시계열 분석 HW 02

2021234640 이종현

Table of Contents

1. Figure 2-3
2. Figure 2-5
3. Figure 2-6
4. Figure 2-7
5. Figure 2-8
6. Figure 2-9
7. Figure 2-10
8. Figure 2-11
9. Figure 2-12
10. Figure 2-13
11. Figure 2-14
12. Figure 2-15
13. Figure 2-16
14. Figure 2-17
15. Exercise 2-3
16. Exercise 2-6
17. Exercise 2-9
18. Appendix
    1. R code script

2. Python code script with graphics

Figure 2–3

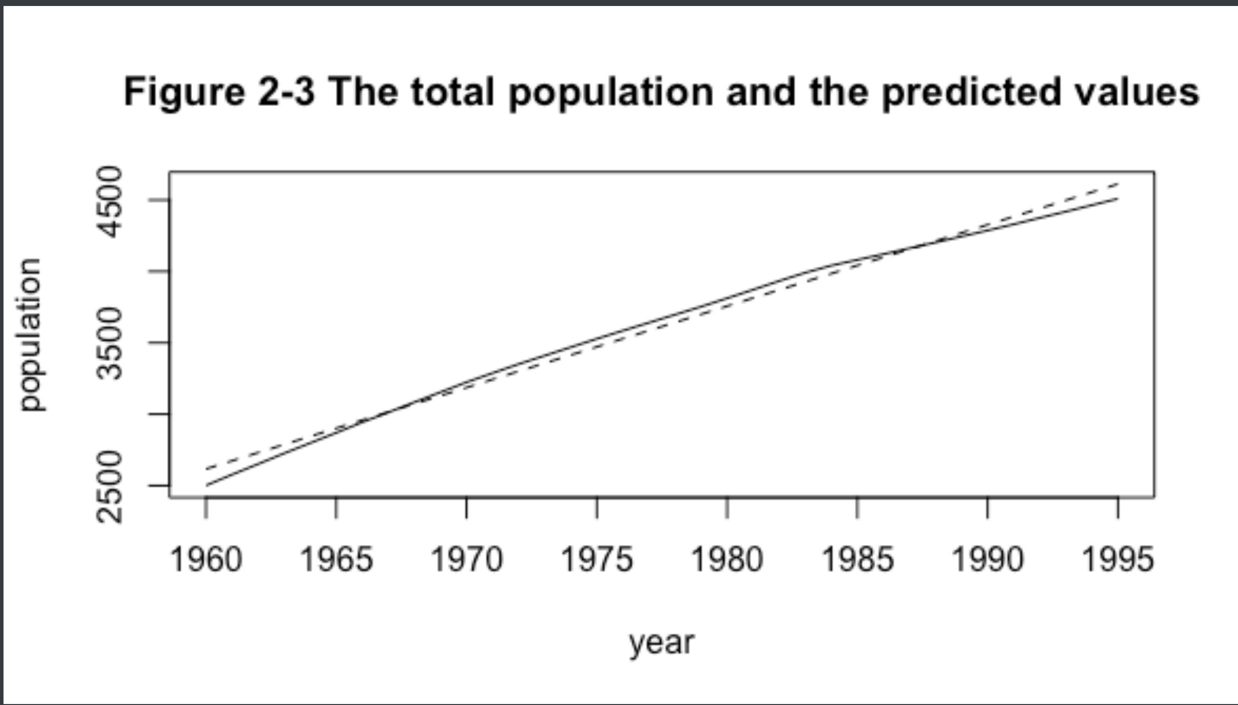


Figure 2–5

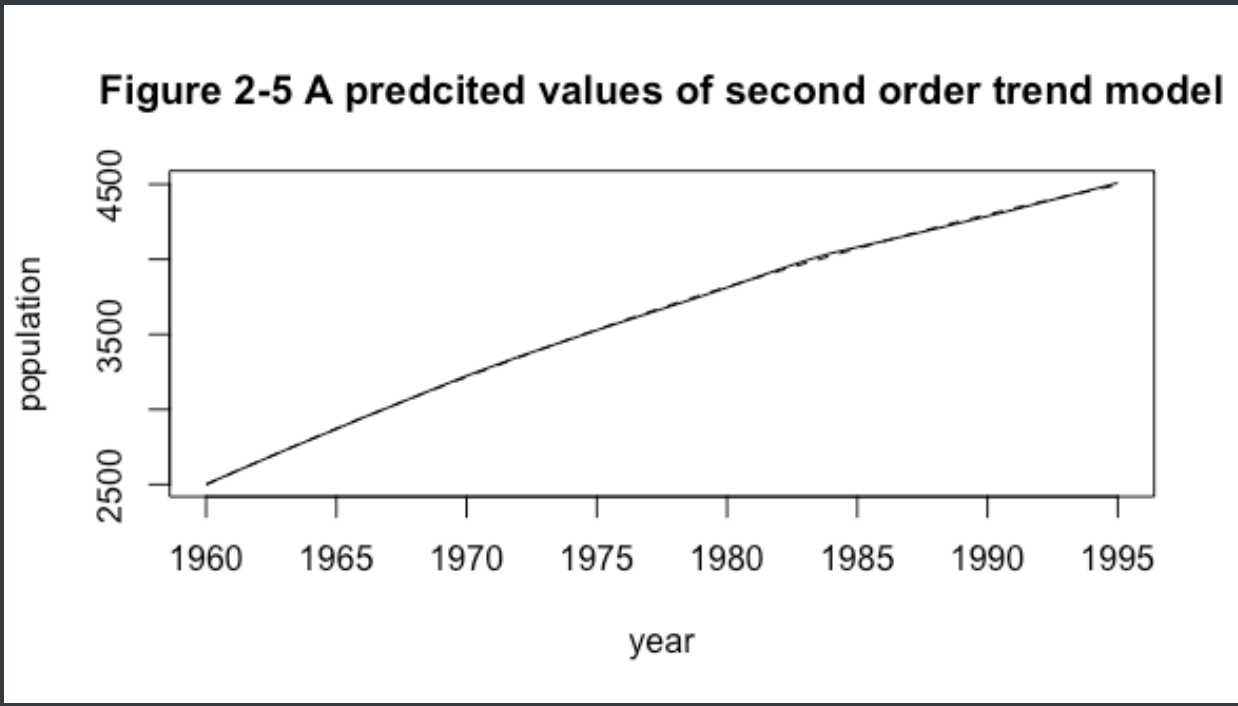


Figure 2-5 A predcited values of second order trend model
population
2500
3500
4500
1960
1965
1970
1975
1980
1985
1990
1995
year

Figure 2–6

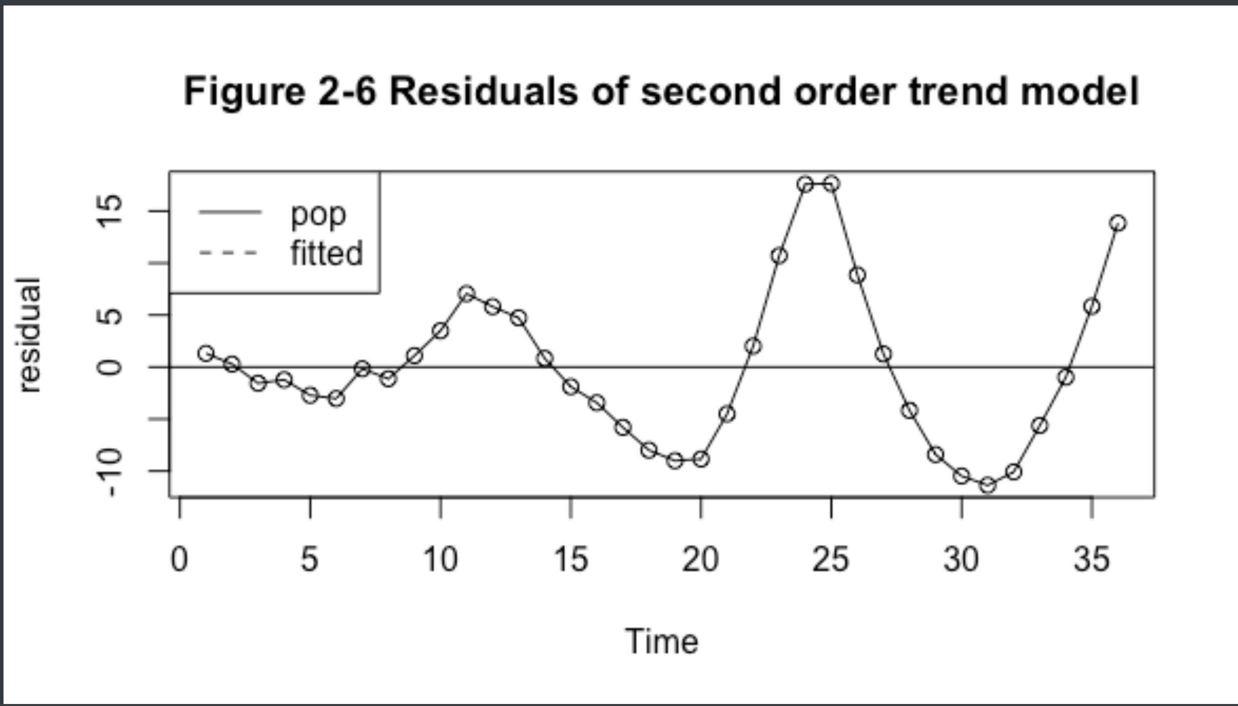


Figure 2-6 Residuals of second order trend model
pop
fitted
residual
-10
0
5
15
0
5
10
15
20
25
30
35
Time

