

OBLIGATORISK OPPGAVE 3

STA-2003-Tidsrekker

24. april 2019

Martin Soria Røvang
Universitetet i Tromsø

Inneholder 11 sider, inkludert forside.

Innhold

1 Oppgave	3
1.1 a)	3
1.2 b)	3
1.3 c-d)	3
1.4 e)	4
2 Oppgave 2	6
2.1 a-b)	6
3 Oppgave 3	6
3.1 a-b)	6
4 Appendix	8
5 Referanser	11

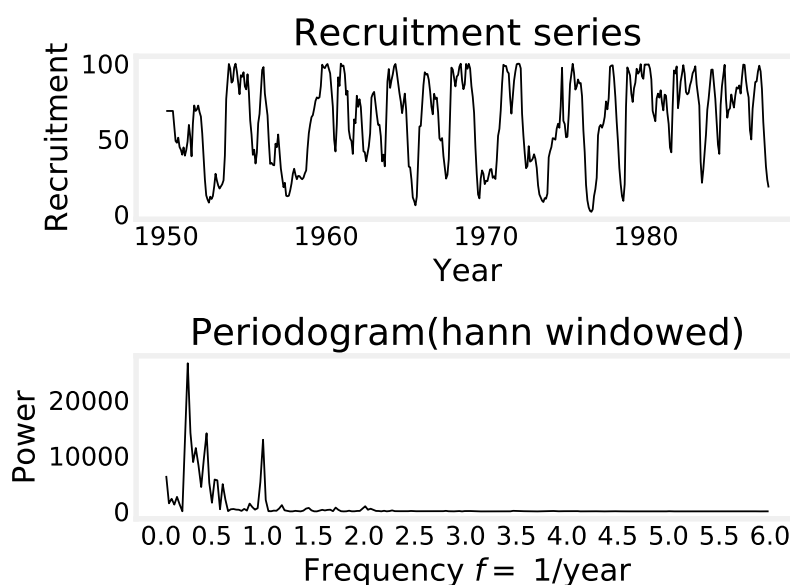
1 Oppgave

1.1 a)

Ved bruk av en vektet periodogram vist i ligning(1.1) kan vi se på energien til tidsrekken. Denne viser at vi har kraftig periode rundt $f = 0.5$ år og på $f = 1$ år, derfor kan man se at det er sesongvariasjoner i tidsrekken.

$$S_{xx}(f) = \frac{dt}{NU} \left| \mathfrak{F} \{ w[n] \cdot x[n] \} \right|^2 \quad (1.1)$$

Her er S_{xx} kraften på frekvenskomponentene, dt er tidssteget (i dette tilfelle $dt = 1$), \mathfrak{F} er Fouriertransformasjonen, $w[n]$ er vinduet (brukt hann vindu) og $x[n]$ er tidsrekken. Resultatet er i figur(1).



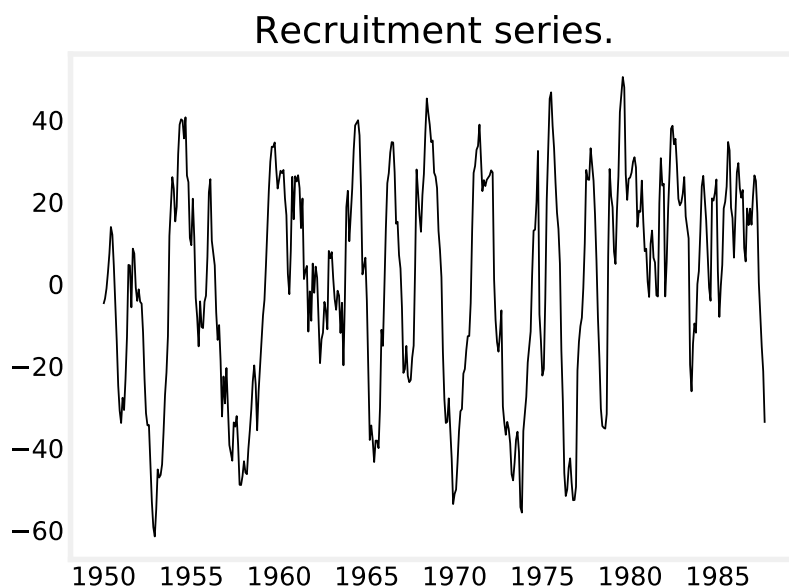
Figur 1: Tidsrekken plottet med vektet periodogram. Her kan man se forskjellige periodisiteter rundt $f = 0.5$ år og en på $f = 1$ år. Frekvensaksen har blitt ganget med [12 måneder/år] for å få enhet 1/år.

