

AALBORG UNIVERSITET

SPECIALEPROJEKT

Modellering og forecast af spotpriserne på el vha. SARFIMA-modeller med regimeskift

Forfattere:

Julie BRANDT
Simon MØLLER

Vejleder:

Esben HØG



AALBORG UNIVERSITY
STUDENT REPORT

11. juni 2014

Aalborg University
Department of Mathematics
Fredrik Bajers Vej 7G, DK-9220 Aalborg Ø, Denmark
Phone +45 99409940, Fax +45 98158129
grubbe@math.aau.dk
www.math.aau.dk

AALBORG UNIVERSITET
INSTITUT FOR MATEMATISKE FAG

TITEL:

Modellering og forecast af spotpriserne på el vha. SARFIMA-modeller med regimeskift

PROJEKTPERIODE:

Fra 3. februar
til 11. juni 2014

PROJEKTGRUPPE:

Julie Brandt

Simon Møller

VEJLEDER:

Esben Høg

OPLAGSTAL: 5

ANTAL SIDER: 124

Abstract

This thesis is concerned with the modelling and forecasting of electricity spot prices in select areas of the Nordpool Spot market. The day-ahead market prices are extracted from five different bidding areas, and these price series form the basis for further investigations.

The spot price data exhibit huge fluctuations and outliers, which are typical for energy prices. However, the prices still tend to revert to a certain mean level. Each bidding area is connected to one or several bidding areas, thus enabling the transmission of electricity between areas. When there is no congestion in an interconnection, prices in the connected areas will be identical, and congestion, on the other hand, will lead to different prices. There are five different connections between the five selected bidding areas, and in approximately 40-90% of the time, prices are identical across two neighboring areas. The data clearly exhibit a seasonal pattern; prices are usually higher on weekdays than in weekends. Furthermore, prices also show a clear seasonal pattern in terms of time of day and month of the year.

The plotting of the sample auto covariance functions of the price series clearly shows the presence of long memory in the data. This motivates the use of a seasonal fractional ARIMA (SARFIMA) model for modelling and forecasting. Initially, the data will be adjusted for seasonality by means of linear regression on linear trend and dummy variables representing time of day, day of week, and month of year. The resulting seasonally adjusted prices are then to be modelled and forecasted using a SARFIMA model.

The influence of congestion in connections between bidding areas prompts a further modification of the SARFIMA model; the states of congestion and non-congestion, respectively, define the two regimes of a regime-switching SARFIMA model, thus allowing a different set of parameters to be estimated for each regime. A Markov chain determines the switching between regimes.

For each connection, a non-regime-switching and a regime-switching SARFIMA model are estimated for the time series of the spot price in each of the two connected areas and the price difference. Because the regimes are observable, the model parameters can be estimated by conditional maximum likelihood.

The implementation of the model parameter estimation has been carried out using the R programming language. The estimation process involves nonlinear numerical optimization powered by the L-BFGS algorithm. Model estimation has proven to be considerably time consuming, especially regarding the estimation of parameters for

the regime-switching models.

The model forecasts have been produced using Monte Carlo simulations.

Finally, forecast results from both the non-regime-switching and the regime-switching SARFIMA model are plotted and examined. In general, the non-regime-switching SARFIMA model quite surprisingly manages to outperform the regime-switching model in terms of forecast accuracy.

Forord

Dette speciale er udarbejdet af gruppe G4-109b på Mat-Øk10, d. 3. februar til d. 11. juni 2014 ved Det Teknisk-Naturvidenskabelige Fakultet, Institut for Matematiske Fag, ved Aalborg Universitet. I dette spacie har vi valgt at arbejde med modelle-ring af spotpriser på el, ved at benytte SARFIMA-modeller med og uden regimeskift. Dette gøres med henblik på at kunne forecaste spotpriserne på el. Specialet er udarbejdet i samarbejde med vejleder Esben Høg, og al data er hentet fra Nord Pool Spot.

Referencer til ligninger er angivet i (), hvorimod referencer til kapitler, afsnit, tabeller og figurer angives uden parenteser. Fodnoternes anvisning er ved et opløftet tal i fed skrift, så der kan skelnes mellem fodnoter og potenser.

Kildehenvisninger bliver angivet med [], hvori der vil stå forfatterens efternavn, udgivelsesår og evt. sidetal. I starten af hvert afsnit angives de primære kilder som afsnittet baserer sig på. Resterende kilder angives løbende.

Slutteligt i projektet er der 4 bilag, der benævnes med store bogstaver. Vigtige R-koder findes i appendiks D.

Plots af resultater er udarbejdet i R, hvor pakken `ggplot2` er blevet benyttet.

Aalborg den 11/6 2014

Julie Brandt

Simon Møller

Indhold

Forord	v
Indledning	1
1 ARFIMA-processer	3
1.1 Long-memory	3
2 Datapræsentation	9
2.1 Klargøring af data	9
2.2 Grafisk præsentation af data	10
3 Estimation af model uden regimeskift	27
3.1 Sæsonrensning af data	27
3.2 Estimation af modeller	32
4 Estimation af modeller med regimeskift	41
4.1 Sæsonrensning af data	41
4.2 Estimation af modeller	42
5 Forecast af elspotpriserne ved brug af modellerne uden regimeskift	49
5.1 Implementering af forecast	49
5.2 Forecast udført vha. modeller uden regimeskift	51
6 Forecast af elspotpriserne ved brug af modellerne med regimeskift	57
6.1 Implementering af forecast udført af modeller med regimeskift	57
6.2 Forecast udført vha. modeller med regimeskift	58
7 Konklusion	71
Litteratur	73

A Koefficienterne for fraktionelle differenser	75
B Tests af enhedsrod	77
C Koefficienter til dummyvariablene i forbindelse med sæsonrensning	81
D R-koder	99

Indledning

Elektricitet er i dag en ressource, de fleste mennesker tager for givet, når de fx. tænder lyset eller ovnen. Samfundet i de fleste lande ville lukke ned, hvis elektriciteten forsvandt, idet de essentielle hjælpemidler, såsom metro, elektroniske betalingssystemer, hospitalsudstyr etc., ville være ubrugelige. Det er vigtigt, at produktionen og forbruget af elektricitet er i lige vægt, idet det ikke er muligt at opbevare store mængder af elektricitet. Derfor skal elektriciteten forbruges så snart den er produceret. Det er således nødvendigt at kunne forudsige, hvor stort forbruget og produktionen vil være i fremtiden.

I de nordiske lande er der blevet etableret en elbørs kaldet Nord Pool Spot. Nord Pool Spot er en fællesnordisk elbørs, som har skabt et marked for elektricitet på tværs af forskellige nordiske lande. De lande, der indgår i Nord Pool Spot, er Danmark, Sverige, Norge, Finland, Estland, Letland og Litauen. Tilsammen er de 7 lande underinddelt i 15 områder. Hvert område er via elkabler forbundet til de nærmestliggende områder. De 15 områder og forbindelserne fremgår af figur 1¹.

Nord Pool Spot har to markeder: *day-ahead*, også kaldet Elspot, og *intraday*, også kaldet Elbas. Idet over 70 %² af det samlede elforbrug i Norden handles på Elspot, fokuseres der i projektet kun på dette marked. På day-ahead-markedet fastsættes priserne et døgn frem i tiden. Dagen inden driftsdøgnet indgiver elleverandørerne deres købsbud, og elproducenterne indgiver deres salgsbud. Timeprisen på hver time i driftsdøgnet udregnes ud fra disse købs- og salgsbud. Dermed er elpriserne et udtryk for lige vægten mellem udbuddet og efterspørgslen.

Idet der er nogle begrænsninger for hvor meget elektricitet, der kan transporteres mellem områderne, er der risiko for, at der kan opstå kapacitetsproblemer, som medfører, at markedet ikke kan udligne priserne. I hvert af de enkelte områder er det nødvendigt at matche udbud og efterspørgsel, under hensyntagen til eksport og import af strøm til og fra naboområder. Når der opstår kapacitetsproblemer, forsøger Nord Pool Spot at flytte mest muligt energi fra de områder, der har overskud, og hen til de områder der er i underskud. Derved bliver priserne i områderne med overskud af energi lavere, og priserne i områderne med underskud stiger.

Nord Pool Spot udgør grundlaget i det markedsbaserede elsystem, som der haves i Norden. Fordelene ved et sådant marked er, at der hele tiden sørges for, at el pro-

¹ Figuren er konstrueret ud fra en tilsvarende figur på Nord Pool Spots hjemmeside.

² Jf. http://energinet.dk/SiteCollectionDocuments/Danske%20dokumenter/El/9258-13_v1_Introduktion%20til%20elmarkedet.eps



Figur 1: Kort over de 15 områder i Nord Pool Spot.

duceres af dem, der kan gøre det billigst. Derudover sørger markedet for en effektiv udnyttelse af produktionen, ved at elektriciteten flyder fra lavprisområder til områder, hvor priserne er høje. Der er endvidere transparente elpriser, dvs. markedet er gennemskueligt.

I dette speciale vil det blive forsøgt at modellere og forecaste energipriser vha. SARFIMA(p, d, q)-processer både med og uden regimeskift. Specialet er inspireret af artiklerne *A regime switching long memory model for electricity prices* og *Directional Congestion and Regime Switching in a Long Memory Model for Electricity Prices*, som begge er skrevet af Niels Haldrup og Morten Ørregaard Nielsen.

Implementeringen er udført på data fra Nord Pool Spot. For at mindske omfanget af specialet, er fem områder i Nord Pool Spot blevet udvalgt til implementeringen. De fem udvalgte områder er: Sydvestnorge (NO2), Vest-/Midtsverige (SE3), Sydsverige (SE4), Vestdanmark (DK1), og Østdanmark (DK2).

Kapitel 1

ARFIMA-processer

I dette kapitel vil teorien bag ARFIMA-processer blive gennemgået. Kapitlet er skrevet på baggrund af [Stoffer og Shumway, 2011, pp. 267-277], [Palma, 2007, pp. 39-55], [Box et al., 2008, pp. 429-430, 433-434], [Beran, 1994, pp. 59-66], [Brockwell og Davis, 1991, pp. 520-522] og [Granger og Joyeux, 1980].

1.1 Long-memory

En kausal ARMA(p, q)-proces kan opstilles som en MA(∞)-proces på følgende form:

$$x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j}. \quad (1.1)$$

Denne proces kaldes en *short memory* proces, idet koefficienterne ψ_j i ligning (1.1) aftager eksponentielt. Så vil autokovariansfunktionen (ACF) for denne short memory proces jf. [Stoffer og Shumway, 2011, pp. 99] ligeledes aftage eksponentielt, dvs. $\rho(h) \rightarrow 0$ med eksponentiel hastighed, når $h \rightarrow \infty$. Idet det vides at ACF for en ARMA(p, q)-proces aftager eksponentielt mod nul, når laglængden går mod uendelig, så eksisterer der en værdi R , $0 < R < 1$, sådan at:

$$|\rho(h)| \leq CR^h \quad h = 1, 2, \dots,$$

hvor $C > 0$ er en konstant.

Hvis der i stedet for haves en tidsrække, hvor ACF ikke aftager eksponentielt, kunne dette umiddelbart give et indtryk af, at det er nødvendigt at tage førstedifferensen, for at opnå stationaritet. Dette er i nogle tilfælde for ekstremt, hvilket kan give en idé om, at der er tale om en *long memory* proces, hvor der, i stedet for førstedifferensen, tages en fraktionel differens. ACF for en long memory proces, som aftager langsomt, har følgende egenskab:

$$\frac{\rho(h)}{Ch^{2d-1}} \rightarrow 1 \quad \text{for } h \rightarrow \infty,$$

hvor $C > 0$ er en konstant, og d er den fraktionelle differensparameter. Hvis en long memory proces skal være stationær, gælder der, jf. [Brockwell og Davis, 1991, pp. 524-526], at $-0.5 < d < 0.5$.

Ifølge [Palma, 2007, p. 40] gælder der for long memory processer, at

$$\sum_{h=-\infty}^{\infty} |\gamma(h)| = \infty,$$

hvor $\gamma(h)$ er autokovariansfunktionen for en stationær proces med laglængde h .

1.1.1 ARFIMA(p, d, q)-proces

En *fractionally integrated autoregressive moving average proces*, også kaldet en ARFIMA(p, d, q)-proces, kan anvendes til at modellere data med long memory egenskaber. En ARFIMA(p, d, q)-proces er på formen:

$$\varphi(B)\nabla^d(x_t - \mu) = \theta(B)w_t \quad \text{for } -0.5 < d < 0.5,$$

hvor μ er middelværdien af x_t , $\nabla^d = (1 - B)^d$ er en fraktionel differensoperator, og w_t er hvid støj med varians σ_w^2 . Parameteren d bestemmer den langsigtede adfærd, hvor p og q og de tilhørende parametre i $\varphi(B)$ og $\theta(B)$ bestemmer den kortsigtede adfærd.

Hvis $d \geq 0$ er et heltal, så kan $(1 - B)^d$ omskrives vha. en binomialudvikling, sådan at følgende haves:

$$(1 - B)^d = \sum_{k=0}^d \binom{d}{k} (-1)^k B^k, \quad (1.2)$$

hvor binomialkoefficienterne er givet ved:

$$\binom{d}{k} = \frac{d!}{k!(d-k)!} = \frac{\Gamma(d+1)}{\Gamma(k+1)\Gamma(d-k+1)}. \quad (1.3)$$

Sidste lighedstegn følger af, at der for heltallige positive d gælder at

$$\Gamma(x+1) = x!$$

$\Gamma(\cdot)$ er gammafunktionen, der er givet ved:

$$\Gamma(x) = \begin{cases} \int_0^\infty t^{x-1} e^{-t} dt, & x > 0, \\ x^{-1} \Gamma(1+x), & x \neq 0, -1, -2, \dots \end{cases}$$

Gammafunktionen $\Gamma(x)$ er defineret for alle reelle tal forskellig fra de negative heltal samt nul. Den fraktionelle differensoperator $(1 - B)^d$ kan defineres for alle reelle tal $d > -1$, således at der vha. en binomialudvikling fås, at

$$(1 - B)^d = \sum_{k=0}^{\infty} \binom{d}{k} (-1)^k B^k. \quad (1.4)$$

I ovenstående ligning må der gælde, at når d er et heltal, så er de første $d + 1$ led forskellige fra nul. I alle de efterfølgende led vil der i nævneren i den sidste brøk i ligning (1.3), indgå en gammafunktion evaluert i et negativt heltal eller 0. I dette tilfælde må ligning (1.4) være lig med ligning (1.2).

ARFIMA(0,d,0)-proces

En ARFIMA(0,d,0)-proces kan skrives som:

$$(1 - B)^d x_t = w_t, \quad \text{for } -0.5 < d < 0.5, \quad (1.5)$$

og denne tidsrække kaldes ofte for fraktionel hvid støj. Ved at anvende binomialudvikling på en ARFIMA(0,d,0)-proces, hvor $d > -1$, fås følgende:

$$w_t = (1 - B)^d x_t = \sum_{k=0}^{\infty} \pi_k B^k x_t = \sum_{k=0}^{\infty} \pi_k x_{t-k}, \quad (1.6)$$

hvor¹

$$\pi_k = \frac{\Gamma(k-d)}{\Gamma(k+1)\Gamma(-d)} = \prod_{0 < j \leq k} \frac{j-1-d}{j}, \quad k = 0, 1, 2, \dots \quad (1.7)$$

Dermed gælder der for π_k , at

$$\pi_k = \frac{k-1-d}{k} \pi_{k-1}.$$

Tilsvarende gælder der for $d < 1$, idet processen skrives som en MA(∞), at

$$x_t = (1 - B)^{-d} w_t = \sum_{k=0}^{\infty} \psi_k B^k w_t = \sum_{k=0}^{\infty} \psi_k w_{t-k}, \quad (1.8)$$

hvor²

$$\psi_k = \frac{\Gamma(k+d)}{\Gamma(k+1)\Gamma(d)} = \prod_{0 < j \leq k} \frac{j-1+d}{j}, \quad k = 0, 1, 2, \dots \quad (1.9)$$

Processerne i ligning (1.6) og (1.8) er stationære, når $-0.5 < d < 0.5$ jf.

[Brockwell og Davis, 1991, p. 521]. Endvidere gælder der for en ARFIMA(0,d,0)-proces, at denne er invertibel på trods af, at processen ikke opfylder at den absolute værdi af koefficienterne summeret er mindre end uendelig. Ydermere haves det, jf. [Brockwell og Davis, 1991, p. 521], at processen er kausal, hvis koefficienterne opfylde, at

$$\sum \pi_k^2 < \infty \quad \text{og} \quad \sum \psi_k^2 < \infty.$$

Det er muligt at bestemme variansen og autokorrelationen for ARFIMA(0,d,0)-processen i ligning (1.5). Jf. [Brockwell og Davis, 1991, pp. 521-524] er variansen og autokorrelationen for den fraktionelle hvide støj givet ved:

$$\begin{aligned} \gamma(0) &= \sigma^2 \frac{\Gamma(1-2d)}{\Gamma^2(1-d)}, \\ \rho(h) &= \frac{\Gamma(h+d)\Gamma(1-d)}{\Gamma(h-d+1)\Gamma(d)} = \prod_{0 < j \leq h} \frac{j-1+d}{j-d}, \quad h = 1, 2, \dots \end{aligned}$$

I det efterfølgende afsnit vil teorien bag estimation af parametrene i en ARFIMA(p, d, q)-proces blive gennemgået.

¹Udledninger af π_k og ψ_k er beskrevet i appendiks A.

²Se fodnote 1.

1.1.2 Estimation af parametre

En ARFIMA(p, d, q)-proces med middelværdi nul kan opstilles på følgende generelle form:

$$\varphi(B)(1 - B)^d x_t = \theta(B)w_t, \quad (1.10)$$

hvor $-0.5 < d < 0.5$ og w_t er hvid støj med middelværdi nul og varians σ_w^2 .

Estimationen af parametrene d , $\boldsymbol{\varphi}$, $\boldsymbol{\theta}$ og σ_w^2 , hvor $\boldsymbol{\varphi}$ og $\boldsymbol{\theta}$ er parametervektorerne for hhv. AR- og MA-delen, kan fx. udføres vha. approksimeret maksimumlikelihoodestimation. Dette gøres ved at opstille ARFIMA-processen i ligning (1.10) på AR(∞)-form: $\pi^*(B)x_t = w_t$. Dermed fås følgende udtryk:

$$x_t - \sum_{k=1}^{\infty} \pi_k^* x_{t-k} = w_t,$$

hvor

$$\pi^*(B) = 1 - \sum_{k=1}^{\infty} \pi_k^* B^k = \theta^{-1}(B)\varphi(B)(1 - B)^d.$$

Vha. relationen

$$\theta(B)\pi^*(B) = \varphi(B)(1 - B)^d \equiv \xi(B),$$

som er ækvivalent med

$$(1 + \theta_1 B + \dots + \theta_q B^q)(1 - \pi_1^* B - \pi_2^* B^2 - \dots) = (1 - \xi_1 B - \xi_2 B^2 - \dots), \quad (1.11)$$

kan π_k^* -koefficienterne i AR(∞)-repræsentationen bestemmes. Dette gøres ved at sammenligne koefficienterne i ligning (1.11), således at

$$\pi_k^* + \theta_1 \pi_{k-1}^* + \dots + \theta_q \pi_{l-q}^* = \xi_k, \quad k = 1, 2, \dots,$$

hvor $\pi_0^* = -1$, og

$$\xi(B) = \varphi(B)(1 - B)^d = 1 - \sum_{k=1}^{\infty} \xi_k B^k. \quad (1.12)$$

Nedenstående eksempel viser hvad der gælder for koefficienterne i en ARFIMA(1,d,1).

Eksempel 1.1

For $-0.5 < d < 0.5$ i en ARFIMA(1,d,1) haves følgende relation:

$$\theta(B)\pi^*(B) = \xi(B),$$

hvilket er ækvivalent med:

$$(1 + \theta_1 B)(1 - \pi_1^* B - \dots) = (1 - \xi_1 B - \dots). \quad (1.13)$$

Ved at sammenligne koefficienterne i ligning (1.13) for $k = 1, 2, \dots$, fås:

$$\begin{aligned} (k=1) : \quad -\xi_1 &= \theta_1 - \pi_1^* \\ (k=2) : \quad -\xi_2 &= -\theta_1 \pi_1^* - \pi_2^* \\ (k=3) : \quad -\xi_3 &= -\theta_1 \pi_2^* - \pi_3^* \\ &\vdots \end{aligned}$$

Ovenstående kan opsummeres til følgende generelle udtryk:

$$\pi_k^* + \theta_1 \pi_{k-1}^* = \xi_k, \quad (1.14)$$

hvor der igen gælder, at $\pi_0^* = -1$.

Nu betragtes relationen i ligning (1.12), hvilket medfører, at

$$\xi(B) = \varphi(B)(1 - B)^d$$

er ækvivalent med

$$(1 - \xi_1 B - \dots) = (1 - \varphi_1 B)(1 + \pi_1 B + \pi_2 B^2 + \dots). \quad (1.15)$$

Ved at sammenligne koefficienterne i ligning (1.15) for $k = 1, 2, \dots$, fås:

$$\begin{aligned} (k=1) : \quad -\xi_1 &= \pi_1 - \varphi_1 \\ (k=2) : \quad -\xi_2 &= \pi_2 - \varphi_1 \pi_1 \\ (k=3) : \quad -\xi_3 &= \pi_3 - \varphi_1 \pi_2 \\ &\vdots \end{aligned}$$

Ovenstående kan opsummeres til følgende udtryk:

$$\xi_k = -\pi_k + \varphi_1 \pi_{k-1}, \quad k \geq 1, \quad (1.16)$$

hvor π_k er givet ved ligning (1.7).

Dvs. i en ARIMA(1,d,1)-proces opfylder π_k^* -koefficienterne udtrykket i ligning (1.14), hvor ξ_k er givet ved relationen i ligning (1.16). \triangle

Ved anvendelse af approksimativ maksimumlikelihoodestimation (MLE) er de truncerede fejl jf. [Box et al., 2008, p. 434] givet ved

$$\varepsilon_t(\boldsymbol{\beta}) = x_t - \sum_{k=1}^{t-1} \pi_k^* x_{t-k}, \quad t = 1, 2, \dots, n. \quad (1.17)$$

Her betragtes ε_t som en funktion af parametervektoren $\boldsymbol{\beta} = (\boldsymbol{\varphi}^\top, \boldsymbol{\theta}^\top, d)^\top$. Estimateet af parametervektoren $\hat{\boldsymbol{\beta}}$ bestemmes ved at minimere kvadratsummen

$$S(\boldsymbol{\beta}) = \sum_{t=1}^n \varepsilon_t^2(\boldsymbol{\beta}). \quad (1.18)$$

Det tilhørende approksimative MLE af variansen σ_w^2 er da givet ved:

$$\hat{\sigma}_w^2 = \frac{S(\hat{\boldsymbol{\beta}})}{n}.$$

Bemærkning:

Når der haves en stor mængde data til rådighed, dvs. n er stor, så bør de første 10-20 led i kvadratsummen i ligning (1.18) udelades, idet approksimationen i ligning (1.17) er dårlig for små værdier af t [Box et al., 2008, p. 434].

Implementering af filteret

I ligning (1.10) kan det observeres, at den fraktionelt differensede tidsrække $(1-B)^d x_t$ følger en ARMA(p, q)-proces. Dermed kan der jf. [Box et al., 2008, p. 434] laves en tidsrække $\{\tilde{x}_t(d)\}$ af de trunkerede værdier af

$$(1 - B)^d x_t \equiv \pi(B)x_t,$$

hvor

$$\tilde{x}_t(d) = x_t + \sum_{k=1}^{t-1} \pi_k(d)x_{t-k}, \quad t = 1, 2, \dots, n. \quad (1.19)$$

Desuden er $-0.5 \leq d < 0.5$ og $\pi_k(d)$ er koefficienterne som defineret i ligningerne (1.6) og (1.7) for et givet d .

Tidsrækken af trunkerede værdier $\tilde{x}_1(d), \dots, \tilde{x}_n(d)$ konstrueres vha. ligning (1.19) for forskellige værdier af d i et gitter, hvori $-0.5 \leq d < 0.5$.

For hver værdi af d i dette gitter findes MLE for parametrene φ , θ og σ_w^2 i ARMA-modellen for tidsrækken $\tilde{x}_1(d), \dots, \tilde{x}_n(d)$. Da vil estimatet \hat{d} være den værdi af d i fornævnte gitter, hvor estimatet $\hat{\sigma}_w^2$ når sit minimum, eller hvor likelihoodfunktionen er maksimeret. For dette \hat{d} fås de tilhørende værdier af estimatorne $\hat{\varphi}$ og $\hat{\theta}$.

Kapitel 2

Datapræsentation

Data benyttet i dette speciale er indhentet fra Nord Pool Spot¹, som er en fælles-nordisk elbørs for landene Sverige, Norge, Danmark, Finland, Estland, Letland og Litauen. Disse lande er tilsammen opdelt i 15 områder, hvor hvert område er forbundet via nogle kabler til forskellige nærtliggende andre områder. Kablerne benyttes til at transportere elektricitet fra et område til et andet. For at afgrænse specialets omfang er fem af de 15 områder blevet udvalgt til den senere databehandling. De fem udvalgte områder er Sydvestnorge (NO2), Vest-/Midtsverige (SE3), Sydsverige (SE4), Vestdanmark (DK1), og Østdanmark (DK2). Der haves 18768 observationer for hvert område. Observationerne er på timebasis og strækker sig over perioden fra den 1. januar 2012 og frem til den 20. februar 2014.

2.1 Klargøring af data

Idet tidsangivelsen i data fra de fem områder er i lokaltid, forekommer der ved skiftet fra normaltid til sommertid en “manglerende” observation, og ved skiftet fra sommertid til normaltid haves en observation “for meget”. Dette medfører, at der ikke er ækvidistante observationer, hvilket er nødvendigt for at kunne anvende tidsrækkestørrelser på data. Dette problem er blevet løst ved at konvertere alle tider til UTC-tid (Greenwich Mean Time). Det bør dog bemærkes, at UTC-tid er en time bagefter dansk normaltid og to timer bagefter dansk sommertid, hvilket muligvis vil have indflydelse på sæsonvariationerne i den senere databehandling. Endvidere er skuddagen den 29. februar 2012 blevet fjernet. Dette gøres for at opnå to på hinanden følgende år med lige mange observationer.

Forud for databehandlingen haves en forventning om, at der vil være en sæsonvariation, som bl.a. vil være baseret på hvilken time på dagen der er tale om, og hvilken ugedag der betragtes. Derfor tilføjes en såkaldt dummyvariabel, som angiver hvilken ugedag observationen tilhører. Yderligere laves der også dummyvariable for hvilken måned og hvilken time på døgnet observationen tilhører.

Der haves yderligere en forventning om at elforbruget er mindre på helligdage, hvilket må påvirke priserne. Derfor opstilles der dummyvariable, der indikerer hvorvidt en pågældende dato er en helligdag. Tabel 2.1 angiver de forskellige helligdage i hhv. Danmark, Sverige og Norge².

¹Spotpriserne kan downloades fra <http://www.nordpoolspot.com/Market-data1/Downloads/Historical-Data-Download1/Data-Download-Page/>

²Kilde: <http://www.timeanddate.com/holidays>

For at få et indblik i hvilke egenskaber og tendenser der er i prisdata, vil data for de fem områder blive afbildet på forskellig vis i det efterfølgende afsnit.

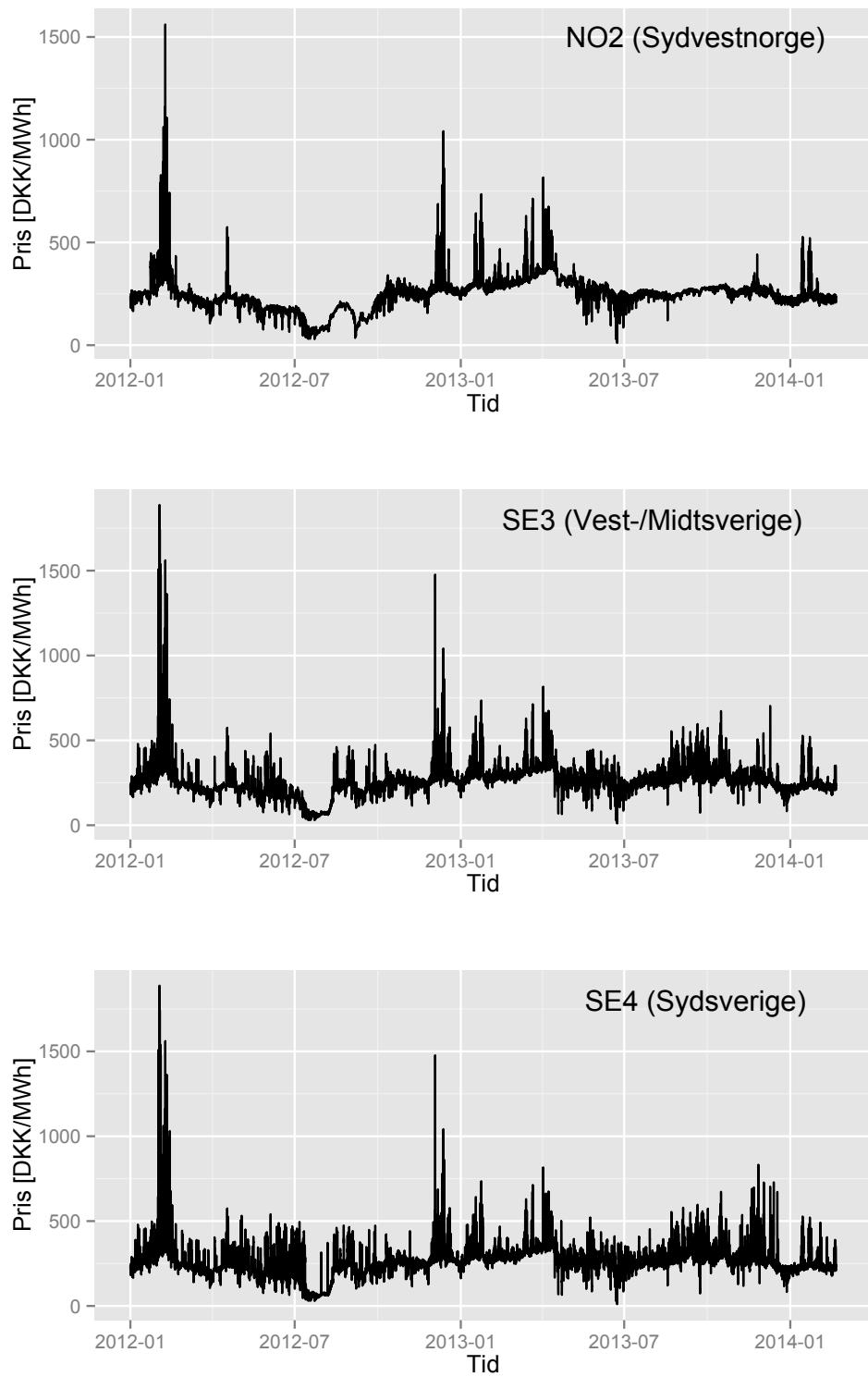
Dag	Betegnelse	Danmark	Sverige	Norge
1. januar 2012	Nytårsdag	✓	✓	✓
6. januar 2012	Helligtrekongersdag		✓	
5. april 2012	Skærtorsdag	✓		✓
6. april 2012	Langfredag	✓	✓	✓
8. april 2012	Påskedag	✓	✓	✓
9. april 2012	2. påskedag	✓	✓	✓
1. maj 2012	Arbejdernes int. kampdag		✓	✓
4. maj 2012	Store bededag	✓		
17. maj 2012	Kristi himmelfartsdag	✓	✓	✓
17. maj 2012	Norges nationaldag			✓
27. maj 2012	Pinsedag	✓	✓	✓
28. maj 2012	2. pinsedag	✓		✓
6. juni 2012	Sveriges nationaldag		✓	
23. juni 2012	Midsommer		✓	
3. november 2012	Allehelgendsdag		✓	
25. december 2012	Juledag	✓	✓	✓
26. december 2012	2. juledag	✓	✓	✓
1. januar 2013	Nytårsdag	✓	✓	✓
6. januar 2013	Helligtrekongersdag		✓	
28. marts 2013	Skærtorsdag	✓		✓
29. marts 2013	Langfredag	✓	✓	✓
31. marts 2013	Påskedag	✓	✓	✓
1. april 2013	2. påskedag	✓	✓	✓
26. april 2013	Store bededag	✓		
1. maj 2013	Arbejdernes int. kampdag		✓	✓
9. maj 2013	Kristi himmelfartsdag	✓	✓	✓
17. maj 2013	Norges nationaldag			✓
19. maj 2013	Pinsedag	✓	✓	✓
20. maj 2013	2. pinsedag	✓		✓
6. juni 2013	Sveriges nationaldag		✓	
22. juni 2013	Midsommer		✓	
2. november 2013	Allehelgendsdag		✓	
25. december 2013	Juledag	✓	✓	✓
26. december 2013	2. juledag	✓	✓	✓
1. januar 2014	Nytårsdag	✓	✓	✓
6. januar 2014	Helligtrekongersdag		✓	

Tabel 2.1: Tabel over de forskellige helligdage i Danmark, Sverige og Norge i årene 2012, 2013 og starten af 2014.

2.2 Grafisk præsentation af data

I dette afsnit præsenteres data vha. forskellige plots. Figur 2.1 viser hvorledes spotpriserne for hhv. Sydvestnorge, Vest-/Midtsverige og Sydsverige har udviklet sig over tidsperioden. Det fremgår af figuren at Sydvestnorge generelt har lavere spotpriser på energi end begge de svenske områder. Alle tre områder viser de karakteristiske egenskaber ved energipriser i form af kortvarige ekstreme priser, også kaldet spikes. Disse spikes indtræffer pludseligt og har en meget kort varighed.

Figur 2.2 viser spotpriserne for hhv. Vestdanmark og Østdanmark. Det øverste plot af spotprisen i Vestdanmark har en ekstremt stor spike, som medfører at figuren er ulæselig, da de andre priser er meget små ift. denne. Derfor vises nedenunder et



Figur 2.1: Spotpriserne for Sydvestnorge, Vest-/Midtsverige og Sydsverige over tid.

udsnit af dette plot. Ud fra figuren fremgår det, at de danske priser varierer meget, og i enkelte perioder opstår der sågar negative priser, hvilket ikke er tilfældet i de øvrige områder. Nord Pool Spot har tilladt negative priser i de to områder DK1 og DK2. Disse negative priser benyttes til at begrænse elproduktionen fra vindmøller om natten, når der er stærk vind, men samtidigt et lavt forbrug af elektricitet³.

For at bekræfte at de store spotpriser, der fremgår af figur 2.2, er korrekte og ikke blot tastefejl e.l., blev Energinet.dk kontaktet pr. mail. Svaret var at disse høje spotpriser rent faktisk var udfaldet den pågældende dag. Ydermere udtrykker disse ekstreme spotpriser, at der ikke var tilstrækkeligt med salgsbud til rådighed i markedet på det pågældende tidspunkt. Derfor indsatte elbørsen en på forhånd bestemt maksimal pris (2000 €/MWh), hvilket begrænser efterspørgslen til udbuddet. Der er tale om en enkeltstående begivenhed, hvor markedsaktørerne ikke var opmærksomme på situationen denne dag. Energinet.dk vurderer derfor at en lignende situation, med så ekstreme spotpriser, er meget usandsynlig. På baggrund af dette er det blevet valgt at udelade disse fem på hinanden følgende observationer af data, og i stedet benyttes der lineær interpolation.

2.2.1 Relationen mellem spotpriserne i forbundne områder

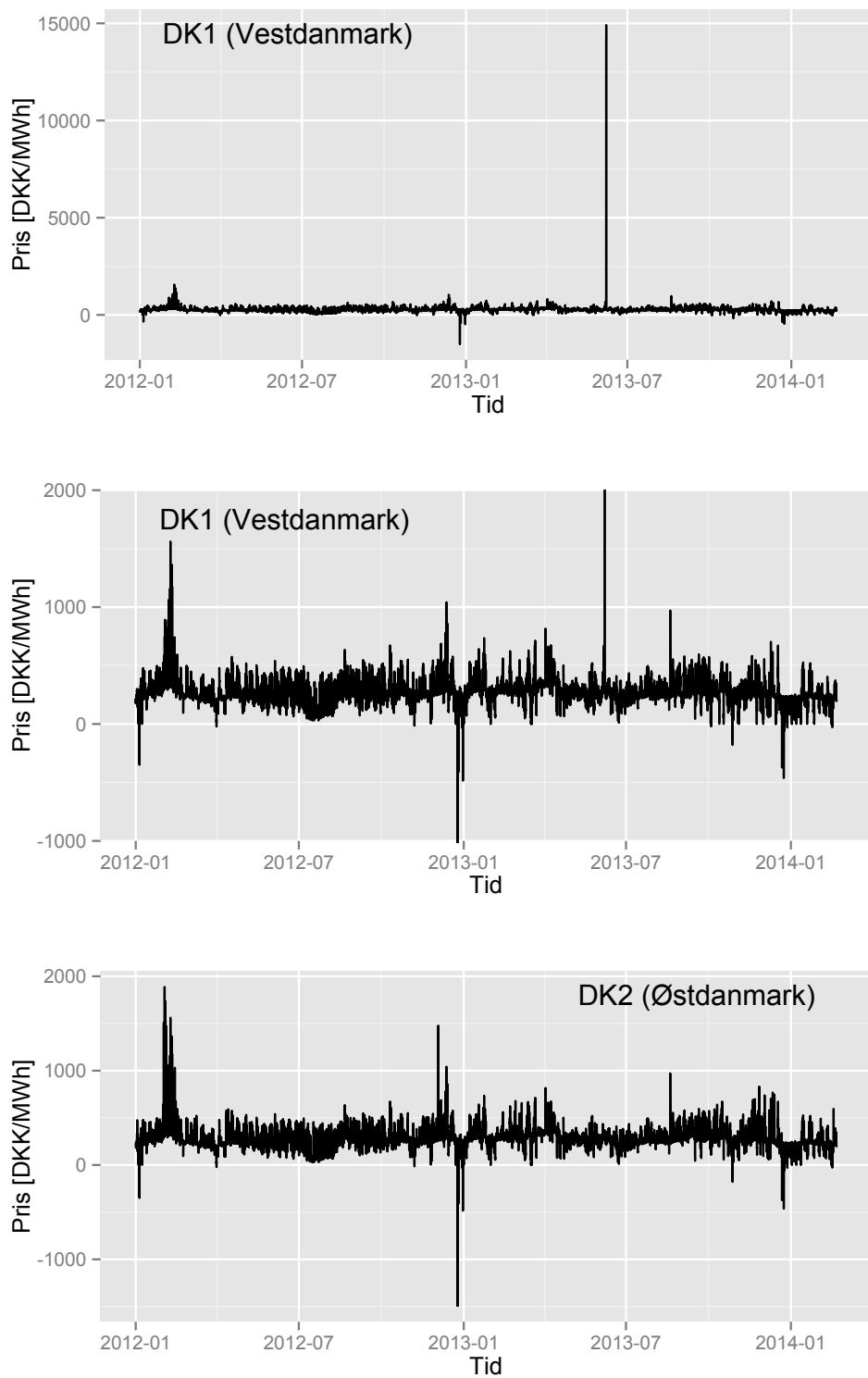
Som tidligere nævnt er de fem områder forbundet med hinanden, idet et givent område er forbundet med et eller flere naboområder. Inden for de fem udvalgte områder haves forbindelserne: NO2-DK1, DK1-SE3, DK1-DK2, DK2-SE4 og SE4-SE3. Disse forbindelser muliggør import/eksport af elektricitet mellem to områder. Som tidligere nævnt afhænger spotprisen på elektricitet af efterspørgsel og produktion i de enkelte områder.

Figurerne 2.3 og 2.4 viser i alt 10 plots af forholdet mellem spotpriserne i to forbundne områder. For hver forbindelse er der to plots, der begge illustrerer sammenhængen mellem spotpriserne i de to forbundne områder. Plottene håndterer begge *overplotting*, hvor plottene til venstre benytter gråtonenuancer, og de til højre benytter farver til at vise hvor mange observationer, der er i et givet punkt. Når priserne ligger på en 45°-linje, er priserne identiske i de to regioner. Dette betyder, at der ikke er kapacitetsproblemer ved denne forbindelse. De punkter som ligger over og under 45°-linjen indikerer situationer, hvori der er opstået et kapacitetsproblem. Hvis punkterne ligger over linjen, er priserne højere i området, der er placeret på andenaksen, og det modsatte er tilfældet, hvis punkterne ligger under linjen.

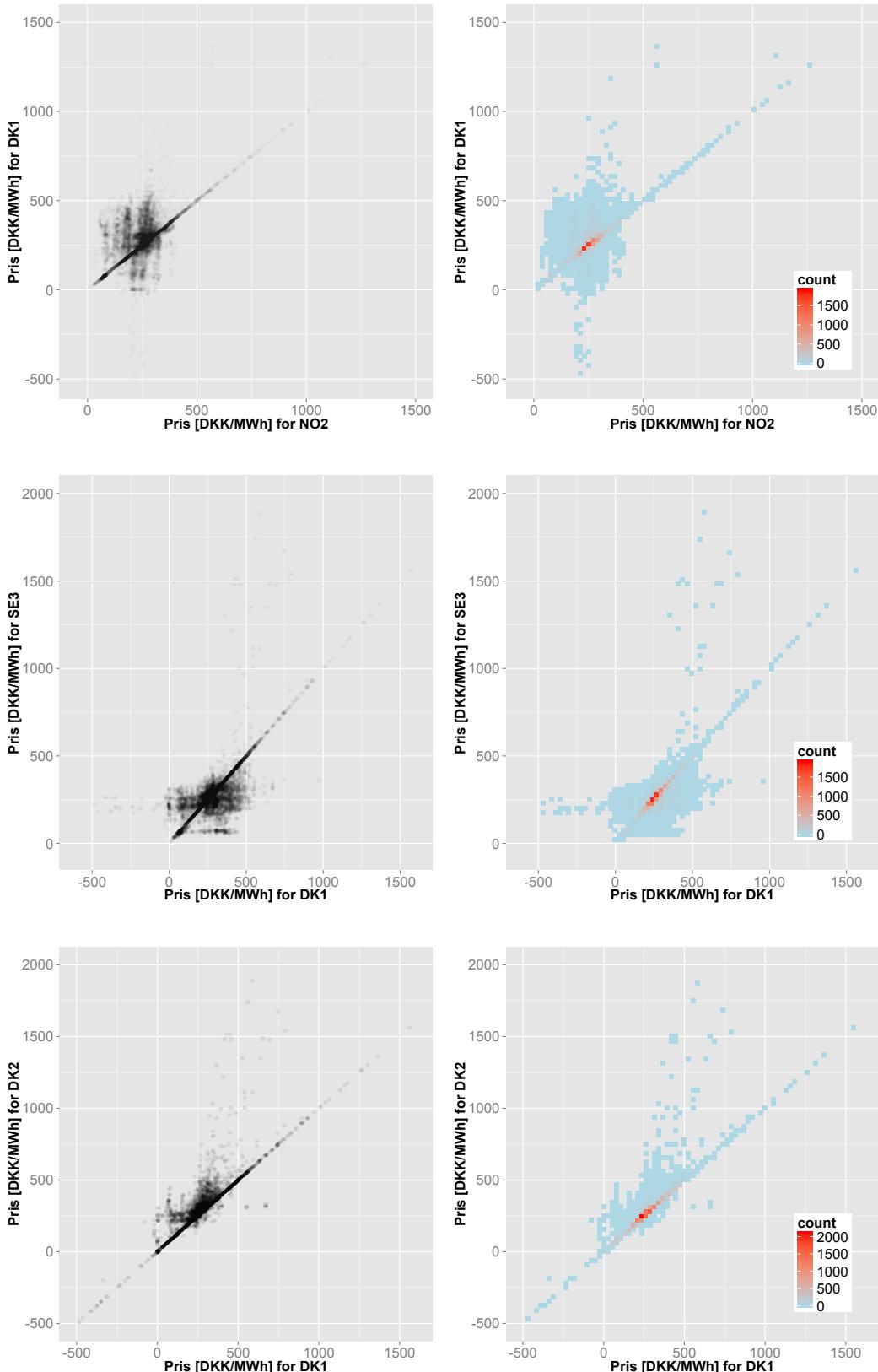
I alle plottene observeres en tydelig 45°-linje. Det bemærkes, at for forbindelsen mellem SE3 og SE4 er de fleste observationer beliggende på 45°-linjen, hvilket vil sige, at spotpriserne oftest er identiske. De øvrige observationer ligger over linjen, hvilket er ensbetydende med at priserne er højere i SE4 end i SE3. For de to områder i Danmark observeres nogenlunde samme tendens, da observationerne oftest er beliggende på 45°-linjen. I tilfælde med ikke-identiske priser vil det oftest være dyrere i DK1.