Figure 2–7

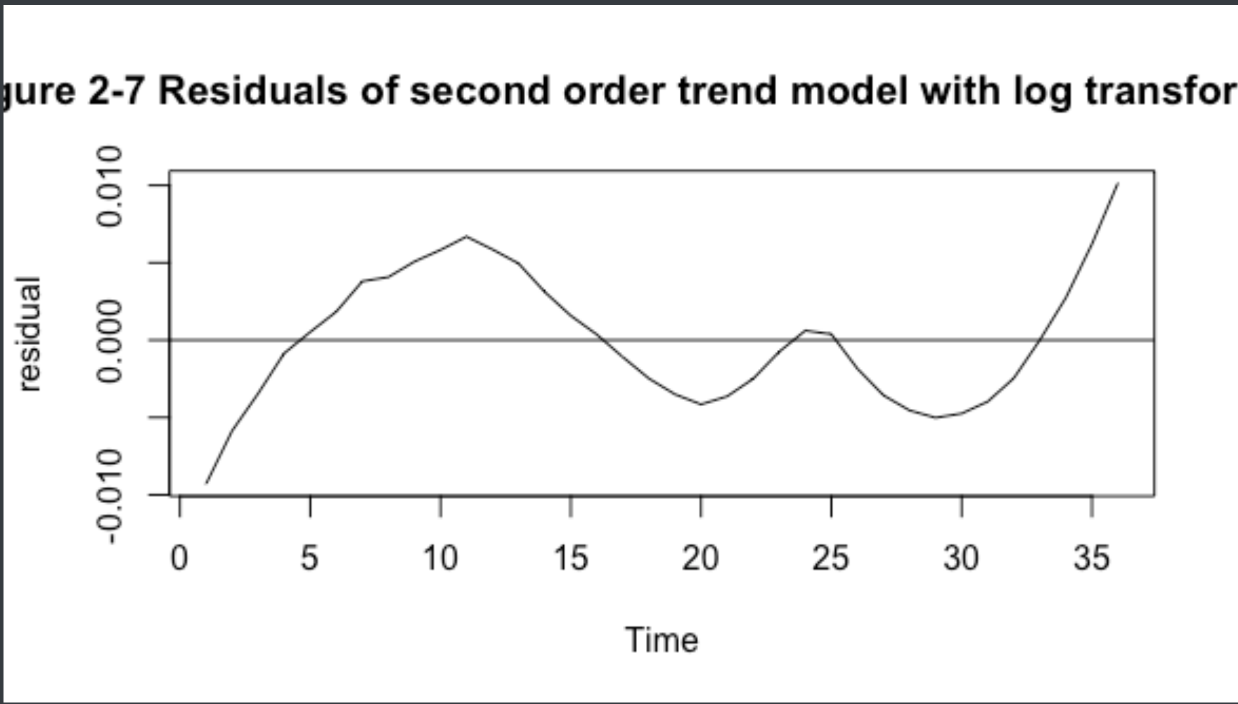


Figure 2–8

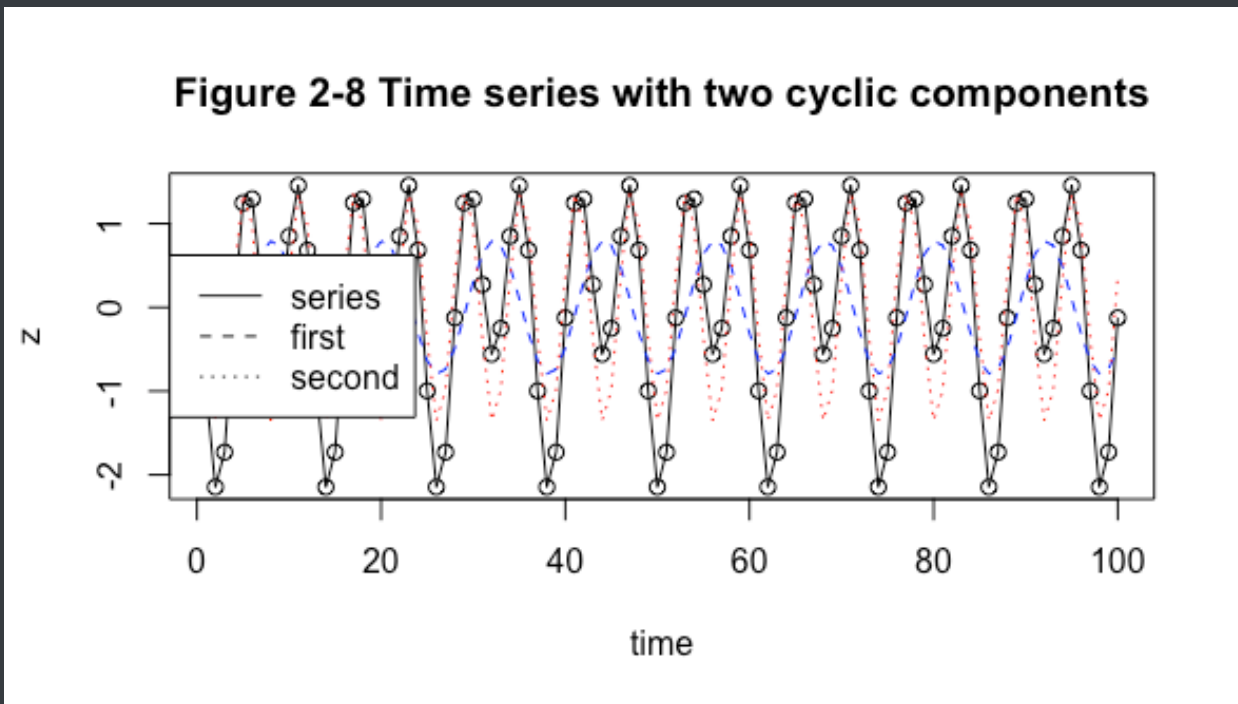


Figure 2–9

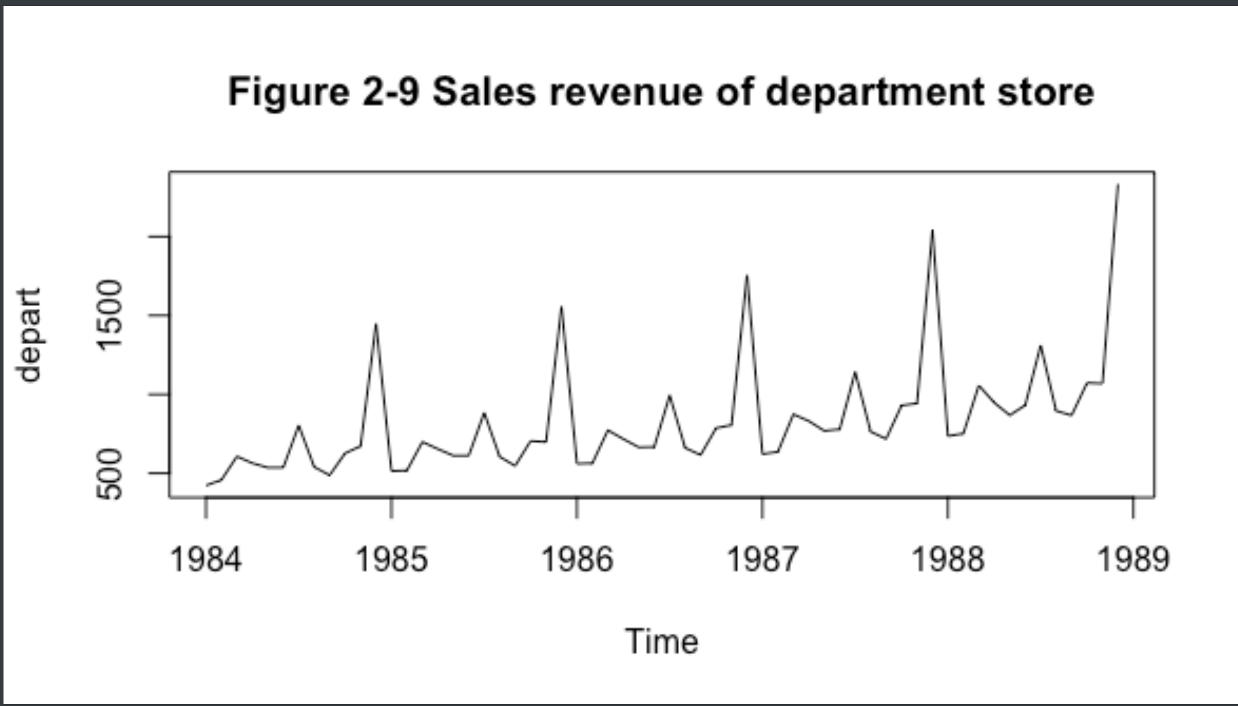


Figure 2–10