1.2 b)

Her trekker vi fra midlere sesongvariasjoner for å gjøre tidsrekken stasjonær. Resultatet er plottet i figur(2).

1.3 c-d)

Vi har plottet av ACF og PACF i figur(3). Her kan man observere at ACF plottet har den karakteristiske AR(0) modellen der det konvergerer mot null når $h \rightarrow 0$, men denne gir ingen indikasjon på orden av AR. I PACF ser man at det er en korrelasjon ved $h = 1$ og $h = 2$, resten ser ut til å



Figur 2: Trukket fra midlere sesongvariasjoner for å gjøre tidsrekken stasjonær.

være hvit støy da dette ligger under 95% konfidensinterval gitt i ligning(1.2). På grunn av den klare indikatoren på korrelasjon ved $h = 1$ og $h = 2$ kan vi si at vi har en AR(2) eller ARMA(2, 0) prosess.

$$\sigma_w = \frac{2}{\sqrt{N}} \quad (1.2)$$

N er lengden på tidsserien.[p. 31 Shumway [2017]]

1.4 e)

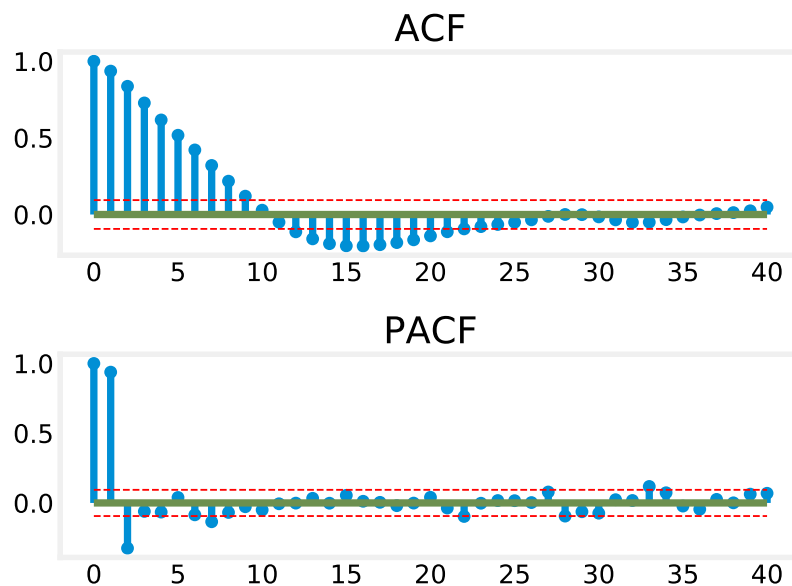
Ved bruk av statsmodels-pakken i python kan vi simulere en ARMA-modell med gitte parametere, i figur(4) ser vi en resultatet fra en ARMA(2,0) modell. Fra utskriften fikk vi modellen gitt i ligning()

$$\hat{x}_n = \alpha + \phi_1 \hat{x}_{n-1} + \phi_2 \hat{x}_{n-2} + \hat{w}_t \quad (1.3)$$

der $\alpha = -0.7019$, $\phi_1 = 1.2483$ og $\phi_2 = -0.3313$ Her har det blitt brukt en *hatt* på tidsrekken for å vise at det er et estimat. Fra utskriften i figur(4) har vi at $\sigma_w = 8.653$, dette er den estimerte variansen på den hvite støyen slik man har fordelingen $N \sim (0, \sigma_w^2)$. Dette kan brukes til å finne *Mean-square prediction error* som er gitt ved ligning(1.4),

$$P_{n+m}^n = E \left[\sigma_w^2 \sum_{j=0}^{m-1} \psi_j^2 \right] \quad (1.4)$$

der ψ er vektene gitt fra modellen slik at $\phi(z)\psi(z) = \theta(z)$. Her er $\psi(z)$ og $\theta(z)$ de karakteristiske ligningene til AR og MA modellen og $\psi(z)$ er vektene gitt ved $\psi(z) = (1 + \psi_1 z + \psi_2 z^2 + \dots + \psi_j z^j + \dots)$



Figur 3: Plot av ACF og PACF for den stasjonære tidsrekken i figur(2)

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

ARMA Model Results

Dep. Variable:

y

No. Observations:

453

Model:

ARMA(2, 0)