For at give et mere præcist billede af hvorledes priserne opfører sig ift. hinanden, er der for hver forbindelse anført antal observationer med identiske priser, samt antal observationer, hvor spotprisen i det ene område er hhv. mindre og større end i det forbundne naboområde. Dette er opsummeret i tabel 2.2. Her bemærkes det, at

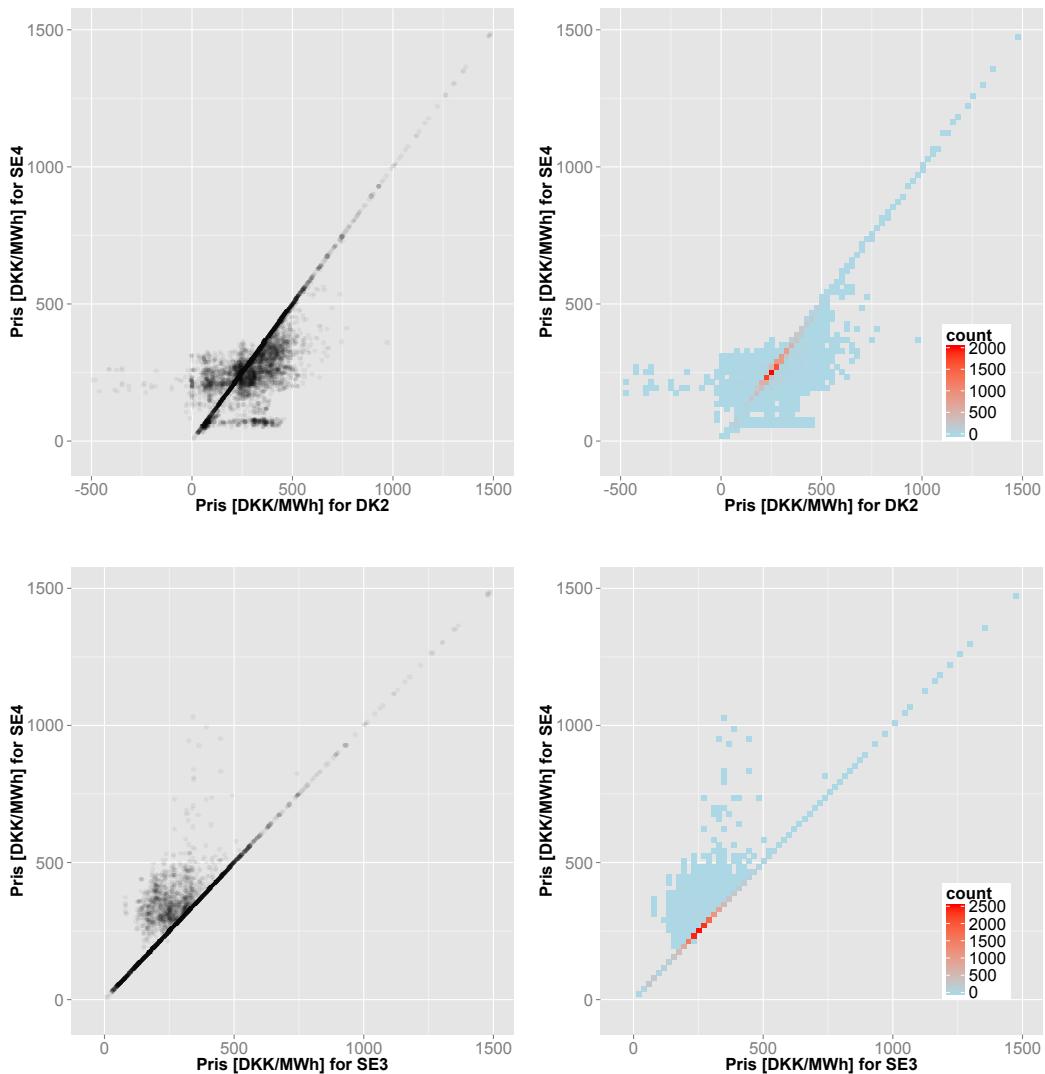
³Kilde: <http://energinet.dk/SiteCollectionDocuments/Engelske%20dokumenter/Klimaogmiljo/Wind%20power%20magazine.eps>



Figur 2.2: Spotpriserne for Vestdanmark og Østdanmark over tid.



Figur 2.3: Scatterplots af spotpriser sammenholdt mellem forbundne regioner.



Figur 2.4: Scatterplots af spotpriser sammenholdt mellem forbundne regioner.

der i fire ud af de fem forbindelser haves flest observationer, hvor priserne i de to forbundne områder er identiske. I forbindelsen NO2-DK1 er der dog flest tilfælde, hvor priserne er forskellige. Ud fra tabellen er det muligt at observere de samme egenskaber, som fremgår af figurerne 2.3 og 2.4. For SE4-SE3-forbindelsen tydeliggør tabellen, at spotpriserne i langt de fleste tilfælde er identiske, og at det kun er i ét tilfælde, at spotprisen er større i SE3 end i SE4.

		Antal observationer	%
NO2 - DK1	NO2 = DK1	8069	43.05%
	NO2 > DK1	3579	19.09%
	NO2 < DK1	7096	37.86%
DK1 - SE3	SE3 = DK1	10578	56.43%
	SE3 > DK1	3934	20.99%
	SE3 < DK1	4232	22.58%
DK1 - DK2	DK1 = DK2	14784	78.87%
	DK1 > DK2	393	2.10%
	DK1 < DK2	3567	19.03%
DK2 - SE4	DK2 = SE4	13928	74.31%
	DK2 > SE4	3477	18.55%
	DK2 < SE4	1339	7.14%
SE3 - SE4	SE4 = SE3	17250	92.03%
	SE4 > SE3	1493	7.97%
	SE4 < SE3	1	0.005%

Tabel 2.2: Oversigt over antal observationer for hhv. identiske og ikke-identiske priser for hver forbindelse.

Tabel 2.3 giver et overblik over hvor ofte spotpriserne er identiske i alle fem områder, dvs. hvor ofte der er én pris på tværs af alle områder. Desuden viser tabellen også hvor ofte der er hhv. to, tre, fire og fem forskellige priser på tværs af områderne.

Antal forskellige priser	Antal gange	%
1	6724	35.87%
2	7140	38.09%
3	4204	22.43%
4	659	3.52%
5	17	0.091%

Tabel 2.3: Oversigt over hvor mange forskellige priser der haves på tværs af områderne, på et givet tidspunkt. Fx. fremgår det, at der i ca. 36% af tiden haves den samme pris i samtlige fem områder.

2.2.2 Sæsonvariationer i elpriserne

Figurerne 2.5 og 2.6 viser to typer plots for hvert område. Det ene er et søjlediagram, som viser gennemsnitsprisen for hver ugedag. Ud fra disse søjlediagrammer fremgår det, at gennemsnitsprisen er lavere på lørdage og søndage, men dette er dog mindre udtalt i området NO2. De lavere priser i weekenderne indikerer, at der haves en ugentlig sæsonvariation.

Det andet plot viser udviklingen i gennemsnitsprisen i løbet af et døgn for hhv. hverdage i januar, weekender/helligdage i januar, hverdage i juli samt weekender/helligdage i juli. Disse plots viser flere forskellige tendenser, hvoraf nogle er overordnede for alle fem områder, mens andre kun gør sig gældende for enkelte områder.

Fælles for dem alle er at priserne er højere på hverdage end i weekender/på helligdage, og at priserne er højere i januar måned, end de er i juli måned. Dette fremgår tydeligt af plottene for område NO2 og SE3. Om sommeren i områderne NO2 og SE3 er variationen i prisen i løbet af et døgn relativt lille. Anderledes forholder det sig i områderne DK1 og DK2, hvor der er en stor variation i prisen i løbet af et døgn i både juli og januar. De højere priser i januar i forhold til juli, indikerer at, der er tale om årlig sæsonvariation.

Endvidere fremgår det, at der for alle fem områder haves to peaks, når kurven for hverdage i januar betragtes. Disse peaks eksisterer ikke i juli, når områderne NO2 og SE3 betragtes. Plottene viser tydeligt, at priserne varierer mest i de to danske områder DK1 og DK2.

Tabel 2.4 viser egenskaber ved priserne i de fem forskellige områder. Alle egenskaber er angivet for hhv. alle dage, hverdage og weekender/helligdage. Af tabellen fremgår middelværdien af prisen for hvert område, og det er tydeligt at middelværdien af priserne er mindre i weekender/på helligdage end på hverdage. Endvidere bør det bemærkes, at den mindste spotpris for hvert område indtræffer i weekenden/på en helligdag, og at maksimumsværdien af spotprisen for de fem områder altid falder på en hverdag. Ydermere betragtes det, at standardafvigelserne er størst på hverdage, hvilket må betyde, at priserne er mere volatile på hverdage end i weekender/på helligdage.

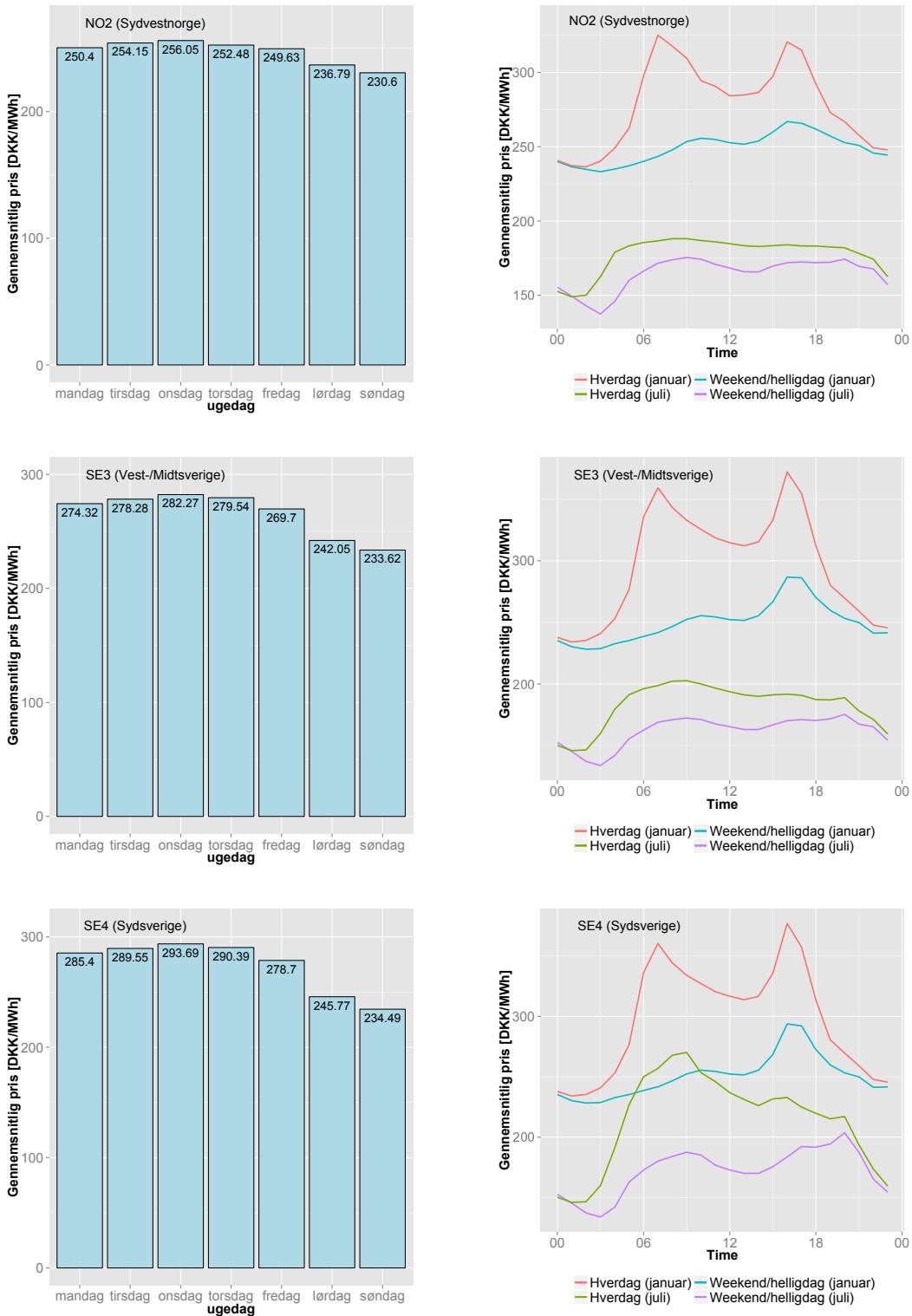
Topstejlheden, der fremgår af tabel 2.4, fortæller at fordelingerne af spotpriserne er mere spids end en normalfordeling, idet værdierne er større end 0. Skævheden måler asymmetri i sandsynlighedsfordelingen. Betragtes værdierne for skævhed for alle fem områder fremgår det, at fordelingerne er højreskæv i hverdage og venstreskæv i weekender/på helligdage.

Område	Middelv.	Median	Std.	Min.	Maks.	Skævh.	Topstejlh.
NO2 alle dage	247.16	249.97	76.57	10.29	1560.99	1.84	18.77
NO2 hverdage	252.72	254.35	81.56	29.16	1560.99	2.19	20.08
NO2 week./helligd.	234.76	240.48	62.31	10.29	769.02	-0.32	1.72
SE3 alle dage	265.68	258.96	94.68	10.29	1887.41	3.26	35.34
SE3 hverdage	278.07	267.52	104.17	29.16	1887.41	3.36	32.83
SE3 week./helligd.	238.10	243.27	60.46	10.29	769.02	-0.51	2.40
SE4 alle dage	274.00	264.61	99.02	10.29	1887.41	2.95	29.44
SE4 hverdage	289.09	276.62	108.43	29.16	1887.41	3.02	27.49
SE4 week./helligd.	240.41	245.26	61.84	10.29	769.02	-0.34	2.99
DK1 alle dage	272.81	265.17	106.77	-1491.92	1560.99	-0.87	32.07
DK1 hverdage	292.98	281.11	105.37	-1118.19	1560.99	0.83	11.54
DK1 week./helligd.	227.61	240.22	95.46	-1491.92	769.02	-6.64	109.68
DK2 alle dage	283.55	272.48	114.80	-1491.92	1887.41	0.42	35.41
DK2 hverdage	304.61	289.14	116.87	-1118.19	1887.41	2.03	21.30
DK2 week./helligd.	236.38	245.55	94.26	-1491.92	769.02	-6.92	117.98

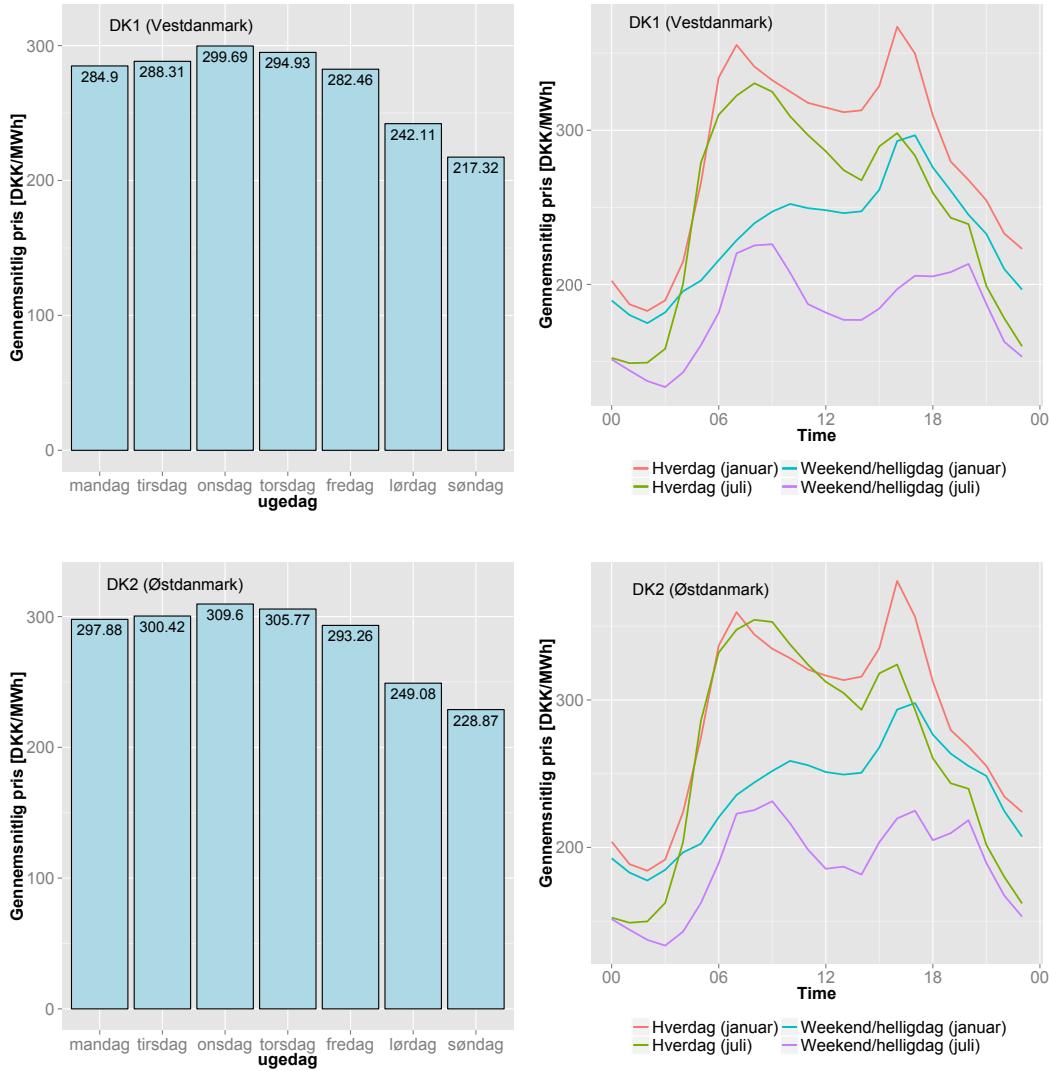
Tabel 2.4: Deskriptiv statistik for hvert af de fem områder for hhv. alle priser, priserne på hverdage og priserne i weekender/på helligdage.

2.2.3 Histogrammer, Q-Q-plots og ACF

Figurerne 2.7 og 2.8 viser histogrammer og Q-Q-plots for spotpriserne i de fem områder. Det fremgår at data ikke er normalfordelte, men dog nogenlunde symmetriske. Ydermere fremgår det af data, at der i områderne DK1 og DK2 opstår tilfælde med



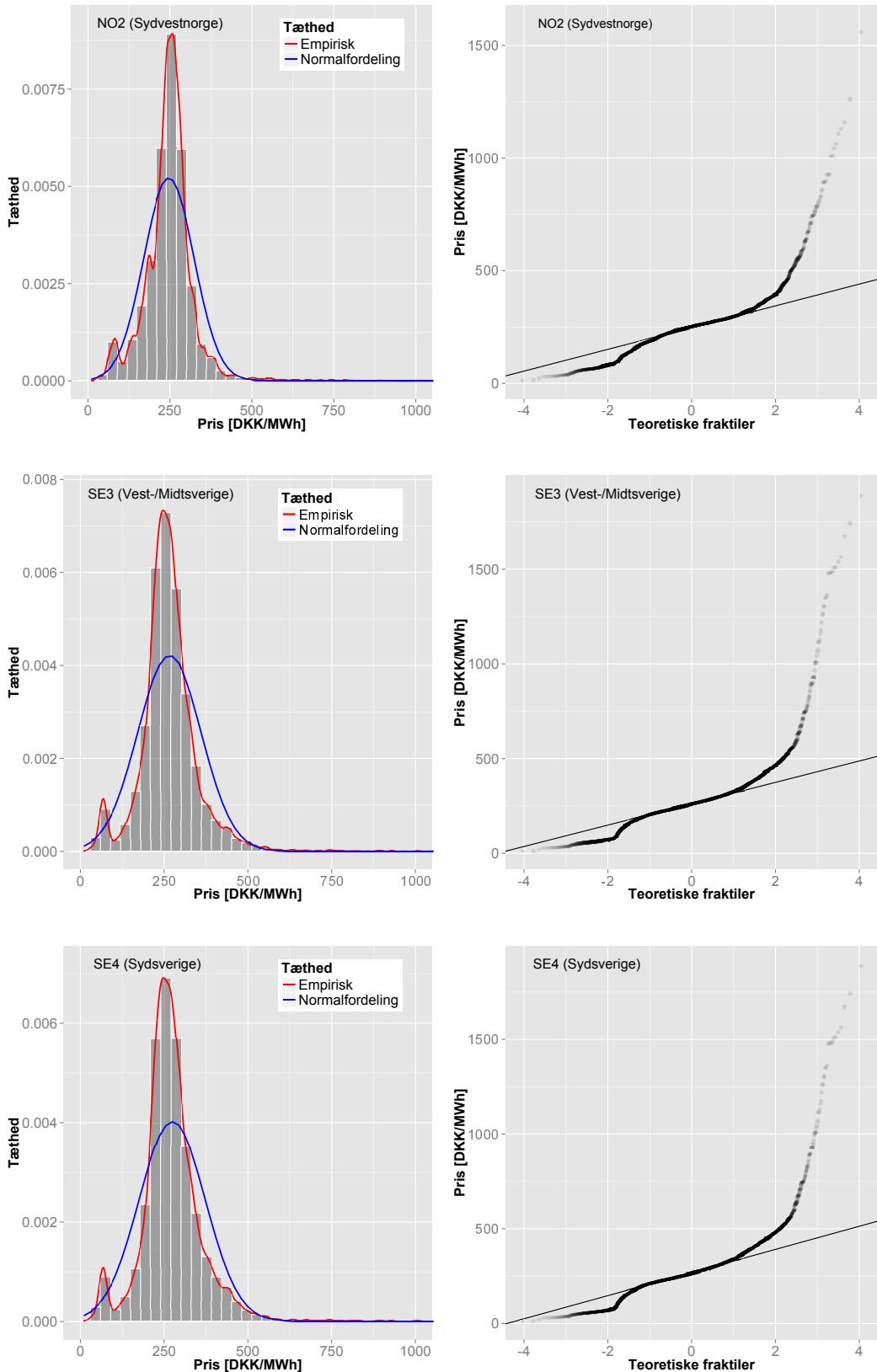
Figur 2.5: Søjlediagram over gennemsnitsprisen på de forskellige ugedage samt plot af udviklingen i den gennemsnitlige pris i løbet af et døgn for hhv. hverdage i januar, weekender/helligdage i januar, hverdage i juli og weekender/helligdage i juli for områderne NO2, SE3 og SE4.



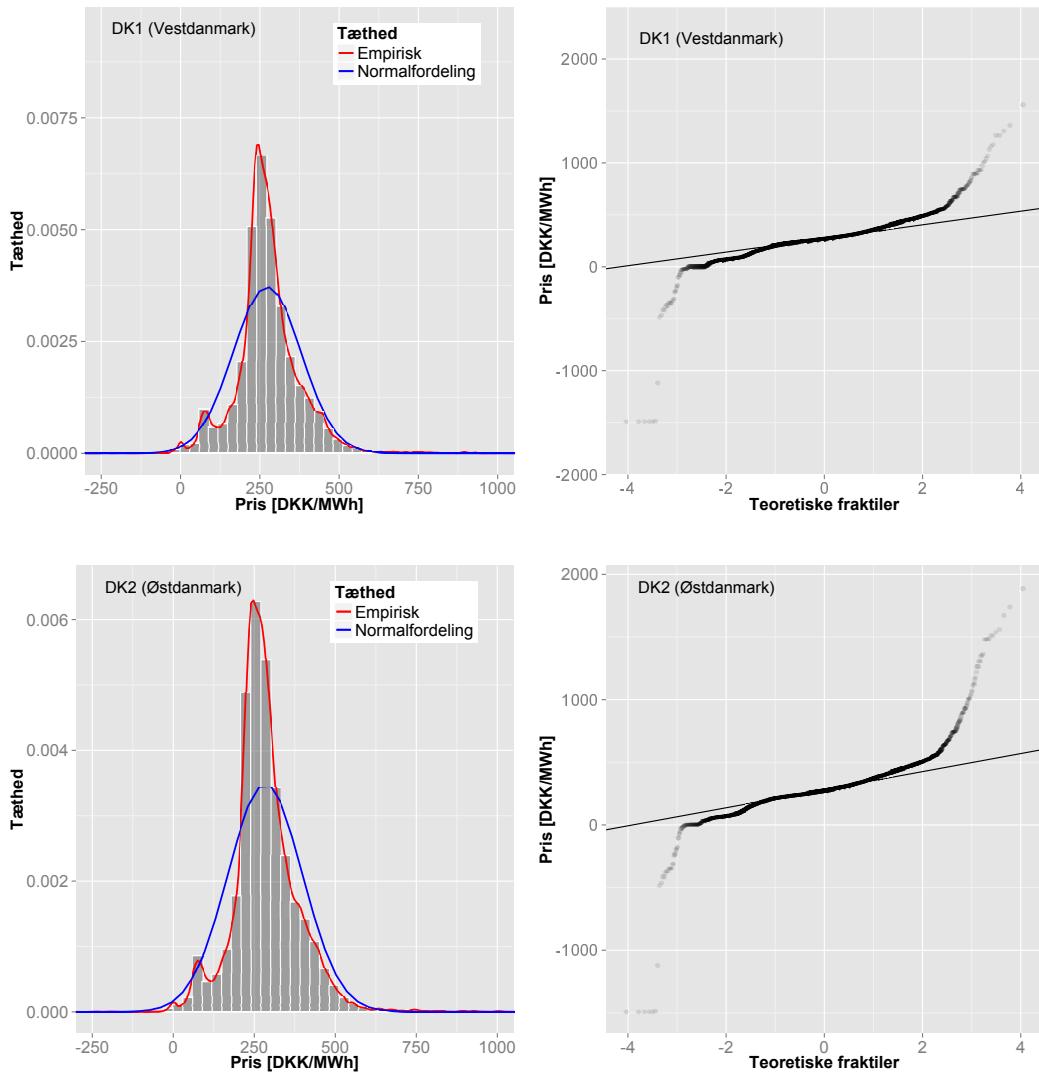
Figur 2.6: Søjlediagram over gennemsnitsprisen på de forskellige ugedage samt plot af udviklingen i den gennemsnitlige pris i løbet af et døgn for hhv. hverdage i januar, weekender/helligdage i januar, hverdage i juli og weekender/helligdage i juli for områderne DK1 og DK2.

negative spotpriser, hvilket ville besværliggøre en log-transformation af data. For at undersøge om en log-transformation ville have en stabilisering effekt på variansen, er de log-transformerede priser blevet plottet. Figur 2.9 indeholder to plots, hvoraf det øverste afbilder de ikke-transformerede priser for område NO2, og det nederste afbilder de log-transformerede priser for område NO2. Det fremgår af figuren, at en log-transformation ikke medfører en stabilisering af variansen. Det samme er gældende for de resterende områder. Derfor fravælges det at transformere data, og der vil i databehandlingen blive arbejdet med de ikke-transformerede priser (DKK/MWh).

Figurerne 2.10-2.12 viser ACF (autokovariansfunktionen) for de fem områder. Disse plots understøtter sammen med figurerne 2.5 og 2.6, at der må være sæsonvariation i data. Ud fra ACF-plottene er det tydeligt, at der må være en daglig sæsonvariation i priserne i alle fem områder. Endvidere fremgår det af ACF for SE3, SE4, DK1 og



Figur 2.7: Histogrammer og Q-Q-plots for spotpriserne i områderne NO2, SE3 og SE4.



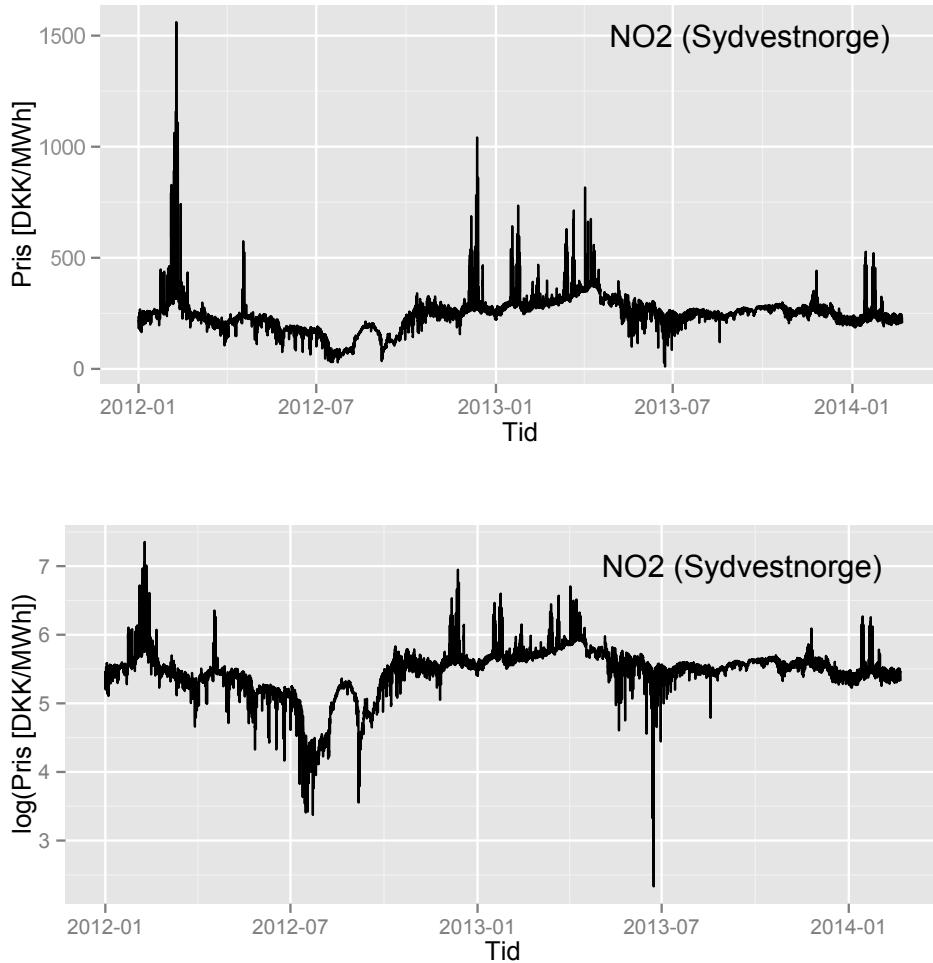
Figur 2.8: Histogrammer og Q-Q-plots for spotpriserne i områderne SE3 og SE4.

DK2, at der også er en ugentlig sæsonvariation, hvilket også fremgik af søjlediagrammerne i figur 2.5 og 2.6.

De store variationer i prisen i områderne DK1 og DK2 fremgår også meget tydeligt af ACF-plottene, hvor der forekommer store peaks i ACF for områderne DK1 og DK2. Endvidere bemærkes det, at der for område NO2 haves små peaks, hvilket må kunne forklares med mindre udtalte prisvariationer. ACF-plottene for de fem områder viser tydeligt, at der er long memory i data.

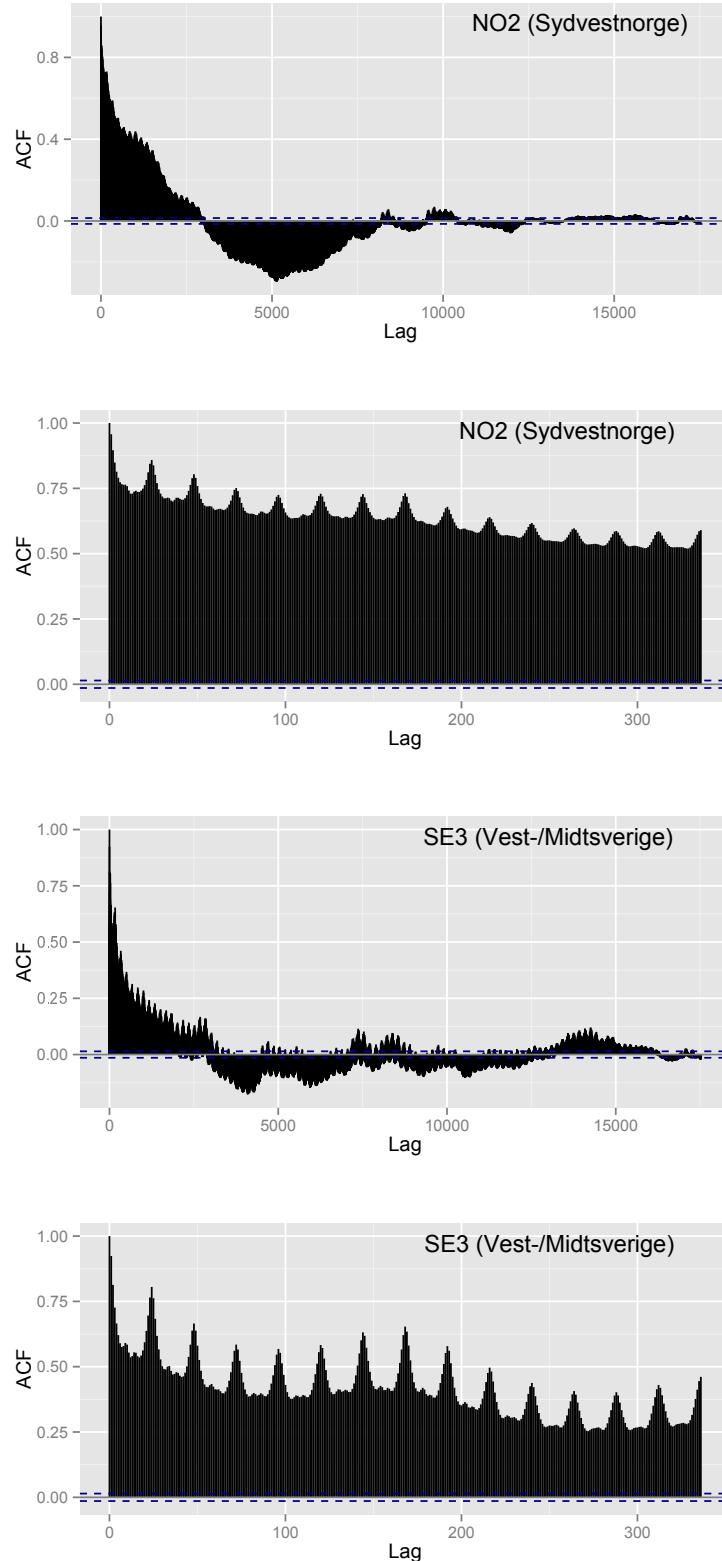
2.2.4 Opsummering af datapræsentationen

Præsentationen af data har givet et indblik i hvilke egenskaber og information data indeholder. Spotprisen opfører sig forskelligt indenfor de forskellige områder. Dog observeres der en række gennemgående tendenser, såsom mindre spotpriser i weekender/på helligdage og højere spotpriser i januar end i juli. Sidstnævnte karakteristika er dog mere udtalt for NO2 og SE3. Et andet gennemgående træk er at spotprisen

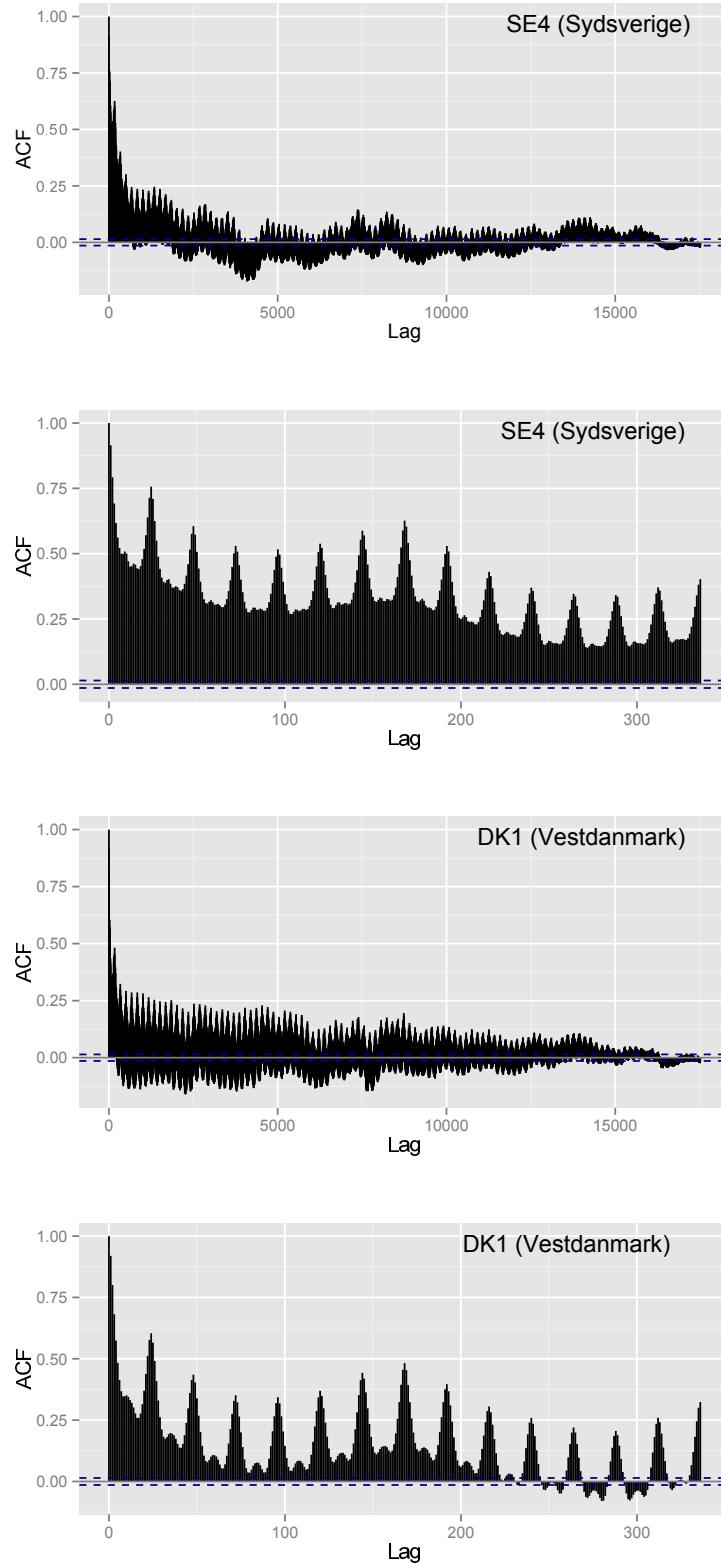


Figur 2.9: Øverste plot viser de ikke-transformerede priser for NO2, og nederste plot viser de log-transformerede priser for NO2

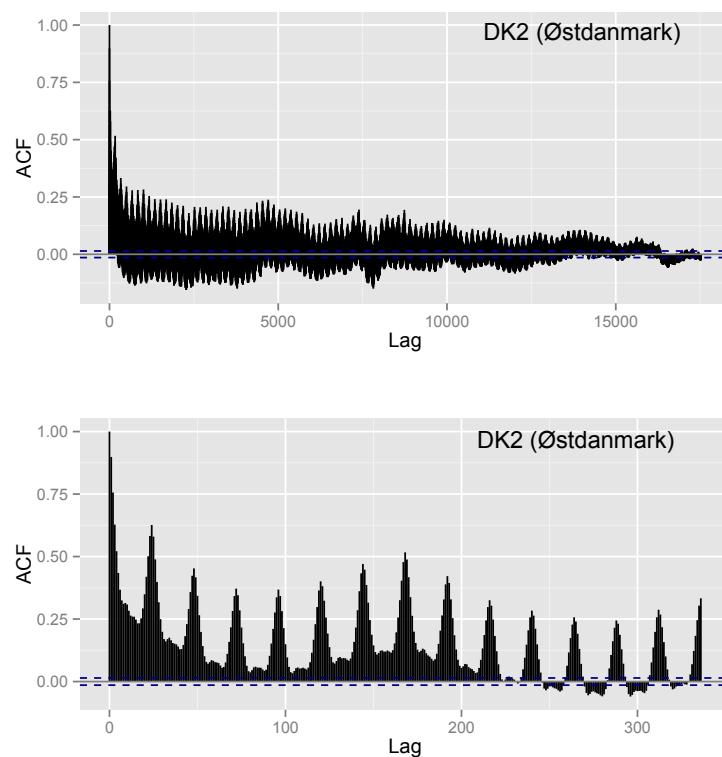
er lav om natten, dvs. tidspunktet på døgnet har betydning for prisen. Dermed er der både daglige, ugentlige og årlige sæsonvariationer i data.



Figur 2.10: Plot af autokovariansfunktionen for NO2 og SE3.



Figur 2.11: Plot af autokovariansfunktionen for SE4 og DK1.



Figur 2.12: Plot af autokovariansfunktionen for DK2.

Kapitel 3

Estimation af model uden regimeskift

I dette og det efterfølgende kapitel vil de forskellige modeller blive estimeret. Implementeringen af modellerne tager udgangspunkt i koderne benyttet af Niels Haldrup og Morten Ørregaard Nielsen i deres artikler [Haldrup og Nielsen, 2006] og [Haldrup og Nielsen, 2005]. Haldrup og Nielsens koder er udført i programmeringsproget Ox, og idet vi ingen erfaring har med dette programmeringssprog, har vi valgt at omarbejde koderne til programmeringssproget R. Først vil det blive forsøgt at estimere en model uden regimeskift, hvorefter en model med regimeskift vil blive forsøgt implementeret.

I det efterfølgende afsnit estimeres modeller uden regimeskift. Der vil blive estimeret en SARFIMA(p, d, q)-proces til at modellere prisen i hvert område, samt en SARFIMA(p, d, q)-proces til at modellere prisdifferensen mellem de to områder i hver forbindelse. Først vil data blive sæsonrenset, hvorefter estimationen af modellerne vil blive foretaget.

3.1 Sæsonrensning af data

Det vil nu blive forsøgt at fjerne sæsonvariationerne i data vha. lineær regression på dummyvariable for hhv. time på døgnet, ugedag samt måned. For at simplificere regressionen er det blevet antaget, at spotprisen på en helligdag opfører sig som spotprisen på en søndag. Derfor betragtes alle dage, der er en helligdag, i det efterfølgende som en søndag. Endvidere er data blevet opdelt således, at det kun er data for årene 2012 og 2013, der renses for sæsonvariationer. Data fra 2014 vil senere blive anvendt til at vurdere forecast. Det er ydermere blevet forsøgt at fjerne sæsonvariationerne ved at anvende trigonometriske funktioner. Disse funktioner resulterede ikke i en bedre forklaringsgrad end den multiple lineære regression på dummyvariablene, så sæsonrensningen er blevet udført vha. multipel lineær regression på dummyvariablene. R-koden til sæsonrensningen fremgår af appendiks D.1.