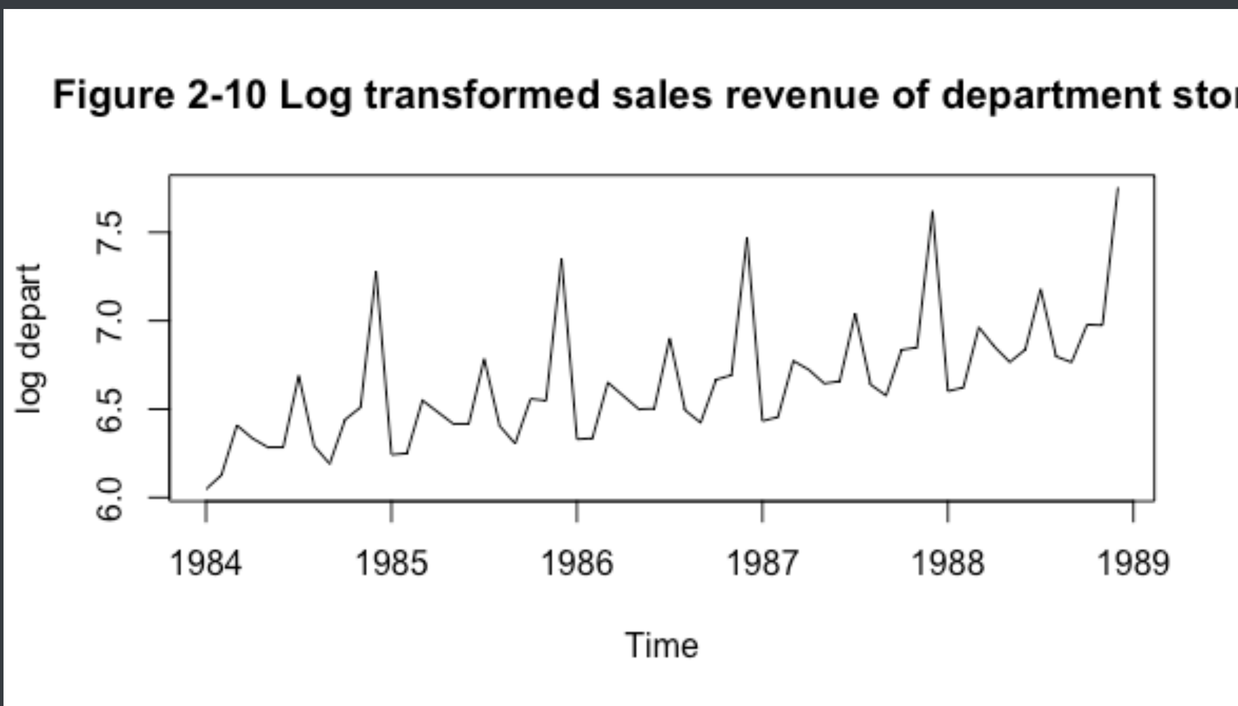


Figure 2–11

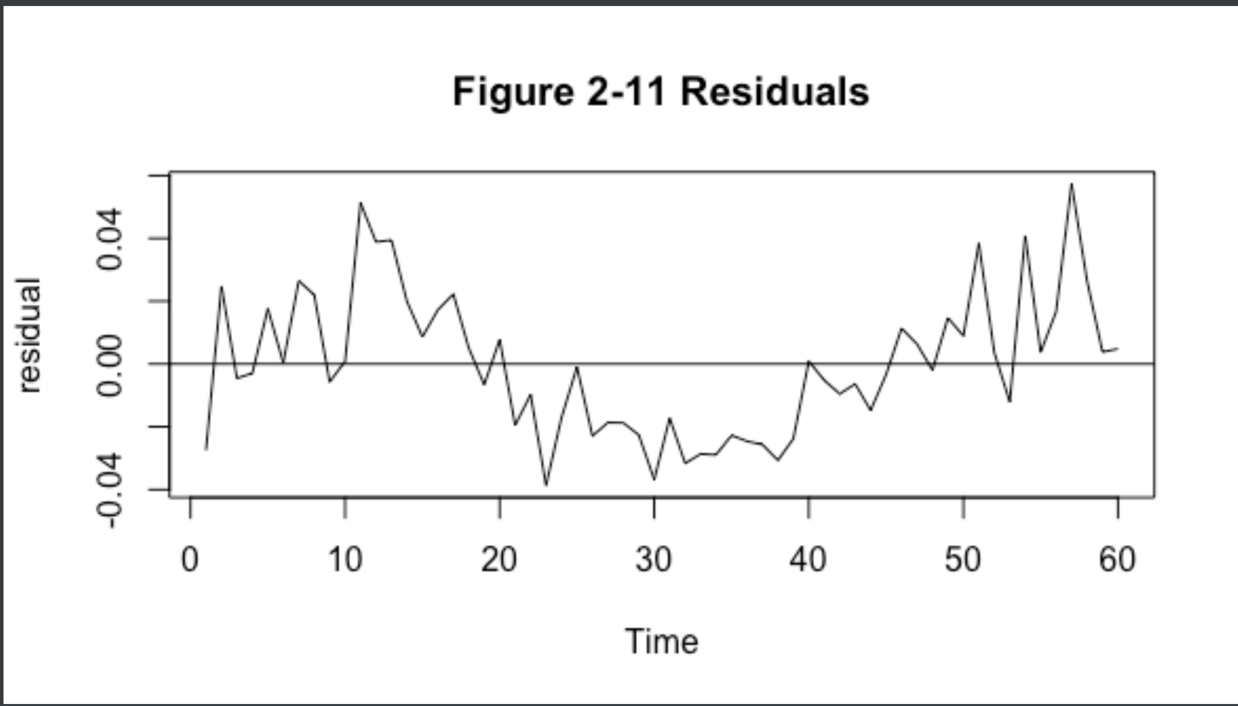


Figure 2–12

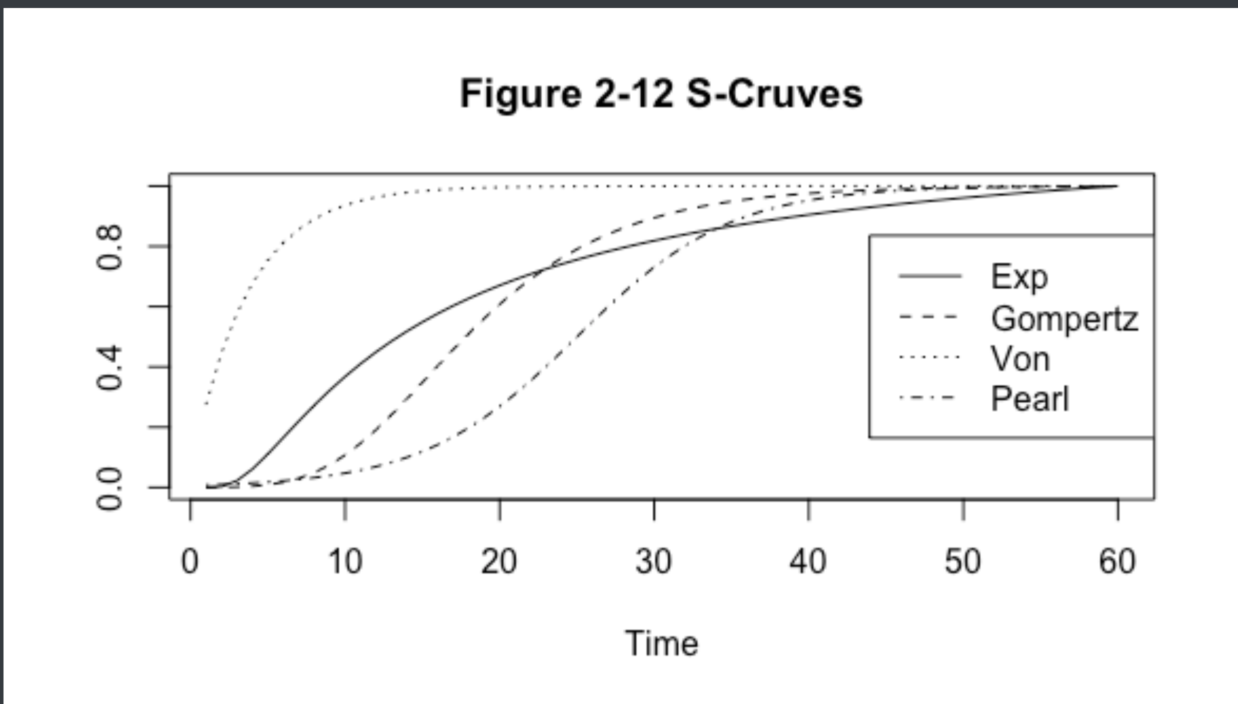


Figure 2–13

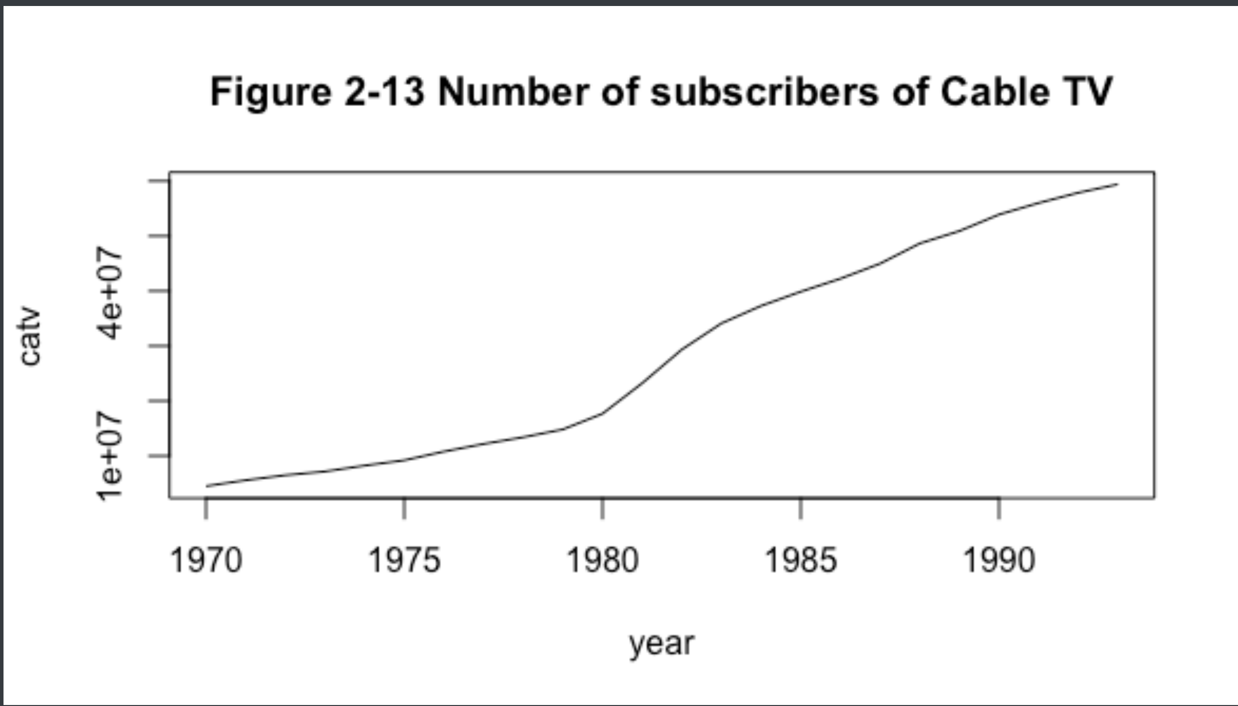


Figure 2–14