Log Likelihood

-1616.790

Method:

css-mle

S.D. of innovations

8.564

Date:

Tue, 16 Apr 2019

AIC

3241.579

Time:

12:27:46

BIC

3258.043

Sample:

0

HQIC

3248.066

coef

std err

z

P>|z|

[0.025

0.975]

const

-0.7019

4.773

-0.147

0.883

-10.057

8.653

ar.L1.y

1.2483

0.044

28.159

0.000

1.161

1.335

ar.L2.y

-0.3313

0.044

-7.471

0.000

-0.418

-0.244

Roots

Real

Imaginary

Modulus

Frequency

AR.1

1.1555

+0.0000j

1.1555

0.0000

AR.2

2.6120

+0.0000j

2.6120

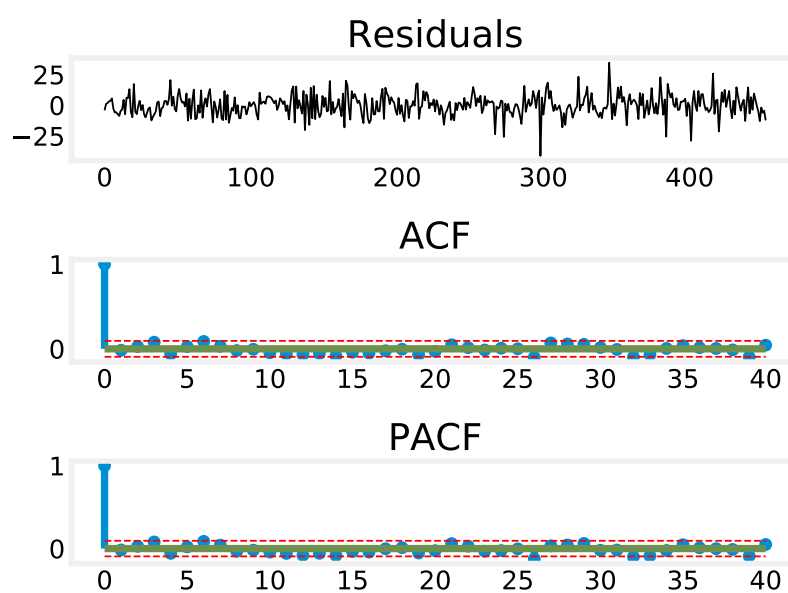
0.0000

Figur 4: Utskrift av ARMA-model resultat gitt fra statsmodels-pakken. Merk her at det har blitt brukt MLE for å estimere parametere. https://www.statsmodels.org/dev/generated/statsmodels.tsa.arima_model.ARMA.html

2 Oppgave 2

2.1 a-b)

Ved å trekke ifra modellen(uten det hvite støyleddet) på den originale tidsserien får vi residuale-
ne/feilene, resultatet er vist i figur(5). Fra plottet av ACF og PACF ser man at alt ligger under
95% konfidensintervallet for hvit støy, dette kan være en indikator på at modellen er god fordi
vi kun står igjen med hvit støy. MERK: Her har vi kun brukt ACF og PACF opp til lag $h = 40$ det
kan være at det ligger noe utafor dette(for eksempel at vi har korrelasjon mellom lag x_t, x_{t+100}),
dette gjelder også for det ACF og PACF i de tidligere oppgavene.

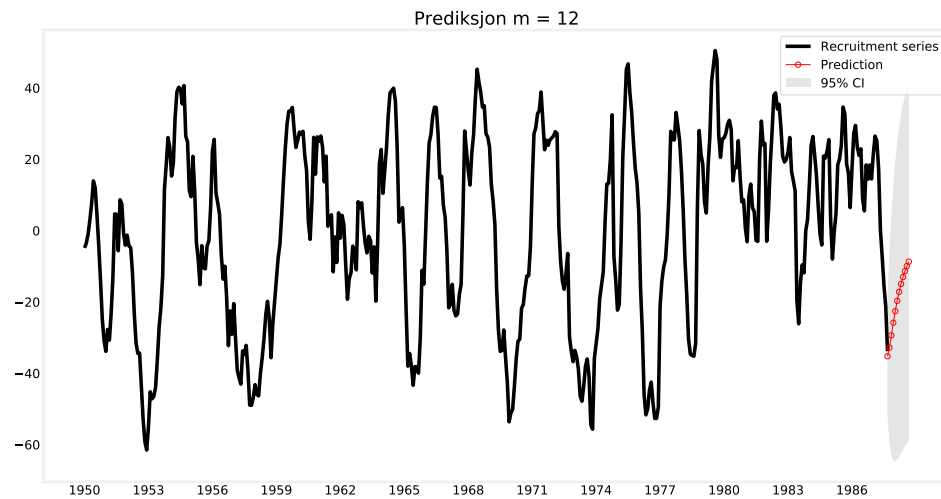


Figur 5: Plot av ACF og PACF av residualene.