Der udføres en multipel lineær regression for hvert område, hvor der tages hensyn til hvilke helligdage, der findes i det pågældende område¹. Endvidere er det blevet forsøgt at lave de multiple lineære regressioner både med og uden lineær trend. Dette gøres, da der på forhånd haves en ide om, at priserne vil stige med tiden grundet inflation. Idet regression med trend resulterede i en bedre forklaringsgrad,

¹Se tabel 2.1.

og at trendene i alle fem tilfælde var yderst signifikante, er trendene medtaget i regressionen. Koefficienterne for alle områder fremgår af appendiks C.

Koefficienterne for område NO2 er angivet i tabel C.1. Det fremgår, at der er en positiv koefficient foran trenden, dvs. at priserne stiger over tid med ca. 0.69 øre/time.

På samme vis er der lavet multipel lineær regression på dummyvariablene tilhørende område SE3. Koefficienter er angivet i tabel C.2. Her fremgår det, at koefficienten foran trenden er positiv, hvilket vil sige, at priserne stiger over tid. Prisstigningen for SE3 er ca. 0.6 øre/time, dvs. prisstigningen er mindre end i område NO2.

Koefficienterne tilhørende den multiple lineære regression for område SE4 er angivet i tabel C.3. Det bemærkes her at koefficienten foran trenden er positiv. Priserne stiger med ca. 0.48 øre/time i SE4.

Den multiple lineære regression for området DK1 har koefficienter som angivet i tabel C.4. Trenden i denne regression har en positiv koefficient, og der er tale om en prisstigning på ca. 0.13 øre/time.

For området DK2 har den lineære multible regression koefficienter som angivet i tabel C.5. Også her er koefficienten foran trenden positiv, og prisstigningen er på ca. 0.17 øre/time.

Ud fra koefficienterne foran trendene bemærkes det, at de danske områder har betydeligt lavere prisstigninger end de tre andre områder. Området NO2 har den største prisstigning på ca. 0.7 øre/time. Indledningsvist blev det forsøgt at sæsonrense hele datasættet inklusive data fra begyndelsen af 2014. Dette resulterede i negative koefficienter foran trendene i DK1 og DK2, og betydeligt lavere, men positive, koefficienter foran trendene ved NO2, SE3 og SE4. Disse negative koefficienter ville betyde, at der var tale om en negativ prisudvikling, hvilket er i strid med en forventning om en positiv prisudvikling grundet inflation. Derfor vil der i det efterfølgende afsnit blive lavet en undersøgelse af, hvorfor disse koefficienter ændres, når data fra begyndelsen af 2014 medtages.

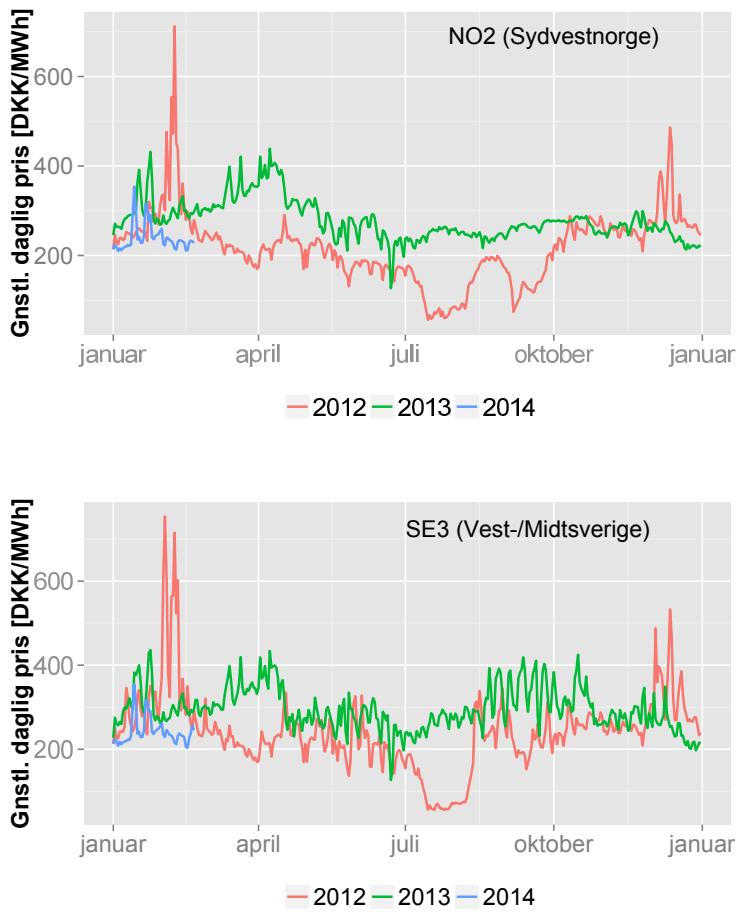
3.1.1 Undersøgelse af den lineære trend

For at få et overblik over hvorledes koefficienterne ændres når data fra 2014 medtages i forbindelse med estimation af sæsonmodellen, angives koefficienterne estimeret hhv. med og uden data fra 2014 i tabel 3.1. Det fremgår af tabellen, at koefficienterne alle bliver betydeligt mindre, hvis data fra 2014 medtages. Dette må betyde, at der må være nogle egenskaber i data for spotpriserne primo 2014, der har en negativ indflydelse på koefficienterne. Derfor vil det blive undersøgt, hvorledes priserne opfører sig primo 2014 sammenlignet med priserne i hhv. 2012 og 2013.

	ekskl. 2014	inkl. 2014
NO2	0.0069	0.0036
SE3	0.0060	0.0025
SE4	0.0048	0.0017
DK1	0.0013	-0.0010
DK2	0.0017	-0.0008

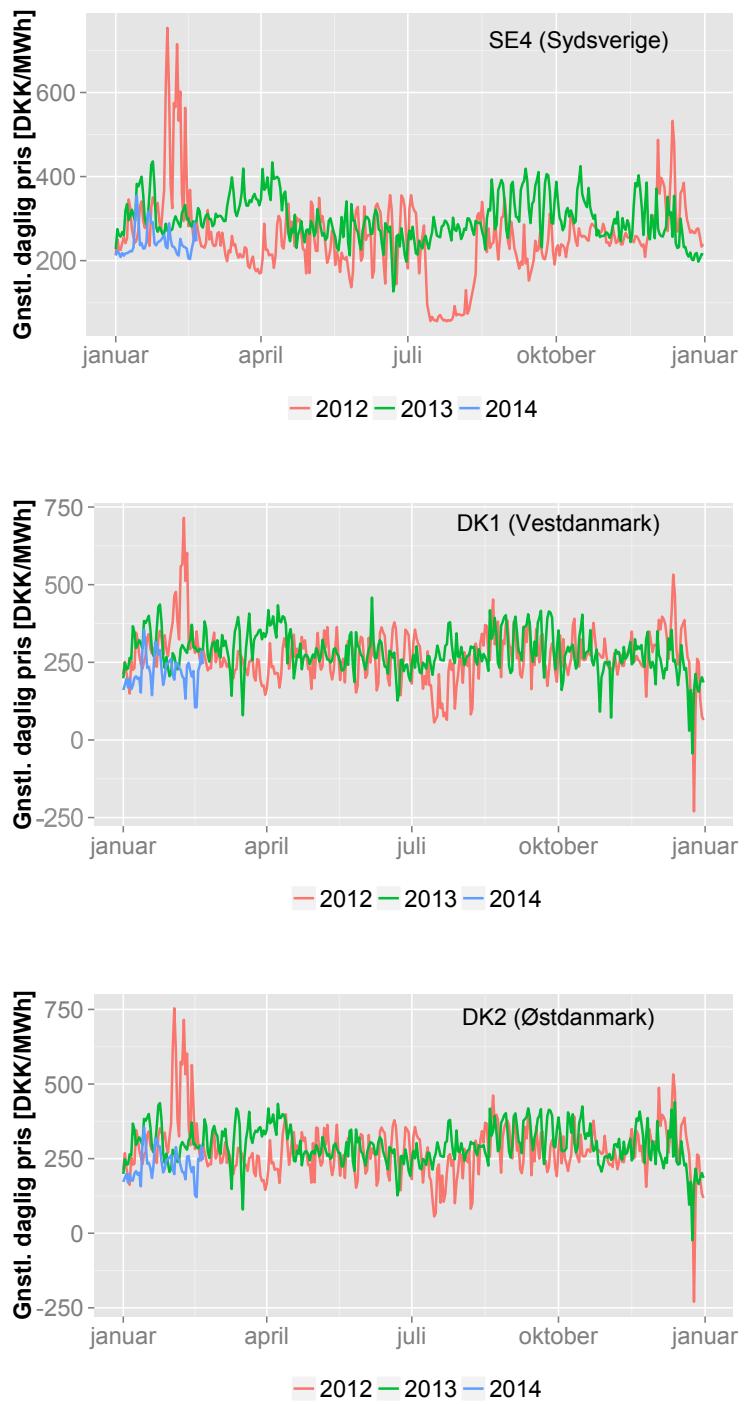
Tabel 3.1: Tabellen angiver koefficienterne foran trendene i de fem forskellige områder. Koefficienterne er estimeret både inklusive og eksklusive data fra 2014.

Figurerne 3.1 og 3.2 illustrerer de gennemsnitlige daglige spotpriser i 2012, 2013 og primo 2014 for hvert område. Her er det tydeligt, at priserne i starten af 2014 er mindre end de to foregående år, når samtlige fem områder betragtes. Endvidere bemærkes det, at for område NO2 ligger de gennemsnitlige daglige spotpriser i år 2013 for det meste over de tilsvarende priser for 2012, og i visse perioder ligger 2013-priserne markant over 2012-niveauet. Tilsvarende tendenser gør sig gældende, når de to svenske områder SE3 og SE4 betragtes. Dog er prisforskellen mellem årene 2012 og 2013 generelt mindre i SE3 og SE4 i forhold til NO2. I de to danske områder DK1 og DK2 er det svært at se, om der skulle være en overordnet prisforskelse mellem de to år 2012 og 2013.



Figur 3.1: Gennemsnitlige daglige priser for Sydvestnorge og Vest-/Midtsverige for hvert af årene 2012, 2013 og primo 2014.

Alle de ovenstående tendenser for de forskellige områder afspejler sig også i størrelsen på koefficienten foran den lineære trend. En mulig årsag til at koefficienterne for den lineære trend i DK1 og DK2 bliver negative, såfremt at data fra primo 2014 medtages, kan være at spotpriserne primo 2014 generelt ligger under priserne i 2012 og 2013.



Figur 3.2: Gennemsnitlige daglige priser for Sydsverige, Vestdanmark og Østdanmark for hvert af årene 2012, 2013 og primo 2014.

3.1.2 Tests af enhedsrødder i det sæsonrensede data

I det følgende undersøges stationaritet vha. forskellige tests på de sæsonrensedde priser i hvert område. De anvendte tests er Dickey-Fuller-, Augmented Dickey-Fuller-, Phillips-Perron- og Kwiatkowski-Phillips-Schmidt-Shin-test. En kort gennemgang af disse tests findes i appendiks B.

Tabel 3.2 viser resultaterne for de forskellige tests for hvert område. DF-, ADF- og PP-testene tager udgangspunkt i den samme nulhypotese om, at tidsrækken har en enhedsrød, og dermed ikke er stationær. Den alternative hypotese er i disse tre tests, at tidsrækken ikke har en enhedsrød, og dermed er stationær. KPSS-testen har derimod en nulhypotese som udtrykker, at tidsrækken er stationær. Den alternative hypotese udtrykker at tidsrækken har en enhedsrød, og dermed ikke er stationær.

Område	Test	Teststørrelse	p-værdi
NO2	DF	-26.1869	< 0.01
	ADF(25)	-7.1384	< 0.01
	PP(14)	-1025.64	< 0.01
	KPSS(30)	1.5004	< 0.01
SE3	DF	-31.63134	< 0.01
	ADF(25)	-7.5955	< 0.01
	PP(14)	-1607.387	< 0.01
	KPSS(30)	0.779	< 0.01
SE4	DF	-32.2432	< 0.01
	ADF(25)	-8.6671	< 0.01
	PP(14)	-1708.715	< 0.01
	KPSS(30)	0.4781	0.04661
DK1	DF	-30.3882	< 0.01
	ADF(25)	-13.6675	< 0.01
	PP(14)	-1596.734	< 0.01
	KPSS(30)	0.2543	> 0.1
DK2	DF	-35.0205	< 0.01
	ADF(25)	-13.0172	< 0.01
	PP(14)	-2117.431	< 0.01
	KPSS(30)	0.1545	> 0.1

Tabel 3.2: Oversigt over resultaterne for de forskellige tests.

Det fremgår af tabel 3.2, at nulhypoteserne for DF-, ADF- og PP-testene i alle områder forkastes, dvs. tidsrækkerne ikke har en enhedsrød (I(1)). Ydermere testes tidsrækkerne for stationaritet vha. KPSS-tests. I KPSS-testene for NO2, SE3 og SE4 forkastes nulhypoteserne, idet p-værdierne er mindre end signifikansniveauet på 5%. Dermed vides det, at tidsrækkerne for områderne NO2, SE3 og SE4 hverken har en enhedsrød (I(1)) eller er stationære. Dette stemmer overens med ideen om long memory. For områderne DK1 og DK2 viser KPSS-testene at nulhypoteserne accepteres, og dermed er tidsrækkerne stationære.

3.1.3 ACF for sæsonrenset data

For at undersøge om sæsonrensningen har haft den ønskede virkning, er ACF for hvert af de fem områder blevet plottet. Det fremgår af ACF-plottene i figurerne 3.3 og 3.4, at sæsonrensningen har haft en tydelig effekt. Dog bemærkes det, at ACF-funktionen ganske vist er blevet udjævnet, men at der fortsat er forholdsvis tydelige peaks for hver 24. time. Derfor vil der i modellen medtages et 24-timers sæsonpolynomium.

Figurerne 3.4 og 3.5 viser, at der er long memory i data for alle områderne. Dette fremgår af plottene, idet autokovariansen ikke er eksponentielt aftagende mod nul.

3.2 Estimation af modeller

Da det er mere kompliceret at estimere parametrene i en ARFIMA-model, der indeholder en MA-del arbejdes der kun med ARFIMA- $(p, d, 0)$ -processer.

Idet ACF-plottene af det sæsonrensede data viste, at der stadig var peaks ved ca. hver 24. time, vil der blive anvendt en ARFIMA (p, d, q) -proces med et 24-timers sæsonpolynomium. Dvs. at der i databehandlingen vil blive anvendt en SARFIMA (p, d, q) -proces på formen:

$$\varphi(B)(1 - aB^{24})(1 - B)^d(x_t - \mu) = \varepsilon_t,$$

hvor $(1 - aB^{24})$ er et 24-timers sæsonpolynomium. Desuden gælder der at $\varepsilon_t \sim nid(0, \sigma^2)$.

For at kunne estimere parametrene i SARFIMA (p, d, q) -modellen opstilles først denne loglikelihoodfunktion, under antagelse om at residualerne $\varepsilon_t \sim nid(0, \sigma^2)$. Loglikelihoodfunktionen er jf. [Haldrup og Nielsen, 2006, p. 14] givet ved

$$L = -\frac{T}{2} \log \left(\frac{1}{T} \sum_{t=1}^T \hat{\varepsilon}_t^2 \right) - \frac{T}{2}(1 + \log(2\pi)),$$

hvor

$$\hat{\varepsilon}_t = \hat{\varphi}(B)(1 - \hat{a}B^{24})(1 - B)^{\hat{d}}(x_t - \hat{\mu}). \quad (3.1)$$

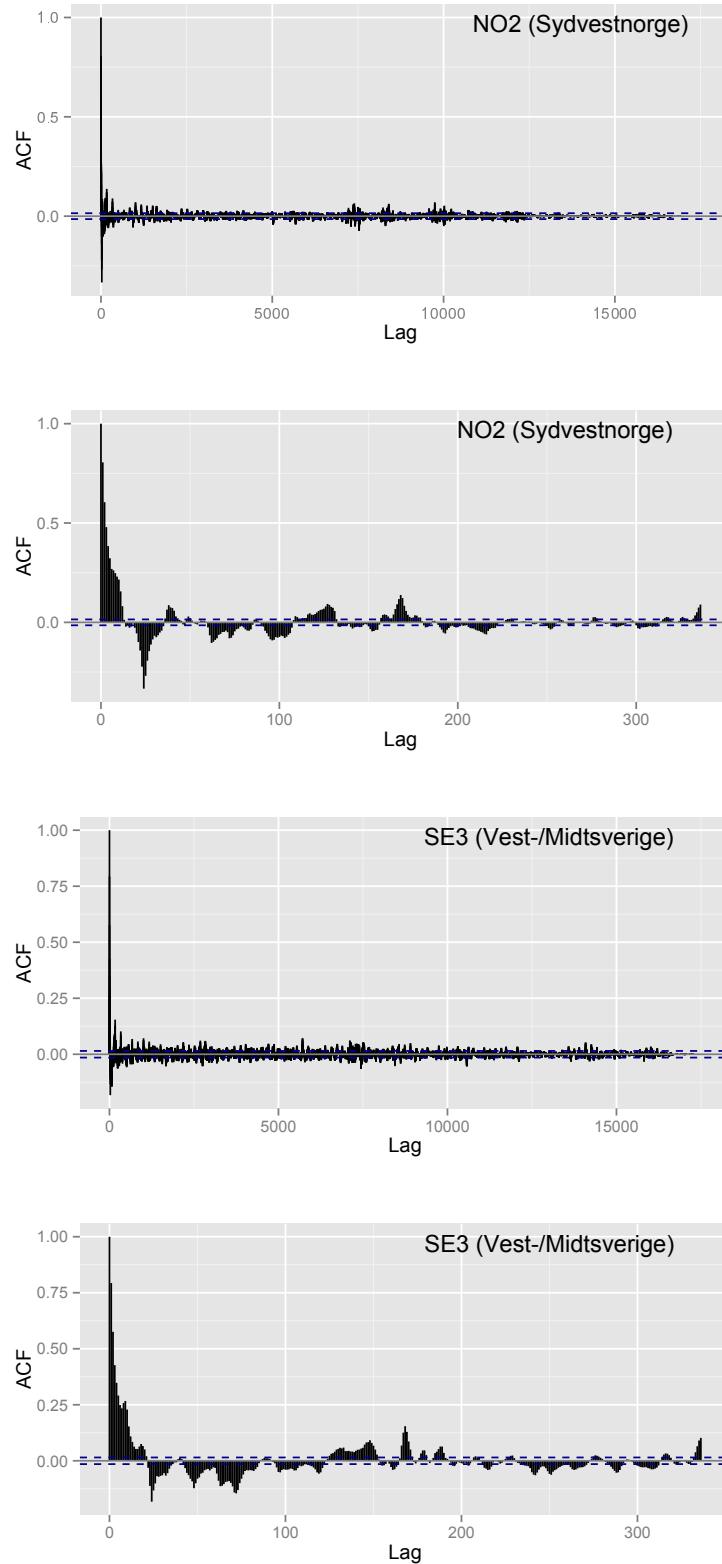
Det ønskes nu at maksimere loglikelihoodfunktionen for at finde ML-estimaterne for parametrene i SARFIMA (p, d, q) -processen. Ved at anvende limited memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS-B)² algoritmen i funktionen `optim` fra R-pakken `stats`, er det muligt at maksimere loglikelihoodfunktionen.

3.2.1 Implementering af loglikelihoodfunktionen

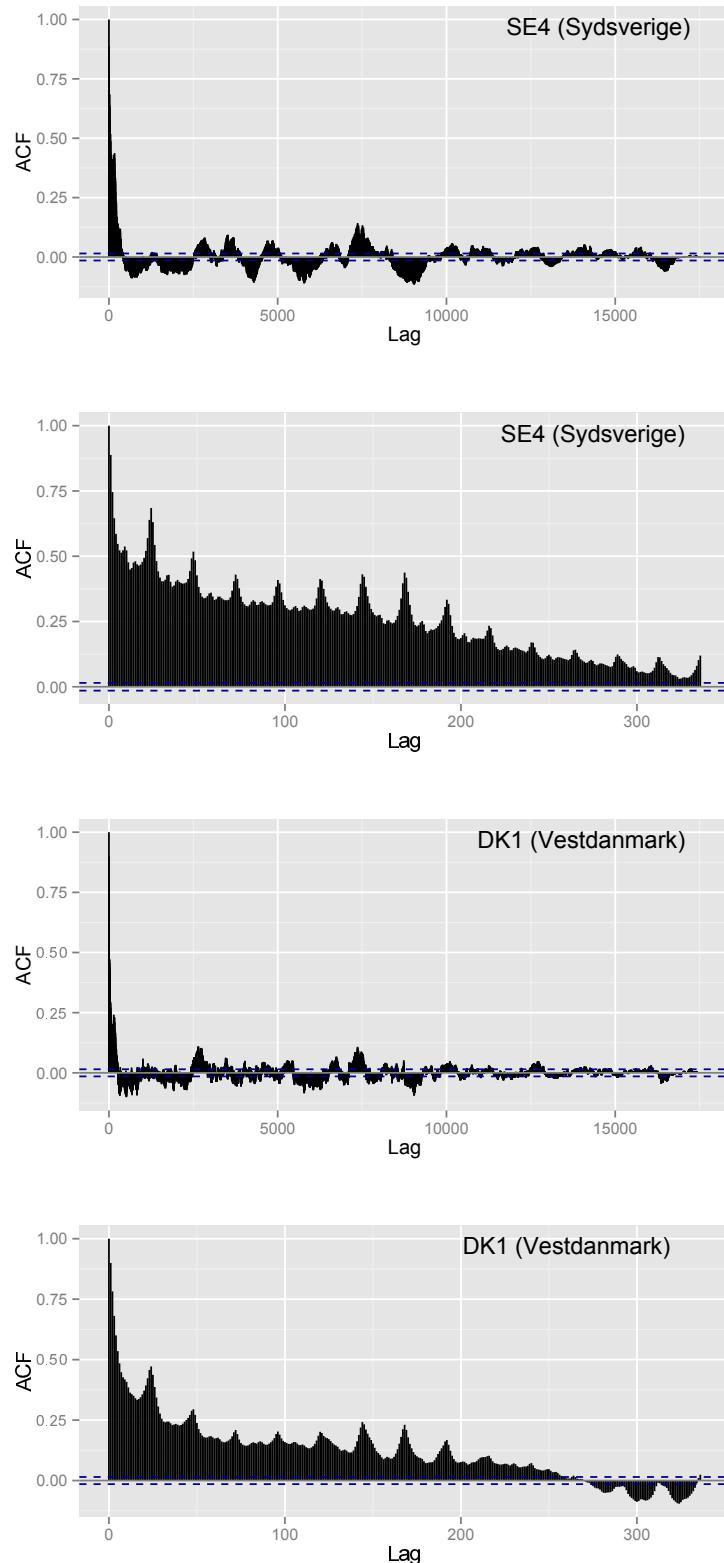
Beregningen af funktionsværdien af loglikelihoodfunktionen er lavet som en funktion i R kaldet `Loglik`. Denne funktion tager som input tidsrækken og en parametervektor indeholdende de p AR-parametre, middelværdien μ , den fraktionelle differensparameter d samt a fra sæsonpolynomiet.

For et givet sæt parameterværdier tager `Loglik`-funktionen først den fraktionelle differens mht. den fraktionelle differensparameter d af data fratrukket middelværdien μ . I Ox-koden har Haldrup og Nielsen anvendt en funktion som hedder `diffpow`, og efter nærmere undersøgelser fandt vi frem til, at funktionen `diffseries` i R-pakken `fracdiff` måtte være en tilsvarende funktion. Det blev dog hurtigt tydeligt, at der er

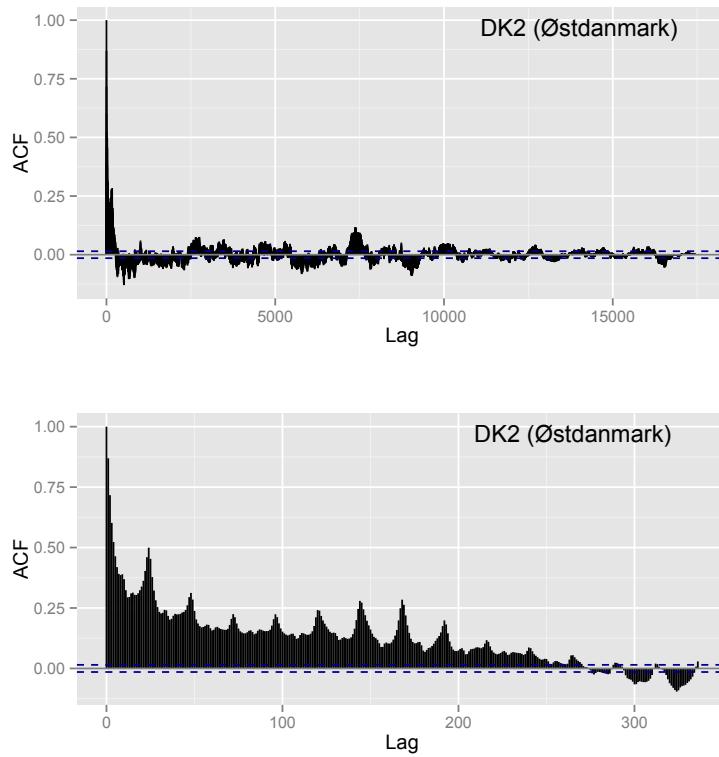
²L-BFGS er en limited udgave af BFGS, som er en Quasi-Newton optimeringsalgoritme. BFGS anvender en approksimation til den inverse Hessematrix til at estimere parametrene. Dvs. BFGS har en $n \times n$ matrix i sin hukommelse, hvor n er antallet af variable i optimeringsproblemet. BFGS er ikke brugbar i situationer med mange variable, da Hessematricen så ville være meget stor. Derfor anvendes L-BFGS i situationer, hvor der haves mange variable. L-BFGS lagrer Hessematricen på komprimeret form, og på denne måde lagrer den kun nogle få vektorer af længde n . L-BFGS husker kun de m sidste opdateringer, hvor m generelt er et lille tal, og ofte er $m < 10$.



Figur 3.3: Plot af autokovariansfunktionen for sæsonrenset data i område NO2 og SE3.



Figur 3.4: Plot af autokovariansfunktionen for sæsonrenset data i område SE4 og DK1.



Figur 3.5: Plot af autokovariansfunktionen for sæsonrenset data i område DK2.

en forskel på de to funktioner. Når vi ændrede på middelværdien i R, havde det ingen indflydelse på de resulterende værdier i den fraktionelt differensede tidsrække, hvormod en ændring af middelværdien i Ox førte til ændrede værdier i den fraktionelt differensede tidsrække. Forskellen mellem de to funktioner består i, at funktionen **diffseries** trækker middelværdien fra data, og derved vil middelværdien, som vi fratrækker, gå ud. Ved at udelade linjen i koden, hvor middelværdien fratrækkes, opnås en funktion, som giver de samme resultater som funktionen **diffpow** i Ox. Denne ”nye“ funktion kan findes i appendiks D.4.

Herefter beregnes residualerne ε_t , som er givet ved ligning (3.1), da disse skal benyttes til at beregne funktionsværdien af loglikelihoodfunktionen. Dette gøres ved at anvende et AR(p)-filter³ på den fraktionelt differensede tidsrække, hvorefter de herefter fremkomne værdier fratrækkes a ganget med de laggede værdier, hvor laglængden er 24. Endeligt opstilles udtrykket for loglikelihoodfunktionen, og **Loglikit**-funktionen returnerer værdien af loglikelihooden.

3.2.2 Estimation af parametrene

Ved at maksimere loglikelihoodfunktionen fås modellens parameterestimater. Der laves derfor en funktion kaldet **ObtainStart**, som tager tidsrækken som input. Koden til **ObtainStart** kan findes i appendiks D.6.

Først opstilles en vektor med startværdier for parametrene i SARFIMA(p, d, q)-processen. Startværdien for parameteren d beregnes i R vha. funktionen **fracdiff**,

³Koden for dette AR-filter kan findes i appendiks D.5.

som ligger i R-pakken af samme navn. Middelværdien μ sættes lig den empiriske middelværdi. Startværdierne for parametrene i AR-delen bestemmes ud fra den fraktionelle differensede serie, hvorfra den empiriske middelværdi er fratrukket, og d er givet ved den fornævnte startværdi for d . Til sidst er startværdien af a sat lig med 0.15.

Vha. funktionen `optim` i R-pakken `stats` maksimeres loglikelihoodfunktionen. Herefter returnerer `ObtainStart`-funktionen parameterestimaterne og den tilhørende funktionsværdi af loglikelihoodfunktionen.

Idet det ønskes at supplere parameterestimaterne med standard errors, er det nødvendigt at finde Hessematrixen tilhørende loglikelihoodfunktionen. Dette gøres idet der haves følgende sammenhæng; Variansen-kovariansmatrixen for et ML-estimat $\hat{\theta}$ kan findes vha. den inverse observerede informationsmatrix

$$\text{var}(\hat{\theta}) = [I(\hat{\theta})]^{-1}.$$

Den observerede informationsmatrix er lig med den negative Hessematrix evaluert i MLE

$$I(\hat{\theta}) = -H(\hat{\theta}).$$

Hessematrixen er en matrix med anden ordens afledede af loglikelihoodfunktionen mht. parametrene

$$H(\theta) = \frac{\partial^2 L(\theta)}{\partial \theta \partial \theta^\top}.$$

Dermed må der gælde, at

$$\begin{aligned} \text{var}(\hat{\theta}) &= [I(\hat{\theta})]^{-1} \\ &= [-H(\hat{\theta})]^{-1} \\ &= \left[-\frac{\partial^2 L(\hat{\theta})}{\partial \hat{\theta} \partial \hat{\theta}^\top} \right]^{-1}. \end{aligned} \tag{3.2}$$

Så er standard errors for $\hat{\theta}$ givet ved kvadratroden af elementerne i diagonalen i varians-kovariansmatrixen i ligning (3.2).

Koden køres med tidsrækkerne for hver af de fem områder; NO2, SE3, SE4, DK1 og DK2 samt for prisdifferensen mellem de to områder i hver forbindelse.

3.2.3 Valg af model

For at kunne vælge hvilke ordener SARFIMA(p, d, q)-processerne skal have, er koden blevet kørt for følgende ordener: $(1, d, 0)$, $(2, d, 0)$, $(3, d, 0)$ og $(4, d, 0)$. Kodens kompleksitet medfører at den er meget tidskrævende at køre, og derfor har vi været nødt til at begrænse os til modeller med relativt få parametre.

For hver forbindelse haves der tre tidsrækker; en tidsrække for den sæsonrensede pris i det ene område og i det andet område, samt en tidsrække for differensen mellem de to sæsonrensede priser. Ved at betragte RMSE, AIC samt residualerne for hver model er det muligt at vurdere, hvilken model der er bedst egnet for hver model i de fem forbindelser.

RMSE

Root-mean-square error (RMSE) mÅler afvigelserne mellem de observerede v rdier og de v rdier der f s vha. modellen. RMSE er givet ved f lgende udtryk:

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (\hat{y}_t - y_t)^2}{n}}.$$

Tabel 3.3 viser de beregnede RMSE, hvor de r de tal viser hvilken model, der er at foretr kke. Det fremg r af tabellen, at det er SARFIMA(4,d,0)-modellerne, der resulterer i de laveste RMSE. Dette resultat er forventligt, idet en model med flere parametre vil give et bedre fit.

		AR(1)	AR(2)	AR(3)	AR(4)
NO2 - DK1	NO2	19.53675	19.29541	19.28524	19.25944
	DK1	36.86394	36.84496	36.84187	36.84176
	Diff.	35.37978	35.37786	35.37443	35.37073
DK1 - SE3	DK1	36.86394	36.84496	36.84187	36.84176
	SE3	29.72542	29.37419	29.36894	29.36636
	Diff.	37.05499	37.04735	37.03907	37.03058
DK1 - DK2	DK1	36.86394	36.84496	36.84187	36.84176
	DK2	43.73022	43.61806	43.61764	43.61247
	Diff.	30.59929	30.47585	30.4794	30.46824
DK2 - SE4	DK2	43.73022	43.61806	43.61764	43.61247
	SE4	33.38904	33.11292	33.10962	33.10888
	Diff.	30.61958	30.61745	30.61452	30.60433
SE4 - SE3	SE4	33.38904	33.11292	33.10962	33.10888
	SE3	29.72542	29.37419	29.3694	29.36636
	Diff.	16.80492	16.75421	16.75163	16.74629

Tabel 3.3: RMSE for hvert omr de i hver forbindelse samt for prisdifferensen (modeller uden regimeskift).

Idet RMSE ikke tager hensyn til overparametrisering, vil AIC-v rdierne nu blive betragtet.

AIC

Akaikes informationskriterium (AIC) er et m l for, hvor egnet en statistisk model er for et givet datas t. Kriteriet straffer overparametrisering, idet antallet af parametre k indg r således, at en ekstra parameter vil medf re en st rre AIC-v rdi. AIC er givet ved f lgende udtryk:

$$AIC = 2k - 2\log(L),$$

hvor k er antal parametre, der skal estimeres i modellen, og L er den maksimerede v rdi af likelihoodfunktionen tilh rende modellen. Den foretrukne model for et datas t, vil v re den model med den mindste AIC-v rdi.

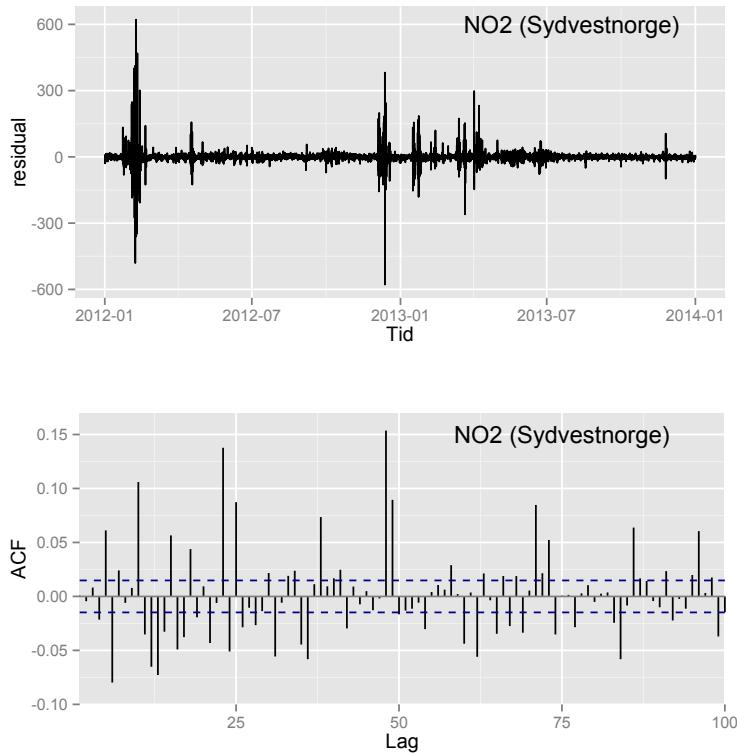
Nedenst ende tabel 3.4 viser de forskellige modellers AIC-v rdi. De r de tal i hver r kke angiver, hvilken model, der har den mindste AIC-v rdi, og dermed hvilken model, der er at foretr kke.

For at unders ge hvorledes residualerne tilh rende de forskellige modeller opf rer sig, plottes ACF for residualerne samt et plot af residualerne over tid. Form let med dette er at unders ge, om residualerne har de samme egenskaber som hvid st j. Idet

		AR(1)	AR(2)	AR(3)	AR(4)
NO2 - DK1	NO2	153885.7	153452.1	153435.6	153390.7
	DK1	176135.1	176119.1	176118.2	176120.0
	Diff.	174695.1	174695.2	174693.8	174692.2
DK1 - SE3	DK1	176135.1	176119.1	176118.2	176120.0
	SE3	168593.0	168178.5	168174.2	168173.1
	Diff.	176316.3	176311.0	176305.2	176299.2
DK1 - DK2	DK1	176135.1	176119.1	176118.2	176120.0
	DK2	182120.5	182032.5	182034.2	182032.0
	Diff.	169608.3	169468.7	169469.6	169463.9
DK2 - SE4	DK2	182120.5	182032.5	182034.2	182032.0
	SE4	172665.8	172376.8	172375.3	172376.5
	Diff.	169631.5	169631.1	169629.8	169620.1
SE4 - SE3	SE4	172665.8	172376.8	172375.3	172376.5
	SE3	168593.0	168178.5	168174.2	168173.1
	Diff.	148607.5	148503.6	148500.2	148491.0

Tabel 3.4: AIC-værdier for hvert område i hver forbindelse samt for prisdifferensen (modeller uden regimeskift).

der observeres de samme tendenser for residualerne på tværs af alle modeller, har vi valgt at vise plots af ACF og residualerne over tid for kun en model.



Figur 3.6: Øverste plot viser residualerne for SARFIMA(4,d,0)-modellen for priserne i NO2 plottet over tid. Nederste plot viser ACF for residualer.

Figur 3.6 afbilder to plots af residualerne for SARFIMA(4,d,0)-modellen for de sæsonrensede priser fra område NO2. Det øverste plot viser residualerne plottet over tid. Det fremgår heraf, at residualerne tydeligvis ikke har konstant varians, og dermed har residualerne ikke de samme egenskaber som hvid støj. Dette kan muligvis

skyldes, at modellen ikke fanger de såkaldte spikes i tidsrækken for elprisen. Det nederste plot viser ACF for residualerne. Her fremgår det, at residualerne ikke er ukorrelerede. Disse problemer forekommer i residualerne for samtlige af de afprøvede modeller. Derfor vælges modellerne ud fra AIC, som jf. [Beran, 1994, p. 11] er det optimale at benytte, når formålet er at forecaste. Dermed udvælges de modeller, hvis AIC-værdi er anført i rødt i tabel 3.4.

De endelige modeller og deres parameterestimater fremgår af tabel 3.5.

		$\hat{\varphi}_1$	$\hat{\varphi}_2$	$\hat{\varphi}_3$	$\hat{\varphi}_4$	d	$\hat{\mu}$	\hat{a}	$\hat{\sigma}$
NO2 - DK1	NO2	0.589 (0.017)	-0.184 (0.009)	0.058 (0.009)	-0.055 (0.008)	0.434 (0.015)	0.000 (16.191)	0.302 (0.008)	19.259 -
	DK1	0.692 (0.023)	-0.029 (0.010)	-0.007 (0.008)	-	0.293 (0.022)	0.000 (15.293)	0.229 (0.007)	36.842 -
	Diff.	0.737 (0.026)	0.005 (0.011)	0.002 (0.009)	-0.015 (0.008)	0.185 (0.025)	0.000 (6.522)	0.158 (0.008)	35.371 -
DK1 - SE3	DK1	0.692 (0.023)	-0.029 (0.010)	-0.007 (0.008)	-	0.293 (0.022)	0.000 (15.293)	0.229 (0.007)	36.842 -
	SE3	0.610 (0.018)	-0.142 (0.009)	-0.027 (0.009)	0.014 (0.008)	0.360 (0.017)	0.000 (18.683)	0.459 (0.007)	29.366 -
	Diff.	0.731 (0.026)	-0.038 (0.011)	0.004 (0.009)	0.022 (0.008)	0.162 (0.025)	0.000 (6.121)	0.272 (0.007)	37.031 -
DK1 - DK2	DK1	0.692 (0.023)	-0.029 (0.010)	-0.007 (0.008)	-	0.293 (0.022)	0.000 (15.293)	0.229 (0.007)	36.842 -
	DK2	0.684 (0.023)	-0.077 (0.010)	-0.005 (0.009)	0.016 (0.008)	0.267 (0.022)	0.000 (14.572)	0.311 (0.007)	43.612 -
	Diff.	0.542 (0.028)	-0.093 (0.009)	0.001 (0.009)	0.026 (0.009)	0.225 (0.027)	0.000 (4.873)	0.282 (0.007)	30.468 -
DK2 - SE4	DK2	0.684 (0.023)	-0.077 (0.010)	-0.005 (0.009)	0.016 (0.008)	0.267 (0.022)	0.000 (14.572)	0.311 (0.007)	43.612 -
	SE4	0.604 (0.018)	-0.121 (0.009)	-0.016 (0.009)	-	0.360 (0.016)	0.000 (20.169)	0.421 (0.007)	33.110 -
	Diff.	0.757 (0.021)	-0.002 (0.011)	0.006 (0.009)	-0.026 (0.008)	0.163 (0.020)	0.000 (4.723)	0.163 (0.008)	30.604 -
SE4 - SE3	SE4	0.604 (0.018)	-0.121 (0.009)	-0.016 (0.009)	-	0.360 (0.016)	0.000 (20.169)	0.421 (0.007)	33.110 -
	SE3	0.610 (0.018)	-0.142 (0.009)	-0.027 (0.009)	0.014 (0.008)	0.360 (0.017)	0.000 (18.683)	0.459 (0.007)	29.366 -
	Diff.	0.620 (0.020)	-0.089 (0.009)	0.032 (0.009)	-0.027 (0.008)	0.304 (0.019)	0.000 (5.582)	0.216 (0.008)	16.746 -

Tabel 3.5: Oversigt over de valgte modeller uden regimeskift, estimerede parametre inklusiv standard errors.

Kapitel 4

Estimation af modeller med regimeskift

I dette kapitel vil det blive forsøgt at estimere en model for hver forbindelse mellem to områder. Dette vil blive gjort vha. af en såkaldt regimeskiftende ARFIMA(p, d, q)-proces. For hver forbindelse estimeres tre SARFIMA(p, d, q)-modeller; en model for hver af de to områder samt en model for prisdifferensen. Der vil blive arbejdet med to regimer i modellerne, hvor regime 0 betegner observationer hvor priserne er ens på tværs af de to områder i en forbindelse, og regime 1 betegner hvor priserne er forskellige. I det efterfølgende afsnit sæsonrensnes data.

4.1 Sæsonrensning af data

Til sæsonrensningen af data opstilles der indledningsvist en regimevektor, som består af 1-taller for observationer under regime 1 og 0'er for observationer under regime 0.

Først betragtes hele datasættet for et enkelt område, der opstilles dummyvariable for hhv. time, ugedag og måned, hvor der alle steder tages hensyn til, om observationen tilhører regime 1 eller regime 0. På denne måde opnås der dobbelt så mange dummyvariable, som der blev anvendt i afsnit 3.1, da der fx. skal være en dummy for måneden februar for både regime 1 og regime 0. Vha. kommandoen `1m` i R laves der lineær regression på dummyvariablene, hvorefter residualernes findes. Det er residualer, der udgør det sæsonrensede data. Dette gøres for begge områder i en forbindelse.

Til sidst betragtes hele datasættet for begge områder. Differensen mellem priserne i de to områder beregnes, og det er disse differenser, der også vil blive sæsonrenset. Idet differensen er nul når regime 0 betragtes, vil alle koefficienter foran en dummyvariable tilhørende regime 0 pr. definition være nul. Derfor udelades alle observationer tilhørende regime 0. Der opstilles dummyvariable for hhv. time, ugedag og måned, og der laves igen lineær regression og residualerne findes.

I appendiks C er koefficienterne tilhørende de forskellige dummyvariable angivet. Tabellerne C.6 - C.10 indeholder koefficienterne for hele datasættet for hvert område, og tabellerne C.11 - C.15 viser koefficienterne tilhørende dummyvariablene når prisdifferensen mellem to områder betragtes. Koden anvendt til denne sæsonrensning kan findes i appendiks D.2.

4.2 Estimation af modeller

I dette afsnit estimeres modellerne med regimeskift for hver forbindelse. Som nævnt i sæsonrensningen af data, er der indledningsvist lavet en regimevektor, som fortæller om priserne i de to områder, der betragtes, er identiske eller forskellige. I den regimeskiftende model estimeres et sæt parametre for SARFIMA-modellen for hvert regime, men for prisdifferensen haves kun ét sæt parametre, idet prisdifferensen pr. definition er nul i regime 0.