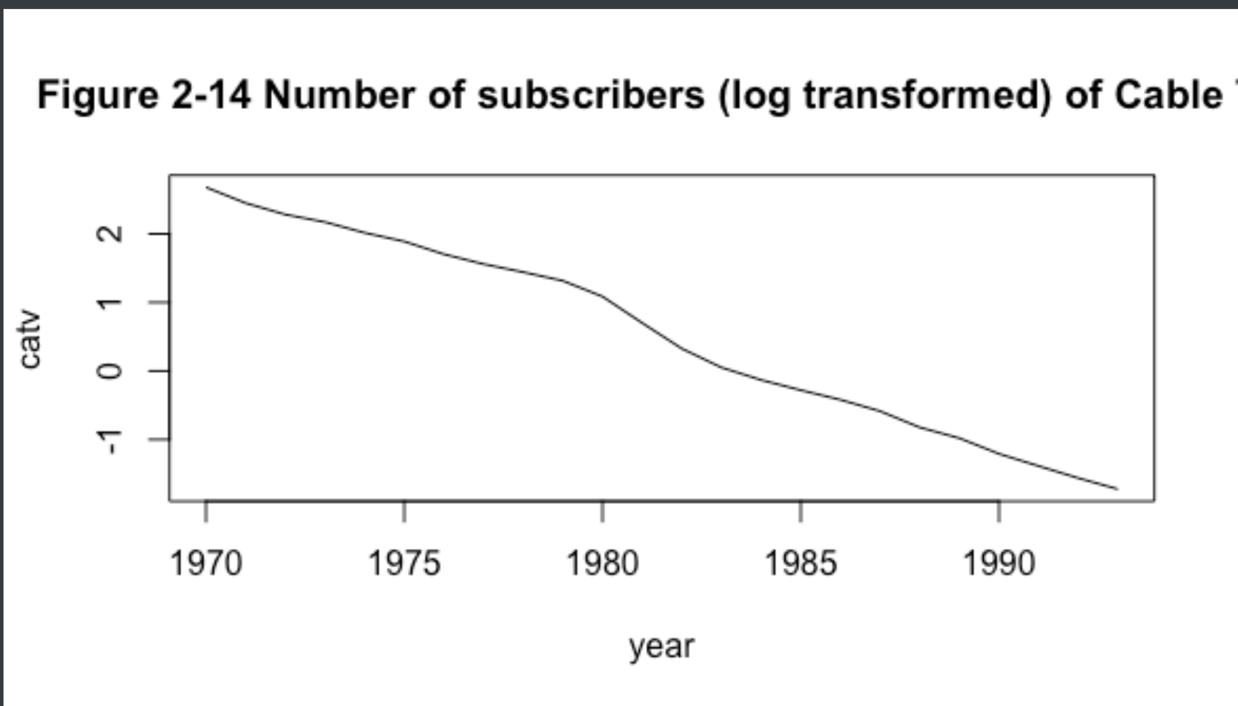


Figure 2–15

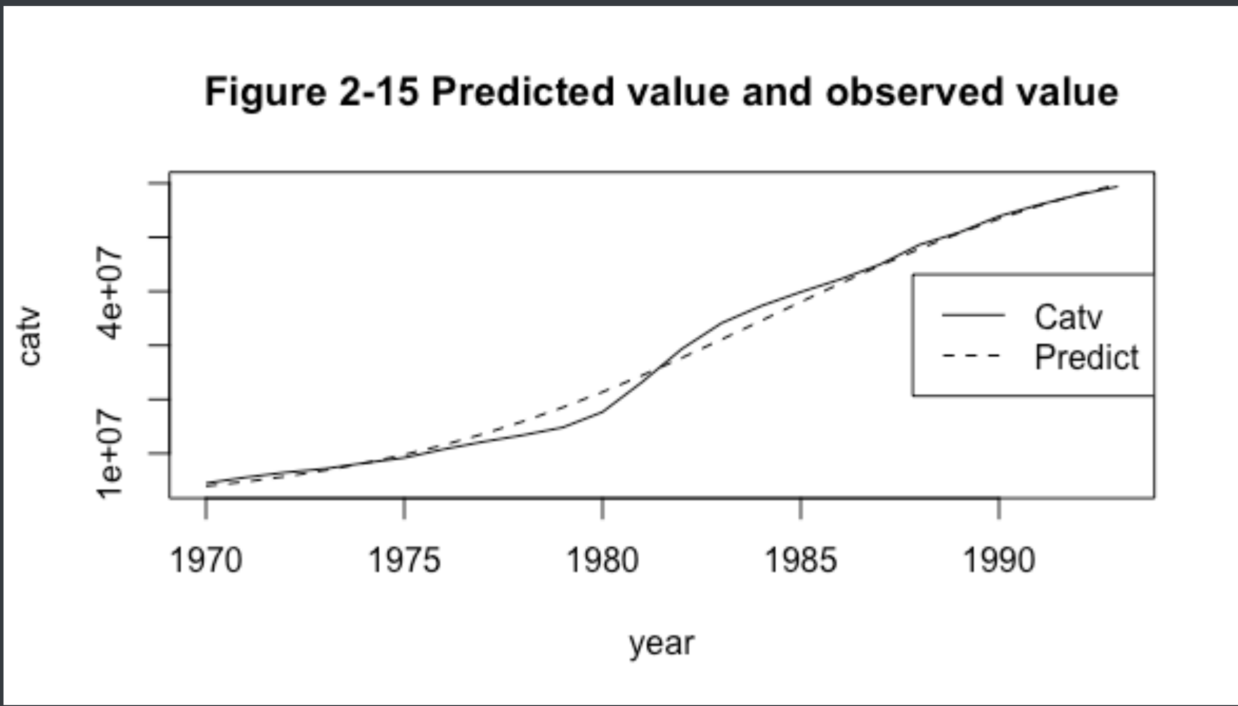


Figure 2–16

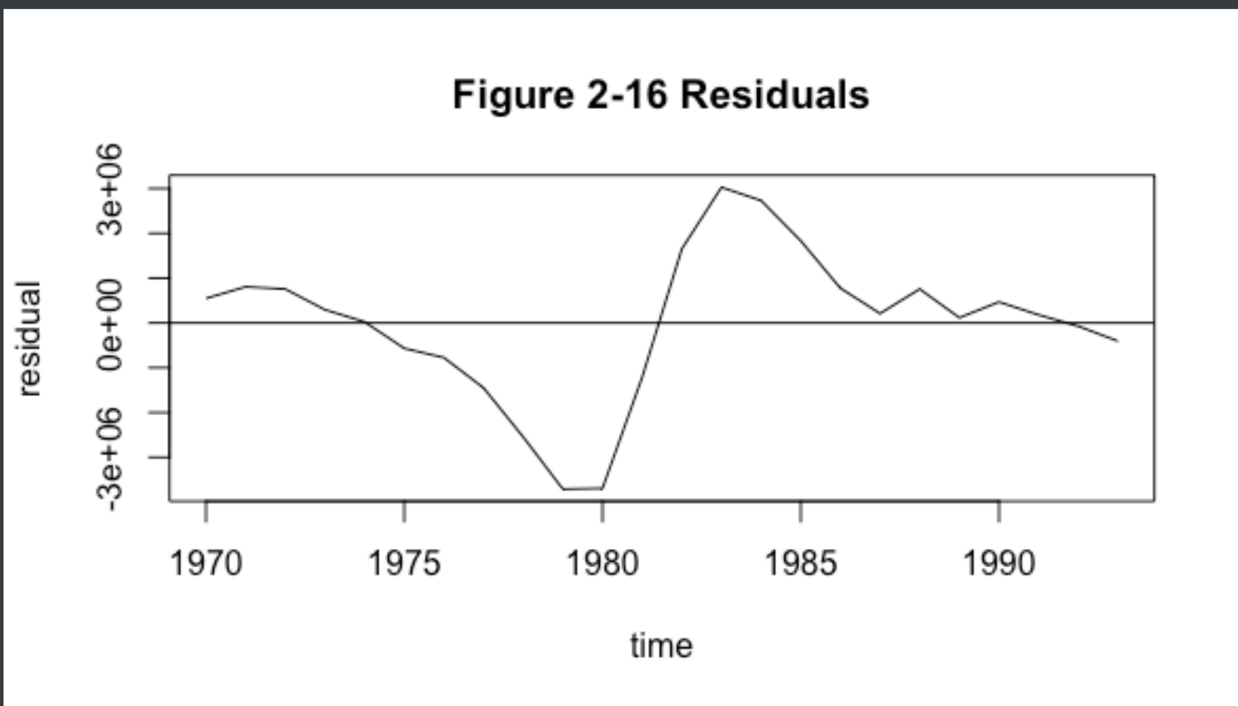
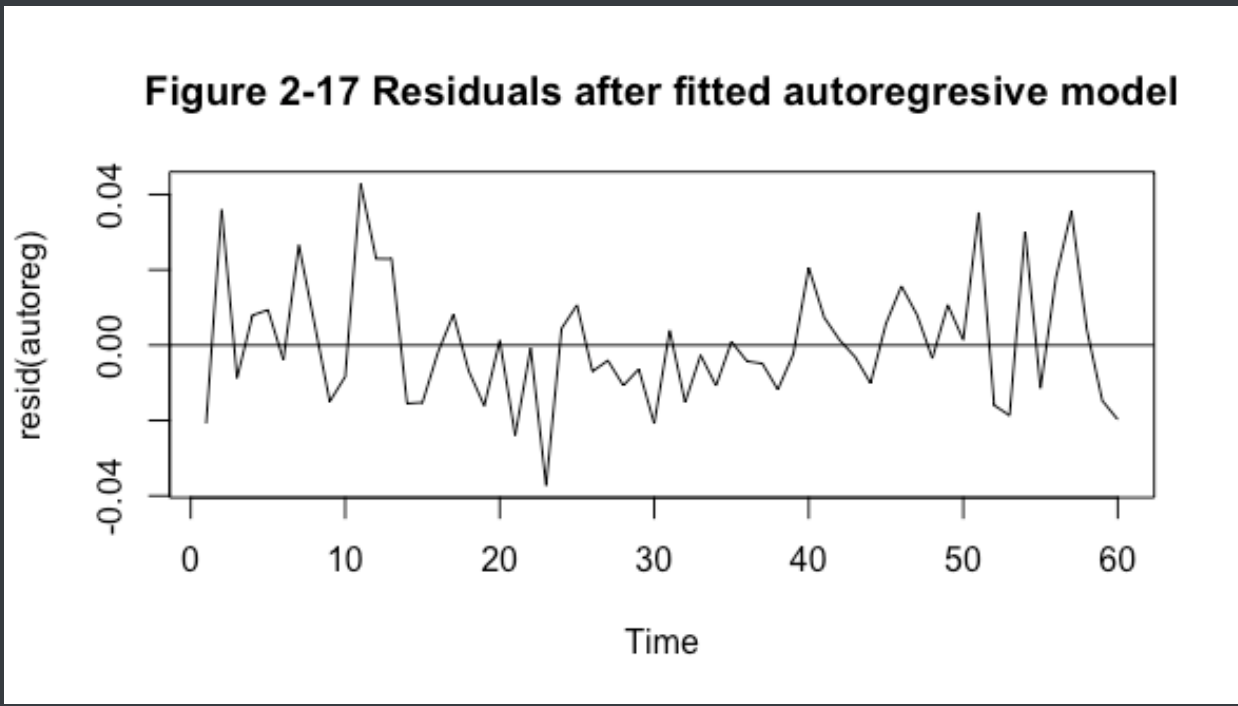