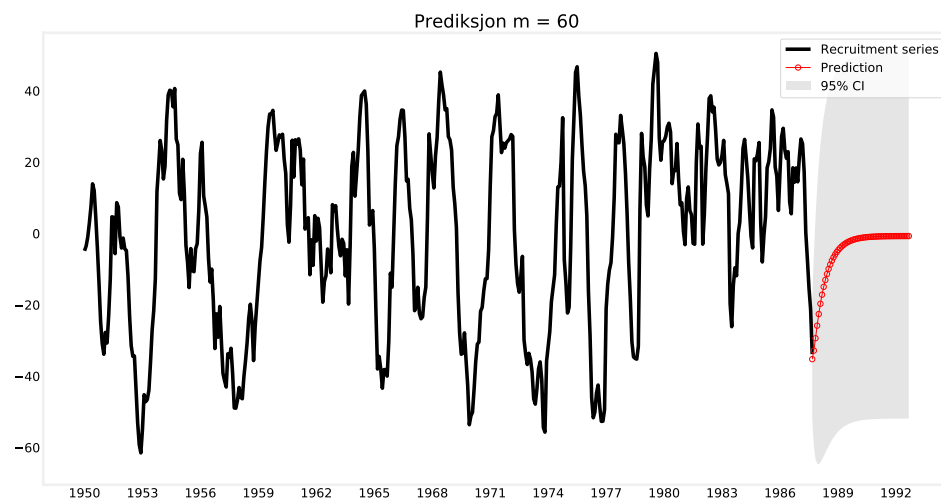
3 Oppgave 3

3.1 a-b)

Prediksjon har blitt gjort med statsmodels sin ARMA-predict funksjon i python. Resultatet med
12 steg prediksjon er vist i figure(6). Feilen konvergerer veldig fort til variansen av tidsrekken
 σ_x som vist i figur(7). Man kan også observere at prediksjonen konvergerer mot gjennomsnittet
av tidsrekken.



Figur 6: Prediksjon med $M = 12$ måneder.



Figur 7: Prediksjon med $M = 60$ måneder. Her kan man se at feilen konvergerer mot variansen til tidsserien når man bruker høy m .

4 Appendix

```
1
2     from statsmodels.tsa.stattools import acf, pacf, ccf
3     from statsmodels.tsa.arima_process import arma2ma, arma2ar
4     import numpy as np
5     import matplotlib.pyplot as plt
6     import pandas as pd
7     import os
8     import statsmodels as sm
9     plt.style.use('fivethirtyeight')
10    plt.rcParams['axes.facecolor']='white'
11    plt.rcParams['savefig.facecolor']='white'
12    plt.rcParams['axes.grid']='off'
13
14    # Load data
15    rec = pd.read_csv('data/rec.txt', delimiter='\t')
16    rec_df = pd.DataFrame(rec)
17    time = np.copy(rec_df['year'])
18    X = np.copy(rec_df['recruitment'])
19
```