SARFIMA(p, d, q)-processen med regimeskift er på følgende form:

$$\varphi_{s_t}(B)(1 - a_{s_t}B^{24})(1 - B)^{d_{s_t}}(x_t - \mu_{s_t}) = \varepsilon_{s_t, t},$$

hvor $\varepsilon_{s_t, t} \sim nid(0, \sigma_{s_t}^2)$, $(1 - a_{s_t}B^{24})$ er et 24-timers polynomium. $s_t = 0, 1$, hvor 0 refererer til de parametre, der er blevet estimeret for data tilhørende regime 0, og 1 refererer til de parametre, der er blevet estimeret for data tilhørende regime 1.

Transitionsmatrix

Transitionsmatricen fortæller hvad sandsynligheden er for at komme fra en tilstand til en anden. I dette tilfælde viser transitionsmatricen hvad sandsynligheden er for at være i regime 0 og blive der, for at være i regime 0 og derefter komme over i regime 1, for at være i regime 1 og komme over i regime 0 og at være i regime 1 og blive der. Den generelle transitionsmatrix med 2 regimer ser ud som følger:

$$P = \begin{bmatrix} p_{00} & 1 - p_{00} \\ 1 - p_{11} & p_{11} \end{bmatrix}.$$

Ved hjælp af formlerne i ligning (4.1) kan p_{00} og p_{11} estimeres ud fra det ikke-sæsonrensede data, idet regimerne er observerbare.

$$\hat{p}_{00} = \frac{n_{00}}{n_{00} + n_{01}} \quad \text{og} \quad \hat{p}_{11} = \frac{n_{11}}{n_{10} + n_{11}}, \quad (4.1)$$

hvor n_{ij} er antal gange, hvor regime i følger regime j .

Ved at anvende formlerne i ligning (4.1) estimeres transitionsmatricerne for hver af de fem forbindelser. Disse matricer er vist i ligning (4.2).

$$\begin{aligned} P_{(\text{NO2-DK1})} &= \begin{bmatrix} 0.86370431 & 0.1362957 \\ 0.09870104 & 0.9012990 \end{bmatrix} \\ P_{(\text{DK1-SE3})} &= \begin{bmatrix} 0.8893052 & 0.1106948 \\ 0.1429319 & 0.8570681 \end{bmatrix} \\ P_{(\text{DK1-DK2})} &= \begin{bmatrix} 0.9363899 & 0.06361011 \\ 0.2397167 & 0.76028330 \end{bmatrix} \\ P_{(\text{DK2-SE4})} &= \begin{bmatrix} 0.9403917 & 0.05960827 \\ 0.1697782 & 0.83022183 \end{bmatrix} \\ P_{(\text{SE4-SE3})} &= \begin{bmatrix} 0.9845608 & 0.01543921 \\ 0.1694102 & 0.83058985 \end{bmatrix} \end{aligned} \quad (4.2)$$

Koden til disse matricer fremgår af appendiks D.3.

4.2.1 Maksimering af loglikelihoodfunktionen

I forbindelse med estimationen af parametrene i de regimeskiftende SARFIMA(p, d, q)-modeller opstilles loglikelihoodfunktionen, under antagelse om at residualerne $\varepsilon_{s_t, t} \sim nid(0, \sigma^2)$. Loglikelihoodfunktionen er jf. [Haldrup og Nielsen, 2006, p. 14] givet ved

$$L = -\frac{T}{2} \log \left(\frac{1}{T} \sum_{t=1}^T \hat{\varepsilon}_{s_t, t}^2 \right) - \frac{T}{2}(1 + \log(2\pi)),$$

hvor

$$\hat{\varepsilon}_{s_t, t} = \hat{\varphi}_{s_t}(B)(1 - \hat{a}_{s_t} B^{24})(1 - B)^{\hat{d}_{s_t}}(x_t - \hat{\mu}_{s_t}).$$

Endvidere haves det, at alle regimer er observerbare, og dette betyder, at residualerne $\hat{\varepsilon}_{s_t, t}$ kan beregnes for et givet sæt parameterværdier, og det er dermed muligt at maksimere loglikelihoodfunktionen.

Parameterestimationen for modellerne med regimeskift kan opdeles i to trin. Første trin er at estimere parametrene i en ikke-regimeskiftende model. Dette gøres ved at anvende samme fremgangsmåde, som beskrevet i afsnit 3.2. De resulterende parameterestimater vil blive benyttet som startværdier i andet trin.

Andet trin består i at estimere parametrene i modellerne med regimeskift. Dvs. der skal estimeres et sæt parametre for hvert regime. Derfor opstilles der nu en ny funktion **Loglik**, som beregner funktionsværdien af loglikelihoodfunktion, og der opstilles en funktion med navnet **ModelEstimates**.

Først opstilles **Loglik**-funktion. Denne funktion tager som input tidsrækken og parametervektoren $\boldsymbol{\eta} = [\boldsymbol{\eta}_1 \ \ \boldsymbol{\eta}_0]$, hvor

$$\boldsymbol{\eta}_1 = [\varphi_{1,1} \ \ \dots \ \ \varphi_{1,p} \ \ d_1 \ \ \mu_1 \ \ a_1]^\top,$$

og

$$\boldsymbol{\eta}_0 = [\varphi_{0,1} \ \ \dots \ \ \varphi_{0,p} \ \ d_0 \ \ \mu_0 \ \ a_0]^\top.$$

Således haves en vektor, hvor de første $p+3$ indgange udgør parametrene i regime 1 og de næste $p+3$ indgange udgør parametrene i regime 0.

For et givet sæt parameterværdier tilhørende regime 1, tager **Loglik**-funktionen først den fraktionelle differens mht. den fraktionelle differensparameter d_1 af data fratrukket middelværdien μ_1 . Herefter beregnes residualerne $\hat{\varepsilon}_{1,t}$. Dette gøres ved at anvende et AR(p)-filter, hvor AR-parametrene tilhører regime 1, på den fraktionelt differen-sede tidsrække, hvorefter de heraf fremkomne værdier fratrækkes a_1 ganget med de laggede værdier, hvor laglængden er 24.

Samme fremgangsmåde anvendes for et givet sæt parameterværdier tilhørende regime 0 med parametrene d_0 , μ_0 og a_0 . Dette resulterer i residualvektoren $\hat{\varepsilon}_{0,t}$.

Herefter laves en ny residualvektor $\hat{\varepsilon}_t$, for hvilken der gælder:

$$\hat{\varepsilon}_t = \sum_{j=0}^1 \hat{\varepsilon}_{j,t}^2 1(s_t = j),$$

hvor $1(\cdot)$ er indikatorfunktionen. Residualvektor $\hat{\varepsilon}_t$ fås ved at gange $\hat{\varepsilon}_{1,t}$ med regimevektoren og lægge $\hat{\varepsilon}_{0,t}$ ganget med (1-regimevektoren) til. På denne måde opnås residualerne for den regimeskiftende SARFIMA-model.

Til sidst opstilles loglikelihoodfunktionen, hvorefter **Loglik**-funktionen returnerer funktionsværdien af loglikelihoodfunktionen.

Når prisdifferensen mellem to forbundne områder betragtes, vil parametrene i regime 0 pr. definition være lig med nul. Derfor opstilles en ny funktion kaldet **Loglik.diff**, som beregner funktionsværdien af loglikelihoodfunktion, når input er prisdifferensen.

Funktionen **ModelEstimates** maksimerer loglikelihoodfunktionen vha. funktionen **optim** i R. Funktionen **ModelEstimates** returnerer herefter parameterestimaterne og den tilhørende funktionsværdi af loglikelihoodfunktionen. Ydermere returneres standard errors for hver parameterestimat vha. en approksimeret Hessematrix. Koden køres med tidsrækkerne for hver af de fem forbindelser; NO2-DK1, DK1-SE3, DK1-DK2, DK2-SE4 og SE4-SE3.

Igen har det været nødvendigt at opstille en separat funktion (**ModelEstimates.diff**) til at maksimere loglikelihoodfunktionen for prisdifferensen. **ModelEstimates.diff**-funktionen returnerer herefter parameterestimaterne for regime 1.

De anvendte koder fremgår af appendiks D.6-D.8.

4.2.2 Valg af modeller

Som tidligere nævnt er modellerne meget avancerede, hvilket medfører, at koden er meget tidskrævende. Derfor er der blevet estimeret modeller med relativt få parametre. Antallet af parametre er meget større, når der haves en regimeskiftende model, idet der for hvert af de to regimer haves samme antal parametre, som i den ikke-regimeskiftende model.

SARFIMA-modellerne med ordenerne $(1,d,0), (2,d,0), (3,d,0)$ og $(4,d,0)$ er først blevet estimeret. Dette gøres som tidligere nævnt for hvert område i en forbindelse, samt for prisdifferensen mellem de to områder. For de regimeskiftende modeller har modelestimationen ofte resulteret i at den inverse Hessematrix har negative værdier i diagonalen, hvilket medfører, at det ikke er muligt at beregne standard errors. Disse problemer med Hessematricen er en indikation for, at optimeringsalgoritmen ikke har fundet de optimale parameterestimater. Dette kan skyldes, at der, grundet mange parametre, haves en meget flad likelihoodfunktion, og at det derfor kan være svært for optimeringsalgoritmen at konvergere. Derfor er det ydermere blevet forsøgt, at estimere samtlige modeller for hver forbindelse, men hvor middelværdien μ i stedet for sættes lig sample mean. Dette reducerer antallet af parametre, der skal estimeres i modellen. Efterfølgende har det været muligt at køre modelestimationerne uden de fornævnte problemer med Hessematricen. Der vil i den videre databehandling derfor blive arbejdet med modellerne, hvor middelværdien sættes lig med sample mean.

For at kunne udvælge de bedst egnede modeller beregnes RMSE for samtlige modeller med regimeskift. RMSE er angivet i tabel 4.1. Igen er de mindste værdier anført i rødt, således at det tydeligt fremgår hvilke modeller, der ift. RMSE, er at foretrække. Ligesom for modellerne uden regimeskift, er det de modeller med flest parametre, der giver mindst RMSE.

Efter at have beregnet RMSE, beregnes AIC-værdierne for hver enkelt model med regimeskift. Disse værdier er anført i tabel 4.2, hvor de mindste værdier er angivet i rødt. Derved fremgår det tydeligt ud fra tabellen, hvilke modeller der er at foretrække.

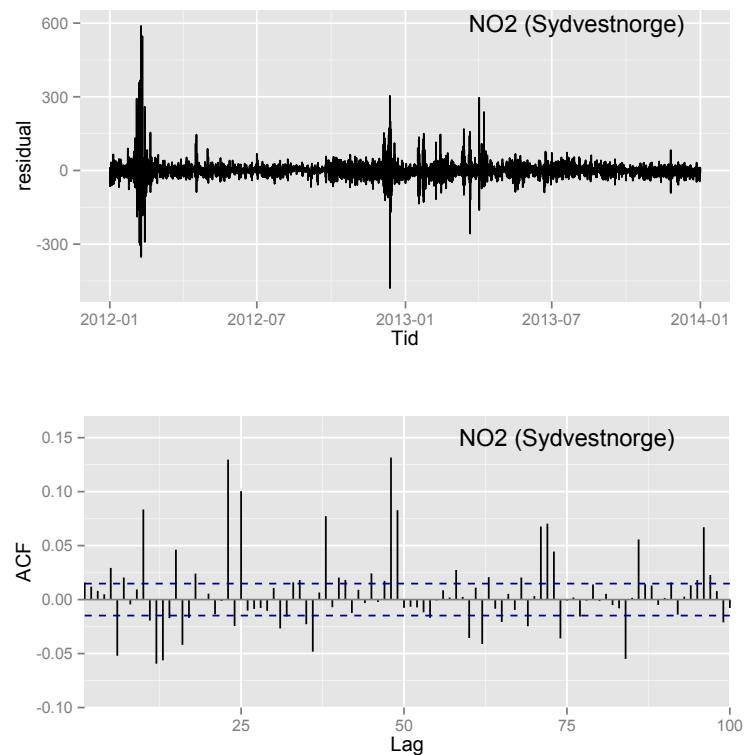
		AR(1)	AR(2)	AR(3)	AR(4)
NO2 - DK1		20.24079	20.11423	20.11111	20.09316
DK1 - SE3		38.16121	38.10941	38.10617	38.10481
DK1 - DK2		33.87504	33.82976	33.82898	33.82685
DK2 - SE4		38.3828	38.30401	38.30144	38.29992
DK2 - SE4		31.59193	31.31679	31.27067	31.27058
SE4 - SE3		36.42243	36.4198	36.41498	36.39999
DK1 - DK2		37.39539	37.37629	37.3637	37.35623
DK1 - DK2		45.03278	44.84932	44.84419	44.78256
DK1 - DK2		28.39798	28.24965	28.20434	28.2031
DK2 - SE4		44.27882	44.13174	44.12645	44.12385
DK2 - SE4		35.99311	35.78901	35.77937	35.7763
SE4 - SE3		28.62742	28.61841	28.61463	28.61448
SE4 - SE3		32.80633	32.45188	32.44713	32.43531
SE4 - SE3		29.8419	29.54584	29.53546	29.53131
SE4 - SE3		13.13215	13.07989	13.04239	13.01292

Tabel 4.1: RMSE for hvert område i hver forbindelse samt for prisdifferensen (modeller med regimeskift).

		AR(1)	AR(2)	AR(3)	AR(4)
NO2 - DK1		155130.3	154914.5	154913	154885.7
DK1 - SE3		177351.1	177307.5	177308.5	177311.3
DK1 - DK2		173170.1	173125.3	173126.5	173126.3
DK2 - SE4		177554.0	177486.0	177487.6	177490.2
DK2 - SE4		170731.0	170428.5	170380.9	170384.8
DK1 - DK2		175710.9	175705.7	175707.7	175695.3
DK1 - DK2		176640.7	176626.8	176619.0	176616.0
DK1 - DK2		183153.0	183014.0	183014.0	182969.8
DK2 - SE4		166990.1	166808.6	166754.4	166754.8
DK2 - SE4		182561.4	182448.8	182448.6	182450.5
DK2 - SE4		175301.4	175106.1	175100.7	175101.7
DK2 - SE4		167272.1	167263.1	167260.4	167262.3
SE4 - SE3		172052.8	171676.1	171675.1	171666.2
SE4 - SE3		168734.0	168388.7	168380.3	168379.4
SE4 - SE3		139963.8	139826.1	139727.5	139650.2

Tabel 4.2: AIC-værdier for hvert område i hver forbindelse samt for prisdifferensen (modeller med regimeskift).

Figur 4.1 viser residualerne fra SARFIMA(4,d,0)-modellen for område NO2. Det øverste afbilder residualerne over tid. Ligesom i tilfældet med de ikke-regimeskiftende modeller fremgår det, at residualerne ikke har konstant varians, og ikke opfører sig som hvid støj. Derfor vælges det også her, at vælge modeller ud fra AIC-værdierne. De valgte modeller med parameterestimater fremgår af tabel 4.3.



Figur 4.1: Øverste plot viser residualerne for SARFIMA(4,d,0)-modellen med regimeskift for priserne i NO2 plottet over tid. Nederste plot viser ACF for residualer.

Kapitel 5

Forecast af elspotpriserne ved brug af modellerne uden regimeskift

I dette kapitel vil det blive forsøgt at foreaste elpriserne 24 timer frem i tiden ved at anvende de SARFIMA(p, d, q)-modeller, som blev valgt i afsnit 3.2.3.

De sæsonrensede elpriser forecastes vha. Monte Carlo simuleringer med brug af de estimerede SARFIMA(p, d, q)-modeller. Da de sæsonrensede elpriser umiddelbart kan være svære at fortolke, vil det være mere interessant at kunne foreaste de faktiske elpriser. Til dette formål anvendes sæsonfunktionen, og ud fra de respektive fremtidige input for ugedag, måned, samt time på døgnet, kan vi bestemme de fremtidige værdier af sæsonfunktionen.

5.1 Implementering af forecast

Der forecastes $k = 24$ skridt ud i fremtiden, dvs. et døgn frem, og der udføres $M = 1000$ Monte Carlo simuleringer, dvs. i forbindelse med hvert prisforecast udføres 1000 MC-stier. For hver af de fem forbindelser udføres et forecast for hvert område, samt for prisdifferensen mellem disse to områder.

Først sammensættes en parametervektor $\hat{\eta} = [\hat{\varphi}_1 \dots \hat{\varphi}_p \hat{d} \hat{\mu} \hat{a}]^\top$ bestående af de estimerede parameterværdier for den valgte SARFIMA-model.

For at opnå de forecastede værdier, føres de n in-sample-observationer x_1, \dots, x_n gennem et SARFIMA-filter;

Først konstrueres en vektor af længde $n+k$, hvori første indgang har værdien 1, mens de resterende indgange indeholder nuller. Vha. den estimerede fraktionelle differens-parameter \hat{d} tages den fraktionelle differens af rækken i denne vektor, og resultatet bliver en vektor indeholdende π_k 'erne fra ligning (1.6). I praksis beregnes den fraktionelle differens sådan, at der kun summeres op til $t-1$, og vektoren er på formen $[\pi_0 \dots \pi_{n+k-1}]^\top$.

Efterfølgende anvendes en funktion, kaldet `polymul`, der tager koefficienterne i to polynomier, der multipliceres med hinanden, som input, og som returnerer koefficienterne i det resulterende polynomium. Denne funktion anvendes her på fornævnte $\pi_k(\hat{d})$ -vektor samt på vektoren $[1 -\hat{\varphi}_1 \dots -\hat{\varphi}_p]^\top$, der indeholder de estimerede AR-parametre i SARFIMA-modellen. Resultatet bliver en vektor af længde $n+k+1$.

Nedenfor vises et eksempel på en sådan vektor, når der haves en SARFIMA(1,d,0)-model:

$$\begin{bmatrix} \pi_0 \\ \pi_1 - \hat{\varphi}_1 \cdot \pi_0 \\ \pi_2 - \hat{\varphi}_1 \cdot \pi_1 \\ \vdots \\ \pi_{n+k-1} - \hat{\varphi}_1 \cdot \pi_{n+k-2} \\ -\hat{\varphi}_1 \cdot \pi_{n+k-1} \end{bmatrix}.$$

De første $n + k$ indgange i denne vektor anvendes i funktionen `polymul` sammen med vektoren med længde 25, som er på formen $[1 \ 0 \ \dots \ 0 \ \hat{a}]^\top$. Denne vektor fungerer som et 24-timers-sæsonfilter. Dette resulterer i en vektor af længde $n+k+24$. Nedenfor vises et eksempel på en sådan vektor, når der haves en SARFIMA(1,d,0)-model:

$$\begin{bmatrix} \pi_0 \\ \pi_1 - \hat{\varphi}_1 \cdot \pi_0 \\ \vdots \\ \pi_{23} - \hat{\varphi}_1 \cdot \pi_{22} \\ \pi_{24} - \hat{\varphi}_1 \cdot \pi_{23} - \hat{a} \cdot \pi_0 \\ \pi_{25} - \hat{\varphi}_1 \cdot \pi_{24} - \hat{a}(\pi_1 - \hat{\varphi}_1 \cdot \pi_0) \\ \vdots \\ \pi_{n+k-1} - \hat{\varphi}_1 \cdot \pi_{n+k-2} - \hat{a}(\pi_{n+k-25} - \hat{\varphi}_1 \cdot \pi_{n+k-26}) \\ -\hat{a}(\pi_{n+k-24} - \hat{\varphi}_1 \cdot \pi_{n+k-25}) \\ \vdots \\ -\hat{a}(\pi_{n+k-1} - \hat{\varphi}_1 \cdot \pi_{n+k-2}) \end{bmatrix}.$$

Indgangene $(2:n+k)$ i denne vektor reverseres, og indgangene multipliceres med -1.

For hver sti der forecastes opstilles en *for-løkke*, som beregner de k forecastede skridt. Første skridt beregnes ved at anvende den reverserede vektor samt en $n + k$ vektor kaldet `lagdata`, der indeholder de n observationer fratrukket den estimerede middelværdi $\hat{\mu}$ efterfulgt af k nuller. Dvs. vektoren har følgende form

$$[x_1 - \hat{\mu} \ \dots \ x_n - \hat{\mu} \ 0 \ \dots \ 0]^\top.$$

Den reverserede vektor med indgange $(k : n + k - 1)$ og `lagdata`-vektoren med indgange $(1:n)$ multipliceres elementvist og summeres, hvorefter $e_1 \sqrt{\hat{\sigma}^2}$, $e_1 \sim N(0, 1)$ tillægges. Ligning (5.1) viser, hvorledes den forecastede værdi x_{n+1} beregnes, hvis der haves en SARFIMA(1,d,0)-model:

$$\begin{aligned} x_{n+1} = & -(\pi_n - \hat{\varphi}_1 \cdot \pi_{n-1} - \hat{a}(\pi_{n-24} - \hat{\varphi}_1 \cdot \pi_{n-25}))x_1 \\ & -(\pi_{n-1} - \hat{\varphi}_1 \cdot \pi_{n-2} - \hat{a}(\pi_{n-25} - \hat{\varphi}_1 \cdot \pi_{n-26}))x_2 \\ & \dots \\ & -(\pi_1 - \hat{\varphi}_1 \cdot \pi_0)x_n + e_1 \sqrt{\hat{\sigma}^2}. \end{aligned} \tag{5.1}$$

Herefter opdateres `lagdata`-vektoren således at det første nul erstattes med den forecastede værdi.

Til at beregne x_{n+2} anvendes samme fremgangsmåde, som blev benyttet til første skridt. Denne gang benyttes indgangene $(k - 1 : n + k - 1)$ i den reverserede vektor

samt indgangene $(1:n+1)$ i `lagdata`-vektoren til at beregne x_{n+2} . Dvs. at x_{n+1} indgår i beregningen af x_{n+2} . Denne procedure gentages, indtil alle k skridt i en sti er beregnet.

De simulerede sæsonrensede priser k skridt frem fyldes ind i en $(k \times M)$ -matrix såjle for såjle, således at hver såjle indeholder én MC-sti, som er et forecast. Her efter sorteres indgangene i hver enkelt række i stigende rækkefølge efter værdi, således at de mindste værdier for hver række er placeret i den første såjle. Dette resulterer i, at den første såjle indeholder de mindste værdier af forecastet på tværs af alle M MC-simulationer. Søjlerne $0.025 * M + 1$ og $0.975 * M + 1$ er hhv. nedre og øvre 95%-fejlbånd tilhørende forecastet. Søjle $0.5 * M + 1$ er punktestimaterne i forecastet.

Koderne anvendt til forecast vha. de ikke-regimeskiftende modeller fremgår af appendiks D.9 - D.12.

I det efterfølgende afsnit vises resultaterne af forecasts foretaget vha. de ikke-regimeskiftende SARFIMA(p, d, q)-modeller.

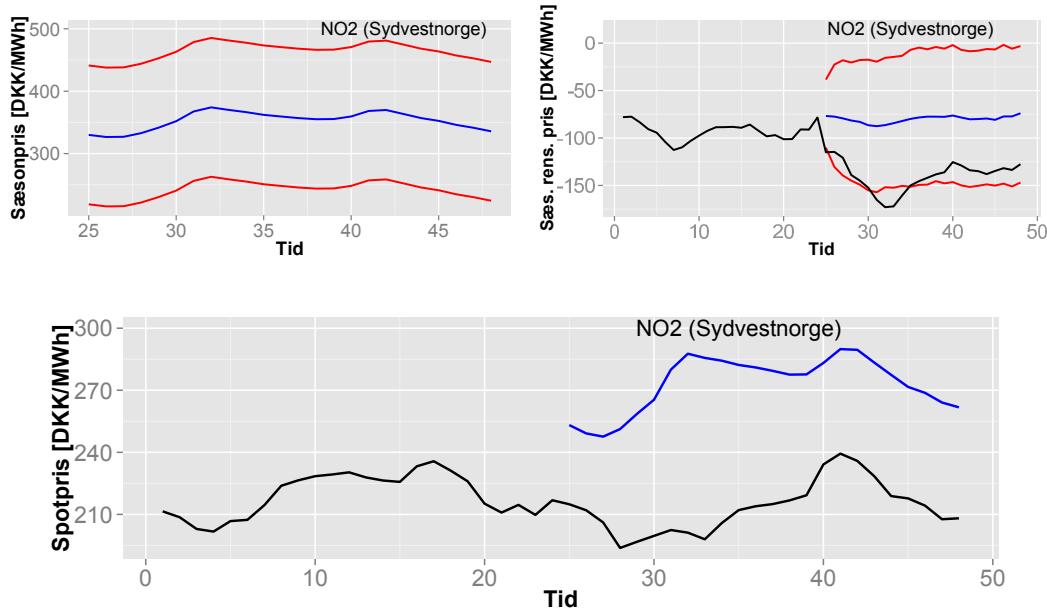
5.2 Forecast udført vha. modeller uden regimeskift

Figurerne 5.1 - 5.10 afbilder de forskellige forecasts, der er blevet produceret vha. modellerne uden regimeskift. Hver enkelt figur indeholder tre plots. Øverst til venstre afbildes de fremtidige værdier af sæsonfunktionen (blå) med 95%-prædiktionsinterval (rød). Øverst til højre haves et plot af de forecastede sæsonrensede priser (blå) med 95%-fejlbånd (rød). Disse er plottet sammen med de faktiske sæsonrensede priser (sort), som er fremkommet ved at fratrække værdien af sæsonfunktionen fra de faktiske priser. Nederst plottes punktestimater for prisforecastet (blå), der er fremkommet ved at addere punktestimaterne i forecastet for de sæsonrensede priser med funktionsværdierne for sæsonfunktionen. Den forecastede pris er plottet sammen med de observerede priser (sort), for at give et indtryk af forecastets præstation. I plottet af de faktiske priser (dvs. inkl. sæsondelen) er der ikke medtaget 95%-fejlbånd. Da prisen er lig summen af sæsonfunktionen og den forecastede sæsonrensede pris, er det ikke umiddelbart muligt at konstruere et fejlbånd, idet det er nødvendigt at tage hensyn til korrelationen mellem sæsonfunktionen og de sæsonrensede priser.

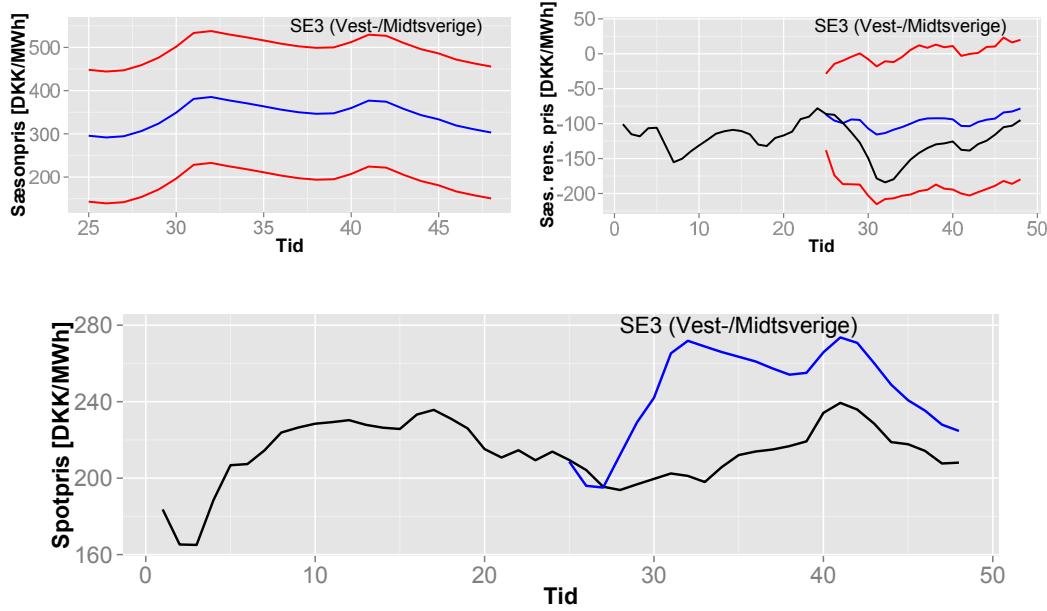
Hver af figurerne 5.1 - 5.5 indeholder et forecast af elprisen i de fem områder. Figur 5.1 afbilder forecastene for området NO2. Her fremgår det, at den forecastede pris og forecastede sæsonrensede pris er højere end de faktiske priser/sæsonrensede priser, hvilket vil sige, at modellen ikke laver et særligt godt forecast. I figur 5.1 fremgik det, at priserne primo 2014 var meget lave i forhold til samme periode i de to tidligere år, og dette kan være en forklaring på, at de forecastede priser er højere.

Figurerne 5.2 og 5.3 afbilder de forskellige forecasts for de to svenske områder SE3 og SE4. Disse to figurer viser de samme tendenser. Både de forecastede sæsonrensede priser og de "forecastede" spotpriser ligger for det meste over de observerede, hvilket vil sige, at modellerne prædkterer en højere pris end den faktisk observerede. Det fremgår, at de forecastede sæsonrensede priser, for både SE3 og SE4, er noget bedre end de forecastede sæsonrensede priser for NO2.

Figurerne 5.4 og 5.5 afbilder de forskellige forecasts for områderne DK1 og DK2. Disse forecasts ligger væsentligt tættere på de observerede priser i forhold til forecastene for de resterende prisområder. Umiddelbart synes modellerne at være gode til at



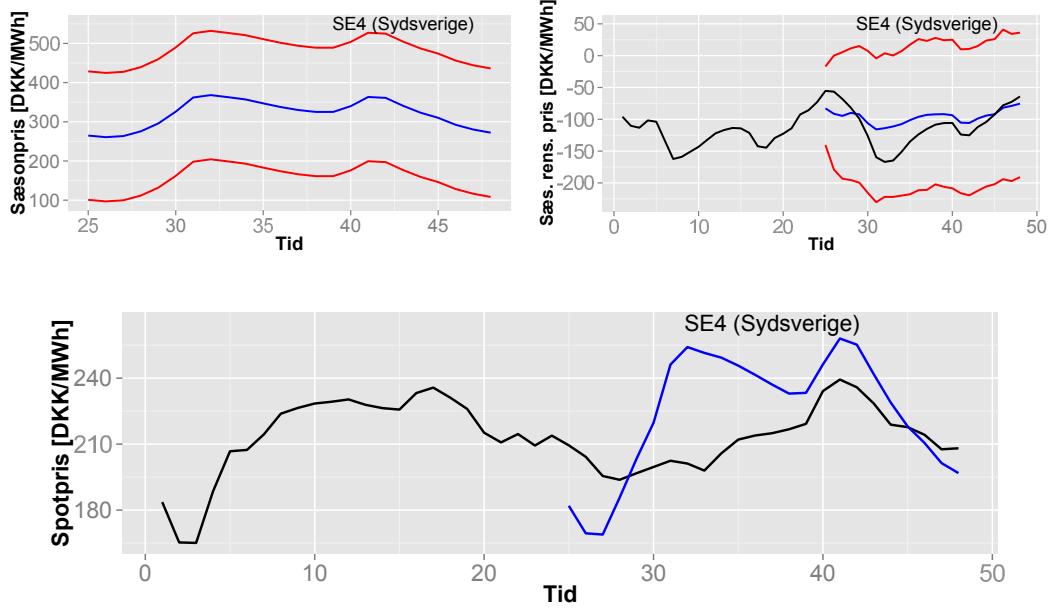
Figur 5.1: Plots af forskellige forecasts for NO2, ved brug af SARFIMA(p, d, q)-modeller uden regimeskift.



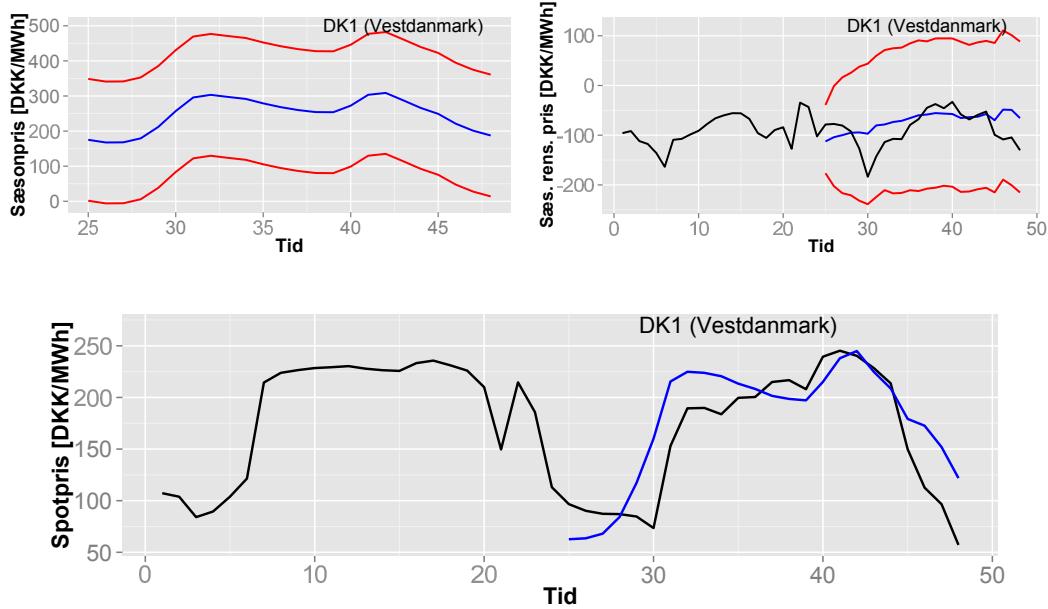
Figur 5.2: Plots af forskellige forecasts for SE3, ved brug af SARFIMA(p, d, q)-modeller uden regimeskift.

opfange de overordnede tendenser i priserne, men de har lidt sværere ved at opfange de små hurtige prisændringer.

Figurerne 5.6 - 5.10 viser forecasts for prisdifferenserne mellem to forbundne områder. Den første figur 5.6 afbilder forecastet for prisdifferensen mellem NO2 og DK1. Her fremgår det af plottet, der viser forecastet af de sæsonrensede prisdifferenser, at modellen ikke laver et godt forecast. Forecastet for den ikke-sæsonrensede prisdiffe-



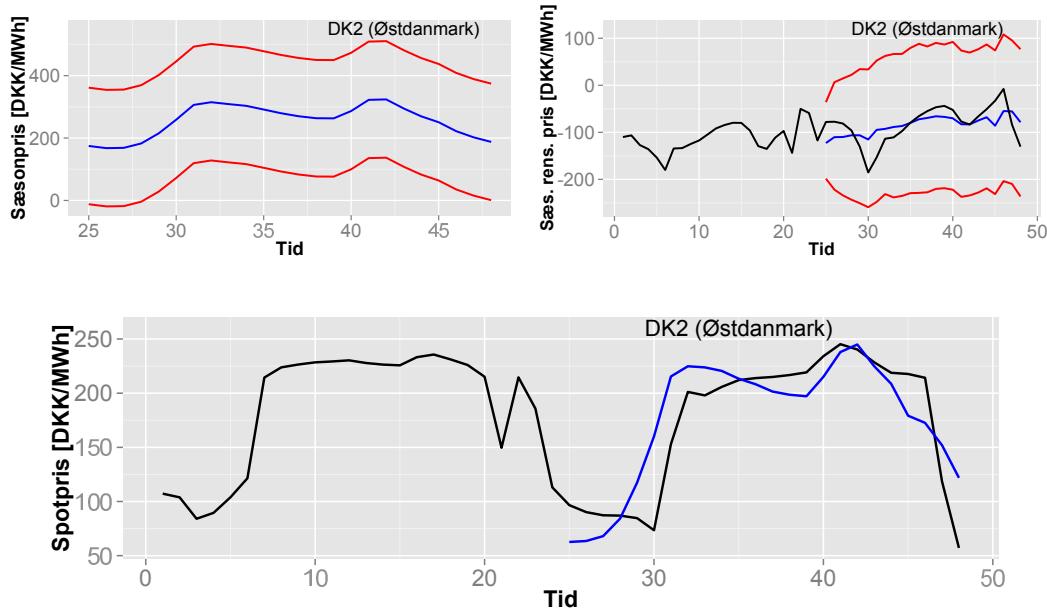
Figur 5.3: Plots af forskellige forecasts for SE4, ved brug af SARFIMA(p, d, q)-modeller uden regimeskift.



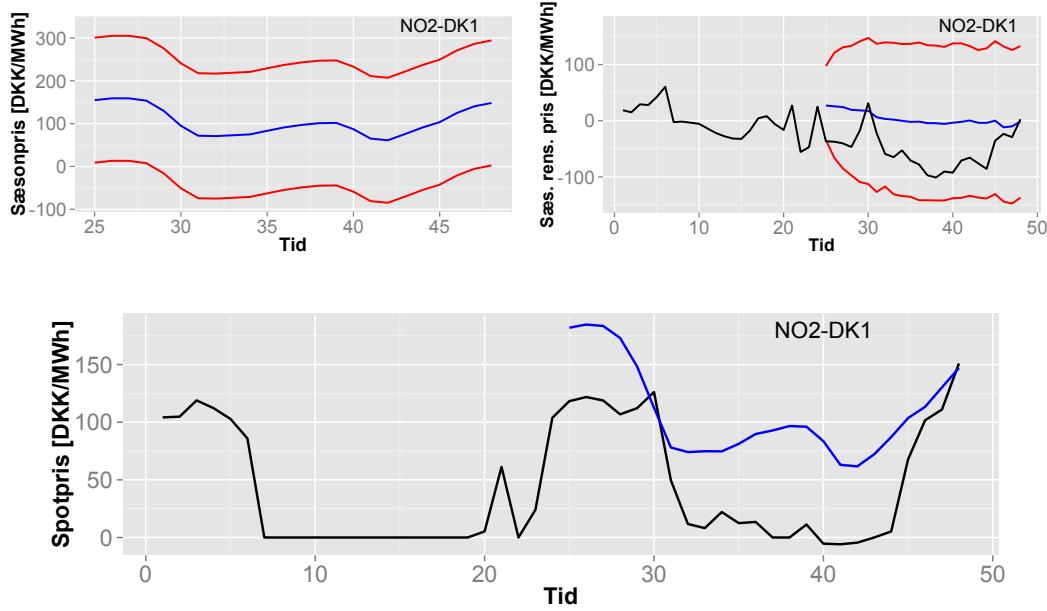
Figur 5.4: Plots af forskellige forecasts for DK1, ved brug af SARFIMA(p, d, q)-modeller uden regimeskift.

rens ser lidt bedre ud. Her udtrykker kurven de rigtige tendenser i stigning og fald, men den forecastede prisdifferens er dog stadig noget større end den observerede.

Figuren 5.7 viser forecastet for prisdifferensen mellem områderne DK1 og SE3. Her er den forecastede prisdifferens mindre end den observerede. Ingen fremgår det, at forecastet af den ikke-sæsonrensedede prisdifferens har samme facon som de observerede priser.



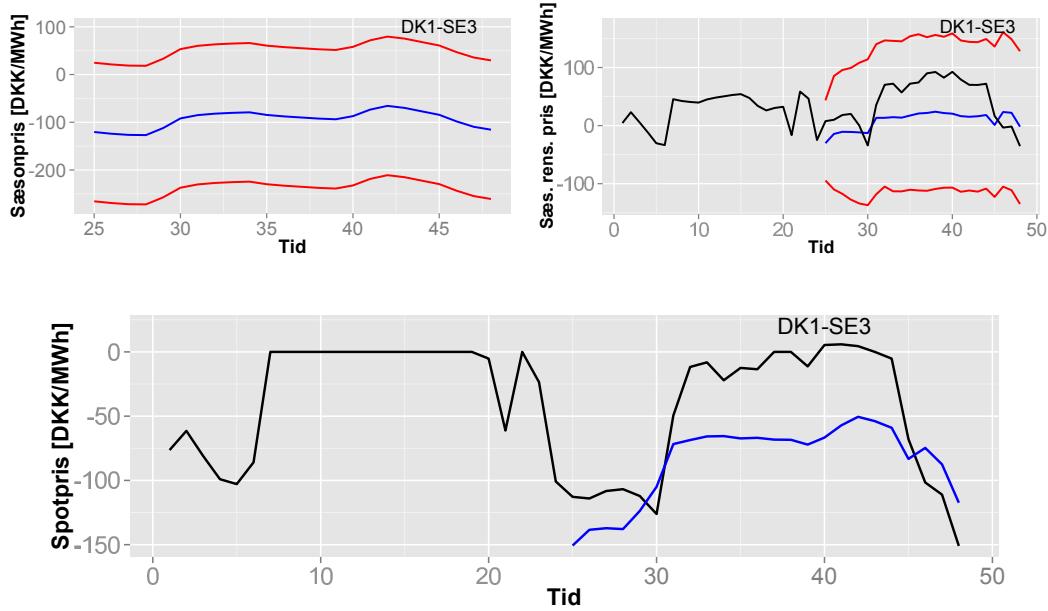
Figur 5.5: Plots af forskellige forecasts for DK2, ved brug af SARFIMA(p, d, q)-modeller uden regimeskift.



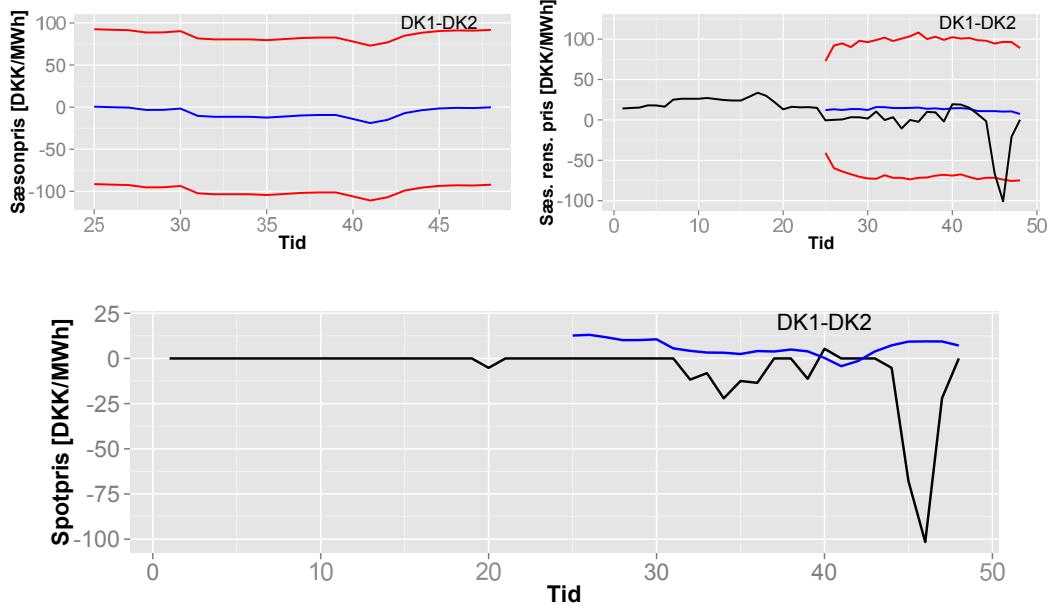
Figur 5.6: Plots af forskellige forecasts for prisdifferensen mellem NO2 og DK1, ved brug af SARFIMA(p, d, q)-modeller uden regimeskift.

Forecastet for prisdifferensen mellem DK1 og DK2 fremgår af figur 5.8. Her ligger den forecastede prisdifferens, både med og uden sæsondelen, ret tæt op ad den observerede prisdifferens, det meste af tiden. Der er dog en nedadgående peak omkring time 46, som den anvendte SARFIMA-model ikke formår at opfange i forecastet.