Figure 2-17



Exercise 2-3

A

Beta 0, Beta 1 의 LSE 추정값은 다음과 같다.

(A)

$x_t = t, \quad \bar{x} = \frac{n+1}{2}$

$$\hat{\beta}_1 = \frac{\sum_{t=1}^{n} (x_t - \bar{x}) z_t}{\sum (x_t - \bar{x})^2}$$

$$\sum \left(t - \frac{n+1}{2}\right)^2$$

$$= \sum \left(t^2 - t\cdot(n+1) + \frac{(n+1)^2}{4}\right)$$

$$\sum_{t=1}^{n} t^2 = \frac{n(n+1)(2n+1)}{6}$$

$$\sum_{t=1}^{n} t = \frac{n(n+1)}{2}$$

$$= \frac{n(n+1)(2n+1)}{6} - \frac{n(n+1)}{2}(n+1) + \frac{n(n+1)^2}{4}$$

$$= \frac{n(n+1)}{2}\left(\frac{2n+1}{3} - n - 1 + \frac{n+1}{2}\right)$$

$$\frac{4n+2-6n-6+3n+3}{6} = \frac{n-1}{6}$$

$$= \frac{n(n^2-1)}{12}$$

$$= \frac{12}{n(n^2-1)} \sum x_t z_t - \frac{n+1}{2} \cdot \frac{12}{n(n^2-1)} \sum z_t$$

$$= \frac{12}{n(n^2-1)} \sum t z_t - \frac{6}{n(n-1)} \sum z_t$$

$$\hat{\beta}_0 = \bar{z} - \hat{\beta}_1 \bar{x}$$

$$= \frac{1}{n}\sum z_t - \left(\frac{12}{n(n^2-1)} \sum t z_t - \frac{6}{n(n-1)} \sum z_t\right)\frac{n(n+1)}{2}$$

$$= \frac{n-1+3n+3}{n(n-1)} \sum z_t - \frac{6}{n(n-1)} \sum t z_t$$

$$= \frac{4n+2}{n(n-1)} \sum z_t - \frac{6}{n(n-1)} \sum t z_t$$

B

Beta 0, Beat 1 의 공분산, 분산은 다음과 같다.

(B)

$$Cov(\hat{\beta}) = \begin{bmatrix} Var(\hat{\beta}_0) & Cov(\hat{\beta}_0, \hat{\beta}_1) \\ Cov(\hat{\beta}_0, \hat{\beta}_1) & Var(\hat{\beta}_1) \end{bmatrix} \Rightarrow \sigma^2 \begin{bmatrix} \frac{1}{n} + \frac{\bar{x}^2}{\Sigma(x_t - \bar{x})^2} & -\frac{\bar{x}}{\Sigma(x_t - \bar{x})^2} \\ -\frac{\bar{x}}{\Sigma(x_t - \bar{x})^2} & \frac{1}{\Sigma(x_t - \bar{x})^2} \end{bmatrix}$$

$$x_t = t, \quad \sum_{k=1}^{n} k = \frac{n(n+1)}{2}, \quad \bar{x} = \frac{n+1}{2}, \quad \sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$$

$$Var(\hat{\beta}_0) = \frac{1}{n} + \frac{\left(\frac{n+1}{2}\right)^2}{\Sigma(x_t - \bar{x})^2} \cdot \sigma_\varepsilon^2 \rightarrow$$ 앞서 $\frac{n(n^2-1)}{12}$ 을 얻음

$$\frac{1}{n} + \frac{\frac{(n+1)^2}{4}}{\frac{n(n^2-1)}{12}} = \frac{1}{n} + \frac{3(n+1)^2}{n(n^2-1)} = \frac{3(n+1)^2}{n(n+1)(n-1)} + \frac{1}{n} = \frac{3(n+1)+n-1}{n(n-1)}$$

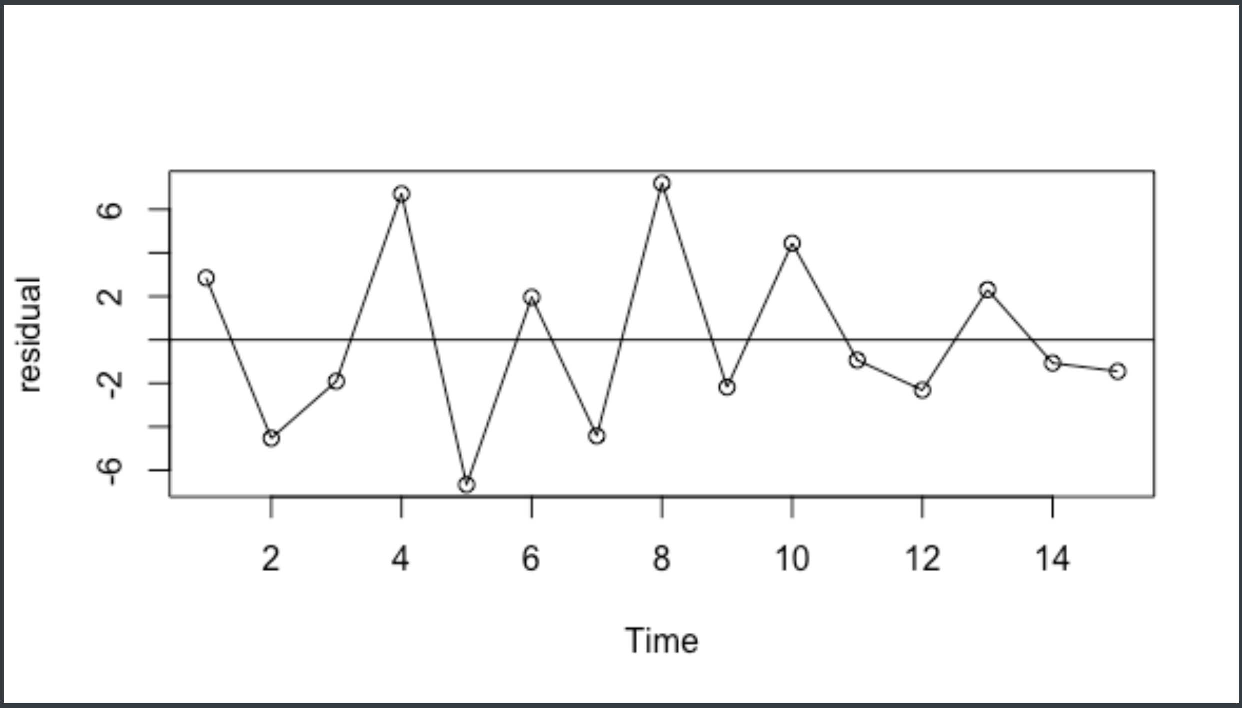
$$= \frac{4n+2}{n(n-1)} = \frac{(2n+1)2}{n(n-1)} \rightarrow Var(\hat{\beta}_0) = \frac{2(2n+1)}{n(n-1)} \sigma_\varepsilon^2$$

$$Cov(\hat{\beta}_0, \hat{\beta}_1) = -\frac{\frac{n+1}{2}}{\frac{n(n^2-1)}{12}} = -\frac{(n+1)\cdot 6}{n(n^2-1)} = \frac{-6}{n(n-1)} \cdot \sigma_\varepsilon^2$$

$$Var(\hat{\beta}_1) = \frac{1}{\Sigma(x_t - \bar{x})^2} \cdot \sigma_\varepsilon^2 = \frac{12}{n(n^2-1)} \cdot \sigma_\varepsilon^2$$

C

모델의 잔차는 다음과 같다. 선형 추세 혹은 주기성이 확인되지 않아 잘 적합되었다고 판단된다.