Figur 8: Load files


```

1  def w_periodogram(x, dt = 1):
2      """Windowed periodogram"""
3      #x = np.pad(x, (0,300), 'constant')
4      N = len(x)
5      n = np.arange(0,N,1)
6      # Hann window
7      window = (1/2)*(1 - np.cos(2*np.pi*n/(N-1)))
8      U = (1/N)*np.sum(window**2)
9      spectrum = np.abs(np.fft.fftshift(np.fft.fft(window*x))**2)
10     spectrum *= (dt/(N*U))
11     freq = np.fft.fftshift(np.fft.fftfreq(N, dt))
12
13     return freq[int(N/2):], spectrum[int(N/2):]
14
15 # Task A
16
17 # Find periodogram
18 freq, periodogram_X = w_periodogram(X)
19
20
21 # Plot data
22 fig, ax = plt.subplots(2,1)
23 ax[0].plot(time, X, color = 'black')
24 ax[0].set_title('Recruitment series')
25 ax[0].set_xlabel('Year')
26 ax[0].set_ylabel('Recruitment')
27 ax[1].plot(freq[2:]*12, periodogram_X[2:], color = 'black')
28 ax[1].set_title('Periodogram(hann windowed)')
29 ax[1].set_xlabel('Frequency $\Delta f = $ year')
30 ax[1].set_ylabel('Power')
31 ax[1].set_xticks([x for x in np.arange(0, 6.5, 1/2)])
32 plt.tight_layout()
33 plt.savefig('rapport/task_a.pdf')
34 plt.show()
35
36 # Task B
37
38 def remove_season(x):
39     C = np.zeros(12)
40     for m in range(0,12):
41         C[m] = np.mean(x[m::12])
42
43     # repeat C to create a periodic signal of equal length or longer than the
44     dataset
45     repC = np.tile(C, int(np.ceil(len(x)/12)))
46     # compute residual (by subtracting periodic signal)
47     X = x - repC[:len(x)]
48     return X
49
50 # Make stationary
51 X_remseason = remove_season(X)
52
53 plt.plot(time, X_remseason, color = 'black')
54 plt.title('Recruitment series.')
55 plt.tight_layout()
56 plt.savefig('rapport/task_b.pdf')
57 plt.show()
58
59 # TASK C
60
61 # make whitenoise Confidens intervall
62 wt_line = 2*np.tile(1/np.sqrt(len(X_remseason)), 41)
63
64 # Plot
65 fig, ax = plt.subplots(2,1)
66 # ax[0].plot(time, X_remseason)
67 # ax[0].set_title('Stasjonre tidsserien')
68 ax[0].stem(acf(X_remseason))
69 ax[0].set_title('ACF')
70 ax[0].plot(wt_line, '--', color = 'red', linewidth = 1); ax[0].plot(-wt_line, '--',
71 color = 'red', linewidth = 1)
72 ax[1].stem(pacf(X_remseason))
73 ax[1].set_title('PACF')
74 ax[1].plot(wt_line, '--', color = 'red', linewidth = 1); ax[1].plot(-wt_line, '--',
75 color = 'red', linewidth = 1)
76 plt.tight_layout()

```

```

1      # Oppgave 2
2
3
4      # Task A
5
6      # Create model.
7      model = sm.tsa.arima_model.ARIMA(X_remseason, order=(2, 0, 0))
8      model_fit = model.fit()
9
10     # Get residuals
11     res = model_fit.resid
12
13     # Plot
14     fig, ax = plt.subplots(3,1)
15     ax[0].plot(res, color = 'black')
16     ax[0].set_title('Residuals')
17     ax[1].stem(acf(res))
18     ax[1].plot(wt_line, '--', color = 'red', linewidth = 1); ax[1].plot(-wt_line, '
19 --', color = 'red', linewidth = 1)
20     ax[1].set_title('ACF')
21     ax[2].stem(pacf(res))
22     ax[2].plot(wt_line, '--', color = 'red', linewidth = 1); ax[2].plot(-wt_line, '
23 --', color = 'red', linewidth = 1)
24     ax[2].set_title('PACF')
25     plt.tight_layout()
26     plt.savefig('rapport/task_2a.pdf')
27     plt.show()

```

Figur 10: Task 2

```

1      # Oppgave 3
2
3
4      # Task A
5
6      year = 5
7      M = 12*year # 12*2 months (2 years)
8      forecast, stderr, conf_int = model_fit.forecast(steps = M)
9      # Default: 95% konfidensintervall
10
11
12     sliced_time = time
13     sliced_X = X_remseason
14     time_forecast = np.linspace(sliced_time[-1], sliced_time[-1] + M/12, M)
15
16     # Plot
17     plt.figure(figsize = [15,8])
18     plt.plot(sliced_time, sliced_X, color = 'black', label = 'Recruitment series')
19     plt.plot(time_forecast, forecast, '-o', mfc='none', color = 'red', linewidth = '
20 1', label = 'Prediction' )
21     plt.fill_between(time_forecast, conf_int[:,0], conf_int[:,1], facecolor = (0, 0,
22 1, 0.2), label = '95% CI')
23     plt.xticks([x for x in np.arange(sliced_time[0], sliced_time[-1]+ M/12, 3)])
24     plt.legend(loc = 'best')
25     plt.title('Prediksjon m = %s'%M)
26     plt.tight_layout()
27     plt.savefig('rapport/task_33.pdf')
28     plt.show()

```

Figur 11: Task 3

5 Referanser

Robert H. Shumway. *Time Series Analysis and its Applications*. Springer texts in statistics, fourth edition, 2017.