Forecastet af prisdifferensen mellem områderne DK2 og SE4 er afbilledet i figur 5.9. Her er den forecastede prisdifferens mindre end den observerede prisdifferens. Forskellen



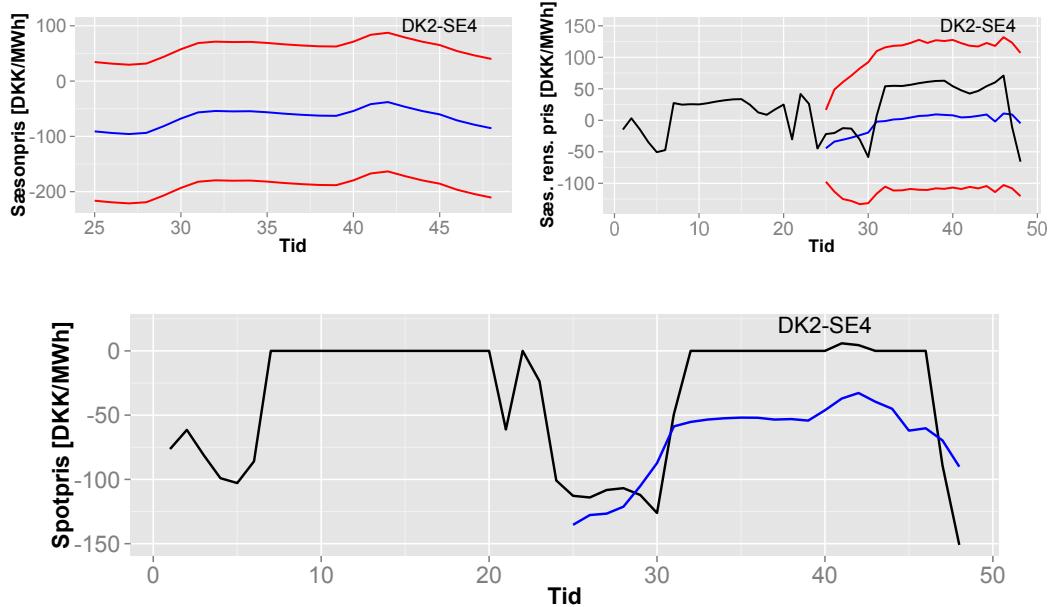
Figur 5.7: Plots af forskellige forecasts for prisdifferensen mellem DK1 og SE3, ved brug af SARFIMA(p, d, q)-modeller uden regimeskift.



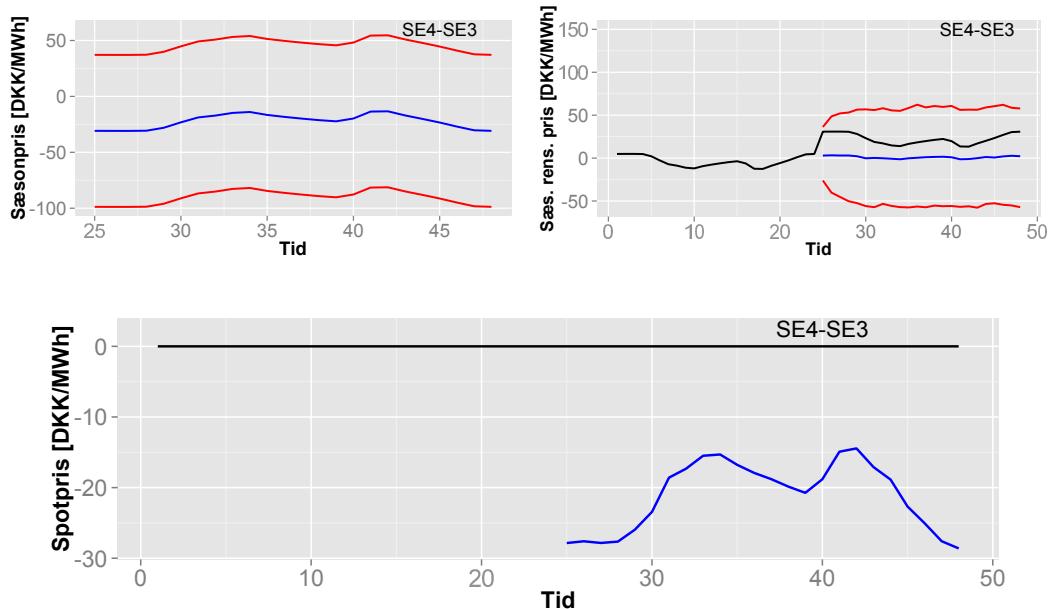
Figur 5.8: Plots af forskellige forecasts for prisdifferensen mellem DK1 og DK2, ved brug af SARFIMA(p, d, q)-modeller uden regimeskift.

ligger på ca. 52 kr. der hvor den er størst omkring time 45.

Den sidste figur 5.10 viser forecastet af prisdifferensen mellem SE4 og SE3. Disse plots skiller sig ud fra de andre, idet de observerede prisdifferenser i de første 24 timer i år 2014 er nul. Dette er ikke i overensstemmelse med forecastet, der viser en negativ prisdifferens, svarende til at priserne i SE3 er større end i SE4.



Figur 5.9: Plots af forskellige forecasts for prisdifferensen mellem DK2 og SE4, ved brug af SARFIMA(p, d, q)-modeller uden regimeskift.



Figur 5.10: Plots af forskellige forecasts for prisdifferensen mellem SE4 og SE3, ved brug af SARFIMA(p, d, q)-modeller uden regimeskift.

Kapitel 6

Forecast af elspotpriserne ved brug af modellerne med regimeskift

I dette kapitel vil det blive forsøgt at foreaste elspotpriserne 24-timer frem ved at anvende de SARFIMA(p, d, q)-modeller, som blev valgt i afsnit 4.2.2.

De sæsonrensede elspotpriser forecastes vha. Monte Carlo simulationer. Ved at anvende de estimerede SARFIMA(p, d, q)-modeller, vil der blive forecastet k skridt frem. Idet det ønskes at lave et forecast for ikke kun de sæsonrensede priser, men også for de ikke-sæsonrensede priser, vil de fremtidige værdier af sæsonfunktionerne blive fundet.

6.1 Implementering af forecast udført af modeller med regimeskift

Der forecastes $k = 24$ skridt ud i fremtiden, dvs. ét døgn frem, og der udføres $M = 1000$ Monte Carlo simuleringer, dvs. i forbindelse med hvert prisforecast udføres 1000 MC-stier. For hver af de fem forbindelser udføres et forecast for hvert område, samt for prisdifferensen.

Implementeringen af forecasts udført af modeller med regimeskift er meget lig implementeringen for modeller uden regimeskift. Først sammensættes to parametervektorer for de estimerede parametre under regime 1 $\hat{\eta}_1$, og for de estimerede parametre under regime 0 $\hat{\eta}_0$:

$$\begin{aligned}\hat{\eta}_1 &= [\hat{\varphi}_{1,1} \dots \hat{\varphi}_{1,p} \hat{d}_1 \hat{\mu}_1 \hat{a}_1]^\top \\ \hat{\eta}_0 &= [\hat{\varphi}_{0,1} \dots \hat{\varphi}_{0,p} \hat{d}_0 \hat{\mu}_0 \hat{a}_0]^\top\end{aligned}$$

Disse to parametervektorer samles til en parametervektor $\hat{\eta} = [\hat{\eta}_1 \hat{\eta}_0]$, således at denne indeholder de estimerede parameterværdier for hvert regime i den valgte SARFIMA(p, d, q)-model.

Idet de regimeskiftende SARFIMA(p, d, q)-modeller har et sæt parametre defineret for hvert regime, er det nødvendigt at “foreaste” regimetilstanden k skridt frem i tiden. Til dette anvendes transitionsmatricerne, der blev estimeret i afsnit 4.2. Indgangene \hat{p}_{00} og \hat{p}_{10} fra transitionsmatricen indsættes i en **markov**-vektor. Derefter opstilles en $(k \times M)$ -matrix **randomus**, som udfyldes med ligefordelte tal.

For hver af de M simulationer foretages der også en simulering af de fremtidige

regimetilstande k skridt frem. Vha. det sidst observerede regime (dvs. 0 eller 1) bestemmes hvorvidt vi efterfølgende befinder os i regime 0 eller regime 1. Såfremt det sidst observerbare regime er regime 0, anvendes p_{00} fra **markov**-vektoren, hvor p_{00} angiver sandsynligheden for at være i regime 0 og blive der. Hvis tallet p_{00} er mindre end det uniformt fordelte tal, der trækkes fra **randomus**, vil det næste regime være 1, ellers 0. Denne fremgangsmåde repeteres, således at regimetilstanden k skridt frem bliver bestemt i hver MC-sti.

Forecastet udføres grundlæggende vha. en lignende fremgangsmåde ift. afsnit 5.1. Dvs. at de n in-sample observationer føres igennem et SARFIMA-filter. I dette tilfælde føres observationerne igennem et SARFIMA-filter, hvori parametrene fra regime 1 først anvendes og derefter parametrene for regime 0. Herved opnås to forskellige reverserede vektorer, hvor den ene er udført vha. de estimerede parametre i regime 1, og den anden er udført vha. de estimerede parametre tilhørende regime 0.

For hver forecast-sti gennemløbes en *ifelse*-betingelse i henhold til den simulerede regime-vektor. Denne betingelse medfører, at hvis vi jf. regime-vektoren befinder os i regime 1, så er det den reverserede vektor, beregnet ud fra parameterestimaterne tilhørende regime 1, der skal anvendes til at beregne den forecastede observation. Ellers er det den reverserede vektor lavet på baggrund af parametrene tilhørende regime 0, der anvendes i beregningerne. Således opnås et forecast k skridt frem i tid, under hensyntagen til regimetilstanden.

Koderne anvendt til forecast vha. de ikke-regimeskiftende modeller fremgår af appendiks D.9, D.10, D.13 og D.14.

6.2 Forecast udført vha. modeller med regimeskift

Figurerne 6.1 - 6.11 afbilder de forskellige forecasts, der er blevet produceret vha. modellerne med regimeskift. Til hver forbindelse er der lavet to figurer. Den første figur indeholder seks plots, som afbilder sæsonfunktionen og den sæsonrensede pris for hvert område samt for prisdifferensen. I venstre spalte er plottene af sæsonfunktionerne placeret. Den faktiske værdi af sæsonfunktionen (sort) er fremkommet vha. det observerede fremtidige regime. Idet sæsonfunktionen afhænger af de fremtidige regimetilstande, er "forecastet" af sæsonfunktionen (blå), samt 95%-prædiktionsinterval i hver sti, blevet beregnet for hver MC-sti vha. de simulerede regimer. Efterfølgende sorteres hver række i $(k \times 3M)$ -matricen der indeholder de fremtidige værdier af sæsonfunktionen, samt 95%-prædiktionsintervallerne. Søjlerne 76 og 2926 vil blive anvendt som 95%-fejlbånd (rød), og søjle 1501 vil udgøre punktestimatet for forecastet af de fremtidige værdier af sæsonfunktionen. Plottene i højre spalte afbilder de forecastede sæsonrensede priser (blå), med 95%-fejlbånd (rød). Disse er plottet sammen med de faktiske sæsonrensede priser (sort), som er fremkommet ved at fratrække værdien af sæsonfunktionen fra de faktiske priser.

Den anden figur afbilder punktestimater for prisforecastet (blå) for hvert område samt for prisdifferensen. Disse er fremkommet ved, at lægge forecastet af den sæsonrensede pris i hver MC-sti sammen med sæsonfunktionen, som er beregnet ud fra de estimerede fremtidige regimer for den pågældende sti. Den forecastede pris er plottet sammen med de observerede priser (sort), for at give et indtryk af forecastets præstation. I plottet af de faktiske priser (dvs. inkl. sæsondelen) er der ikke medtaget 95%-fejlbånd. Da prisen er lig summen af sæsonfunktionen og den fore-

castede sæsonrensede pris, er det ikke umiddelbart muligt at konstruere et fejlbånd, idet det er nødvendigt at tage hensyn til korrelationen mellem sæsonfunktionen og de sæsonrensede priser. For de regimeskiftende modellers vedkommende udgør den fremtidige, men ukendte, regimetilstand en yderligere kilde til usikkerhed.

For at kunne vurdere hvilken type model, der giver det bedste forecast, er RMSE blevet beregnet for forecastene af priserne med sæson. Dette er gjort for både de regimeskiftende og de ikke-regimeskiftende modeller. Resultatet heraf er angivet i tabel 6.1, hvor laveste RMSE for hver prisserie er anført i rødt. Det fremgår af tabellen, at modellerne uden regime-skifte i de fleste tilfælde præsterer bedst målt på RMSE.

		Modeller uden regimeskift	Modeller med regimeskift
NO2 - DK1	NO2	61.33107	92.77073
	DK1	36.38963	39.52433
	Diff.	63.38748	65.27847
DK1 - SE3	DK1	36.38963	45.63545
	SE3	39.33558	45.37079
	Diff.	46.93899	70.81262
DK1 - DK2	DK1	36.38963	33.78555
	DK2	33.08095	32.26755
	Diff.	30.40058	26.31875
DK2 - SE4	DK2	33.08095	60.793
	SE4	26.49156	58.01871
	Diff.	43.82147	67.6511
SE4 - SE3	SE4	26.49156	39.40748
	SE3	39.33558	63.01483
	Diff.	21.75182	0.000

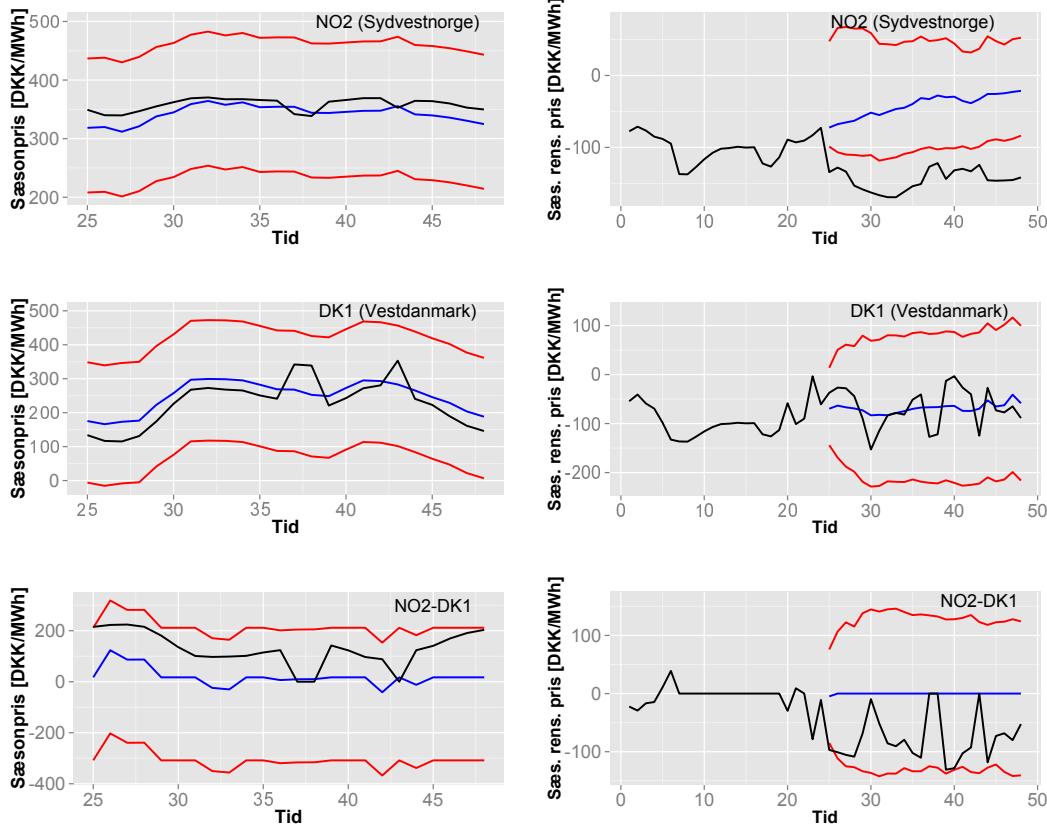
Tabel 6.1: RMSE for forecastene af de faktiske priser vha. både regimeskiftende og ikke-regimeskiftende modeller.

Figurerne 6.1 og 6.2 afbilder de forskellige forecast for forbindelsen NO2-DK1. Det fremgår af figurerne, at forecastet af priserne med sæson for område NO2 ligger højere end de faktiske priser. Sammenlignes der med forecastet af priserne med sæson for NO2 i figur 5.1, bemærkes det, at forecastet udført vha. modellen med regimeskift præsterer dårligere end forecastet udført vha. modellen uden regimeskift.

Forecastet af priserne med sæson for område DK1 ligger tæt på de faktiske priser. Dette var dog også i høj grad tilfældet ved anvendelse af den ikke-regimeskiftende model, hvis forecast fremgår af figur 5.4.

Betrages forecastet af prisdifferensen mellem NO2 og DK1 bemærkes det, at modellen med regimeskift i perioder ligger forholdsvis tæt på den faktiske prisdifference. Sammenlignes dette med resultatet fra den ikke-regimeskiftende model i figur 5.6, præsterer modellen med regimeskift bedre end modellen uden i timerne omkring 32 til 44. Dog er modellen uden regimeskift bedre målt på RMSE.

Forecastene tilhørende forbindelsen DK1-SE3 er vist i figurerne 6.3 og 6.4. Her ligger forecastet af priserne med sæson for området DK1 forholdsvis tæt op ad de faktiske priser. Dette forecast kan sammenlignes med både prisforecastet for området DK1 i forbindelsen NO2-DK1, udført vha. en regimeskiftende model (figur 6.2), og med prisforecastet for område DK1 udført vha. en ikke-regimeskiftende model, hvilket fremgår af figur 5.4. Alle tre prisforecasts ligger relativt tæt på de faktiske observationer, og det er umiddelbart ikke muligt at se hvilken model, der har præsteret det bedste prisforecast. Betragtes RMSE fra tabel 6.1, fremgår det, at modellen uden



Figur 6.1: Plots af forskellige forecasts for forbindelsen NO2-DK1, ved brug af SARFIMA(p, d, q)-modeller med regimeskift.

regimeskift præsterer det bedste forecast.

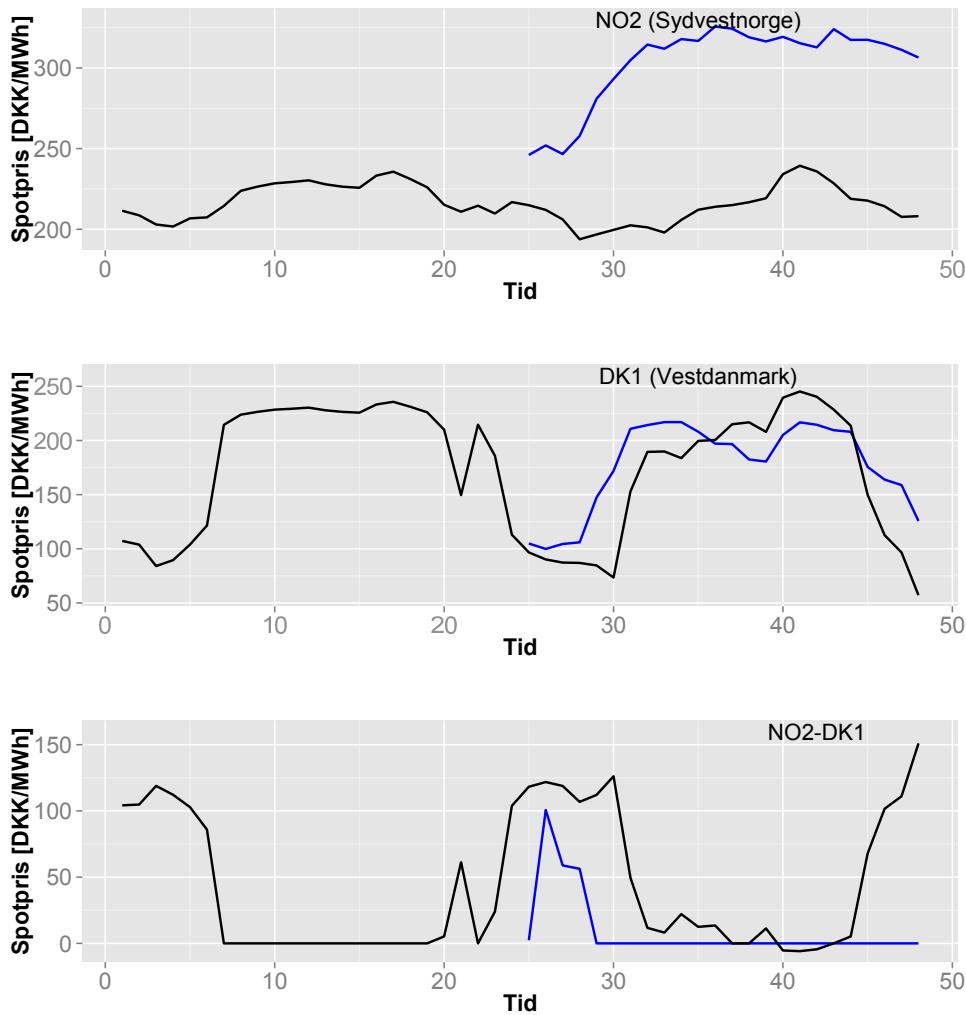
De forecastede priser for område SE3, er generelt noget højere end de observerede priser. Dette var også tilfældet i forecastet udført vha. modellen uden regimeskift, som fremgår af figur 5.2. Ud fra figurerne alene, er det ikke klart, hvilken model der giver det bedste forecast. Modellen uden regimeskift har dog lavest RMSE.

Betrages RMSE for den forecastede prisdifferens mellem områderne DK1 og SE4 er modellen uden regimeskift at foretrække. Men i perioden time 32 til 44 ligger forecastet fra den regimeskiftende model meget tæt på den observerede prisdifferens.

Figurerne 6.5 og 6.6 afbilder forecastene for forbindelsen DK1-DK2. Modellerne med regimeskift har mindste RMSE i denne forbindelse. Forecastene for begge de danske områder ligger tæt på de observerede priser. Dette var også tilfældet ved modellerne uden regimeskift, som fremgår af figurerne 5.4 og 5.5.

Den forecastede prisdifferens mellem områderne DK1 og DK2 er nul i alle 24 timer. Dette var ikke tilfældet i forecastet fra den ikke regimeskiftende model i figur 5.8. Denne forskel er forventelig, idet modellen med regimeskift tager højde for de to forskellige regimer, og sandsynligheden for at vi befinner os i hvert regime.

Figurerne 6.7 - 6.9 afbilder de forskellige forecasts for forbindelsen DK2-SE4. Det fremgår af figur 6.7, at fejlbåndene for forecastet af den sæsonrensede pris for området DK2 hurtigt går mod ekstreme værdier. Derfor er der taget et udsnit af dette plot, som fremgår af figur 6.8.



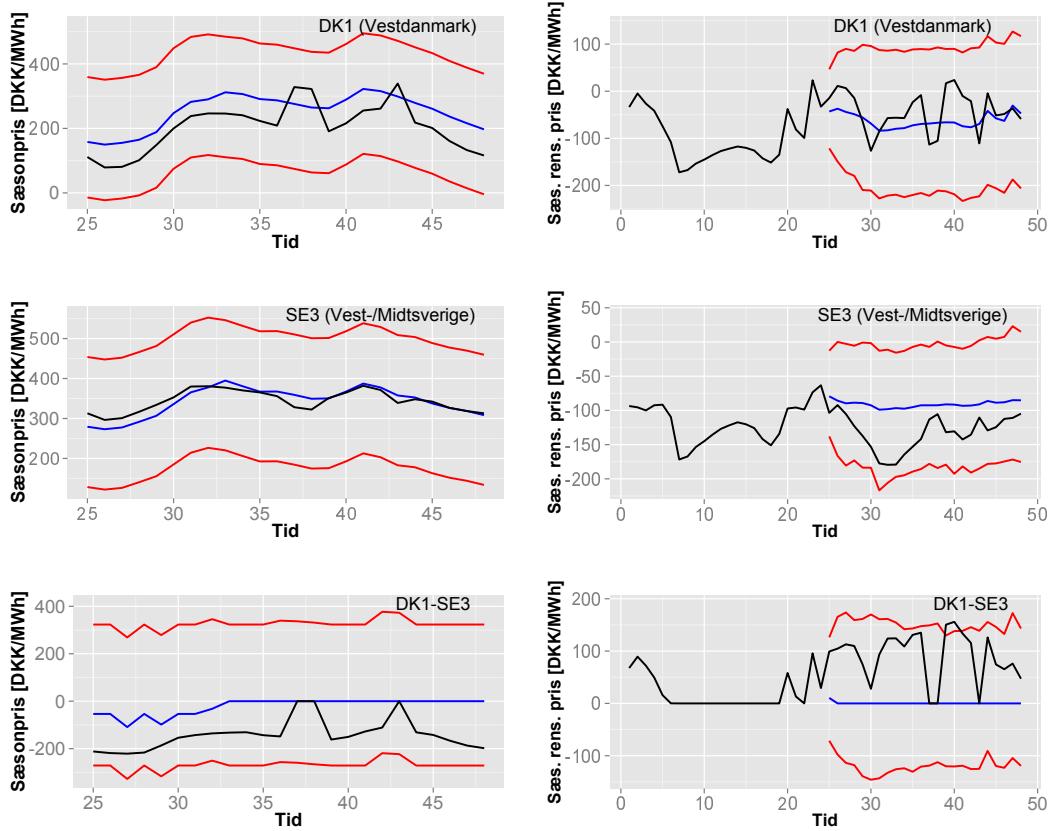
Figur 6.2: Plots af forskellige forecasts for forbindelsen NO2-DK1, ved brug af SARFIMA(p, d, q)-modeller med regimeskift.

Forecastet for priserne i området DK2 ligger som regel over de observerede priser. Dette er væsentligt dårligere end DK2 forecastet fra den ikke-regimeskiftende model (figur 5.5). Det samme gør sig gældende for de to forecasts for SE4.

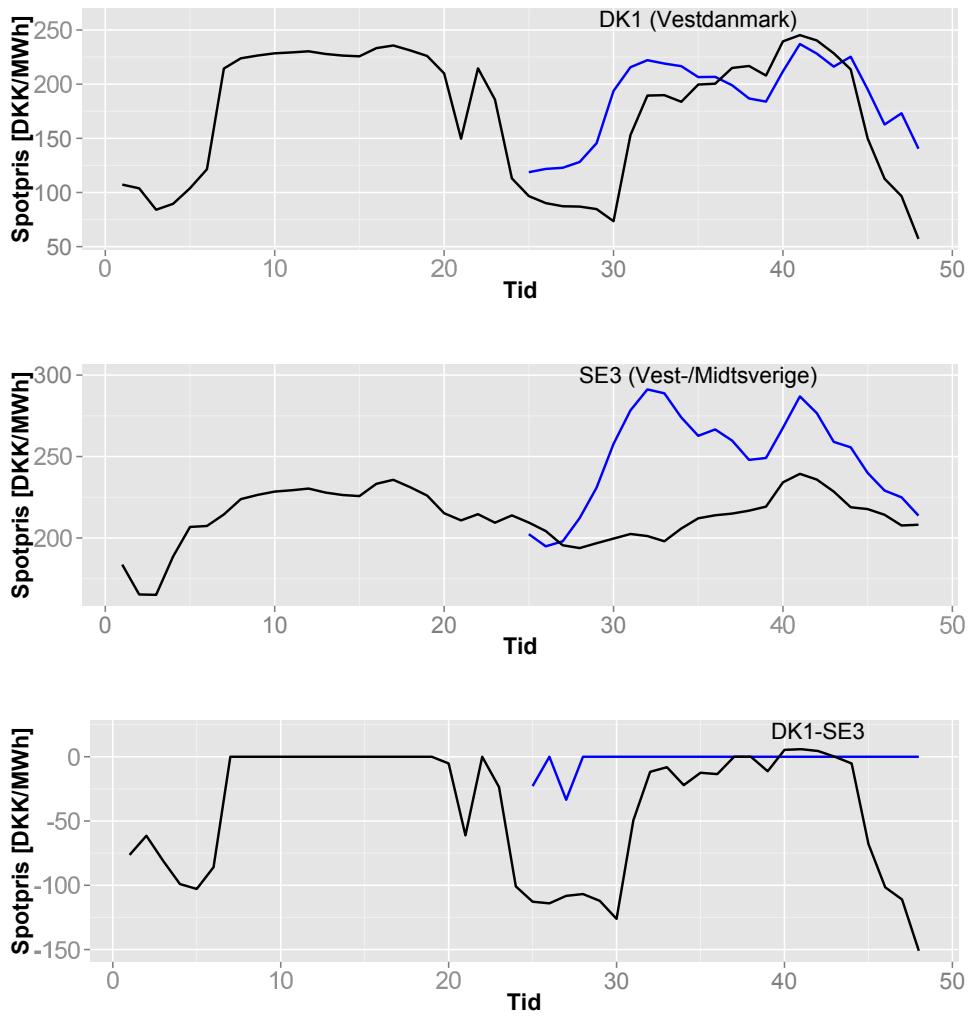
Ligesom ved forbindelsen DK1-DK2, ligger forecastet af prisdifferensen mellem DK2 og SE4 for det meste på nul. Dette kan som tidligere nævnt ikke rigtigt lade sig gøre i den ikke regimeskiftende model.

Figurerne 6.10 og 6.11 afbilder forecasts for forbindelse SE4-SE3. Ingen ligger de forecastede priser for SE4 og SE3 over de observerede. Dette var også tilfældet ved forecastene fra de ikke-regimeskiftende modeller (figurer 5.2 og 5.3), hvor forecastfejlene var lidt mindre.

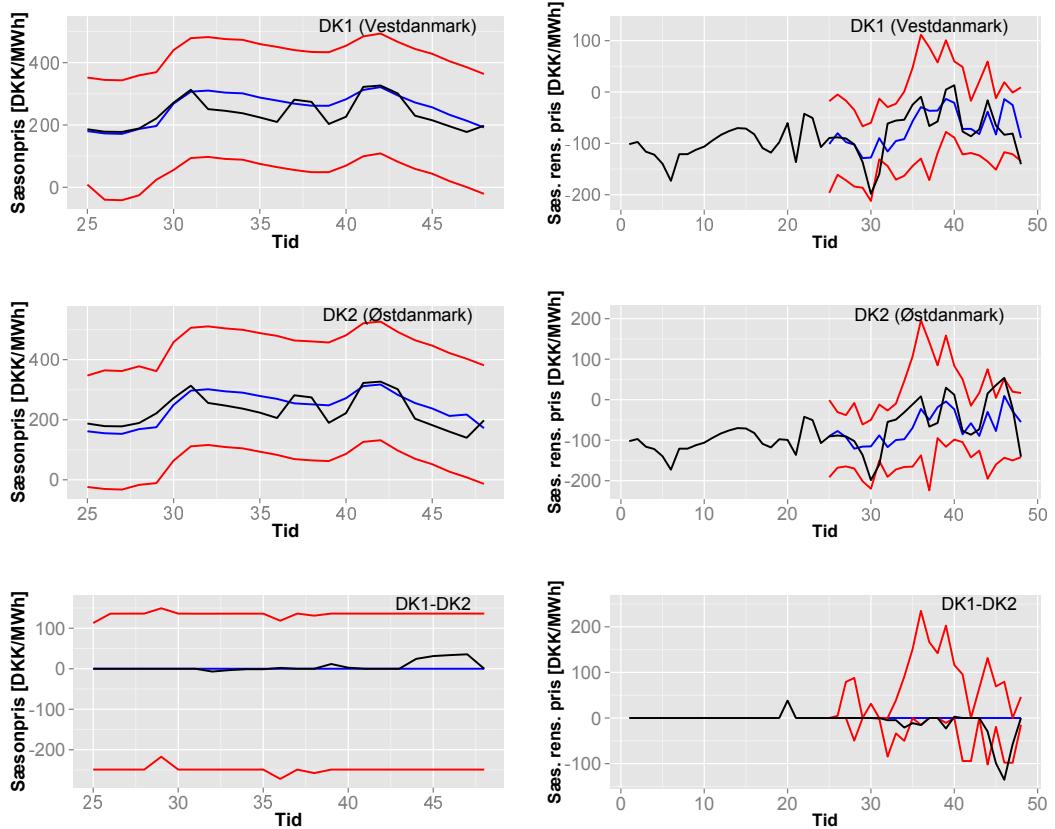
Den regimeskiftende model forecaster en udvikling i prisdifferensen, der er fuldstændig i overensstemmelse med den observerede prisdifference. Dette skyldes at modellen tager hensyn til, at sandsynligheden for at forbindelsen SE4-SE3 er i regime 0, er meget høj. Dette er ikke muligt i den ikke-regimeskiftende model (5.10), hvor forecastet af prisdifferensen afviger fra den observerede prisdifference.



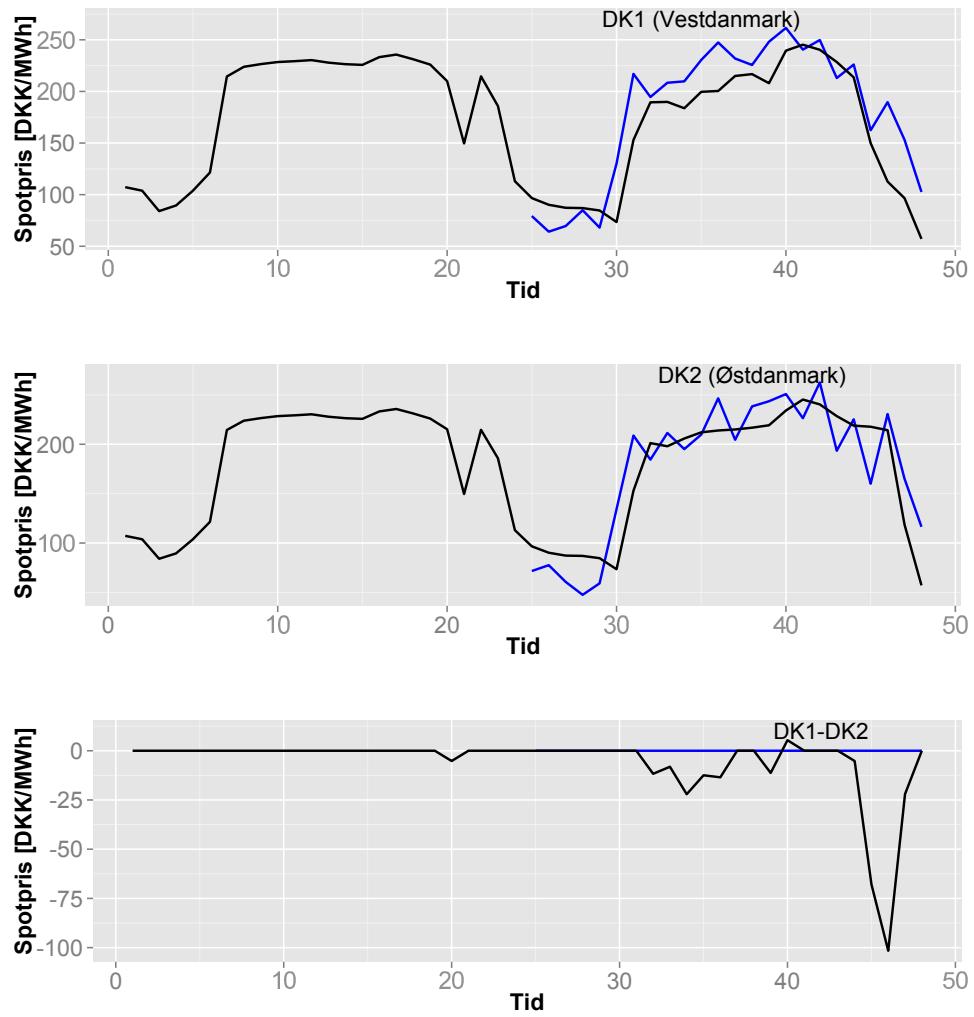
Figur 6.3: Plots af forskellige forecasts for forbindelsen DK1-SE3, ved brug af SARFIMA(p, d, q)-modeller med regimeskift.



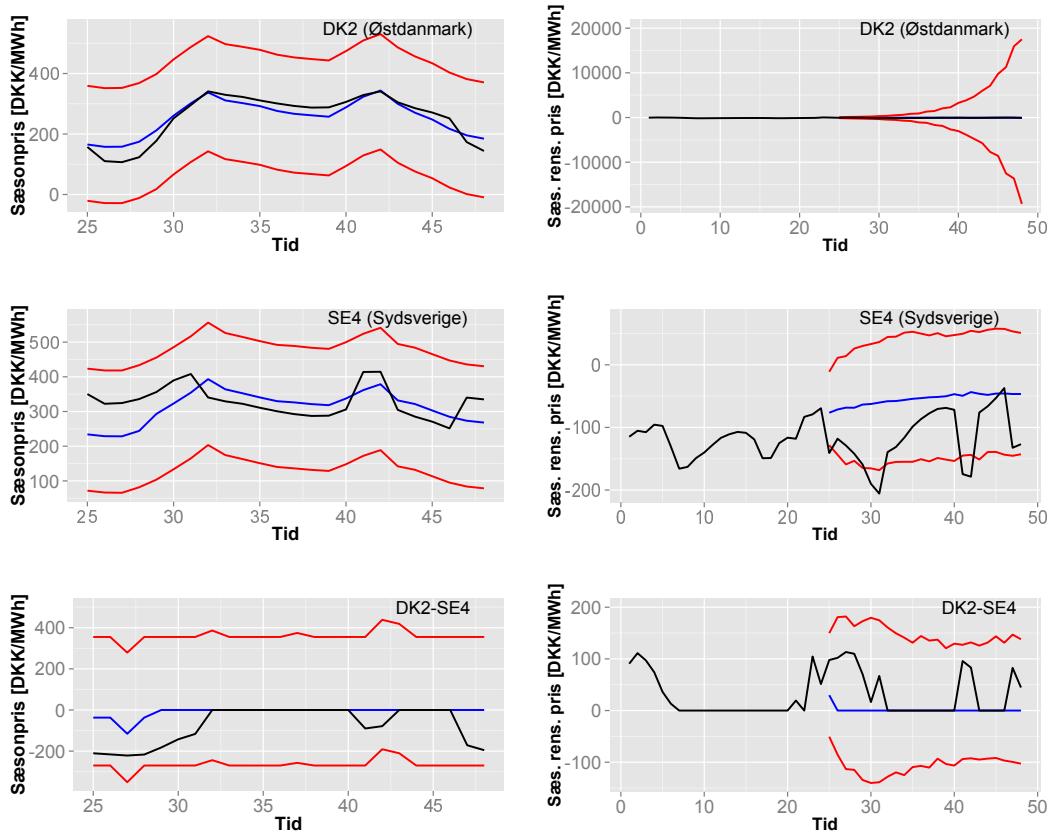
Figur 6.4: Plots af forskellige forecasts for forbindelsen DK1-SE3, ved brug af SARFIMA(p, d, q)-modeller med regimeskift.



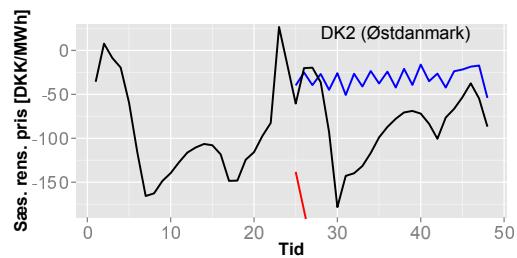
Figur 6.5: Plots af forskellige forecasts for forbindelsen DK1-DK2, ved brug af SARFIMA(p, d, q)-modeller med regimeskift.



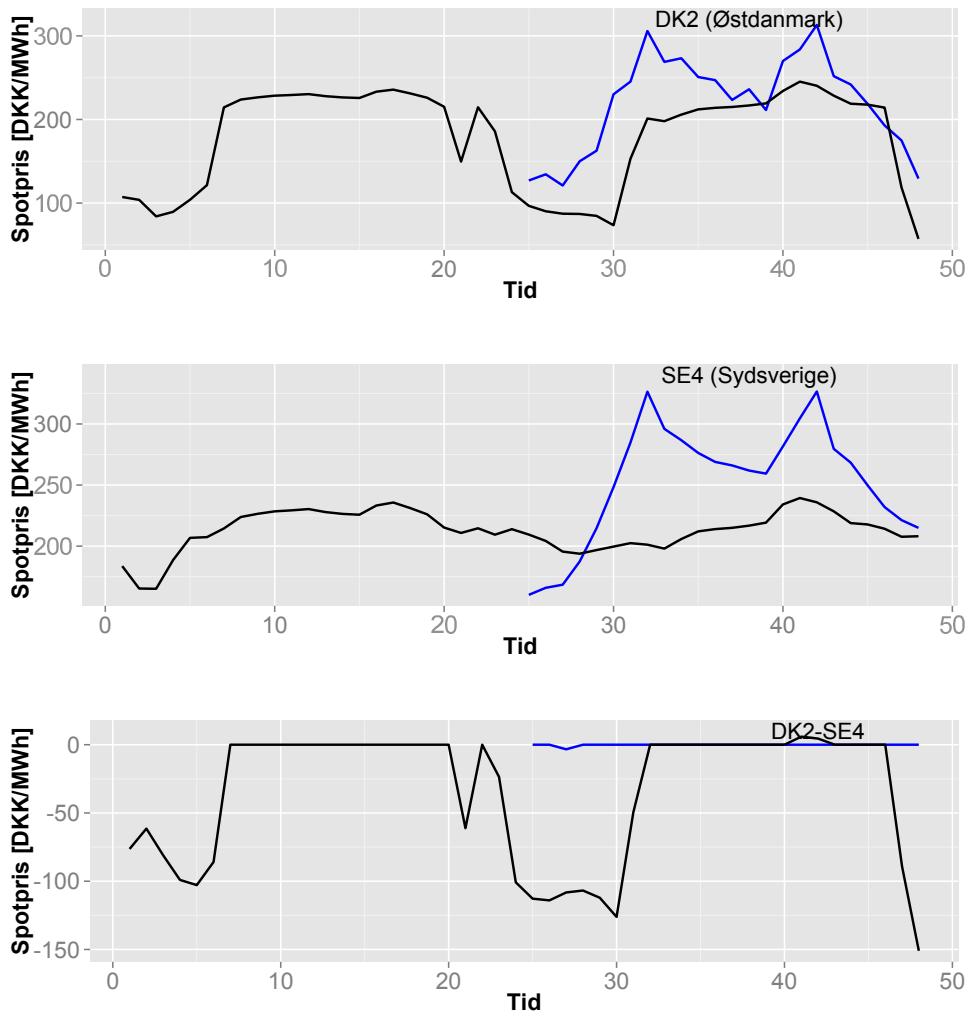
Figur 6.6: Plots af forskellige forecasts for forbindelsen DK1-DK2, ved brug af SARFIMA(p, d, q)-modeller med regimeskift.