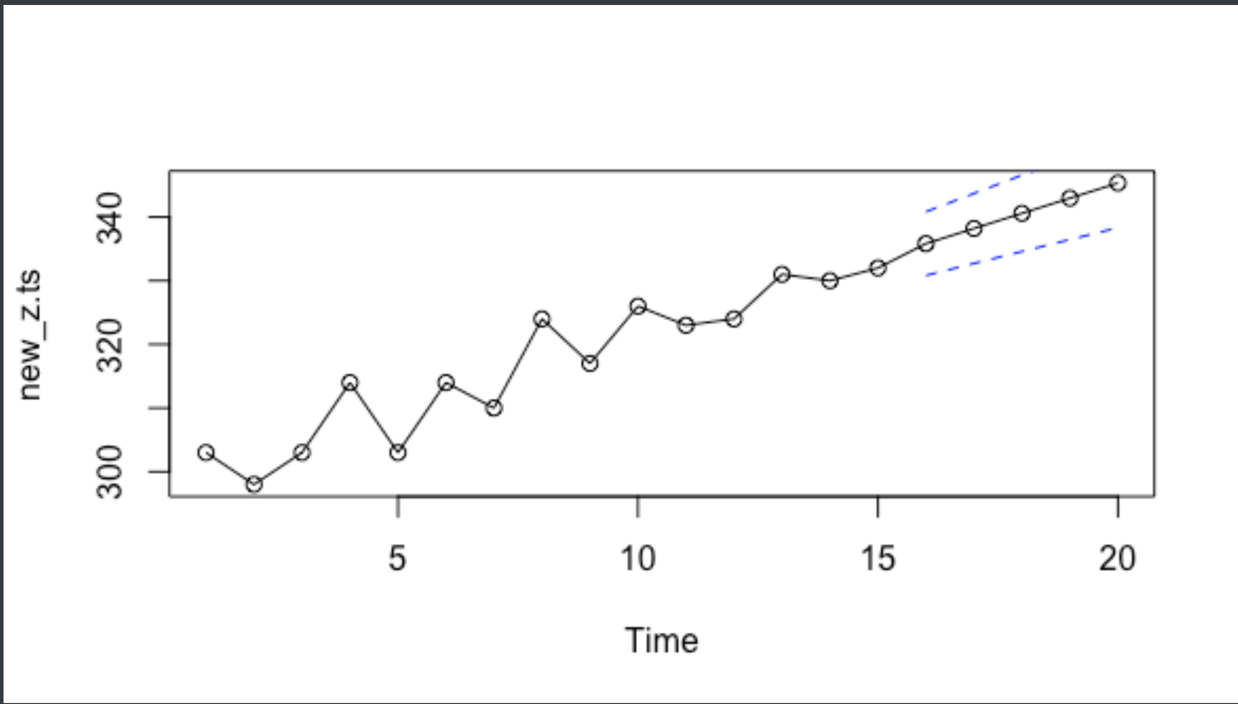
D

미래 5 시점에 대한 예측값은 다음과 같다.

```
[1] 335.8286 338.2071 340.5857 342.9643 345.3429
```

E

미래 시점에 대한 예측값과 그의 신뢰 구간은 다음과 같다.

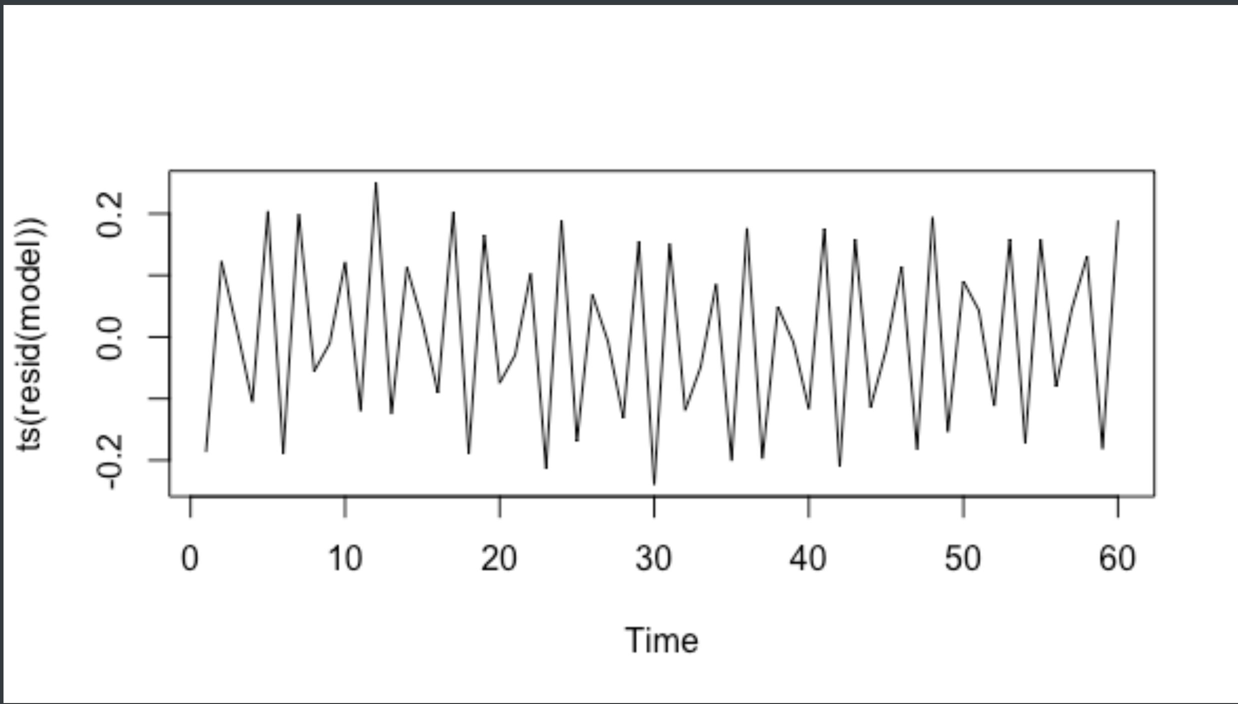


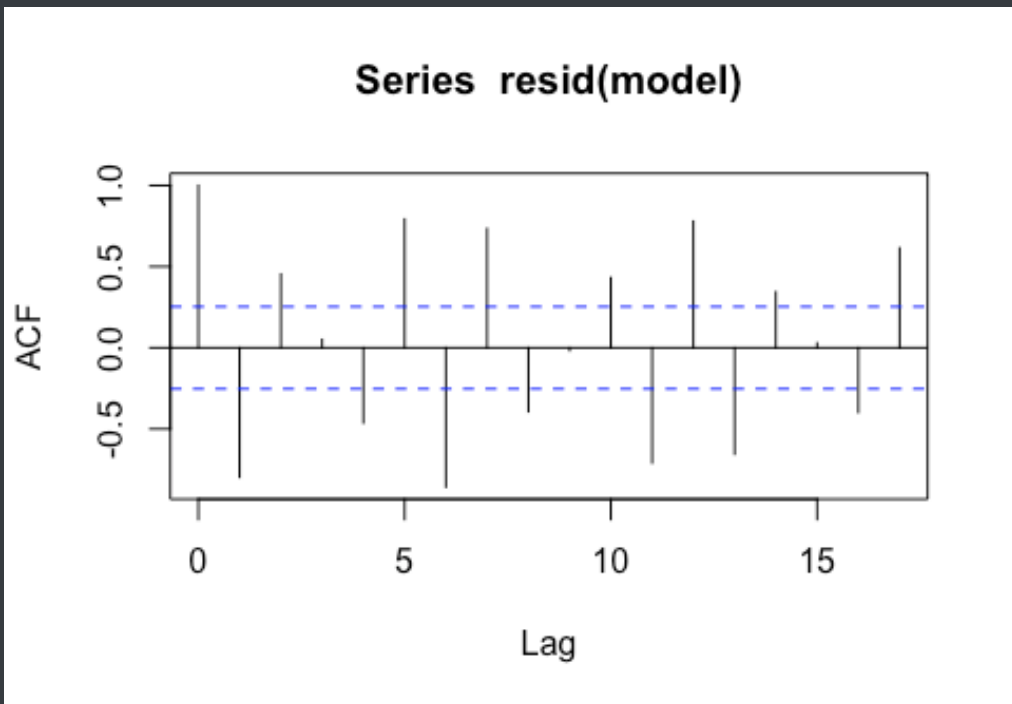
Exercise 2-6

모형 적합을 위해 다음과 같은 삼각 함수를 사용하였다.

```
x1 = sin(2 * pi * t / 12)
x2 = cos(2 * pi * t / 12)
x3 = sin(2 * pi * t / 12*2)
x4 = cos(2 * pi * t / 12*2)
x5 = sin(2 * pi * t / 12*3)
x6 = cos(2 * pi * t / 12*3)
x7 = sin(2 * pi * t / 12*4)
x8 = cos(2 * pi * t / 12*4)
x9 = sin(2 * pi * t / 12*6)
x10 = cos(2 * pi * t / 12*6)
```