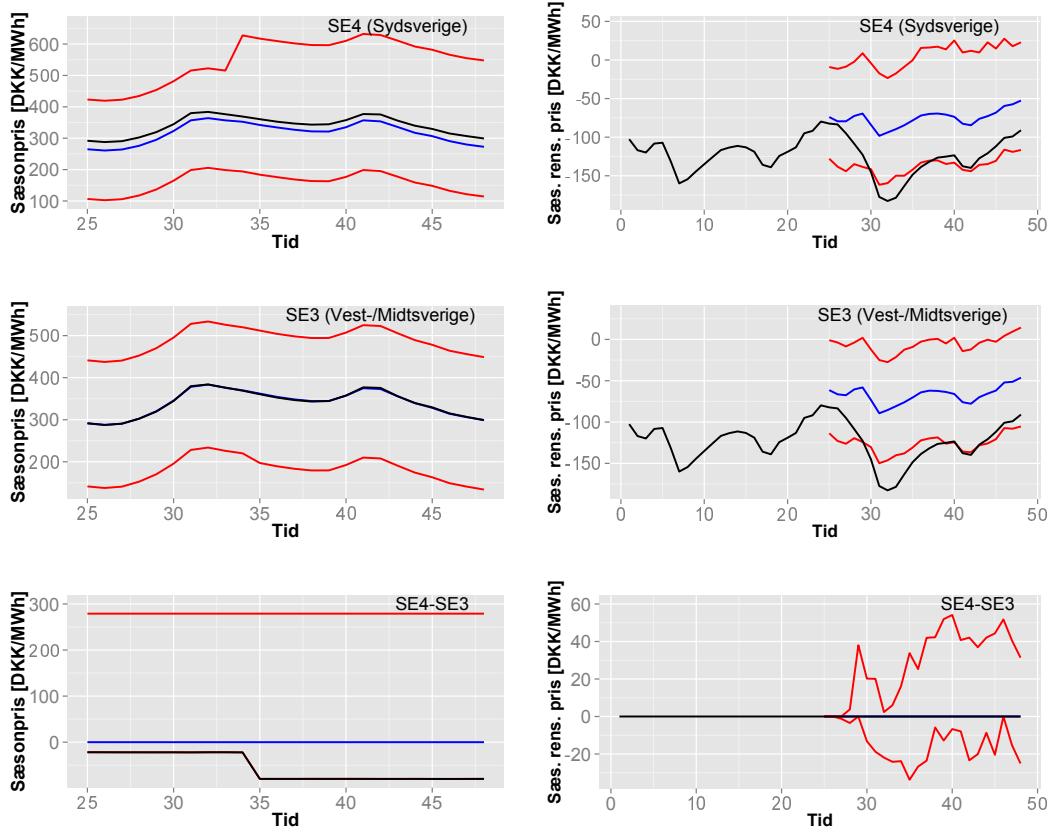
Figur 6.7: Plots af forskellige forecasts for forbindelsen DK2-SE4, ved brug af ARFIMA(p, d, q)-modeller med regimeskift.



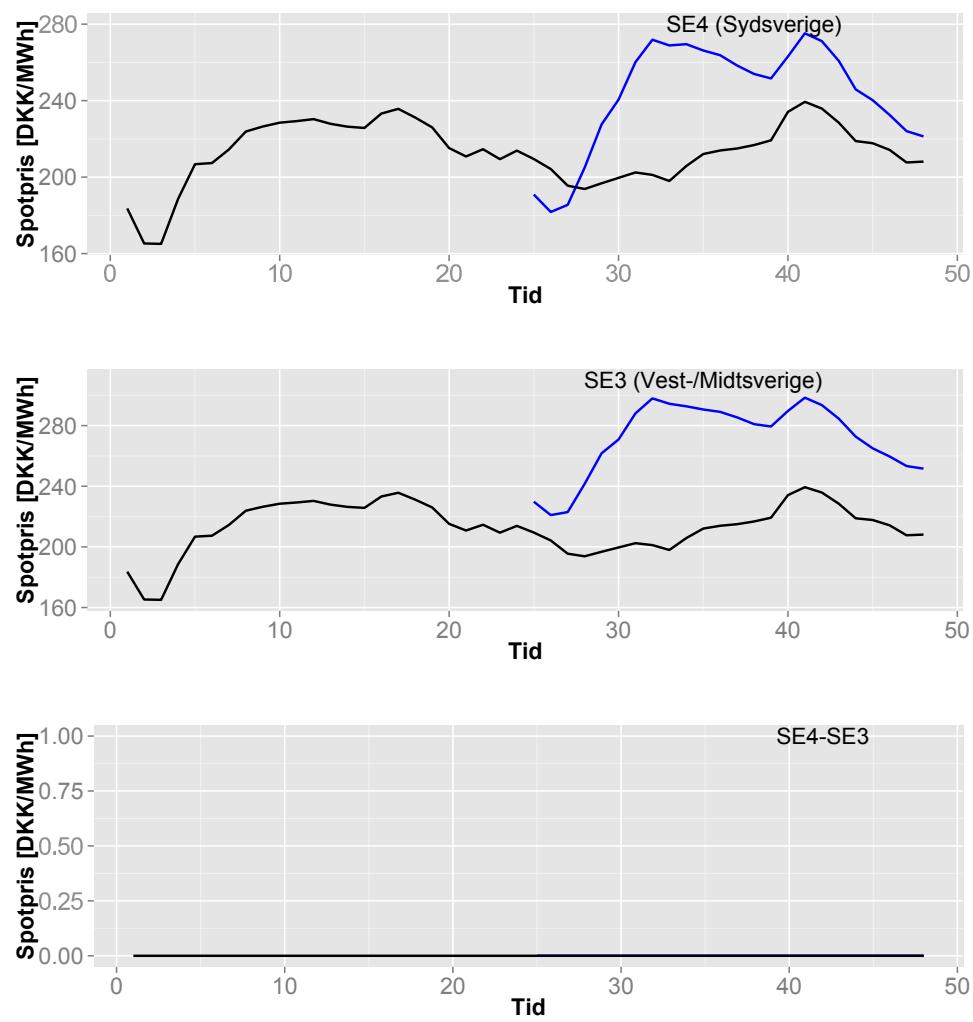
Figur 6.8: Forecast af de sæsonrensede priser for område DK2



Figur 6.9: Plots af forskellige forecasts for forbindelsen DK2-SE4, ved brug af SARFIMA(p, d, q)-modeller med regimeskift.



Figur 6.10: Plots af forskellige forecasts for forbindelsen SE4-SE3, ved brug af SARFIMA(p, d, q)-modeller med regimeskift.



Figur 6.11: Plots af forskellige forecasts for forbindelsen SE4-SE3, ved brug af SARFIMA(p, d, q)-modeller med regimeskift.

Kapitel 7

Konklusion

I dette forløb har vi forsøgt at modellere og forecaste elpriser vha. regimeskiftende og ikke-regimeskiftende SARFIMA(p, d, q)-modeller. Dette speciale har været inspireret af Niels Haldrup og Morten Ørregaard Nielsens artikler *A regime switching long memory model for electricity prices* og *Directional Congestion and Regime Switching in a Long Memory Model for Electricity Prices*.

Data til specialet er hentet fra Nord Pool Spot, hvor elpriserne er frit tilgængelige. For at mindske specialets omfang, har vi valgt kun at arbejde med de fem områder; NO2, SE3, SE4, DK1 og DK2. Imellem disse fem områder er der fem forbindelser, hvorigennem det er muligt at overføre elektricitet. De fem forbindelser er: NO2-DK1, DK1-SE3, DK1-DK2, DK2-SE4 og SE4-SE3. Specialets mål er at kunne modellere og forecaste elpriserne i disse forbindelser ved at anvende SARFIMA(p, d, q)-modeller med og uden regimeskift. Herefter ville vi sammenligne modellernes forecastpræstationer.

Implementeringen af modelestimation og forecast, tager udgangspunkt i koder anvendt af Haldrup og Nielsen i deres artikler. De oprindelige koder er udført i programmeringssproget Ox. Idet vi ikke før har arbejdet med Ox, har vi omarbejdet disse til programmeringssproget R. Dette var en yderst tidskrævende proces, og det har undertiden været nødvendigt selv at programmere diverse funktioner.

Oprindeligt ville vi gerne have implementeret en model med tre regimer, men desværre viste det sig at R-koderne, hvori modellerne med to regimer bliver estimeret, er meget tidskrævende, idet der er mange parametre at estimere. En model med tre regimer ville blot nødvendiggøre estimation af endnu flere ukendte parametre, hvilket ville gøre koden endnu mere tidskrævende. Derfor har vi været nødt til at begrænse os til modellerne med to regimer.

I forbindelse med parameterestimationen for de regimeskiftende modeller, opstod der problemer i optimeringsprocessen, idet den inverse Hessematrix indeholdt negative værdier i diagonalen. Dette medførte, at det ikke var muligt at beregne standard errors for alle parametrene i de regimeskiftende modeller. Desuden var problemerne med Hessematricen en indikation for at optimeringsalgoritmen ikke havde fundet de optimale parameterestimater. Derfor valgte vi at sætte middelværdiparameteren i modellerne lig med sample mean, således at der blev færre parametre, der skulle estimeres.

Forecastene er lavet vha. Monte Carlo simuleringer, hvor vi lavede 1000 MC-stier, der var 24 skridt lange.

Gennem hele forløbet har vi haft en forventning om, at modellerne med regimeskift, ville give bedre forecasts end modellerne uden regimeskift. Dette viste sig dog ikke at være tilfældet. I 11 ud af 15 tilfælde, var det modellerne uden regimeskift, der gav det bedste forecast målt på RMSE. Ud fra RMSE er det faktisk kun de tre forecasts for forbindelsen DK1-DK2, samt forecastet for prisdifferensen mellem SE4 og SE3, hvor de regimeskiftende modeller er at foretrække.

Ved at se på de forskellige forecasts, fremgik det, at modellerne med regimeskift i flere tilfælde i perioder lå meget tæt op af de faktiske priser. Endvidere er modellerne med regimeskift i stand til at opnå en forecastet prisdifferens på nul.

Ud fra vores undersøgelser, er det umiddelbart svært at argumentere for brugen af de regime-skiftende SARFIMA-modeller, idet disse i langt de fleste tilfælde ikke leverer et bedre forecast, end de ikke-regimeskiftende SARFIMA-modeller. Et yderligere argument imod de regimeskiftende modeller er, at disse tager uforholdsmaessigt lang tid at estimere, og forecastene er ligeledes mere tidskrævende. I de tilfælde, hvor SARFIMA-modellerne giver rimelige forecasts af den ikke-sæsonrensede pris, kan dette skyldes, at sæsonfunktionen i høj grad har opfanget variationen i prisen.

Endvidere viste diverse plots af residualerne fra både de ikke-regimeskiftende og de regimeskiftende SARFIMA-modeller, at disse ikke formår at opfange de store, pludselige outliers i data, de såkaldte “spikes”, der er karakteristiske for energipriser.

Litteratur

- Beran, J. (1994). *Statistics for Long-Memory Processes*. Chapman & Hall, 1. udgave. ISBN: 0-412-04901-5.
- Box, G. E. P., Jenkins, G. M. og Reinsel, G. C. (2008). *Time Series Analysis: Forecasting and Control*. John Wiley & Sons Inc, 4. udgave. ISBN: 978-0-470-27284-8.
- Brockwell, P. J. og Davis, R. A. (1991). *Time Series: Theory and Methods*. Springer-Verlag, 2. udgave. ISBN: 0-387-97429-6.
- Granger, C. W. J. og Joyeux, R. (1980). *An Introduction to Long-Memory Time Series Models and Fractional Differencing*. Journal of Time Series Analysis, 1:15-29.
- Haldrup, N. og Nielsen, M. . (2005). *A regime switching long memory model for electricity prices*. Journal of Econometrics, 135:349-376.
- Haldrup, N. og Nielsen, M. . (2006). *Directional Congestion and Regime Switching in a Long Memory Model for Electricity Prices*. Nonlinear Analysis of Electricity Prices, 10:1-22.
- Nielsen, H. B. (2005). *Non-Stationary Time Series and Unit Root Tests*. Set den 22. april 2014. URL: http://www.econ.ku.dk/metrics/Econometrics2_05_II/Slides/08_unitroottests_2pp.pdf.
- Palma, W. (2007). *Long-Memory Time Series - Theory and Methods*. John Wiley & Sons, Inc, 1. udgave. ISBN: 978-0-470-11402-5.
- Stoffer, D. S. og Shumway, R. H. (2011). *Time Series Analysis and Its Applications*. Springer Science+Business Media, LLC, 3. udgave. ISBN: 978-1-4419-7864-6.

LITTERATUR

Bilag A

Koefficienterne for fraktionelle differenser

I dette appendiks vises det, hvorledes ligningerne (1.7) fremkommer. For at udlede ligning (1.9) benyttes samme fremgangsmåde, hvor $-d$ indsættes på d 's plads.

Der haves følgende gammafunktion:

$$\Gamma(x) = \begin{cases} \int_0^\infty t^{x-1} e^{-t} dt, & x > 0, \\ x^{-1} \Gamma(1+x), & x \neq 0, -1, -2, \dots \end{cases} \quad (\text{A.1})$$

Følgende rekursion gælder for alle x :

$$\Gamma(x+1) = x\Gamma(x), \quad (\text{A.2})$$

og for alle heltallige positive x gælder der, at $\Gamma(x+1) = x!$ Det fremgår af ligning (A.1), at der for negative ikke-heltallige x , gælder følgende rekursion:

$$\Gamma(x) = \frac{1}{x} \Gamma(x+1). \quad (\text{A.3})$$

Hvis k er et positivt heltal og $-1 < d < 1$, så vil $k-d > 0$, og vha. rekursionsligningen (A.2) opstilles følgende:

$$\begin{aligned} \Gamma(k-d) &= (k-d-1)\Gamma(k-d-1) \\ \Gamma(k-d-1) &= (k-d-2)\Gamma(k-d-2) \\ \Gamma(k-d-2) &= (k-d-3)\Gamma(k-d-3) \\ &\vdots \\ \Gamma(k-d-(k+1)) &= -d\Gamma(-d). \end{aligned}$$

Ud fra ovenstående følger det, at $\Gamma(k-d)$ kan udtrykkes ved

$$\Gamma(k-d) = (k-d-1)(k-d-2)(k-d-3) \dots (-d)\Gamma(-d). \quad (\text{A.4})$$

Det bemærkes nu, at $d - k + 1 < 1$ og den omvendte rekursion i ligning (A.3) giver:

$$\begin{aligned}\Gamma(d - k + 1) &= \frac{1}{d - k + 1} \Gamma(d - k + 2) \\ \Gamma(d - k + 2) &= \frac{1}{d - k + 2} \Gamma(d - k + 3) \\ \Gamma(d - k + 3) &= \frac{1}{d - k + 3} \Gamma(d - k + 4) \\ &\vdots \\ \Gamma(d - k + k) &= \Gamma(d) = \frac{1}{d} \Gamma(d + 1).\end{aligned}$$

På samme måde som før, er det nu muligt at opstille et udtryk for $\Gamma(d - k + 1)$, givet ved

$$\Gamma(d - k + 1) = \frac{1}{(d - k + 1)} \frac{1}{(d - k + 2)} \frac{1}{(d - k + 3)} \cdots \frac{1}{d} \Gamma(d).$$

Ved at gange samtlige k brøker med -1 , fås følgende:

$$\Gamma(d - k + 1) = (-1)^k \frac{1}{(k - d - 1)} \frac{1}{(k - d - 2)} \frac{1}{(k - d - 3)} \cdots \frac{1}{(-d)} \Gamma(d). \quad (\text{A.5})$$

Ved at sammenligne med ligning (A.4), følger det umiddelbart, at udtrykket i nævnerne i ligning (A.5) kan erstattes med $\Gamma(k - d) \frac{1}{\Gamma(-d)}$. Således fås der at

$$\begin{aligned}\Gamma(d - k + 1) &= (-1)^k \frac{\Gamma(d + 1)}{\Gamma(k - d) \frac{1}{\Gamma(-d)}} \\ &= (-1)^k \frac{\Gamma(d + 1) \Gamma(-d)}{\Gamma(k - d)}.\end{aligned} \quad (\text{A.6})$$

Som nævnt i afsnit 1.1.1, er binomialkoefficienterne givet ved:

$$\binom{d}{k} = \frac{d!}{k!(d - k)!} = \frac{\Gamma(d + 1)}{\Gamma(k + 1)\Gamma(d - k + 1)}.$$

Betrages binomialkoefficienten og udtrykket i ligning (A.6), bemærkes det, at udtrykket i ligning (A.6) kan indsættes i nævneren af binomialkoefficienten, således at følgende fås:

$$\begin{aligned}\binom{d}{k} &= \frac{\Gamma(d + 1)}{\Gamma(k + 1)(-1)^k \frac{\Gamma(d + 1) \Gamma(-d)}{\Gamma(k - d)}} \\ &= \frac{\Gamma(d + 1) \Gamma(k - d)}{\Gamma(k + 1)(-1)^k \Gamma(d + 1) \Gamma(-d)} \\ &= (-1)^k \frac{\Gamma(k - d)}{\Gamma(k + 1) \Gamma(-d)}.\end{aligned}$$

Hermed er udtrykket i ligning (1.7) blevet udledt.

Bilag B

Tests af enhedsrod

I dette appendiks forklares kort teorien bag de forskellige tests, som er blevet anvendt i dette projekt. Appendikset er skrevet på baggrund af [Stoffer og Shumway, 2011, pp. 279-280] og [Nielsen, 2005]

Dickey-Fuller (DF) og Augmented Dickey-Fuller (ADF)

Denne test anvendes til at undersøge, om der findes en enhedsrod i en autoregressiv model, og dermed teste om modellen er stationær.

For en stokastisk proces x_t på formen

$$x_t = \alpha_0 + \varphi x_{t-1} + w_t, \quad \text{for } t = 1, 2, \dots,$$

hvor ε_t er hvid støj, gælder der, at processen er ikke-stationær hvis $\varphi = 1$ og stationær, hvis $|\varphi| < 1$. For den stokastiske proces x_t testes nulhypotesen

$$H_0 : \varphi = 1^1$$

med den alternative hypotese

$$H_a : |\varphi| < 1^2.$$

Ved at sætte $\gamma = \varphi - 1$ fås følgende:

$$\Delta x_t = \alpha_0 + (\varphi - 1)x_{t-1} + w_t = \alpha_0 + \gamma x_{t-1} + w_t.$$

Nulhypotesen og den alternative hypotese er nu givet ved:

$$H_0 : \gamma = 0 \quad \text{og} \quad H_a : \gamma \neq 0.$$

Herefter regresses Δx_t på x_{t-1} og konstanten α_0 . Vha. OLS eller maksimumlikelihood fås estimatorne $\hat{\alpha}_0$ og $\hat{\gamma}$. Efterfølgende kan en teststørrelse opstilles:

$$\tau = \frac{\hat{\gamma}}{se(\hat{\gamma})}.$$

Denne teststørrelse er ikke t-fordelt, men følger i stedet en særlig fordeling i henhold til en Dickey-Fuller-tabel.

¹Når $\varphi = 1$, er der tale om en random walk med drift.

²Når $|\varphi| < 1$, er der tale om en AR(1)-proces.

Der er tale om en nedre ensidet test, hvori nulhypotesen forkastes, når værdien af teststørrelsen er mindre end den kritiske værdi i henhold til det valgte signifikansniveau.

Der findes tre overordnede udgaver af testen³:

$$1 : \Delta x_t = \gamma x_{t-1} + \varepsilon_t \quad (\text{test for random walk})$$

$$2 : \Delta x_t = \alpha_0 + \gamma x_{t-1} + \varepsilon_t \quad (\text{test for random walk med drift})$$

$$3 : \Delta x_t = \alpha_0 + \alpha_1 t + \gamma x_{t-1} + \varepsilon_t \quad (\text{test for random walk med drift. trend}).$$

Nulhypotesen og den alternative hypotese er givet ved:

$$H_0 : \varphi = 1 \quad \text{og} \quad H_a : |\varphi| < 1.$$

Den udvidede Dickey-Fuller-test (ADF) benytter samme fremgangsmåde som DF-testen. Forskellen består i at Δx_t i ADF-testen regresses på x_{t-1} , konstanten α_0 , den lineære trend, samt k laggede værdier $\Delta x_{t-1}, \dots, \Delta x_{t-k}$:

$$\Delta x_t = \alpha_0 + \alpha_1 t + \gamma x_{t-1} + \delta_1 \Delta x_{t-1} + \dots + \delta_k \Delta x_{t-k} + \varepsilon_t.$$

De laggede værdier af førstdifferenserne af x_t introduceres for at tage højde for problemer med seriell korrelation i den fittede regressionsmodel, som benyttes i DF-testen. Antallet af lags afhænger af, hvor mange observationer der er i datasættet. R anvender som default k beregnet ved $(n - 2)^{1/3}$, hvor der afrundes ned til nærmeste heltal.

Phillips-Perron (PP)

PP-testen tager udgangspunkt i DF-testen, og PP-testen anvender regressionsligningen

$$x_t = \alpha_0 + \alpha_1 t + \gamma x_{t-1} + \varepsilon_t.$$

Som i tilfældet med DF-testen kan α_0 og den lineære trend udelades⁴. R anvender som default k beregnet ved $4 \cdot (n/100)^{1/4}$, hvor der afrundes ned til nærmeste heltal.

PP-testen tager ligesom ADF-testen højde for seriell korrelation i den fittede regressionsmodel. PP-testen laver en ikke-parametrisk korrektion af teststørrelsen i DF-testen.

Kwiatkowski-Phillips-Schmidt-Shin (KPSS)

KPSS-testen tester nulhypotesen om at tidsrækken er stationær, hvilket er modsat ift. nulhypoteserne i DF-, ADF- og PP-testene. Udgangspunktet er modellen

$$x_t = \alpha_0 + \alpha_1 t + \mu_t + u_t,$$

hvor u_t er stationær og kan være heteroskedastisk. μ_t er en random walk, givet ved:

$$\mu_t = \mu_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim WN(0, \sigma_\varepsilon^2).$$

³Sidstnævnte version anvendes i R-pakken `tseries` i funktionen `adf.test`.

⁴I R-pakken `tseries` i funktionen `pp.test` indgår regressionsmodellen inkl. konstant og lineær trend.

Nulhypotesen siger at x_t er stationær, og er givet ved:

$$H_0 : \sigma_\varepsilon^2 = 0,$$

hvilket må medføre, at μ_t er en konstant, og dermed er x_t stationær. Den alternative hypotese er givet ved:

$$H_a : \sigma_\varepsilon^2 > 0.$$

R anvender som default k beregnet ved $3\sqrt{n}/13$, hvor der afrundes ned til nærmeste heltal. KPSS teststørrelsen er en Lagrange multiplier teststørrelse, også kaldet score teststørrelse.

Bilag C

Koefficienter til dummyvariablene i forbindelse med sæsonrensning

I dette appendiks er koefficienterne til dummyvariablene i forbindelse med sæsonrensningen anført. Tabellerne C.1 - C.5 viser koefficienterne tilhørende sæsonrensningen, når der haves en model uden regimeskift. Tabellerne C.6 - C.15 viser koefficienterne i forbindelse med en model med regimeskift.

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	229.3339	2.7772	82.58	0.0000
tid	0.0069	0.0001	71.18	0.0000
time01	-3.3651	2.9665	-1.13	0.2566
time02	-3.0515	2.9665	-1.03	0.3037
time03	2.8500	2.9665	0.96	0.3367
time04	11.6368	2.9665	3.92	0.0001
time05	21.9568	2.9665	7.40	0.0000
time06	37.5263	2.9665	12.65	0.0000
time07	44.0708	2.9665	14.86	0.0000
time08	39.9946	2.9665	13.48	0.0000
time09	36.4096	2.9665	12.27	0.0000
time10	32.0708	2.9665	10.81	0.0000
time11	29.3697	2.9665	9.90	0.0000
time12	26.8310	2.9665	9.04	0.0000
time13	25.0467	2.9665	8.44	0.0000
time14	25.3154	2.9665	8.53	0.0000
time15	29.4440	2.9665	9.93	0.0000
time16	38.2527	2.9665	12.90	0.0000
time17	39.7600	2.9665	13.40	0.0000
time18	33.3282	2.9665	11.24	0.0000
time19	26.7963	2.9665	9.03	0.0000
time20	22.3641	2.9665	7.54	0.0000
time21	15.9260	2.9665	5.37	0.0000
time22	11.2155	2.9665	3.78	0.0002
time23	5.5701	2.9654	1.88	0.0604
ugedag.NOtirsdag	1.3267	1.6247	0.82	0.4142
ugedag.NOonsdag	4.6935	1.6339	2.87	0.0041
ugedag.NOtorsdag	2.3404	1.6377	1.43	0.1530
ugedag.NOfredag	-1.7291	1.6296	-1.06	0.2887
ugedag.NOlørdag	-14.2059	1.6173	-8.78	0.0000
ugedag.NOsøndag	-21.1124	1.5583	-13.55	0.0000
maanedfebruar	28.9551	2.1340	13.57	0.0000
maanedmarts	-14.7540	2.0832	-7.08	0.0000
maanedapril	-4.3977	2.1067	-2.09	0.0369

time17	149.1526	4.9790	29.96	0.0000
time18	120.1602	4.9790	24.13	0.0000
time19	94.9859	4.9790	19.08	0.0000
time20	76.0059	4.9790	15.27	0.0000
time21	47.3834	4.9790	9.52	0.0000
time22	27.5699	4.9790	5.54	0.0000
time23	12.9779	4.9773	2.61	0.0091
ugedag.DKtirsdag	4.9479	2.7208	1.82	0.0690
ugedag.DKonsdag	12.8956	2.7363	4.71	0.0000
ugedag.DKtorsdag	9.4042	2.7487	3.42	0.0006
ugedag.DKfredag	-5.5456	2.7424	-2.02	0.0432
ugedag.DKlørdag	-51.6190	2.7145	-19.02	0.0000
ugedag.DKsøndag	-74.8724	2.6189	-28.59	0.0000
maanedfebruar	55.6322	3.5818	15.53	0.0000
maanedmarts	-17.7177	3.4964	-5.07	0.0000
maanedapril	5.6266	3.5372	1.59	0.1117
maanedmaj	-19.4781	3.5217	-5.53	0.0000
maanedjuni	-31.9071	3.5671	-8.94	0.0000
maanedjuli	-59.7561	3.5598	-16.79	0.0000
maanedaugust	0.3876	3.5872	0.11	0.9140
maanedseptember	12.8618	3.6438	3.53	0.0004
maanedoktober	-8.1235	3.6495	-2.23	0.0260
maanednovember	-22.5570	3.7149	-6.07	0.0000
maaneddecember	-37.1402	3.7294	-9.96	0.0000

Tabel C.5: Koefficienterne til den multiple lineære regression for område DK2. Denne har en forklaringsgrad R^2 på 0.3293.

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	228.5247	2.7436	83.29	0.0000
tid.regime0	0.0057	0.0002	36.18	0.0000
time23.regime0	10.2326	3.6682	2.79	0.0053
time01.regime0	0.2798	3.6568	0.08	0.9390
time02.regime0	1.1010	3.6691	0.30	0.7641
time03.regime0	8.5598	3.6988	2.31	0.0207
time04.regime0	17.9681	3.9055	4.60	0.0000
time05.regime0	30.2336	4.1954	7.21	0.0000
time06.regime0	58.0236	4.4214	13.12	0.0000
time07.regime0	74.5593	4.3187	17.26	0.0000
time08.regime0	69.2492	4.3136	16.05	0.0000
time09.regime0	59.7498	4.2842	13.95	0.0000
time10.regime0	51.6808	4.2648	12.12	0.0000
time11.regime0	46.5361	4.1972	11.09	0.0000
time12.regime0	41.7429	4.1652	10.02	0.0000
time13.regime0	38.7201	4.1335	9.37	0.0000
time14.regime0	39.8343	4.1337	9.64	0.0000
time15.regime0	47.4457	4.3553	10.89	0.0000
time16.regime0	70.8987	4.6663	15.19	0.0000
time17.regime0	75.5696	4.5330	16.67	0.0000
time18.regime0	54.0404	4.2601	12.69	0.0000
time19.regime0	36.9777	4.0450	9.14	0.0000
time20.regime0	26.2433	3.9674	6.61	0.0000
time21.regime0	16.7932	3.7520	4.48	0.0000
time22.regime0	17.0572	3.6759	4.64	0.0000
lørdag.regime0	-20.6841	2.3793	-8.69	0.0000
søndag.regime0	-32.3859	2.2727	-14.25	0.0000
tirsdag.regime0	8.7889	2.5127	3.50	0.0005
onsdag.regime0	19.4575	2.5425	7.65	0.0000

BILAG C. KOEFFICIENTER TIL DUMMYVARIABLENE I FORBINDELSE MED
SÆSONRENSNING

torsdag.regime0	11.6738	2.5935	4.50	0.0000
fredag.regime0	-0.6773	2.5063	-0.27	0.7870
december.regime0	-52.2042	3.1934	-16.35	0.0000
februar.regime0	25.1731	2.7700	9.09	0.0000
marts.regime0	-29.2397	2.7835	-10.50	0.0000
april.regime0	-1.5018	2.9847	-0.50	0.6148
maj.regime0	-67.8207	2.9810	-22.75	0.0000
juni.regime0	-94.4412	3.1498	-29.98	0.0000
juli.regime0	-135.3252	2.9178	-46.38	0.0000
august.regime0	-128.1894	4.3248	-29.64	0.0000
september.regime0	-77.6321	6.0719	-12.79	0.0000
oktober.regime0	-48.0226	3.0818	-15.58	0.0000
november.regime0	-61.4478	3.0013	-20.47	0.0000
tid.regime1	0.0078	0.0001	61.35	0.0000
time23.regime1	1.1984	3.9484	0.30	0.7615
time01.regime1	-7.0904	3.9673	-1.79	0.0739
time02.regime1	-6.6992	3.9497	-1.70	0.0899
time03.regime1	-0.9857	3.9144	-0.25	0.8012
time04.regime1	7.5048	3.7347	2.01	0.0445
time05.regime1	16.2192	3.5979	4.51	0.0000
time06.regime1	26.7344	3.5313	7.57	0.0000
time07.regime1	27.0819	3.5652	7.60	0.0000
time08.regime1	23.7728	3.5682	6.66	0.0000
time09.regime1	23.2584	3.5768	6.50	0.0000
time10.regime1	21.0441	3.5792	5.88	0.0000
time11.regime1	19.4094	3.6006	5.39	0.0000
time12.regime1	18.2700	3.6094	5.06	0.0000
time13.regime1	17.0478	3.6232	4.71	0.0000
time14.regime1	17.3161	3.6220	4.78	0.0000
time15.regime1	20.6661	3.5489	5.82	0.0000
time16.regime1	25.0781	3.4788	7.21	0.0000
time17.regime1	23.7938	3.5052	6.79	0.0000
time18.regime1	22.3931	3.5786	6.26	0.0000
time19.regime1	20.2435	3.6681	5.52	0.0000
time20.regime1	18.8237	3.7095	5.07	0.0000
time21.regime1	14.9832	3.8642	3.88	0.0001
time22.regime1	4.9734	3.9445	1.26	0.2074
lørdag.regime1	-11.7556	2.0708	-5.68	0.0000
søndag.regime1	-14.7372	2.0258	-7.27	0.0000
tirsdag.regime1	-3.0935	1.9806	-1.56	0.1183
onsdag.regime1	-4.2968	1.9838	-2.17	0.0303
torsdag.regime1	-2.4208	1.9721	-1.23	0.2196
fredag.regime1	-2.6988	1.9924	-1.35	0.1756
december.regime1	-69.9616	2.8885	-24.22	0.0000
februar.regime1	27.3193	2.9906	9.14	0.0000
marts.regime1	-4.5266	2.8035	-1.61	0.1064
april.regime1	-7.4580	2.7016	-2.76	0.0058
maj.regime1	-50.8051	2.7184	-18.69	0.0000
juni.regime1	-89.0264	2.7130	-32.81	0.0000
juli.regime1	-130.4790	2.8552	-45.70	0.0000
august.regime1	-112.4487	2.5417	-44.24	0.0000
september.regime1	-119.1153	2.5297	-47.09	0.0000
oktober.regime1	-72.0842	2.8581	-25.22	0.0000
november.regime1	-79.7617	3.0215	-26.40	0.0000

Tabel C.6: Koefficienterne til den multiple lineære regression for område NO2 med to regimer. Denne har en forklaringsgrad R² på 0.503.

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	228.8788	3.7443	61.13	0.0000
tid.regime0	0.0057	0.0002	29.86	0.0000
time23.regime0	10.7265	4.8527	2.21	0.0271
time01.regime0	-0.8735	4.8834	-0.18	0.8580
time02.regime0	-0.3184	4.8573	-0.07	0.9477
time03.regime0	6.0989	4.8493	1.26	0.2085
time04.regime0	16.0028	5.0367	3.18	0.0015
time05.regime0	30.2610	5.3443	5.66	0.0000
time06.regime0	54.5072	5.3927	10.11	0.0000
time07.regime0	64.9646	5.2728	12.32	0.0000
time08.regime0	61.5828	5.2277	11.78	0.0000
time09.regime0	54.7340	5.2393	10.45	0.0000
time10.regime0	47.9725	5.3046	9.04	0.0000
time11.regime0	42.6746	5.2550	8.12	0.0000
time12.regime0	37.8408	5.2442	7.22	0.0000
time13.regime0	35.5536	5.2330	6.79	0.0000
time14.regime0	36.4624	5.2685	6.92	0.0000
time15.regime0	41.7098	5.3595	7.78	0.0000
time16.regime0	60.2545	5.4340	11.09	0.0000
time17.regime0	65.1133	5.3394	12.19	0.0000
time18.regime0	51.9232	5.1926	10.00	0.0000
time19.regime0	38.1144	5.0381	7.57	0.0000
time20.regime0	28.8068	5.0179	5.74	0.0000
time21.regime0	20.0779	4.8702	4.12	0.0000
time22.regime0	17.1367	4.8434	3.54	0.0004
lørdag.regime0	-18.0249	2.9549	-6.10	0.0000
søndag.regime0	-30.2776	2.8400	-10.66	0.0000
tirsdag.regime0	8.4982	3.0985	2.74	0.0061
onsdag.regime0	13.1527	3.0994	4.24	0.0000
torsdag.regime0	10.2964	3.1624	3.26	0.0011
fredag.regime0	-3.8054	3.0727	-1.24	0.2156
december.regime0	-53.1873	4.0994	-12.97	0.0000
februar.regime0	23.9053	3.5617	6.71	0.0000
marts.regime0	-31.0775	3.4373	-9.04	0.0000
april.regime0	-13.0265	3.6697	-3.55	0.0004
maj.regime0	-67.8380	3.6380	-18.65	0.0000
juni.regime0	-95.9602	3.8932	-24.65	0.0000
juli.regime0	-130.9252	3.7914	-34.53	0.0000
august.regime0	-92.5042	7.0134	-13.19	0.0000
september.regime0	-76.9537	8.0624	-9.54	0.0000
oktober.regime0	-45.5604	3.7850	-12.04	0.0000
november.regime0	-60.8399	3.7477	-16.23	0.0000
tid.regime1	0.0063	0.0002	32.01	0.0000
time23.regime1	2.9153	5.6687	0.51	0.6071
time01.regime1	-7.6815	5.6167	-1.37	0.1714
time02.regime1	-1.5728	5.6637	-0.28	0.7812
time03.regime1	18.4672	5.6846	3.25	0.0012
time04.regime1	44.2146	5.4247	8.15	0.0000
time05.regime1	75.2025	5.1682	14.55	0.0000
time06.regime1	112.0533	5.1413	21.79	0.0000
time07.regime1	113.7169	5.2271	21.76	0.0000
time08.regime1	104.0886	5.2661	19.77	0.0000
time09.regime1	97.8087	5.2543	18.61	0.0000
time10.regime1	88.9437	5.2015	17.10	0.0000
time11.regime1	80.8099	5.2325	15.44	0.0000
time12.regime1	73.3092	5.2324	14.01	0.0000
time13.regime1	68.6867	5.2428	13.10	0.0000
time14.regime1	69.5528	5.2145	13.34	0.0000
time15.regime1	86.5375	5.1565	16.78	0.0000

BILAG C. KOEFFICIENTER TIL DUMMYVARIABLENE I FORBINDELSE MED
SÆSONRENSNING

time16.regime1	101.6204	5.1156	19.86	0.0000
time17.regime1	94.2563	5.1726	18.22	0.0000
time18.regime1	76.6063	5.2849	14.50	0.0000
time19.regime1	63.7449	5.4446	11.71	0.0000
time20.regime1	53.5603	5.4634	9.80	0.0000
time21.regime1	31.7765	5.6584	5.62	0.0000
time22.regime1	11.8488	5.6929	2.08	0.0374
lørdag.regime1	-43.0768	3.1511	-13.67	0.0000
søndag.regime1	-46.9140	2.9671	-15.81	0.0000
tirsdag.regime1	0.7103	2.9124	0.24	0.8073
onsdag.regime1	7.8182	2.9602	2.64	0.0083
torsdag.regime1	3.4881	2.8959	1.20	0.2284
fredag.regime1	-3.8770	2.9774	-1.30	0.1929
december.regime1	-58.9570	4.2221	-13.96	0.0000
februar.regime1	81.7747	4.6193	17.70	0.0000
marts.regime1	-6.1959	4.8185	-1.29	0.1985
april.regime1	-43.8132	4.1700	-10.51	0.0000
maj.regime1	-62.3559	4.1913	-14.88	0.0000
juni.regime1	-91.4684	4.0283	-22.71	0.0000
juli.regime1	-170.5938	4.0819	-41.79	0.0000
august.regime1	-88.5751	3.5573	-24.90	0.0000
september.regime1	-61.5464	3.6069	-17.06	0.0000
oktober.regime1	-63.0157	4.4437	-14.18	0.0000
november.regime1	-98.9065	4.7919	-20.64	0.0000

Tabel C.7: Koefficienterne til den multiple lineære regression for område SE3 med to regimer. Denne har en forklaringsgrad R^2 på 0.39.

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	231.7721	3.9851	58.16	0.0000
tid.regime0	0.0056	0.0002	26.13	0.0000
time23.regime0	8.2596	5.1631	1.60	0.1097
time01.regime0	-3.5055	5.1955	-0.67	0.4999
time02.regime0	-2.8942	5.1676	-0.56	0.5754
time03.regime0	3.6283	5.1628	0.70	0.4822
time04.regime0	13.6689	5.3872	2.54	0.0112
time05.regime0	28.2796	5.7781	4.89	0.0000
time06.regime0	53.7238	5.8991	9.11	0.0000
time07.regime0	65.8709	5.7956	11.37	0.0000
time08.regime0	63.4525	5.8098	10.92	0.0000
time09.regime0	56.2851	5.8061	9.69	0.0000
time10.regime0	47.7467	5.8262	8.20	0.0000
time11.regime0	42.2708	5.7475	7.35	0.0000
time12.regime0	37.1622	5.7184	6.50	0.0000
time13.regime0	34.8408	5.7052	6.11	0.0000
time14.regime0	35.9791	5.7253	6.28	0.0000
time15.regime0	41.1611	5.8339	7.06	0.0000
time16.regime0	61.4031	5.9679	10.29	0.0000
time17.regime0	67.2653	5.8958	11.41	0.0000
time18.regime0	53.1924	5.7624	9.23	0.0000
time19.regime0	36.9826	5.5597	6.65	0.0000
time20.regime0	26.4287	5.4791	4.82	0.0000
time21.regime0	17.1006	5.2709	3.24	0.0012
time22.regime0	14.6853	5.1628	2.84	0.0045
lørdag.regime0	-18.5462	3.2149	-5.77	0.0000
søndag.regime0	-32.2012	3.0744	-10.47	0.0000
tirsdag.regime0	9.0724	3.3843	2.68	0.0074
onsdag.regime0	15.2542	3.4046	4.48	0.0000

torsdag.regime0	12.2295	3.5005	3.49	0.0005
fredag.regime0	-3.0985	3.3666	-0.92	0.3574
december.regime0	-52.0040	4.4050	-11.81	0.0000
februar.regime0	22.8828	3.8551	5.94	0.0000
marts.regime0	-31.6572	3.6870	-8.59	0.0000
april.regime0	-7.5240	4.0509	-1.86	0.0633
maj.regime0	-68.2594	3.9490	-17.29	0.0000
juni.regime0	-93.1655	4.3422	-21.46	0.0000
juli.regime0	-129.8002	4.2579	-30.48	0.0000
august.regime0	-91.1688	7.4785	-12.19	0.0000
september.regime0	-75.3409	8.5892	-8.77	0.0000
oktober.regime0	-46.4346	4.0344	-11.51	0.0000
november.regime0	-60.6799	4.0175	-15.10	0.0000
tid.regime1	0.0043	0.0002	21.16	0.0000
time23.regime1	5.9723	6.0171	0.99	0.3209
time01.regime1	-3.7701	5.9617	-0.63	0.5271
time02.regime1	2.0650	6.0114	0.34	0.7312
time03.regime1	20.8419	6.0279	3.46	0.0005
time04.regime1	47.2214	5.7310	8.24	0.0000
time05.regime1	83.3353	5.4358	15.33	0.0000
time06.regime1	124.2250	5.3773	23.10	0.0000
time07.regime1	127.7570	5.4370	23.50	0.0000
time08.regime1	120.2315	5.4313	22.14	0.0000
time09.regime1	115.9745	5.4304	21.36	0.0000
time10.regime1	105.4792	5.4157	19.48	0.0000
time11.regime1	95.1794	5.4540	17.45	0.0000
time12.regime1	86.7172	5.4653	15.87	0.0000
time13.regime1	79.8282	5.4755	14.58	0.0000
time14.regime1	79.3406	5.4622	14.53	0.0000
time15.regime1	99.2615	5.4066	18.36	0.0000
time16.regime1	121.1603	5.3452	22.67	0.0000
time17.regime1	114.2248	5.3776	21.24	0.0000
time18.regime1	92.2802	5.4478	16.94	0.0000
time19.regime1	77.4127	5.5924	13.84	0.0000
time20.regime1	63.5829	5.6624	11.23	0.0000
time21.regime1	40.7010	5.8694	6.93	0.0000
time22.regime1	15.7034	6.0239	2.61	0.0091
lørdag.regime1	-48.6612	3.2498	-14.97	0.0000
søndag.regime1	-57.7792	3.1023	-18.62	0.0000
tirsdag.regime1	0.2203	3.0261	0.07	0.9420
onsdag.regime1	7.0506	3.0589	2.30	0.0212
torsdag.regime1	0.5827	2.9893	0.19	0.8454
fredag.regime1	-7.8242	3.0764	-2.54	0.0110
december.regime1	-34.4204	4.4447	-7.74	0.0000
februar.regime1	90.9852	4.6620	19.52	0.0000
marts.regime1	5.1252	4.9163	1.04	0.2972
april.regime1	-25.9585	4.2016	-6.18	0.0000
maj.regime1	-44.4500	4.3323	-10.26	0.0000
juni.regime1	-36.5030	4.1270	-8.84	0.0000
juli.regime1	-118.7331	4.1353	-28.71	0.0000
august.regime1	-76.2243	3.7657	-20.24	0.0000
september.regime1	-47.1640	3.8140	-12.37	0.0000
oktober.regime1	-43.7113	4.6719	-9.36	0.0000
november.regime1	-52.3410	4.9807	-10.51	0.0000

Tabel C.8: Koefficienterne til den multiple lineære regression for område SE4 med to regimer. Denne har en forklaringsgrad R^2 på 0.3681.

BILAG C. KOEFFICIENTER TIL DUMMYVARIABLENE I FORBINDELSE MED
SÆSONRENSNING

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	229.6523	4.0949	56.08	0.0000
tid.regime0	0.0056	0.0002	24.39	0.0000
time23.regime0	10.0370	5.6279	1.78	0.0745
time01.regime0	2.2451	5.4925	0.41	0.6827
time02.regime0	3.2773	5.5526	0.59	0.5550
time03.regime0	7.5842	5.7367	1.32	0.1862
time04.regime0	17.6948	6.2748	2.82	0.0048
time05.regime0	33.1232	6.5993	5.02	0.0000
time06.regime0	64.2199	6.4096	10.02	0.0000
time07.regime0	73.9174	6.1523	12.01	0.0000
time08.regime0	66.6138	6.0819	10.95	0.0000
time09.regime0	57.7658	6.0853	9.49	0.0000
time10.regime0	49.6258	6.0627	8.19	0.0000
time11.regime0	44.6509	5.9706	7.48	0.0000
time12.regime0	41.2752	5.9124	6.98	0.0000
time13.regime0	37.9836	5.8955	6.44	0.0000
time14.regime0	38.3998	6.0024	6.40	0.0000
time15.regime0	45.5419	6.4062	7.11	0.0000
time16.regime0	70.7802	6.6870	10.58	0.0000
time17.regime0	70.4733	6.4468	10.93	0.0000
time18.regime0	52.3130	6.1622	8.49	0.0000
time19.regime0	37.5985	6.0156	6.25	0.0000
time20.regime0	26.3977	6.0241	4.38	0.0000
time21.regime0	17.7656	5.7645	3.08	0.0021
time22.regime0	15.5065	5.7565	2.69	0.0071
lørdag.regime0	-16.2782	3.5208	-4.62	0.0000
søndag.regime0	-27.6454	3.3878	-8.16	0.0000
tirsdag.regime0	6.7259	3.6880	1.82	0.0682
onsdag.regime0	19.7393	3.7852	5.21	0.0000
torsdag.regime0	9.4887	3.7736	2.51	0.0119
fredag.regime0	1.6115	3.6867	0.44	0.6620
december.regime0	-48.2296	4.7964	-10.06	0.0000
februar.regime0	33.2858	4.1298	8.06	0.0000
marts.regime0	-28.9248	4.1852	-6.91	0.0000
april.regime0	-2.7491	4.6011	-0.60	0.5502
maj.regime0	-72.5249	4.5648	-15.89	0.0000
juni.regime0	-97.4722	4.7525	-20.51	0.0000
juli.regime0	-146.0247	4.3294	-33.73	0.0000
august.regime0	-118.8483	5.7623	-20.63	0.0000
september.regime0	-86.5270	7.3662	-11.75	0.0000
oktober.regime0	-56.2060	4.4954	-12.50	0.0000
november.regime0	-67.2913	4.5093	-14.92	0.0000
tid.regime1	-0.0003	0.0002	-1.81	0.0708
time23.regime1	12.1331	5.7532	2.11	0.0350
time01.regime1	-17.0346	5.9217	-2.88	0.0040
time02.regime1	-18.5990	5.8420	-3.18	0.0015
time03.regime1	-2.7280	5.6620	-0.48	0.6300
time04.regime1	40.4432	5.3634	7.54	0.0000
time05.regime1	92.5388	5.2685	17.56	0.0000
time06.regime1	133.8058	5.3317	25.10	0.0000
time07.regime1	138.9139	5.4402	25.53	0.0000
time08.regime1	134.1935	5.4779	24.50	0.0000
time09.regime1	131.5486	5.4763	24.02	0.0000
time10.regime1	116.9608	5.4827	21.33	0.0000
time11.regime1	107.1847	5.5230	19.41	0.0000
time12.regime1	98.1170	5.5519	17.67	0.0000
time13.regime1	89.4926	5.5662	16.08	0.0000
time14.regime1	87.0727	5.5039	15.82	0.0000
time15.regime1	109.0473	5.3304	20.46	0.0000

time16.regime1	138.0043	5.2551	26.26	0.0000
time17.regime1	146.6637	5.3255	27.54	0.0000
time18.regime1	128.2196	5.4343	23.59	0.0000
time19.regime1	106.9152	5.5019	19.43	0.0000
time20.regime1	89.0392	5.4941	16.21	0.0000
time21.regime1	56.0402	5.6477	9.92	0.0000
time22.regime1	27.6757	5.6454	4.90	0.0000
lørdag.regime1	-54.2827	3.1274	-17.36	0.0000
søndag.regime1	-89.7970	3.0446	-29.49	0.0000
tirsdag.regime1	-0.9775	2.9955	-0.33	0.7442
onsdag.regime1	6.6242	2.9762	2.23	0.0260
torsdag.regime1	4.3993	3.0077	1.46	0.1436
fredag.regime1	-11.6008	3.0369	-3.82	0.0001
december.regime1	-61.0300	4.3020	-14.19	0.0000
februar.regime1	50.0639	4.5267	11.06	0.0000
marts.regime1	-8.3500	4.1843	-2.00	0.0460
april.regime1	4.0515	3.9802	1.02	0.3087
maj.regime1	0.0154	4.0173	0.00	0.9969
juni.regime1	-3.8917	4.0405	-0.96	0.3355
juli.regime1	-11.7994	4.3050	-2.74	0.0061
august.regime1	22.9809	3.8392	5.99	0.0000
september.regime1	21.8417	3.8037	5.74	0.0000
oktober.regime1	-8.9146	4.3425	-2.05	0.0401
november.regime1	-19.4045	4.4979	-4.31	0.0000

Tabel C.9: Koefficienterne til den multiple lineære regression for område DK1 med to regimer. Denne har en forklaringsgrad R² på 0.4085.

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	222.2957	4.4389	50.08	0.0000
tid.regime0	0.0058	0.0003	22.66	0.0000
time23.regime0	13.5392	5.9346	2.28	0.0225
time01.regime0	3.5356	5.9161	0.60	0.5501
time02.regime0	4.3559	5.9360	0.73	0.4631
time03.regime0	11.8648	5.9840	1.98	0.0474
time04.regime0	21.0954	6.3184	3.34	0.0008
time05.regime0	33.3738	6.7878	4.92	0.0000
time06.regime0	61.0000	7.1535	8.53	0.0000
time07.regime0	77.6776	6.9871	11.12	0.0000
time08.regime0	72.3514	6.9790	10.37	0.0000
time09.regime0	62.8602	6.9316	9.07	0.0000
time10.regime0	54.7377	6.9002	7.93	0.0000
time11.regime0	49.5330	6.7902	7.29	0.0000
time12.regime0	44.7252	6.7384	6.64	0.0000
time13.regime0	41.7061	6.6870	6.24	0.0000
time14.regime0	42.7186	6.6869	6.39	0.0000
time15.regime0	50.5271	7.0464	7.17	0.0000
time16.regime0	74.1157	7.5498	9.82	0.0000
time17.regime0	78.6571	7.3337	10.73	0.0000
time18.regime0	56.9969	6.8917	8.27	0.0000
time19.regime0	40.0333	6.5437	6.12	0.0000
time20.regime0	29.3750	6.4184	4.58	0.0000
time21.regime0	19.9942	6.0701	3.29	0.0010
time22.regime0	20.3222	5.9471	3.42	0.0006
lørdag.regime0	-19.2724	3.8492	-5.01	0.0000
søndag.regime0	-30.5707	3.6801	-8.31	0.0000
tirsdag.regime0	9.0836	4.0401	2.25	0.0246
onsdag.regime0	20.8958	4.1132	5.08	0.0000

BILAG C. KOEFFICIENTER TIL DUMMYVARIABLENE I FORBINDELSE MED
SÆSONRENSNING

torsdag.regime0	13.1073	4.1957	3.12	0.0018
fredag.regime0	0.5761	4.0640	0.14	0.8873
december.regime0	-51.0195	5.1659	-9.88	0.0000
februar.regime0	26.9531	4.4815	6.01	0.0000
marts.regime0	-27.4779	4.5033	-6.10	0.0000
april.regime0	0.5381	4.8295	0.11	0.9113
maj.regime0	-67.6520	4.8137	-14.05	0.0000
juni.regime0	-93.0962	5.0943	-18.27	0.0000
juli.regime0	-133.7837	4.7199	-28.34	0.0000
august.regime0	-126.6179	6.9968	-18.10	0.0000
september.regime0	-76.4097	9.8217	-7.78	0.0000
oktober.regime0	-46.3881	4.9857	-9.30	0.0000
november.regime0	-60.0889	4.8552	-12.38	0.0000
tid.regime1	-0.0003	0.0002	-1.55	0.1216
time23.regime1	8.6900	6.3879	1.36	0.1737
time01.regime1	-15.9268	6.4183	-2.48	0.0131
time02.regime1	-16.2876	6.3899	-2.55	0.0108
time03.regime1	1.7188	6.3329	0.27	0.7861
time04.regime1	45.6473	6.0421	7.55	0.0000
time05.regime1	101.2178	5.8209	17.39	0.0000
time06.regime1	151.5474	5.7132	26.53	0.0000
time07.regime1	155.6571	5.7680	26.99	0.0000
time08.regime1	148.8279	5.7729	25.78	0.0000
time09.regime1	146.0913	5.7869	25.25	0.0000
time10.regime1	133.9873	5.7908	23.14	0.0000
time11.regime1	120.6836	5.8253	20.72	0.0000
time12.regime1	110.5846	5.8394	18.94	0.0000
time13.regime1	102.1676	5.8617	17.43	0.0000
time14.regime1	102.4804	5.8597	17.49	0.0000
time15.regime1	128.6062	5.7416	22.40	0.0000
time16.regime1	162.3162	5.6283	28.84	0.0000
time17.regime1	164.4454	5.6709	29.00	0.0000
time18.regime1	139.0279	5.7895	24.01	0.0000
time19.regime1	115.5317	5.9344	19.47	0.0000
time20.regime1	95.7866	6.0012	15.96	0.0000
time21.regime1	61.7901	6.2516	9.88	0.0000
time22.regime1	29.2625	6.3815	4.59	0.0000
lørdag.regime1	-58.7141	3.3501	-17.53	0.0000
søndag.regime1	-89.2719	3.2824	-27.20	0.0000
tirsdag.regime1	0.9028	3.2029	0.28	0.7781
onsdag.regime1	6.0811	3.1995	1.90	0.0574
torsdag.regime1	1.6481	3.1905	0.52	0.6055
fredag.regime1	-9.3658	3.2333	-2.90	0.0038
december.regime1	-35.7007	4.6740	-7.64	0.0000
februar.regime1	108.6706	4.8382	22.46	0.0000
marts.regime1	1.5623	4.5358	0.34	0.7305
april.regime1	12.7310	4.3736	2.91	0.0036
maj.regime1	3.2605	4.3987	0.74	0.4586
juni.regime1	-13.6924	4.3887	-3.12	0.0018
juli.regime1	-12.8786	4.6188	-2.79	0.0053
august.regime1	25.7857	4.1116	6.27	0.0000
september.regime1	28.6074	4.0925	6.99	0.0000
oktober.regime1	22.3445	4.6233	4.83	0.0000
november.regime1	6.1510	4.8876	1.26	0.2082

Tabel C.10: Koefficienterne til den multiple lineære regression for område DK2 med to regimer. Denne har en forklaringsgrad R^2 på 0.4068.

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.2092	5.9728	-0.04	0.9721
tid.regime1	0.0080	0.0002	40.53	0.0000
time23.regime1	-11.9302	6.4412	-1.85	0.0640
time01.regime1	7.5379	6.6045	1.14	0.2538
time02.regime1	9.0420	6.5306	1.38	0.1662
time03.regime1	0.1947	6.3692	0.03	0.9756
time04.regime1	-34.4906	6.0984	-5.66	0.0000
time05.regime1	-79.6017	6.0161	-13.23	0.0000
time06.regime1	-114.2312	6.0730	-18.81	0.0000
time07.regime1	-117.9435	6.1662	-19.13	0.0000
time08.regime1	-116.5428	6.2024	-18.79	0.0000
time09.regime1	-113.6992	6.2022	-18.33	0.0000
time10.regime1	-100.5661	6.2130	-16.19	0.0000
time11.regime1	-91.4402	6.2519	-14.63	0.0000
time12.regime1	-83.9888	6.2803	-13.37	0.0000
time13.regime1	-75.4926	6.2871	-12.01	0.0000
time14.regime1	-73.0407	6.2258	-11.73	0.0000
time15.regime1	-92.2957	6.0643	-15.22	0.0000
time16.regime1	-118.4596	5.9971	-19.75	0.0000
time17.regime1	-126.9823	6.0615	-20.95	0.0000
time18.regime1	-110.2648	6.1626	-17.89	0.0000
time19.regime1	-91.8549	6.2259	-14.75	0.0000
time20.regime1	-74.5301	6.2074	-12.01	0.0000
time21.regime1	-45.4279	6.3462	-7.16	0.0000
time22.regime1	-24.1953	6.3422	-3.81	0.0001
lørdag.regime1	41.2671	3.2788	12.59	0.0000
søndag.regime1	74.7856	3.1926	23.43	0.0000
tirsdag.regime1	-2.5612	3.1481	-0.81	0.4159
onsdag.regime1	-10.9184	3.1353	-3.48	0.0005
torsdag.regime1	-6.7040	3.1532	-2.13	0.0335
fredag.regime1	7.9018	3.1737	2.49	0.0128
december.regime1	-11.0803	4.5335	-2.44	0.0145
februar.regime1	-36.9642	4.8009	-7.70	0.0000
marts.regime1	3.9663	4.4436	0.89	0.3721
april.regime1	-5.8467	4.2354	-1.38	0.1675
maj.regime1	-50.3404	4.2634	-11.81	0.0000
juni.regime1	-81.7467	4.2873	-19.07	0.0000
juli.regime1	-107.8168	4.5352	-23.77	0.0000
august.regime1	-131.8828	4.0754	-32.36	0.0000
september.regime1	-137.8909	4.0394	-34.14	0.0000
oktober.regime1	-56.0568	4.5694	-12.27	0.0000
november.regime1	-55.1988	4.7247	-11.68	0.0000

Tabel C.11: Koefficienterne til den multiple lineære regression for prisforskellen mellem områderne NO2 og DK1. Denne har en forklaringsgrad R^2 på 0.5089.

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-22.7479	8.3284	-2.73	0.0063
tid.regime1	-0.0076	0.0003	-27.62	0.0000
time23.regime1	13.9347	8.6197	1.62	0.1060
time01.regime1	-6.7338	8.7833	-0.77	0.4433
time02.regime1	-9.1641	8.6198	-1.06	0.2878
time03.regime1	-4.5062	8.2636	-0.55	0.5856
time04.regime1	25.4009	8.0105	3.17	0.0015
time05.regime1	57.9767	8.0665	7.19	0.0000

BILAG C. KOEFFICIENTER TIL DUMMYVARIABLENE I FORBINDELSE MED
SÆSONRENSNING

time06.regime1	69.2202	8.1429	8.50	0.0000
time07.regime1	76.2916	8.3248	9.16	0.0000
time08.regime1	79.5643	8.3895	9.48	0.0000
time09.regime1	81.1720	8.3970	9.67	0.0000
time10.regime1	68.7023	8.4014	8.18	0.0000
time11.regime1	63.4655	8.4672	7.50	0.0000
time12.regime1	59.0079	8.5001	6.94	0.0000
time13.regime1	54.2099	8.5426	6.35	0.0000
time14.regime1	50.4025	8.4356	5.97	0.0000
time15.regime1	61.6160	8.1992	7.51	0.0000
time16.regime1	84.5062	8.0791	10.46	0.0000
time17.regime1	101.0228	8.2200	12.29	0.0000
time18.regime1	93.4568	8.3757	11.16	0.0000
time19.regime1	80.8352	8.3647	9.66	0.0000
time20.regime1	69.8312	8.3640	8.35	0.0000
time21.regime1	45.2167	8.6047	5.25	0.0000
time22.regime1	24.9797	8.5579	2.92	0.0035
lørdag.regime1	-20.1652	4.2857	-4.71	0.0000
søndag.regime1	-56.9255	4.1570	-13.69	0.0000
tirsdag.regime1	6.0067	4.2737	1.41	0.1599
onsdag.regime1	5.8189	4.2447	1.37	0.1705
torsdag.regime1	1.3897	4.1956	0.33	0.7405
fredag.regime1	4.3412	4.2224	1.03	0.3039
december.regime1	4.8683	6.4058	0.76	0.4473
februar.regime1	-5.5631	7.2124	-0.77	0.4405
marts.regime1	19.4076	6.3019	3.08	0.0021
april.regime1	43.7044	6.5828	6.64	0.0000
maj.regime1	71.1578	6.3694	11.17	0.0000
juni.regime1	112.3899	6.4038	17.55	0.0000
juli.regime1	133.6850	6.1142	21.86	0.0000
august.regime1	119.1217	5.9823	19.91	0.0000
september.regime1	99.3433	6.1583	16.13	0.0000
oktober.regime1	52.6408	6.3046	8.35	0.0000
november.regime1	74.9158	6.7054	11.17	0.0000

Tabel C.12: Koefficienterne til den multiple lineære regression for prisforskellen mellem områderne DK1 og SE3. Denne har en forklaringsgrad R² på 0.3808.

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-11.3129	11.5703	-0.98	0.3283
tid.regime1	0.0024	0.0004	6.16	0.0000
time23.regime1	7.6794	12.1490	0.63	0.5274
time01.regime1	-9.4299	13.3096	-0.71	0.4787
time02.regime1	-5.1087	12.7006	-0.40	0.6875
time03.regime1	-3.9220	11.5743	-0.34	0.7347
time04.regime1	6.9757	11.0446	0.63	0.5277
time05.regime1	7.9781	11.3116	0.71	0.4807
time06.regime1	-35.6026	11.5474	-3.08	0.0021
time07.regime1	-41.6466	11.5971	-3.59	0.0003
time08.regime1	-38.1560	11.4961	-3.32	0.0009
time09.regime1	-35.8753	11.4111	-3.14	0.0017
time10.regime1	-35.9211	11.2716	-3.19	0.0015
time11.regime1	-32.6365	11.3249	-2.88	0.0040
time12.regime1	-29.1546	11.4330	-2.55	0.0108
time13.regime1	-22.8136	11.3267	-2.01	0.0441
time14.regime1	-23.0410	11.2761	-2.04	0.0411
time15.regime1	-32.1484	11.0555	-2.91	0.0037

time16.regime1	-56.2565	11.2724	-4.99	0.0000
time17.regime1	-57.3692	11.6390	-4.93	0.0000
time18.regime1	-32.4586	12.0794	-2.69	0.0072
time19.regime1	-10.3182	11.9517	-0.86	0.3880
time20.regime1	-3.4526	11.9661	-0.29	0.7730
time21.regime1	-1.0346	11.9620	-0.09	0.9311
time22.regime1	0.8986	11.7726	0.08	0.9392
lørdag.regime1	19.9962	5.5257	3.62	0.0003
søndag.regime1	4.7207	5.3577	0.88	0.3783
tirsdag.regime1	-2.8781	5.3524	-0.54	0.5908
onsdag.regime1	3.9094	5.4178	0.72	0.4706
torsdag.regime1	4.2681	5.2711	0.81	0.4182
fredag.regime1	1.2632	5.3175	0.24	0.8122
december.regime1	-60.2275	7.9429	-7.58	0.0000
februar.regime1	-107.0373	8.1765	-13.09	0.0000
marts.regime1	-30.3936	8.3804	-3.63	0.0003
april.regime1	-34.1586	8.3375	-4.10	0.0000
maj.regime1	-10.6989	8.7399	-1.22	0.2210
juni.regime1	18.8798	8.9325	2.11	0.0346
juli.regime1	-62.6186	8.4473	-7.41	0.0000
august.regime1	-38.2058	8.1504	-4.69	0.0000
september.regime1	-58.3853	8.4582	-6.90	0.0000
oktober.regime1	-69.2905	8.1526	-8.50	0.0000
november.regime1	-69.4864	8.1852	-8.49	0.0000

Tabel C.13: Koefficienterne til den multiple lineære regression for prisforskellen mellem områderne DK1 og DK2. Denne har en forklaringsgrad R^2 på 0.1539.

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-32.2463	10.8111	-2.98	0.0029
tid.regime1	-0.0064	0.0004	-16.69	0.0000
time23.regime1	15.1548	10.7326	1.41	0.1580
time01.regime1	-5.5152	10.5825	-0.52	0.6023
time02.regime1	-10.9634	10.4934	-1.04	0.2962
time03.regime1	-6.1254	10.4531	-0.59	0.5579
time04.regime1	28.1284	10.5627	2.66	0.0078
time05.regime1	68.0201	10.3160	6.59	0.0000
time06.regime1	94.1411	10.1243	9.30	0.0000
time07.regime1	108.1497	10.3776	10.42	0.0000
time08.regime1	106.7138	10.6148	10.05	0.0000
time09.regime1	109.1853	10.7528	10.15	0.0000
time10.regime1	104.5973	10.8783	9.62	0.0000
time11.regime1	97.7115	11.0911	8.81	0.0000
time12.regime1	92.7110	11.1790	8.29	0.0000
time13.regime1	86.3545	11.3683	7.60	0.0000
time14.regime1	79.9460	11.3351	7.05	0.0000
time15.regime1	99.3014	10.5048	9.45	0.0000
time16.regime1	120.6764	9.9254	12.16	0.0000
time17.regime1	132.0355	9.9578	13.26	0.0000
time18.regime1	115.4386	10.2514	11.26	0.0000
time19.regime1	98.0113	10.4708	9.36	0.0000
time20.regime1	86.7226	10.6895	8.11	0.0000
time21.regime1	56.7386	11.2724	5.03	0.0000
time22.regime1	39.0819	11.1405	3.51	0.0005
lørdag.regime1	-18.9895	5.7554	-3.30	0.0010
søndag.regime1	-66.4161	5.7353	-11.58	0.0000
tirsdag.regime1	6.3974	5.5613	1.15	0.2501

BILAG C. KOEFFICIENTER TIL DUMMYVARIABLENE I FORBINDELSE MED
SÆSONRENSNING

onsdag.regime1	9.4843	5.5546	1.71	0.0878
torsdag.regime1	1.0807	5.5909	0.19	0.8467
fredag.regime1	7.4207	5.6573	1.31	0.1897
december.regime1	-29.4879	9.1760	-3.21	0.0013
februar.regime1	55.6586	9.6633	5.76	0.0000
marts.regime1	2.7376	8.3023	0.33	0.7416
april.regime1	30.8281	9.0926	3.39	0.0007
maj.regime1	58.1873	8.6932	6.69	0.0000
juni.regime1	51.5060	10.0905	5.10	0.0000
juli.regime1	124.2114	8.3639	14.85	0.0000
august.regime1	110.2253	8.0291	13.73	0.0000
september.regime1	96.5733	8.4369	11.45	0.0000
oktober.regime1	69.5887	8.7324	7.97	0.0000
november.regime1	74.2104	9.1137	8.14	0.0000

Tabel C.14: Koefficienterne til den multiple lineære regression for prisforskellen mellem områderne DK2 og SE4. Denne har en forklaringsgrad R² på 0.4301.

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	105.2498	65.9587	1.60	0.1108
tid.regime1	-0.0122	0.0006	-20.88	0.0000
time23.regime1	9.8596	91.6805	0.11	0.9144
time01.regime1	-32.7578	91.4967	-0.36	0.7204
time02.regime1	-36.9157	91.4967	-0.40	0.6867
time03.regime1	-10.8347	79.2916	-0.14	0.8913
time04.regime1	-1.8065	65.9685	-0.03	0.9782
time05.regime1	43.8829	65.4482	0.67	0.5027
time06.regime1	53.7714	65.2639	0.82	0.4101
time07.regime1	58.6595	65.2247	0.90	0.3686
time08.regime1	60.1814	65.1816	0.92	0.3560
time09.regime1	64.1636	65.1819	0.98	0.3251
time10.regime1	54.8530	65.2115	0.84	0.4004
time11.regime1	54.5595	65.2601	0.84	0.4033
time12.regime1	45.3926	65.2838	0.70	0.4870
time13.regime1	33.9005	65.2895	0.52	0.6037
time14.regime1	29.6501	65.3306	0.45	0.6500
time15.regime1	44.5492	65.2835	0.68	0.4951
time16.regime1	74.9731	65.1999	1.15	0.2504
time17.regime1	73.8857	65.1841	1.13	0.2572
time18.regime1	48.2551	65.1928	0.74	0.4593
time19.regime1	37.0351	65.2364	0.57	0.5703
time20.regime1	33.0825	65.4480	0.51	0.6133
time21.regime1	14.4990	66.0048	0.22	0.8262
time22.regime1	8.9675	72.4429	0.12	0.9015
lørdag.regime1	-19.7517	7.7748	-2.54	0.0112
søndag.regime1	-83.9312	9.8614	-8.51	0.0000
tirsdag.regime1	3.3734	5.6517	0.60	0.5507
onsdag.regime1	13.9059	5.8544	2.38	0.0177
torsdag.regime1	5.7876	5.8452	0.99	0.3223
fredag.regime1	4.0172	5.9740	0.67	0.5014
december.regime1	103.6264	13.2824	7.80	0.0000
februar.regime1	7.3974	12.9944	0.57	0.5693
marts.regime1	-71.0867	14.9582	-4.75	0.0000
april.regime1	-48.8760	12.2552	-3.99	0.0001
maj.regime1	-18.7269	12.7092	-1.47	0.1408
juni.regime1	35.3876	11.6110	3.05	0.0023
juli.regime1	73.4785	11.5678	6.35	0.0000

august.regime1	-7.0767	14.0115	-0.51	0.6136
september.regime1	72.3129	14.8369	4.87	0.0000
oktober.regime1	8.6323	26.8766	0.32	0.7481
november.regime1	143.5136	12.9599	11.07	0.0000

Tabel C.15: Koefficienterne til den multiple lineære regression for prisforskellen mellem områderne SE4 og SE3. Denne har en forklaringsgrad R^2 på 0.3553.

BILAG C. KOEFFICIENTER TIL DUMMYVARIABLENE I FORBINDELSE MED
SÆSONRENSNING

Bilag D

R-koder

Dette appendiks indeholder et udvalg af de anvendte R-koder fra projektet.

Estimation af modeller

```
1 AdjustSeason <- function(omraade){
2
3   data <- spotpris.ny
4
5   for(t in unique(data$time)) {
6     data[paste("time",t,sep="")] <- ifelse(data$time==t,1,0) # Laver en dummy-vektor for
        hver time (-1)
7   }
8
9   eval(parse(text=
10   paste("
11   for(t in unique(data$ugedag .,substr(omraade,start=1,stop=2),"))
12   data[t] <- ifelse(data$ugedag .,substr(omraade,start=1,stop=2),"==t,1,0) # Laver en
        dummy-vektor for hver ugedag (-1)
13   ),sep="")))
14
15   for(t in unique(data$maaned)) {
16     data[t] <- ifelse(data$maaned==t,1,0) # Laver en dummy-vektor for hver måned (-1)
17   }
18
19   seasonreg <- data[,-c(1:17,20,45,51)]
20
21   pris <- eval(parse(text=paste("data$",omraade,sep="")))
22
23   regressors <- cbind.data.frame(pris,seasonreg) # Laver matrix med regressorer
24
25   season <- lm(pris~.,data=regressors) # Laver lineær regression
26
27   print(summary(season))
28   print(xtable(season))
29
30   adjusted <- season$residuals
31   return(as.numeric(adjusted))
32 }
33
34 ##########
35
36 AdjustSeason.diff <- function(omraade1,omraade2){
37 }
```

```
38 data <- spotpris.ny
39
40 eval(parse(text=paste('data$ugedag[which(data$ugedag.', substr(omraade1,start=1,stop
41 =2),'=="søndag" | data$ugedag.', substr(omraade1,start=1,stop=2),'"=="søndag")] <-
42 "søndag"',sep="")))) # Hvis det er søndag i det ene regime er det det også i det
43 andet
44
45 for(t in unique(data$time)) {
46 data[paste("time",t,sep="")] <- ifelse(data$time==t,1,0) # Laver en dummy-vektor for
47     hver time (-1)
48 }
49
50 for(t in unique(data$ugedag)) {
51 data[t] <- ifelse(data$ugedag==t,1,0) # Laver en dummy-vektor for hver ugedag (-1)
52 }
53
54 seasonreg <- data[,-c(1:17,20,45,51)]
55
56 diff <- eval(parse(text=paste("data$",omraade1,"-data$",omraade2,sep="")))
57
58 regressors <- cbind.data.frame(diff,seasonreg) # Laver matrix med regressorer
59
60 season <- lm(diff~.,data=regressors) # Laver lineær regression
61
62 print(summary(season))
63 print(xtable(season))
64
65 adjusted <- season$residuals
66
67 adjusted.ny <- numeric(length(diff))
68
69 for (i in 1:length(adjusted)){
70 adjusted.ny[as.numeric(names(adjusted)[i])] <- adjusted[i]
71 }
72
73 return(adjusted)
74 }
```

R-kode D.1: Sæsonrensning af data til modellerne uden regimeskift.

```
1 AdjustSeason <- function(omraade){
2
3 data <- spotpris.ny
4
5 for(t in unique(data$time)) {
6 data[paste("time",t,sep="")] <- ifelse(data$time==t,1,0) # Laver en dummy-vektor for
7     hver time (-1)
8 }
9 eval(parse(text=
10 paste(
11 for(t in unique(data$ugedag.),substr(omraade,start=1,stop=2),") {
12 data[t] <- ifelse(data$ugedag.,substr(omraade,start=1,stop=2),"==t,1,0) # Laver en
13     dummy-vektor for hver ugedag (-1)
14 }",sep="")))
```

```

16
17 for(t in unique(data$maaned)) {
18 data[t] <- ifelse(data$maaned==t,1,0) # Laver en dummy-vektor for hver måned (-1)
19 }
20
21
22 seasonreg <- data[,-c(1:17,20,45,51)]
23
24 pris <- eval(parse(text=paste("data$",omraade,sep="")))
25
26 regressors <- cbind.data.frame(pris,seasonreg*(1-regime),seasonreg*regime) # Laver
   matrix med regressorer for regime 1 og 0
27
28 for(i in 2:42){
29 colnames(regressors)[i] <- paste(colnames(regressors)[i],".regime0",sep="")
30 }
31 for(i in 43:83){
32 colnames(regressors)[i] <- paste(colnames(regressors)[i],".regime1",sep="")
33 }
34
35 season <- lm(pris~.,data=regressors) # Laver lineær regression
36
37 print(summary(season))
38 print(xtable(season))
39
40 adjusted <- season$residuals
41 return(as.numeric(adjusted))
42 }
43
44 #####
45
46 AdjustSeason.diff <- function(omraade1,omraade2){
47
48 data <- spotpris.ny
49
50 eval(parse(text=paste('data$ugedag[which(data$ugedag.',substr(omraade1,start=1,stop
      =2),'=="søndag" | data$ugedag.',substr(omraade1,start=1,stop=2),'=="søndag")] <-
      "søndag",sep="')))) # Hvis det er søndag i et af områderne er det også søndag i
      det andet.
51
52 for(t in unique(data$time)) {
53 data[paste("time",t,sep="")] <- ifelse(data$time==t,1,0) # Laver en dummy-vektor for
      hver time (-1)
54 }
55
56 for(t in unique(data$ugedag)) {
57 data[t] <- ifelse(data$ugedag==t,1,0) # Laver en dummy-vektor for hver ugedag (-1)
58 }
59
60 for(t in unique(data$maaned)) {
61 data[t] <- ifelse(data$maaned==t,1,0) # Laver en dummy-vektor for hver måned (-1)
62 }
63
64 seasonreg <- data[,-c(1:17,20,45,51)]
65
66 diff <- eval(parse(text=paste("data$",omraade1,"-data$",omraade2,sep="")))
67
68 regressors <- cbind.data.frame(diff,seasonreg*regime) # laver matrix med regressorer
      for regime 1
69
70 for(i in 2:42){

```

```
71 colnames(regressors)[i] <- paste(colnames(regressors)[i], ".regime1", sep="")
72 }
73
74 rownames(regressors) <- seq(length=nrow(regressors))
75
76 regressors$diff[which(diff==0)] <- NA
77
78 season <- lm(diff~., data=regressors) # Laver lineær regression
79
80 print(summary(season))
81 print(xtable(season))
82
83 adjusted <- season$residuals
84
85 adjusted.ny <- numeric(length(diff))
86
87 for (i in 1:length(adjusted)){
88 adjusted.ny [as.numeric(names(adjusted)[i])] <- adjusted[i]
89 }
90
91 return(adjusted.ny)
92 }
```

R-kode D.2: Sæsonrensning af data til modellerne med regimeskift.

```
1 Estimate_transition<-function(regime){
2 lag.ny<-function(regime){
3 lag <- numeric()
4 lag[1] <- 1
5 for (i in 2:length(regime)){
6 lag[i]<-regime[i-1]
7 }
8 return(lag)
9 }
10
11 Pmat<-matrix(nrow=2,ncol=2)
12 Pmat[1,2]<-length(which(regime-lag.ny(regime)==-1))/(length(regime)-sum(regime))
13 Pmat[2,1]<-length(which(regime-lag.ny(regime)==1))/sum(regime)
14 Pmat[1,1]<-1-Pmat[1,2]
15 Pmat[2,2]<-1-Pmat[2,1]
16
17 return(Pmat)
18 }
```

R-kode D.3: Beregning af transitionsmatrix

```
1 diffseries.ny <- function (x, d)
2 {
3   x <- as.data.frame(x)
4   names(x) <- "series"
5   x <- x$series
6   if (NCOL(x) > 1)
7     stop("only implemented for univariate time series")
8   if (any(is.na(x)))
9     stop("NAs in x")
10  n <- length(x)
11  stopifnot(n >= 2)
12  #x <- x - mean(x)
13  PI <- numeric(n)
14  PI[1] <- -d
15  for (k in 2:n) {
```

```

16     PI[k] <- PI[k - 1] * (k - 1 - d)/k
17   }
18   ydiff <- x
19   for (i in 2:n) {
20     ydiff[i] <- x[i] + sum(PI[1:(i - 1)] * x[(i - 1):1])
21   }
22   ydiff
23 }

```

R-kode D.4: Den modificerede diffseries-funktion.

```

1 ar.filter <- function(data,phi){
2 filtered <- numeric(length(data)) # Laver en vektor med nuller i
3 for (i in (length(phi)+1):length(filtered)){
4 filtered[i] <- data[i]-sum(phi*data[(i-1):(i-length(phi))]) # Finder residualerne
   for hver indgang i Filtered-vektoren.
5 }
6 return(filtered)
7 }

```

R-kode D.5: AR-filter.

```

1 Loglikt <- function(param,data){
2 ar <- param[1:length(ar)] # Fortæller hvor ar-parametrene er i parametervektoren
3 d <- param[length(ar)+1] # Fortæller hvor d parameteren er i parametervektoren
4 mu <- param[length(ar)+2] # Fortæller hvor mu-parameteren er i parametervektoren
5
6 if(length(ar_w)!=0){
7 arw <- param[length(ar)+3] # Fortæller hvor ar_w er i parametervektoren
8 }
9
10 dseries <- diffseries.ny(data-mu,d) # Tager den fraktionelle differens af data
11 epsilon <- ar.filter(dseries,ar) # Finder residualerne
12
13 if(length(ar_w)!=0){
14 epsilon <- epsilon-arw*lagged(epsilon,24) # Indfører sæsonpolynomiet
15 }
16
17 T <- length(data)
18 logl <- -0.5*log(2*pi)-0.5*log(sum(epsilon^2)/length(epsilon))-0.5*epsilon^2/(sum(
   epsilon^2)/length(epsilon)) # Opstiller loglikelihoodfunktionen
19 logl <- sum(logl) # Beregner funktionsværdien
20 return(logl)
21 }
22
23 #####
24
25 ObtainStart <- function(data,omraade){
26 eta[length(ar)+1] <- fracdiff(data)$d # Startværdi for d
27 eta[length(ar)+2] <- mean(data) # Startværdi for mu
28 eta[1:length(ar)] <- ar.yw(diffseries.ny(data-eta[length(ar)+2],eta[length(ar)+1]),
   aic=FALSE,order=length(ar))$ar # Startværder for ar-koefficienterne
29
30 if(length(ar_w)!=0){
31 eta[length(ar)+3] <- 0.15 # Startværdi for a jf. Haldrup og Nielsen
32 }
33
34 p <- optim(eta,Loglikt,data=data,hessian=TRUE,method="L-BFGS-B",control=list(fnscale
   =-1)) # Maksimerer loglikelihoodfunktionen
35
36 funcval <- p$value

```

```
37 eta <- p$par
38 OI <- try(solve(-p$hessian))
39 se <- try(sqrt(diag(OI))) # Beregner standard error
40
41 print(paste0("Log likelihood: ", funcval))
42
43 print( paste0("ar par. = ", eta[1:length(ar)]," std.errors = ",se[1:length(ar)] ) )
44 print( paste0("d par. = ", eta[length(ar)+1]," std.error = ",se[length(ar)+1]) )
45 print( paste0("mu par. = ", eta[length(ar)+2]," std.error = ",se[length(ar)+2]) )
46
47 if(length(ar_w)!=0){
48 print( paste0("ar_w par. = ", eta[length(ar)+3]," std.error = ",se[length(ar)+3]) )
49 }
50
51 # Funktion der får residualerne ud
52 getepsilon <- function(param,data){
53 ar <- param[1:length(ar)]
54 d <- param[length(ar)+1]
55 mu <- param[length(ar)+2]
56
57 if(length(ar_w)!=0){
58 arw <- param[length(ar)+3]
59 }
60
61 dseries <- diffseries.ny(data-mu,d)
62 epsilon <- ar.filter(dseries,ar)
63
64 if(length(ar_w)!=0){
65 epsilon <- epsilon-arw*lagged(epsilon,24)
66 }
67
68 return(epsilon)
69 }
70
71
72 epsilon <- getepsilon(eta,data)
73
74 var <- sum(epsilon^2)/length(epsilon)
75
76 print( paste0("Residual s.d. = ", sqrt(var)) )
77
78 if(length(ar_w)==0){
79 output <- list(log.lik=funcval,ar.par=eta[1:length(ar)],ar.std.err=se[1:length(ar)],
80 d.par=eta[length(ar)+1],d.std.err=se[length(ar)+1],mu.par=eta[length(ar)+2],mu.
81 std.err=se[length(ar)+2],resid.std.err=sqrt(var))
82 }
83
84 if(length(ar_w)!=0){
85 output <- list(log.lik=funcval,ar.par=eta[1:length(ar)],ar.std.err=se[1:length(ar)],
86 d.par=eta[length(ar)+1],d.std.err=se[length(ar)+1],mu.par=eta[length(ar)+2],mu.
87 std.err=se[length(ar)+2],ar_w.par=eta[length(ar)+3],ar_w.std.err=se[length(ar)
88 +3],resid.std.err=sqrt(var))
89 }
90
91 output.text <- sapply(names(output),function(x) paste(x,paste(output[[x]],collapse=""))
92
93 lapply(output.text, write, paste0("Data/RS/",omraade,"_connection_",omraade1,
94 omraade2,"_obtain_start_resultater.txt"), append=TRUE, ncolumns=1000)
```

```

91 | write.csv(epsilon,file=paste0("Data/RS/",omraade,"_connection_",omraade1,omraade2,"_
|   obtain_start_residualer.csv"),row.names=FALSE)
92 |
93 | return(eta)
94 |

```

R-kode D.6: Funktion som estimerer startværdier for parametrene.

```

1 # Opstiller en funktion der beregner funktionsværdien for loglikelihoodfunktionen
2 Loglik <- function(param,data){
3 # Fortæller hvor de forskellige parametre fra regime 1 er placeret i
4 # parametervektoren
5 ar1 <- param[1:length(ar)]
6 d1 <- param[length(ar)+1]
7 mu1 <- param[length(ar)+2]
8
9 if(length(ar_w)!=0){
10 ar_w1 <- param[length(ar)+3]
11 }
12 dseries1 <- diffseries.ny(data-mu1,d1) # Tager den fraktionelle differens ved at
# anvende mu og d fra regime 1
13 epsilon1 <- ar.filter(dseries1,ar1) # Finder residualerne ift. regime 1
14 if(length(ar_w)!=0){
15 epsilon1 <- epsilon1-ar_w1*lagged(epsilon1,24) # Indfører sæsonpolynomiet fra regime
# 1
16 }
17
18 # Fortæller hvor de forskellige parametre fra regime 1 er placeret i
# parametervektoren
19 ar0 <- param[(length(ar)+3+length(ar_w)):(2*length(ar)+2+length(ar_w))]
20 d0 <- param[2*length(ar)+3+length(ar_w)]
21 mu0 <- param[2*length(ar)+4+length(ar_w)]
22
23 if(length(ar_w)!=0){
24 ar_w0 <- param[2*length(ar)+5+length(ar_w)]
25 }
26
27 dseries0 <- diffseries.ny(data-mu0,d0) # Tager den fraktionelle differens ved at
# anvende mu og d fra regime 0
28 epsilon0 <- ar.filter(dseries0,ar0) # Finder residualerne ift. regime 0
29 if(length(ar_w)!=0){
30 epsilon0 <- epsilon0-ar_w0*lagged(epsilon0,24) # Indfører sæsonpolynomiet fra regime
# 0
31 }
32
33 epsilon <- epsilon1*regime+epsilon0*(1-regime) # Opstiller den endelige
# epsilonvektor, som indeholder residualerne tilhørende regime 1 og regime 0
34
35 logl <- -0.5*log(2*pi)-0.5*log(sum((epsilon1*regime)^2)/length(epsilon))+sum((
# epsilon0*(1-regime))^2)/length(epsilon))-0.5*epsilon^2/(sum((epsilon1*regime)^2)/
# length(epsilon)+sum((epsilon0*(1-regime))^2)/length(epsilon)) # Udtrykket for
# loglikelihoodfunktionen
36 logl <- sum(logl) # Funktionsværdien af loglikelihoodfunktionen
37
38 return(logl)
39 }
40
41 # Loglikelihoodfunktions til prisdifferenserne (indeholder kun regime 1, da
# parameterne pr. definition er 0 i regime 0)
42 Loglik.diff <- function(param,data){

```

```
43 f Fortæller hvor de forskellige parametre er i parametervektoren
44 ar1 <- param[1:length(ar)]
45 d1 <- param[length(ar)+1]
46 mu1 <- param[length(ar)+2]
47
48 if(length(ar_w)!=0){
49 ar_w1 <- param[length(ar)+3]
50 }
51
52 dseries1 <- diffseries.ny(data-mu1,d1) # Tager den fraktionelle differens
53 epsilon1 <- ar.filter(dseries1,ar1) # Finder residualerne for regime 1
54 epsilon0 <- numeric(length(epsilon1)) # Residualerne i regime 0 er 0
55
56 if(length(ar_w)!=0){
57 epsilon1 <- epsilon1-ar_w1*lagged(epsilon1,24) # Indfører sæsonpolynomiet
58 }
59
60 epsilon <- epsilon1*regime+epsilon0*(1-regime) # Den endelige epsilonvektor består
       af både regim1 og regime 0
61
62 logl <- -0.5*log(2*pi)-0.5*log(sum((epsilon1*regime)^2)/length(epsilon))-0.5*epsilon
       ^2/(sum((epsilon1*regime)^2)/length(epsilon)) # Udtrykket for
       loglikelihoodfunktionen
63 logl <- sum(logl) # Beregner funktionsværdien af loglikelihoodfunktionen
64 return(logl)
65 }
66
67 #####
68 # Funktion til at beregne parameterestimaterne
69 ModelEstimates <- function(data,initval,omraade){
70 eta <- c(initval,initval) # Laver parametervektoren dobbelt så lang
71
72 #mu1 <- mean(data[which(regime==1)]) # Anvendes i delen hvor mu er sat lig med
       sample mean
73 #mu0 <- mean(data[which(regime==0)]) # Anvendes i delen hvor mu er sat lig med
       sample mean
74
75 p <- optim(eta,Loglik,data=data,hessian=TRUE,method="L-BFGS-B",control=list(fnscale
       =-1)) # Maksimerer loglikelihoodfunktionen
76
77 funcval <- p$value
78
79 eta <- p$par
80
81 OI <- try(-solve(p$hessian))
82 se <- try(sqrt(diag(OI))) # Beregner standard eroor ud fra Hessematrix
83
84 # Udskriver alle estimatorer samt standard errors
85 print(paste0("Log likelihood: ", funcval))
86
87 print("Regime 0 estimator: ")
88
89 print(paste0("ar par. = ", eta[(length(ar)+3+length(ar_w)):(2*length(ar)+2+length(
       ar_w))]," std.errors = ",se[(length(ar)+3+length(ar_w)):(2*length(ar)+2+length(
       ar_w))]))
90 print(paste0("d par. = ", eta[2*length(ar)+3+length(ar_w)]," std.error = ",se[2*
       length(ar)+3+length(ar_w)])
91 print(paste0("mu par. = ", eta[2*length(ar)+4+length(ar_w)]," std.error = ",se[2*
       length(ar)+4+length(ar_w)])
92
93 if(length(ar_w)!=0){
```

```

94 print( paste0("ar_w par. = ", eta[2*length(ar)+5+length(ar_w)], " std.error = ", se[2*
95   length(ar)+5+length(ar_w)]) )
96
97 # Funktion der får residualerne ud for regime 0
98 getepsilon0 <- function(param,data){
99 ar0 <- param[(length(ar)+3+length(ar_w)):(2*length(ar)+2+length(ar_w))]
100 d0 <- param[2*length(ar)+3+length(ar_w)]
101 mu0 <- param[2*length(ar)+4+length(ar_w)]
102
103 if(length(ar_w)!=0){
104 ar_w0 <- param[2*length(ar)+5+length(ar_w)]
105 }
106
107 dseries0 <- diffseries.ny(data-mu0,d0)
108 epsilon0 <- ar.filter(dseries0,ar0)
109
110 if(length(ar_w)!=0){
111 epsilon0 <- epsilon0-ar_w0*lagged(epsilon0,24)
112 }
113
114 return(epsilon0)
115 }
116
117 epsilon0 <- getepsilon0(eta,data)
118
119 var0 <- sum((epsilon0*(1-regime))^2)/sum(1-regime)
120
121 # Udskriver residualerne samt deres standard error for regime 0
122 print( paste0("Residual s.d. = ", sqrt(var0)) )
123
124 print("Regime 1 estimator: ")
125
126 print( paste0("ar par. = ", eta[1:length(ar)], " std.errors = ", se[1:length(ar)]) )
127 print( paste0("d par. = ", eta[length(ar)+1], " std.error = ", se[length(ar)+1]) )
128 print( paste0("mu par. = ", eta[length(ar)+2], " std.error = ", se[length(ar)+2]) )
129
130 if(length(ar_w)!=0){
131 print( paste0("ar_w par. = ", eta[length(ar)+3], " std.error = ", se[length(ar)+3]) )
132 }
133
134 # Funktion der får residualerne ud til regime 1
135 getepsilon1 <- function(param,data){
136 ar1 <- param[1:length(ar)]
137 d1 <- param[length(ar)+1]
138 mu1 <- param[length(ar)+2]
139
140 if(length(ar_w)!=0){
141 ar_w1 <- param[length(ar)+3]
142 }
143
144 dseries1 <- diffseries.ny(data-mu1,d1)
145 epsilon1 <- ar.filter(dseries1,ar1)
146
147 if(length(ar_w)!=0){
148 epsilon1 <- epsilon1-ar_w1*lagged(epsilon1,24)
149 }
150
151 return(epsilon1)
152 }
153