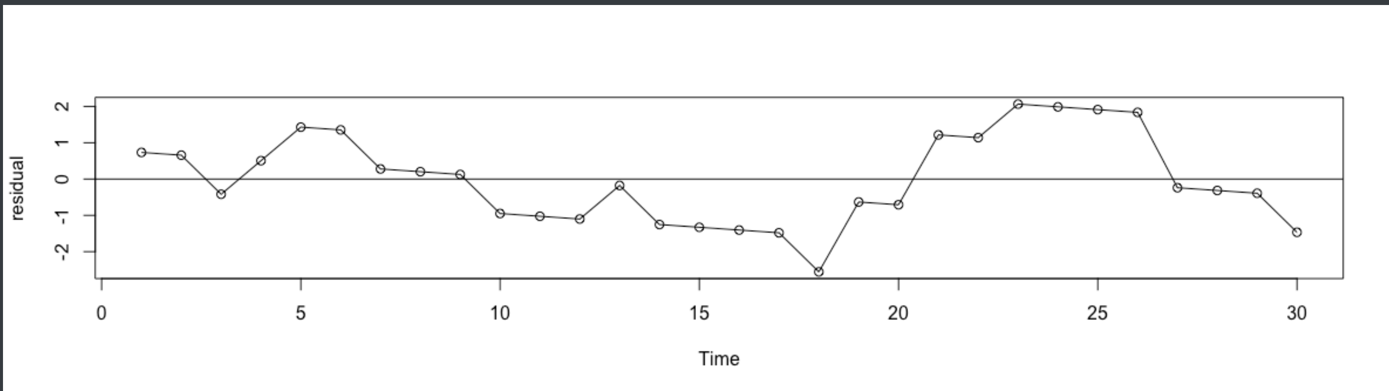
잔차 분석 결과는 다음과 같다. 잔차의 선형 추세가 확인되지 않았다. 그러나 acf 함수 등을 확인해보았을 때, 여전히 잔차에 주기성이 있는 것으로 확인된다. 따라서 다른 주기 함수들이 필요할 것으로 추정된다. 교재에서 indicator 함수를 이용하여 각 월별로 총 12개의 주기 함수를 넣었는데, 이것이 문제가 된 듯 하다.



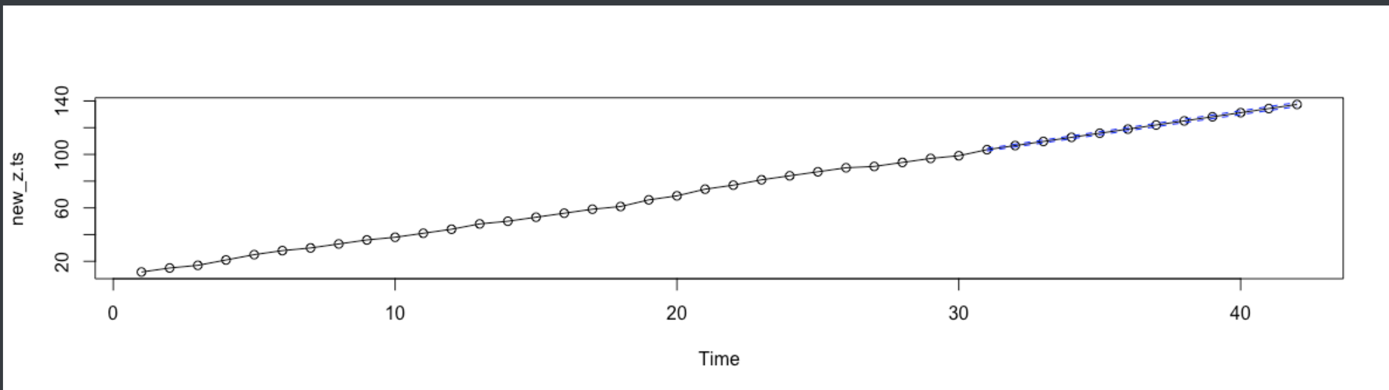


Exercise 2-9

모형 적합 후 잔차는 다음과 같다. 또한 잔차는 선형 추세 혹은 주기성은 발견되지 않았다.



이때 추정 결과와 그의 신뢰 구간은 다음과 같다.



Appendix

R code

```
rm(list = ls())
setwd("Workspace/2022-Fall_TimeSeriesAnalysis/")

## Example 2.1
library(lmtest)

z = scan("data/population.txt")
pop = round(z / 10000)
pop = ts(pop, start=c(1960))
```

```r
t = 1:length(pop)
t2 = t * t
m1 = lm(pop ~ t)
dwtest(m1)
summary(m1)

ts.plot(pop, fitted(m1), xlab="year", ylab="population", lty=1:2,
        main="Figure 2-3 The total population and the predicted values")
acf(resid(m1), main="Residuals of ACF")

m2 = lm(pop ~ t + t2)
summary(m2)

ts.plot(pop, fitted(m2), xlab="year", ylab="population", lty=1:2,
        main="Figure 2-5 A predcited values of second order trend
model")
ts.plot(resid(m2), type="o", ylab="residual",
        main="Figure 2-6 Residuals of second order trend model")
abline(h=0)
legend("topleft", legend=c("pop", "fitted"), lty=1:2)
acf(resid(m2), main="Residuals of ACF")

lnpop = log(pop)
m3 = lm(lnpop ~ t + t2)
dwtest(m3)
summary(m3)

ts.plot(lnpop, fitted(m3), xlab="year", ylab="log population", lty=1:2,
        main="A predcited values of second order trend model with log
transformation")
legend("topleft", legend=c("lnpop", "fitted"), lty=1:2)
ts.plot(resid(m3), ylab="residual",
        main="Figure 2-7 Residuals of second order trend model with log
transformation")
abline(h=0)
acf(resid(m3), main="Residual of ACF")
```

```r
## Example 2.2
library(astsa)
z = scan("data/depart.txt")
dep = ts(z, frequency=12, start=c(1984, 1))
ts.plot(dep, ylab="depart",
        main="Figure 2-9 Sales revenue of department store")

lndep = log(dep)
ts.plot(lndep, ylab="log depart",
        main="Figure 2-10 Log transformed sales revenue of department store")
trend = time(lndep) - 1984
y = factor(cycle(lndep))
reg = lm(lndep ~ 0 + trend + y)
dwtest(reg)
summary(reg)
model.matrix(reg)
resid = ts(resid(reg), start=c(1984, 1), frequency=12)
ts.plot(resid(reg), ylab="residual",
        main="Figure 2-11 Residuals")
abline(h=0)
acf2(resid(reg), main="Residuals of ACF & PCAF")

## Example 2.3
z = scan("data/catv.txt")
k = 70000000
t = 1:length(z)
year = t + 1969
catv = ts(z, start=c(1970))
lncatv = log(k / catv - 1)

ts.plot(catv, xlab="year", ylab="catv",
        main="Figure 2-13 Number of subscribers of Cable TV")
ts.plot(lncatv, xlab="year", ylab="catv",
```

```
        main="Figure 2-14 Number of subscribers (log transformed) of
Cable TV")
fit = lm(lncatv ~ t)
summary(fit)
pred = k / (exp(fitted(fit)) + 1)
resid = catv - pred
y = data.frame(catv, pred)
fig = ts(y, start=c(1970))
ts.plot(fig, xlab="year", ylab="catv", lty=1:2,
        main="Figure 2-15 Predicted value and observed value")
legend("right", legend =c("Catv", "Predict"), lty=1:2)
ts.plot(resid, xlab="time", ylab="residual",
        main="Figure 2-16 Residuals")
abline(h=0)
acf2(resid, main="Residuals of ACF & PCAF")

## Example 2.4
dept = scan("data/depart.txt")
n = 1:length(dept)
time = ts(n, frequency=12, start=c(1984, 1))
dept.ts = ts(dept, frequency=12, start=c(1984, 1))
lndept = log(dept.ts)
y = factor(cycle(time))
fit = lm(lndept ~ 0 + time + y)
anova(fit)
summary(fit)

resid = ts(resid(fit), start=c(1984, 1), frequency=12)
acf2(resid, main="Residuals of ACF & PACF")
autoreg = arima(residuals(fit), order=c(3, 0, 0))
summary(autoreg)
plot(resid(autoreg), main="Figure 2-17 Residuals after fitted
autoregresive model")
abline(h=0)

## Figure 2.8
```

```r
n = 100
t = 1:n
a1 = -0.8
a2 = 1.4
phi1 = pi / 8
phi2 = 3 * pi/4
first = a1 * sin(pi * t / 6 + phi1)
second = a2 * sin(pi * t / 3 + phi2)
z = first + second
plot(z, type="o", lty=1, xlab="time", ylab="z",
     main="Figure 2-8 Time series with two cyclic components")
lines(first, lty=2, col="blue")
lines(second, lty=3, col="red")
legend("left", legend=c("series", "first", "second"), lty=1:3)

fig = data.frame(z, first, second)
ts.plot(fig, col=c("black", "blue", "red"), lty=1:3,
        xlab="time", ylab="z",
        main="Figure 2-8 Time series with two cyclic components")
legend("right", legend=c("series", "first", "second"))

## Figure 2.12
# S-curve
b0 = 0.2
b1 = -12
t = 1:60
z1 = exp(b0 + b1 / t)

# gompertz
b0 = 10
b1 = 0.15
k = 1
z2 = k * exp(-b0 * exp(-b1 * t))

# von
b0 = 0.95
```

```
b1 = 0.09
z3 = (1 - b0 * exp((-b1) * t)^3)

# Pear 1
b0 = 5
b1 = -0.2
k = 1
z4 = k / (1 + exp(b0 + b1 * t))

z = data.frame(z1, z2, z3, z4)
z.ts = ts(z)
ts.plot(z.ts, lty=1:4, main="Figure 2-12 S-Cruves")
legend("right", legend=c("Exp", "Gompertz", "Von", "Pearl"), lty=1:4)

## Exersize 2.2
z = c(303, 298, 303, 314, 303, 314, 310, 324, 317, 326, 323, 324,331,
330, 332)
t = 1:length(z)
z.ts = ts(z)
model = lm(z ~ t)
ts.plot(resid(model), type="o", ylab="residual")
abline(h=0)

acf(resid(model))

new_data = data.frame(t=c(16,17,18,19,20))
pred = data.frame(predict(model, newdata=new_data,
interval='confidence'))

new_z = c(z, pred$fit)
new_z.ts = ts(new_z)
ts.plot(new_z.ts, type="o")
lines(new_data$t, pred$lwr, lty=2, col="blue")
lines(new_data$t, pred$upr, lty=2, col="blue")

## Exercise 2.6
```