```

```

154 epsilon1 <- getepsilon1(eta,data)
155
156 var1 <- sum((epsilon1*regime)^2)/sum(regime)
157
158 # Udskriver residualer samt deres standard errors for regime 1
159 print(paste0("Residual s.d. = ", sqrt(var1)) )
160
161 if(length(ar_w)==0){
162   output <- list(log.lik=funcval,ar0.par=eta[(length(ar)+2+1):(2*length(ar)+2)],ar0.
163     std.err=se[(length(ar)+2+1):(2*length(ar)+2)],d0.par=eta[2*length(ar)+3],d0.std.
164     err=se[2*length(ar)+3],mu0.par=eta[2*length(ar)+4],mu0.std.err=se[2*length(ar)
165     +4],resid0.std.err=sqrt(var0),ar1.par=eta[1:length(ar)],ar1.std.err=se[1:length(
166     ar)],d1.par=eta[length(ar)+1],d1.std.err=se[length(ar)+1],mu1.par=eta[length(ar)
167     +2],mu1.std.err=se[length(ar)+2],resid1.std.err=sqrt(var1))
168 }
169
170 if(length(ar_w)!=0){
171   output <- list(log.lik=funcval,ar0.par=eta[(length(ar)+3+length(ar_w)):(2*length(ar)
172     +2+length(ar_w))],ar0.std.err=se[(length(ar)+3+length(ar_w)):(2*length(ar)+2+
173     length(ar_w))],d0.par=eta[2*length(ar)+3+length(ar_w)],d0.std.err=se[2*length(ar
174     +3+length(ar_w)),mu0.par=eta[2*length(ar)+4+length(ar_w)],mu0.std.err=se[2*
175     length(ar)+4+length(ar_w)],ar_w0.par=eta[2*length(ar)+5+length(ar_w)],ar_w0.std.
176     err=se[2*length(ar)+5+length(ar_w)],resid0.std.err=sqrt(var0),ar1.par=eta[1:
177     length(ar)],ar1.std.err=se[1:length(ar)],d1.par=eta[length(ar)+1],d1.std.err=se[
178     length(ar)+1],mu1.par=eta[length(ar)+2],mu1.std.err=se[length(ar)+2],ar_w1.par=
179     eta[length(ar)+3],ar_w1.std.err=se[length(ar)+3],resid1.std.err=sqrt(var1))
180 }
181
182 output.text <- sapply(names(output),function(x) paste(x,paste(output[[x]],collapse=""))
183
184 lapply(output.text, write, paste0("Data/RS/",omraade,"_connection_",omraade1,
185   omraade2,"_model_estimates.txt"), append=TRUE, ncolumns=1000)
186
187 epsilon <- epsilon1*regime+epsilon0*(1-regime)
188
189 write.csv(epsilon,file=paste0("Data/RS/",omraade,"_connection_",omraade1,omraade2,"_
190   model_residualer.csv"),row.names=FALSE)
191
192 return(eta)
193 }
194
195 #####
196 # Denne funktion anvendes når det er prisdifferensen
197 ModelEstimates.diff <- function(data,initval){
198   eta <- initval # Parametervektoren
199
200   p <- optim(eta,Loglik.diff,data=data,hessian=TRUE,method="L-BFGS-B",control=list(
201     fnscale=-1)) # Maksimerer loglikelihoodfunktionen
202
203   funcval <- p$value
204
205   eta <- p$par
206
207   OI <- try(-solve(p$hessian))
208   se <- try(sqrt(diag(OI))) # Beregner standard errors vha Hessematrix
209
210   # Udskriver alle parameterestimaterne samt deres standard errors
211   print(paste0("Log likelihood: ", funcval))

```

```

198 print("Regime 0 estimator er lig nul ")
199
200 print("Regime 1 estimator: ")
201
202 print( paste0("ar par. = ", eta[1:length(ar)], " std.errors = ", se[1:length(ar)]) )
203 print( paste0("d par. = ", eta[length(ar)+1], " std.error = ", se[length(ar)+1]) )
204 print( paste0("mu par. = ", eta[length(ar)+2], " std.error = ", se[length(ar)+2]) )
205
206
207 if(length(ar_w)!=0){
208 print( paste0("ar_w par. = ", eta[length(ar)+3], " std.error = ", se[length(ar)+3]) )
209 }
210
211 # Får residualerne ud for regime 1
212 getepsilon1 <- function(param,data){
213 ar1 <- param[1:length(ar)]
214 d1 <- param[length(ar)+1]
215 mu1 <- param[length(ar)+2]
216
217 if(length(ar_w)!=0){
218 ar_w1 <- param[length(ar)+3]
219 }
220
221 dseries1 <- diffseries.ny(data-mu1,d1)
222 epsilon1 <- ar.filter(dseries1,ar1)
223
224 if(length(ar_w)!=0){
225 epsilon1 <- epsilon1-ar_w1*lagged(epsilon1,24)
226 }
227
228 return(epsilon1)
229 }
230
231 epsilon1 <- getepsilon1(eta,data)
232
233 var1 <- sum((epsilon1*regime)^2)/sum(regime)
234
235 # Udskriver residualerne samt deres standard errors
236 print( paste0("Residual s.d. = ", sqrt(var1)) )
237
238 if(length(ar_w)==0){
239 output <- list(log.lik=funcval,ar1.par=eta[1:length(ar)],ar1.std.err=se[1:length(ar)],
240                 d1.par=eta[length(ar)+1],d1.std.err=se[length(ar)+1],mu1.par=eta[length(ar)+2],
241                 mu1.std.err=se[length(ar)+2],resid1.std.err=sqrt(var1))
242 }
243
244 if(length(ar_w)!=0){
245 output <- list(log.lik=funcval,ar1.par=eta[1:length(ar)],ar1.std.err=se[1:length(ar)],
246                 d1.par=eta[length(ar)+1],d1.std.err=se[length(ar)+1],mu1.par=eta[length(ar)+2],
247                 mu1.std.err=se[length(ar)+2],ar_w1.par=eta[length(ar)+3],ar_w1.std.err=se[length(ar)+3],
248                 resid1.std.err=sqrt(var1))
249 }
250
251 output.text <- sapply(names(output),function(x) paste(x,paste(output[[x]],collapse="")))
252
253 lapply(output.text, write, paste0("Data/RS/Diff_connection_",omraade1,omraade2,"_",
254                                     "model_estimates.txt"), append=TRUE, ncolumns=1000)
255
256 epsilon <- epsilon1*regime # Residualvektor
257

```

```
252 write.csv(epsilon,file=paste0("Data/RS/Diff_connection_ ",omraade1,omraade2,"_model_
253   residualer.csv"),row.names=FALSE)
254 return(eta)
255
256 }
```

R-kode D.7: Funktion som estimerer parametrene i modellerne.

```
1 library(fracdiff)
2 library(xtable)
3 library(numDeriv)
4
5 #setwd("C:/Users/Simon/Documents/P10 SVN/Projekt/Rapport")
6 #setwd("C:/Users/Julie/Desktop/P10/Projekt/Rapport")
7
8 source("Data/dummies.R")
9 source("Data/RS/obtaintransition.R")
10 source("Data/RS/seasonadjust.R")
11 source("Data/RS/diffseries_ny.R")
12 source("Data/RS/ar.filter.R")
13 source("Data/RS/lagged.R")
14 source("Data/RS/obtainstart.R")
15 #source("Data/RS/obtainstart_uden_mu.R")
16 source("Data/RS/obtainestimates.R")
17 #source("Data/RS/obtainestimates_uden_mu.R")
18
19 spotpris.ny <- spotpris.ny[-which(format(spotpris.ny$Date.UTC,"%Y")=="2014"),]
20
21
22 spotpris.ny$tid <- seq(0,nrow(spotpris.ny)-1,1)
23
24 colnames(spotpris.ny)[6] <- "N02"
25
26
27 #####
28
29 ar <- numeric(1)
30 ar_w <- TRUE
31 ma <- numeric(0)
32 d <- 0.3
33 mu <- 0.1
34 eta <- c(ar,ma,d,mu)
35 #eta <- c(ar,ma,d)
36
37 if(ar_w==TRUE){
38 eta <- c(eta,0.10)
39 }
40
41 if(ar_w==FALSE){
42 ar_w <- numeric(0)
43 }
44
45 omraade1 <- "N02"
46
47 omraade2 <- "DK1"
48
49 obs1 <- eval(parse(text=(paste("spotpris.ny$",omraade1,sep=""))))
# Indlæser
# priserne i område 1
50 obs2 <- eval(parse(text=(paste("spotpris.ny$",omraade2,sep=""))))
# Indlæser
# priserne i område 2
```

```

51
52 diff <- obs1-obs2 # Finder prisdifferensen
53
54 regime <- ifelse(diff!=0,1,0) # Laver regimevektoren
55
56 P <- Estimate_transition(regime) # Henter transitionsmatrix
57
58 obs1 <- AdjustSeason(omraade1) # Sæsonrenser data fra område 1
59 obs2 <- AdjustSeason(omraade2) # Sæsonrenser data fra område 2
60 diff <- AdjustSeason.diff(omraade1,omraade2) # Sæsonrenser prisdifferensen
61
62 ptm <- proc.time()
63 eta1 <- ObtainStart(obs1,omraade1) # Finder startestimater til område 1
64 proc.time() - ptm
65
66 ptm <- proc.time()
67 eta2 <- ObtainStart(obs2,omraade2) # Finder startestimater til områder 2
68 proc.time() - ptm
69
70 ptm <- proc.time()
71 eta0 <- ObtainStart(diff,"diff") # Finder startestimater for prisdifferensen
72 proc.time() - ptm
73
74 ptm <- proc.time()
75 mod1 <- ModelEstimates(obs1,eta1,omraade1) # Finder mpdelestimerne til område 1
76 proc.time() - ptm
77
78 ptm <- proc.time()
79 mod2 <- ModelEstimates(obs2,eta2,omraade2) # Finder modelestimerne til område 2
80 proc.time() - ptm
81
82 ptm <- proc.time()
83 mod0 <- ModelEstimates.diff(diff,eta0) # Finder modelestimerne til prisdifferensen
84 proc.time() - ptm

```

R-kode D.8: Hovedfilen til parameterestimation.

Forecasts

```

1 # Sorterings-funktion: sorterer en matrix rækkevis efter størrelsen på værdierne
2 sortr <- function(matrix){
3   for (i in 1:nrow(matrix))
4   {
5     matrix[i,] <- sort(matrix[i,],decreasing=FALSE)
6   }
7   return(matrix)
8 }

```

R-kode D.9: Funktion som sorterer en matrix rækkevis efter størrelse.

```

1 # Polymul-funktionen tager koefficienterne i to polynomier, der multipliceres med
  # hinanden, som input, og som returnerer koefficienterne i det resulterende
  # polynomium
2 polymul = function(a,b)
3 {
4   # Definere den korte vektor som x og den lange vektor som y. Hvis de er lige lange
  # er rækkefølgen ligegyldig
5   if (length(a) <= length(b))
6   {

```

```
7     x = a
8     y = b
9 }
10 else
11 {
12     x = b
13     y = a
14 }

15
16     # Beregner koefficienterne
17 # Calculate the coefficients
18 m = length(x)
19 n = length(y)
20
21 out = numeric(length = n+m-1) # Tom vektor til output
22
23 for (i in 1:m)
24     out[i] = sum(x[1:i] * y[i:1])
25
26 if ( n > m )
27     for (i in (m+1):n)
28         out[i] = sum(x[1:m] * y[i:(i-m+1)])
29
30 for (i in (n+1):(n+m-1))
31     out[i] = sum(x[(i-n+1):m] * y[n:(i-m+1)])
32
33 return(out)
34 }
```

R-kode D.10: Funktion som tager koefficienterne i to polynomier, der multipliceres med hinanden, som input, og som returnerer koefficienterne i det resulterende polynomium

```
1 # Funktion som beregner de forecastede værdier
2 MCforc_u <- function(data,resid,param){
3 ar_u <- param[1:length(ar)]
4 d_u <- param[length(ar)+1]
5 mu_u <- param[length(ar)+2]
6
7 if(length(ar_w)!=0){
8 arw <- param[length(ar)+3]
9 }
10 forc <- numeric(forc_steps)
11
12 arcoefs1 <- diffseries.ny(c(1,numeric(length(resid)-1)),d_u) # Tager den
    fraktionelle differens af en vektor indeholdende et 1-tal og nuller
13 arcoefs1 <- polymul(arcoefs1,c(1,-ar_u))[1:length(resid)] # Beregner pi'erne
14 arcoefs1 <- polymul(arcoefs1,c(1,rep(0,23),-arw))[1:length(resid)] # benytter
    polymul igen
15 arcoefs1 <- -rev(arcoefs1[2:length(arcoefs1)]) # Fjerne første indgang og reverserer
    vektoren
16
17
18 for (i in 0:(forc_steps-1)){
19 lagdata <- c((data-mu_u),forc)
20 forc[i+1] <- sum(arcoefs1[(forc_steps-i):(length(arcoefs1))]*lagdata[1:(length(data)
    +i)])+resid[length(data)+i+1] # Fylder forecast-vektoren
21 }
22
23 forc <- forc+mu_u
24 }
```

```
25 return(forc)
26 }
```

R-kode D.11: Funktion der beregner de forecastede værdier (modeller uden regimeskift).

```
1 #setwd("C:/Users/Simon/Documents/P10 SVN/Projekt/Rapport")
2 #setwd("C:/Users/Julie/Desktop/P10/Projekt/Rapport")
3
4 library(fracdiff)
5 library(xtable)
6 library(numDeriv)
7
8 source("Data/Forecast/seasonadjust_uden_regime.R")
9 source("Data/Forecast/diffseries_ny.R")
10 source("Data/Forecast/ar.filter.R")
11 source("Data/Forecast/lagged.R")
12 source("Data/Forecast/MC-forecast.R")
13 source("Data/Forecast/sortr.R")
14 source("Data/Forecast/Resid_u_gen.R")
15 source("Data/Forecast/MCforc_u.R")
16 source("Data/Forecast/pulymut.R")
17 source("Data/dummies.R")
18
19 #####
20
21 spotpris.ny$tid <- seq(0,nrow(spotpris.ny)-1,1)
22
23 colnames(spotpris.ny)[6] <- "N02"
24
25 forc_steps <- 24
26 reps <- 1000
27
28 omraade1 <- "N02"
29
30 omraade2 <- "DK1"
31
32 obs1 <- eval(parse(text=(paste("spotpris.ny$",omraade1,sep="")))) # Indlæser
     priserne i område 1
33 obs2 <- eval(parse(text=(paste("spotpris.ny$",omraade2,sep="")))) # Indlæser
     priserne i område 2
34
35 diff <- obs1-obs2
36
37 obs1 <- AdjustSeason(omraade1,forc_steps)$adjusted # Sæsonrenser data fra område 1
38 obs2 <- AdjustSeason(omraade2,forc_steps)$adjusted # Sæsonrenser data fra område 2
39 diff <- AdjustSeason.diff(omraade1,omraade2,forc_steps)$adjusted # Sæsonrenser
     prisdifferensen
40
41
42 set.seed(63120)
43 randoms <- matrix(rnorm(forc_steps*reps),nrow=forc_steps,ncol=reps) # Matrix med
     normalfordelte tal
44 randomus <- matrix(runif(forc_steps*reps),nrow=forc_steps,ncol=reps) # Matrix med
     uniformt fordelte tal
45
46 #####
47 # Område 1
48
49 # De forskellige estimerede parametre
50 ar <- c(0.588912473935061,-0.184376338233785,
```

```
51      0.0580669517956083,-0.0552979894245369)
52 mu <- -3.84222433234156e-05
53 d <- 0.434318805367567
54 var <- 19.2594401474238
55 ar_w=TRUE
56
57 eta <- c(ar,d,mu)
58 #eta <- c(ar,d)
59
60 if(ar_w==TRUE){
61 eta <- c(eta,0.30178104051449)
62 }
63
64 if(ar_w==FALSE){
65 ar_w <- numeric(0)
66 }
67
68 # Funktion der laver forecast
69 frecas <- function(data,param,var,randoms ,forc_steps){
70 eps <- Resid_u_gen(data,param) # Laver residual-vektor
71
72 forecastsims <- matrix(rep(0,nrow(randoms)*ncol(randoms)),nrow=nrow(randoms),ncol=
    ncol(randoms)) # Matrix med nuller til de forecastede værdier
73
74 for (k in 1:reps){
75 forecastsims[,k] <- MCforc_u(data,c(eps,randoms[,k]*sqrt(var)),param) # Fylder
    matricen ud med de forecastede værdier
76 }
77
78 forecastsims <- sortr(forecastsims) # Sorterer matricen række for række
79
80 forecasts <- cbind(forecastsims[,floor(0.025*reps)+1],forecastsims[,floor(0.5*reps)
    +1],forecastsims[,floor(0.975*reps)+1]) # Fejl bånd samt forecast
81
82 output <- as.data.frame(forecasts)
83 print("output =")
84
85 print(output)
86
87 return(output)
88 }
89
90 obs1_fore <- frecas(obs1,eta,var,randoms ,forc_steps)
91
92 season.predicted.1 <- AdjustSeason(omraade1,forc_steps)$predict.season # Prediktion
    af sæsonmodellen
93
94 out1 <- cbind(obs1_fore,season.predicted.1)
95
96 write.csv(out1,file=paste0("Data/Forecast/",omraade1,"_connection_",omraade1,
    omraade2,"_uden_RS.csv"),row.names=FALSE)
97
98 ##########
99 #Område 2
100
101 # De forskellige estimerede parametre
102 ar <- c(0.691505443810371,-0.0289242261656109,
103     -0.00711854289623273)
104 mu <- -2.81398755057092e-05
105 d <- 0.292629401537492
106 var <- 36.8418746916429
```

```

107 ar_w=TRUE
108
109 eta <- c(ar,d,mu)
110 #eta <- c(ar,d)
111
112 if(ar_w==TRUE){
113 eta <- c(eta,0.22883806068438)
114 }
115
116 if(ar_w==FALSE){
117 ar_w <- numeric(0)
118 }
119
120 obs2_fore <- freqas(obs2,eta,var,randoms,forc_steps) # Udskriver forecastet
121
122 season.predicted.2 <- AdjustSeason(omraade2,forc_steps)$predict.season # Predikterer
    sæsonfunktionen
123
124 out2 <- cbind(obs2_fore,season.predicted.2)
125
126 write.csv(out2,file=paste0("Data/Forecast/",omraade2,"_connection_",omraade1,
    omraade2,"_uden_RS.csv"),row.names=FALSE)
127
128 ##########
129 #Differensen
130
131 # De forskellige estimerede parametre
132 ar <- c(0.737026586211997,0.00507048919767234,
    0.00168045207170804,-0.0146310153205996)
133 mu <- 3.31889849649896e-05
134 d <- 0.185019865411597
135 var <- 35.3707288667047
136 ar_w=TRUE
137
138
139 eta <- c(ar,d,mu)
140 #eta <- c(ar,d)
141
142 if(ar_w==TRUE){
143 eta <- c(eta,0.15781431129445)
144 }
145
146 if(ar_w==FALSE){
147 ar_w <- numeric(0)
148 }
149
150 diff_fore <- freqas(diff,eta,var,randoms,forc_steps) # Udskriver forecastet
151
152 season.predicted.diff <- AdjustSeason.diff(omraade1,omraade2,forc_steps)$predict.
    season # Predikterer sæsonfunktionen
153
154 out.diff <- cbind(diff_fore,season.predicted.diff)
155
156 write.csv(out.diff,file=paste0("Data/Forecast/Diff_connection_",omraade1,omraade2,"_"
    "uden_RS.csv"),row.names=FALSE)

```

R-kode D.12: Hovedfilen til forecast ved anvendelse af modeller uden regimeskift.

```

1 # Funktion til forecasts
2 MCforc <- function(data,resid,param){
3 regime <- regime.ny
4 ar1 <- param[1:length(ar1)]

```

```
5 d1 <- param[length(ar1)+1]
6 mu1 <- param[length(ar1)+2]
7
8 if(ar_w==TRUE){
9 ar_w1 <- param[length(ar1)+3]
10 }
11
12 ar0 <- param[(length(ar)+3+length(ar_w)):(2*length(ar)+2+length(ar_w))]
13 d0 <- param[2*length(ar)+3+length(ar_w)]
14 mu0 <- param[2*length(ar)+4+length(ar_w)]
15
16 if(length(ar_w)!=0){
17 ar_w0 <- param[2*length(ar)+5+length(ar_w)]
18 }
19
20 forc <- numeric(forc_steps)
21
22 # For regime 0
23 arcoefs0 <- diffseries.ny(c(1,numeric(nrow(resid)-1)),d0)
24 arcoefs0 <- polymul(arcoefs0,c(1,-ar0))[1:nrow(resid)]
25 arcoefs0 <- polymul(arcoefs0,c(1,rep(0,23),-ar_w0))[1:nrow(resid)]
26 arcoefs0 <- -rev(arcoefs0[2:length(arcoefs0)])
27
28 # For regime 1
29 arcoefs1 <- diffseries.ny(c(1,numeric(nrow(resid)-1)),d1)
30 arcoefs1 <- polymul(arcoefs1,c(1,-ar1))[1:nrow(resid)]
31 arcoefs1 <- polymul(arcoefs1,c(1,rep(0,23),-ar_w1))[1:nrow(resid)]
32 arcoefs1 <- -rev(arcoefs1[2:length(arcoefs1)])
33
34
35
36 for (i in 0:(forc_steps-1)){
37 if(regime[length(regime)-forc_steps+i+1]==1){
38 lagdata <- c((data-mu1),forc)
39 forc[i+1] <- sum(arcoefs1[(forc_steps-i):(length(arcoefs1))]*lagdata[1:(length(data)
+ i)]+resid[length(data)+i+1,2] # Fylder forecast-vektor
40 }
41 else{
42 lagdata <- c((data-mu0),forc)
43 forc[i+1] <- sum(arcoefs0[(forc_steps-i):(length(arcoefs1))]*lagdata[1:(length(data)
+ i)]+resid[length(data)+i+1,1]
44 }
45 }
46 forc <- forc + mu1*regime[(length(data)+1):length(regime)]+mu0*(1-regime[(length(
data)+1):length(regime)])
47
48 return(forc)
49 }
50
51
52 #####
53 # Funktion til forecast af dif
54 MCforc.diff <- function(data,resid,param){
55
56 regime <- regime.ny
57
58 ar1 <- param[1:length(ar1)]
59 d1 <- param[length(ar1)+1]
60 mu1 <- param[length(ar1)+2]
61
62 ar_w0 <- 0
```

```

63
64 if(ar_w==TRUE){
65 ar_w1 <- param[length(ar1)+3]
66 }
67
68 forc <- numeric(forc_steps)
69
70 # Regime 0
71 arcoefs0 <- diffseries.ny(c(1,numeric(nrow(resid)-1)),d0)
72 arcoefs0 <- polymul(arcoefs0,c(1,-ar0))[1:nrow(resid)]
73 arcoefs0 <- polymul(arcoefs0,c(1,rep(0,23),-ar_w0))[1:nrow(resid)]
74 arcoefs0 <- -rev(arcoefs0[2:length(arcoefs0)])
75
76 # Regime 1
77 arcoefs1 <- diffseries.ny(c(1,numeric(nrow(resid)-1)),d1)
78 arcoefs1 <- polymul(arcoefs1,c(1,-ar1))[1:nrow(resid)]
79 arcoefs1 <- polymul(arcoefs1,c(1,rep(0,23),-ar_w1))[1:nrow(resid)]
80 arcoefs1 <- -rev(arcoefs1[2:length(arcoefs1)])
81
82 for (i in 0:(forc_steps-1)){
83 if(regime[length(regime)-forc_steps+i+1]==1){
84 lagdata <- c((data-mu1),forc)
85 forc[i+1] <- sum(arcoefs1[(forc_steps-i):(length(arcoefs1))]*lagdata[1:(length(data)
86 +i)]+resid[length(data)+i+1,2]) # Fylder forecast-vektor
87 }
88 else{
89 forc[i+1]<-0
90 }
91 forc <- forc + mu1*regime[(length(data)+1):length(regime)]
92
93 return(forc)
94 }
```

R-kode D.13: Funktion der beregner de forecastede værdier (modeller med regimeskift).

```

1 #setwd("C:/Users/Simon/Documents/P10 SVN/Projekt/Rapport")
2 #setwd("C:/Users/Julie/Desktop/P10/Projekt/Rapport")
3
4 library(fracdiff)
5 library(xtable)
6 library(numDeriv)
7
8 source("Data/Forecast/seasonadjust.R")
9 source("Data/Forecast/diffseries_ny.R")
10 source("Data/Forecast/ar.filter.R")
11 source("Data/Forecast/lagged.R")
12 source("Data/Forecast/MC-forecast.R")
13 source("Data/Forecast/sortr.R")
14 source("Data/Forecast/Resid_u_gen.R")
15 source("Data/Forecast/MCforc_u.R")
16 source("Data/Forecast/MCforc.R")
17 source("Data/Forecast/polymul.R")
18 source("Data/Forecast/Season_predict.R")
19 source("Data/dummies.R")
20
21 ##########
22
23 spotpris.ny$tid <- seq(0,nrow(spotpris.ny)-1,1)
24
```

```
25 colnames(spotpris.ny)[6] <- "N02"
26
27 omraade1 <- "N02"
28
29 omraade2 <- "DK1"
30
31 obs1 <- eval(parse(text=(paste("spotpris.ny$",omraade1,sep="")))) #Indlæser priserne
   i område 1
32 obs2 <- eval(parse(text=(paste("spotpris.ny$",omraade2,sep="")))) #Indlæser priserne
   i område 2
33
34 diff <- obs1-obs2
35
36 regime <- ifelse(diff!=0,1,0)
37
38 obs1 <- AdjustSeason(omraade1)
39 obs2 <- AdjustSeason(omraade2)
40 diff <- AdjustSeason.diff(omraade1,omraade2)
41
42 markov <- c(0.5,0.6)
43
44 forc_steps <- 24
45 reps <- 10
46
47 set.seed(63120)
48 randoms <- matrix(rnorm(forc_steps*reps),nrow=forc_steps,ncol=reps) # Matrix med
   normalfordelte værdier
49 randomus <- matrix(runif(forc_steps*reps),nrow=forc_steps,ncol=reps) # Matrix med
   uniformt fordelte værdier
50
51 ######
52 # Indlæser de estimerede parametre for regime 0
53 ar0 <- c(0.46621)
54 mu0 <- -0.187013
55 d0 <- 0.39449
56
57 eta0 <- c(ar0,d0,mu0)
58
59 ar_w=TRUE
60
61 if(ar_w==TRUE){
62 eta0 <- c(eta0,0.3256)
63 }
64
65 # Indlæser de estimerede parametre for regime 1
66 ar1 <- c(0.46621)
67 mu1 <- -0.187013
68 d1 <- 0.39449
69
70 eta1 <- c(ar1,d1,mu1)
71
72 if(ar_w==TRUE){
73 eta1 <- c(eta1,0.3256)
74 }
75
76 eta <- c(eta1,eta0) # Parametervektoren
77
78 if(ar_w==FALSE){
79 ar_w <- numeric(0)
80 }
```

```

82 var <- c(0.11830,0.256984)^2
83
84 eps <- Resid_gen(obs1,eta) # Residualerne
85
86 regime.ny <- c(regime[1:17521],rep(0,forc_steps)) # Regimevektoren med nuller i
87 enden
88
89 forecastsims <- matrix(rep(0,nrow(randoms)*ncol(randoms)),nrow=nrow(randoms),ncol=
90 ncol(randoms)) # Matrix med nuller til de forecastede MC-stier
91
92 regime.forc <- matrix(rep(0,nrow(randoms)*ncol(randoms)),nrow=nrow(randoms),ncol=
93 ncol(randoms)) # Regime-matrix med nuller til de forecastede regime-stier
94
95 for (k in 1:reps){
96   for (j in 1:forc_steps){
97     regime.ny[length(regime.ny)-forc_steps+j] <- ifelse(markov[regime.ny[length(regime.
98 ny)-forc_steps-1+j]+1]<randomus[j,k],1,0) # Forecaster de nye regimer
99     regime.forc[j,k] <- regime.ny[length(regime.ny)-forc_steps+j]
100   }
101   forecastsims[,k] <- MCforc(obs1,rbind(cbind(eps,eps),cbind(randoms[,k]*sqrt(var[1]),
102 randoms[,k]*sqrt(var[2]))),eta) # Forecaster priserne
103 }
104
105 season.forc <- matrix(rep(0,nrow(randoms)*ncol(randoms)),nrow=nrow(randoms),ncol=
106 ncol(randoms)) # Matrix med nuller
107
108 for (k in 1:reps){
109   season.forc[,k] <- Season.predict(omraade1,forc_steps,regime.forc[,k])$fit # Kommer
110   de fremtidige værdier af sæsonfunktionen ind i season_forc givet de forecastede
111   regimer
112 }
113
114 pris.med.season <- matrix(rep(0,nrow(randoms)*ncol(randoms)),nrow=nrow(randoms),ncol=
115 =ncol(randoms)) # Matrix med nuller
116
117 for (k in 1:reps){
118   pris.med.season[,k] <- forecastsims[,k] + season.forc[,k] # Lægger sæsonen og den
119   sæsonrensede pris sammen
120 }
121
122 season.med.predict <- matrix(rep(0,nrow(randoms)*ncol(randoms)*3),nrow=nrow(randoms)
123 ,ncol=ncol(randoms)*3) # Matrix med nuller
124
125 for (k in 1:reps){
126   season.med.predict[, (k*3-2):(k*3)] <- as.matrix(Season.predict(omraade1,forc_steps,
127   regime.forc[,k])) # For hver regimesti beregnes værdi af sæsonfunktionen samt
128   dens 95%-prædiktionsintervaller
129 }
130
131 forecastsims <- sortr(forecastsims) # Sorterer matricen
132
133 forecasts <- cbind(forecastsims[,floor(0.025*reps)+1],forecastsims[,floor(0.5*reps)
134 +1],forecastsims[,floor(0.975*reps)+1]) # Finder fejlbånd samt forecastet
135
136 output.deseason <- as.data.frame(forecasts)
137 print("output deseason=")
138
139 print(output.deseason)
140
141 write.csv(output.deseason,file=paste0("Data/Forecast/",omraade1,"_connection_",
142 omraade1,omraade2,"_med_RS_deseason.csv"),row.names=FALSE)
143
144 pris.med.season <- sortr(pris.med.season)

```

```
128  
129 output.med.season <- as.data.frame(pris.med.season[,floor(0.5*reps)+1])  
130  
131 print("output med season=")  
132  
133 print(output.med.season)  
134  
135 write.csv(output.med.season,file=paste0("Data/Forecast/",omraade1,"_connection_",
    omraade1,omraade2,"_med_RS_med_season.csv"),row.names=FALSE)  
136  
137 season.med.predict <- sortr(season.med.predict)  
138  
139 season.med.predict.band <- cbind(season.med.predict[,floor(0.025*reps*3)+1],season.
    med.predict[,floor(0.5*reps*3)+1],season.med.predict[,floor(0.975*reps*3)+1])  
140  
141 output.season.predict <- as.data.frame(season.med.predict.band)  
142  
143 print("season med baand =")  
144  
145 print(output.season.predict)  
146  
147 write.csv(output.season.predict,file=paste0("Data/Forecast/",omraade1,"_connection_"
    ,omraade1,omraade2,"_med_RS_season_med_baand.csv"),row.names=FALSE)  
148  
149 #####  
150 # Indlæser de estimerede parametre for regime 0  
151 ar0 <- c(0.46621)  
152 mu0 <- -0.187013  
153 d0 <- 0.39449  
154  
155 eta0 <- c(ar0,d0,mu0)  
156  
157 ar_w=TRUE  
158  
159 if(ar_w==TRUE){  
160 eta0 <- c(eta0,0.3256)  
161 }  
162  
163 # Indlæser de estimerede parametre for regime 1  
164 ar1 <- c(0.46621)  
165 mu1 <- -0.187013  
166 d1 <- 0.39449  
167  
168 eta1 <- c(ar1,d1,mu1)  
169  
170 if(ar_w==TRUE){  
171 eta1 <- c(eta1,0.3256)  
172 }  
173  
174 eta <- c(eta1,eta0) # Parametervektoren  
175  
176 if(ar_w==FALSE){  
177 ar_w <- numeric(0)  
178 }  
179  
180  
181 var <- c(0.11830,0.256984)^2  
182  
183 eps <- Resid_gen(obs2,eta) # Residualerne  
184  
185 regime.ny <- c(regime[1:17521],rep(0,forc_steps)) #Regimevektoren med nuller i enden
```

```

186 forecastsims <- matrix(rep(0,nrow(randoms)*ncol(randoms)),nrow=nrow(randoms),ncol=
187   ncol(randoms)) # Matrix med nuller til de forecastede MC-stier
188
189 regime.forc <- matrix(rep(0,nrow(randoms)*ncol(randoms)),nrow=nrow(randoms),ncol=
190   ncol(randoms))# Regime-matrix med nuller til de forecastede regime-stier
191
192 for (k in 1:reps){
193   for (j in 1:forc_steps){
194     regime.ny[length(regime.ny)-forc_steps+j] <- ifelse(markov[regime.ny[length(regime.
195       ny)-forc_steps-1+j]+1]<randomus[j,k],1,0) # Forecaster de nye regimer
196     regime.forc[j,k] <- regime.ny[length(regime.ny)-forc_steps+j]
197   }
198   forecastsims[,k] <- MCforc(obs2, rbind(cbind(eps,eps), cbind(randoms[,k]*sqrt(var[1]),
199     randoms[,k]*sqrt(var[2]))),eta) # Forecaster priserne
200 }
201
202 season.forc <- matrix(rep(0,nrow(randoms)*ncol(randoms)),nrow=nrow(randoms),ncol=
203   ncol(randoms)) # Matrix med nuller
204
205 for (k in 1:reps){
206   season.forc[,k] <- Season.predict(omraade2,forc_steps,regime.forc[,k])$fit # Kommer
207   de fremtidige værdier af sæsonfunktionen ind i season_forc givet de forecastede
208   regimer
209 }
210
211 pris.med.season <- matrix(rep(0,nrow(randoms)*ncol(randoms)),nrow=nrow(randoms),ncol=
212   ncol(randoms)) # Matrix med nuller
213 for (k in 1:reps){
214   pris.med.season[,k] <- forecastsims[,k] + season.forc[,k] # Lægger sæsonen og den
215   sæsonrensede pris sammen
216 }
217
218 season.med.predict <- matrix(rep(0,nrow(randoms)*ncol(randoms)*3),nrow=nrow(randoms)
219   ,ncol=ncol(randoms)*3) # Matrix med nuller
220 for (k in 1:reps){
221   season.med.predict[,,(k*3-2):(k*3)] <- as.matrix(Season.predict(omraade2,forc_steps,
222     regime.forc[,k])) # For hver regimesti beregnes værdi af sæsonfunktionen samt
223   dens 95%-prædiktionsintervaller
224 }
225
226 forecastsims <- sortr(forecastsims) # Sorterer matricen
227
228 forecasts <- cbind(forecastsims[,floor(0.025*reps)+1],forecastsims[,floor(0.5*reps)
229   +1],forecastsims[,floor(0.975*reps)+1]) # Finder fejlbånd samt forecastet
230
231 output.deseason <- as.data.frame(forecasts)
232 print("output deseason=")
233
234 print(output.deseason)
235
236 write.csv(output.deseason,file=paste0("Data/Forecast/",omraade2,"_connection_",
237   omraade1,omraade2,"_med_RS_deseason.csv"),row.names=FALSE)
238
239 pris.med.season <- sortr(pris.med.season)
240
241 output.med.season <- as.data.frame(pris.med.season[,floor(0.5*reps)+1])
242
243 print("output med season=")
244
245 print(output.med.season)

```

```
233 write.csv(output.med.season,file=paste0("Data/Forecast/",omraade2,"_connection_",
234   omraade1,omraade2,"_med_RS_med_season.csv"),row.names=FALSE)
235
236 season.med.predict <- sort(season.med.predict)
237
238 season.med.predict.band <- cbind(season.med.predict[,floor(0.025*reps*3)+1],season.
239   med.predict[,floor(0.5*reps*3)+1],season.med.predict[,floor(0.975*reps*3)+1])
240
241 output.season.predict <- as.data.frame(season.med.predict.band)
242
243 print("season med baand =")
244
245 print(output.season.predict)
246
247 write.csv(output.season.predict,file=paste0("Data/Forecast/",omraade2,"_connection_"
248   ,omraade1,omraade2,"_med_RS_season_med_baand.csv"),row.names=FALSE)
249 #####
250 #Diff
251 # Indlæser de estimerede parametre for regime 0
252 ar0 <- numeric(24)
253 mu0 <- 0
254 d0 <- 0
255
256 eta0 <- c(ar0,d0,mu0)
257
258 ar_w=TRUE
259
260 if(ar_w==TRUE){
261 eta0 <- c(eta0,0)
262 }
263
264 # Indlæser de estimerede parametre for regime 1
265 ar1 <- c(0.46621)
266 mu1 <- -0.187013
267 d1 <- 0.39449
268
269 eta1 <- c(ar1,d1,mu1)
270
271 if(ar_w==TRUE){
272 eta1 <- c(eta1,0.3256)
273 }
274
275 eta <- c(eta1,eta0) # Parametervektoren
276
277 if(ar_w==FALSE){
278 ar_w <- numeric(0)
279 }
280
281 var <- c(0,0.256984)^2
282
283 eps <- Resid_gen.diff(diff,eta) # Residualerne
284
285 regime.ny <- c(regime[1:17521],rep(0,forc_steps)) # Regimevektoren med nuller i
286   enden
287
288 forecastsims <- matrix(rep(0,nrow(randoms)*ncol(randoms)),nrow=nrow(randoms),ncol=
289   ncol(randoms)) # Matrix med nuller til de forecastede MC-stier
```

```

289 regime.forc <- matrix(rep(0,nrow(randoms)*ncol(randoms)),nrow=nrow(randoms),ncol=
  ncol(randoms)) # Regime-matrix med nuller til de forecastede regime-stier
290
291 for (k in 1:reps){
292   for (j in 1:forc_steps){
293     regime.ny[length(regime.ny)-forc_steps+j] <- ifelse(markov[regime.ny[length(regime.
      ny)-forc_steps-1+j]+1]<randomus[j,k],1,0) # Forecaster de nye regimer
294     regime.forc[j,k] <- regime.ny[length(regime.ny)-forc_steps+j]
295   }
296   forecastsims[,k] <- MCforc.diff(diff, rbind(cbind(eps,eps), cbind(randoms[,k]*sqrt(var
      [1]),randoms[,k]*sqrt(var[2]))),eta) # Forecaster priserne
297 }
298
299 season.forc <- matrix(rep(0,nrow(randoms)*ncol(randoms)),nrow=nrow(randoms),ncol=
  ncol(randoms)) # Matrix med nuller
300
301 for (k in 1:reps){
302   season.forc[,k] <- Season.predict.diff(omraade1,omraade2,forc_steps,regime.forc[,k])
      $fit # Kommer de fremtidige værdier af sæsonfunktionen ind i season_forc givet
      de forecastede regimer
303 }
304
305 pris.med.season <- matrix(rep(0,nrow(randoms)*ncol(randoms)),nrow=nrow(randoms),ncol=
  ncol(randoms)) # Matrix med nuller
306 for (k in 1:reps){
307   pris.med.season[,k] <- forecastsims[,k] + season.forc[,k] # Lægger sæsonen og den
      sæsonrensede pris sammen
308 }
309
310 season.med.predict <- matrix(rep(0,nrow(randoms)*ncol(randoms)*3),nrow=nrow(randoms)
  ,ncol=ncol(randoms)*3) # Matrix med nuller
311 for (k in 1:reps){
312   season.med.predict[, (k*3-2):(k*3)] <- as.matrix(Season.predict.diff(omraade1,
      omraade2,forc_steps,regime.forc[,k])) # For hver regimesti beregnes værdi af
      sæsonfunktionen samt dens 95%-prædiktionsintervaller
313 }
314
315 forecastsims <- sortr(forecastsims) # Sorterer matricen
316
317 forecasts <- cbind(forecastsims[,floor(0.025*reps)+1],forecastsims[,floor(0.5*reps)
  +1],forecastsims[,floor(0.975*reps)+1]) # Finder fejlbånd samt forecastet
318
319 output.deseason <- as.data.frame(forecasts)
320 print("output deseason=")
321
322 print(output.deseason)
323
324 write.csv(output.deseason,file=paste0("Data/Forecast/Diff_connection_",omraade1,
  omraade2,"_med_RS_deseason.csv"),row.names=FALSE)
325
326 pris.med.season <- sortr(pris.med.season)
327
328 output.med.season <- as.data.frame(pris.med.season[,floor(0.5*reps)+1])
329
330 print("output med season=")
331
332 print(output.med.season)
333
334 write.csv(output.med.season,file=paste0("Data/Forecast/Diff_connection_",omraade1,
  omraade2,"_med_RS_med_season.csv"),row.names=FALSE)
335

```

```
336 season.med.predict <- sort(season.med.predict)
337
338 season.med.predict.band <- cbind(season.med.predict[,floor(0.025*reps*3)+1],season.
339   med.predict[,floor(0.5*reps*3)+1],season.med.predict[,floor(0.975*reps*3)+1])
340
341 output.season.predict <- as.data.frame(season.med.predict.band)
342
343 print("season med baand =")
344
345 print(output.season.predict)
346 write.csv(output.season.predict,file=paste0("Data/Forecast/Diff_connection_",
347   omraade1,omraade2,"_med_RS_season_med_baand.csv"),row.names=FALSE)
```

R-kode D.14: Hovedfilen til forecast ved anvendelse af modeller med regimeskift.