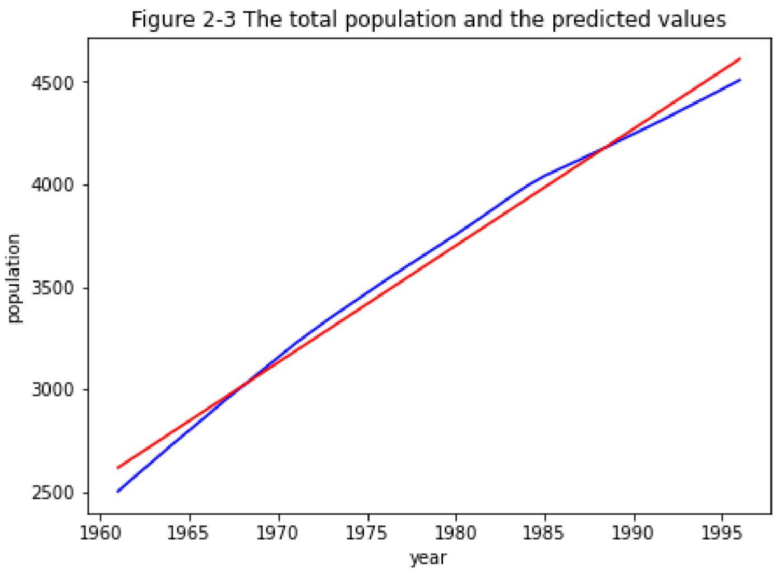
```r
dept = scan("data/depart.txt")
t = 1:length(dept)
dept.ts = ts(dept, frequency=12, start=c(1984, 1))
lndept = log(dept.ts)

ts.plot(dept.ts)

x1 = sin(2 * pi * t / 12)
x2 = cos(2 * pi * t / 12)
x3 = sin(2 * pi * t / 12*2)
x4 = cos(2 * pi * t / 12*2)
x5 = sin(2 * pi * t / 12*3)
x6 = cos(2 * pi * t / 12*3)
x7 = sin(2 * pi * t / 12*4)
x8 = cos(2 * pi * t / 12*4)
x9 = sin(2 * pi * t / 12*6)
x10 = cos(2 * pi * t / 12*6)

model = lm(lndept ~ t + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 +
x10)
summary(model)

ts.plot(ts(resid(model)))
acf(resid(model))

## Exercise 2.9
book = scan("data/book.txt")
t = 1:length(z)

model = lm(book ~ t)
summary(model)

ts.plot(resid(model), type="o", ylab="residual")
abline(h=0)

new_data = data.frame(t=c(31:42))
```

```
pred = data.frame(predict(model, newdata=new_data,
interval='confidence'))

new_z = c(z, pred$fit)
new_z.ts = ts(new_z)
ts.plot(new_z.ts, type="o")
lines(new_data$t, pred$lwr, lty=2, col="blue")
lines(new_data$t, pred$upr, lty=2, col="blue")
```

Python

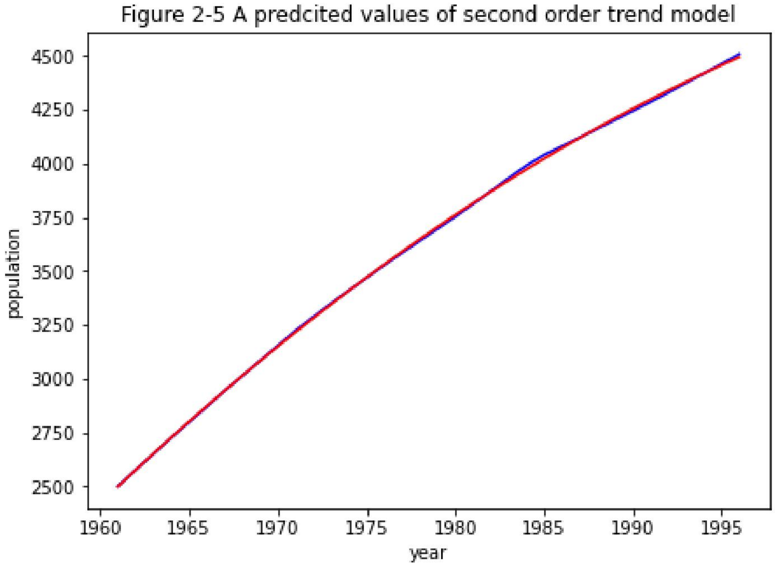
In [1]:
```python
import math
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.dates as mdates

from sklearn import linear_model
```

In [2]:
```python
# Example 2.1
z = []

with open('../data/population.txt') as f:
    for line in f.readlines():
        for elem in line.rstrip().split(" "):
            if len(elem):
                z.append(float(elem))

pop = np.array(z)
pop = np.round(pop / 10000)
ln_pop = np.log(z)
t = np.array(range(len(z)))
t2 = t * t

m1 = linear_model.LinearRegression()
m1.fit(t.reshape(-1, 1), pop)

z_ts = pd.DataFrame(np.vstack([pop, m1.intercept_ + m1.coef_*t]).T,
    index=pd.date_range("1960-01-01", periods=len(z), freq="y"),
    columns=["Zt", "Xt"]
)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(z_ts['Zt'], 'b')
ax.plot(z_ts['Xt'], 'r')
ax.xaxis.set_major_formatter(mdates.DateFormatter('%Y'))
ax.xaxis.set_major_locator(mdates.YearLocator(5))
ax.set_xlabel("year")
ax.set_ylabel("population")
ax.set_title("Figure 2-3 The total population and the predicted values")
plt.show()
```

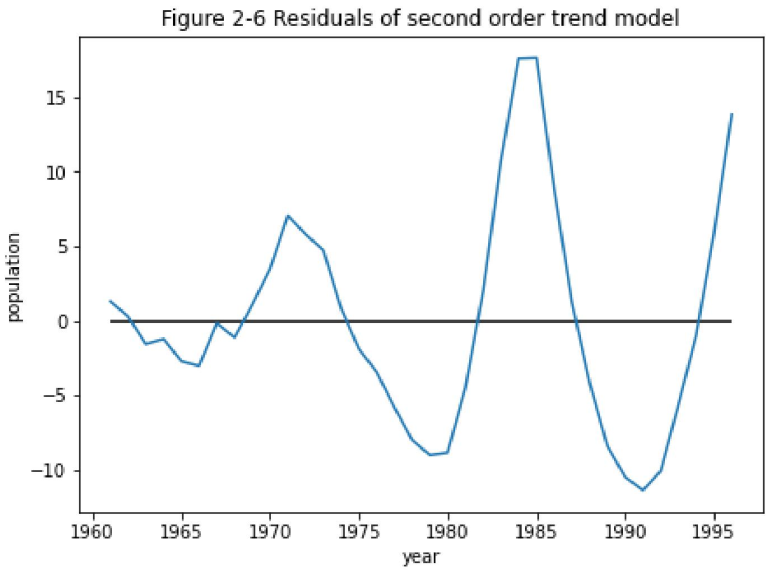


```
In [3]: X = np.vstack([t, t2]).T
m2 = linear_model.LinearRegression()
m2.fit(X, pop)

z_ts = pd.DataFrame(np.hstack([pop.reshape(-1, 1), (m2.intercept_ + np.sum(X * m2
    index=pd.date_range("1960-01-01", periods=len(z), freq="y"),
    columns=["Zt", "Xt"]
)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(z_ts['Zt'], 'b')
ax.plot(z_ts['Xt'], 'r')
ax.xaxis.set_major_formatter(mdates.DateFormatter('%Y'))
ax.xaxis.set_major_locator(mdates.YearLocator(5))
ax.set_xlabel("year")
ax.set_ylabel("population")
ax.set_title("Figure 2-5 A predcited values of second order trend model")
plt.show()
```

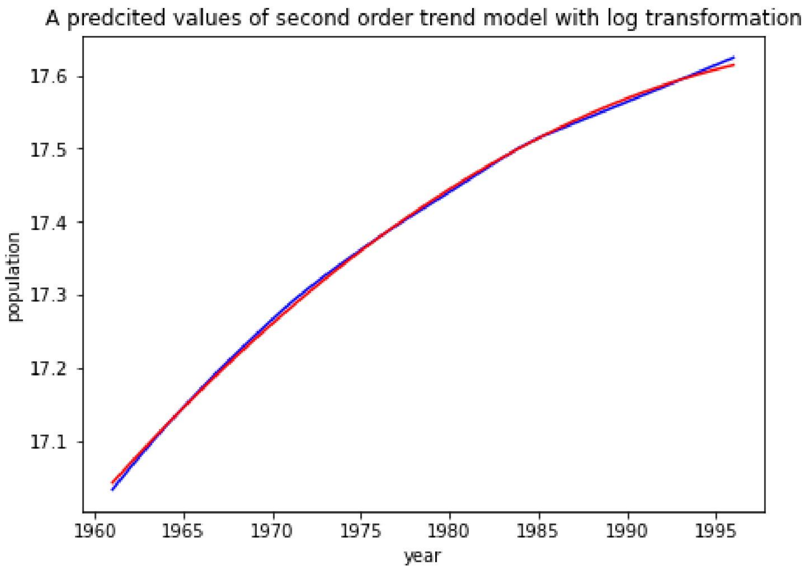
Figure 2-5 A predcited values of second order trend model



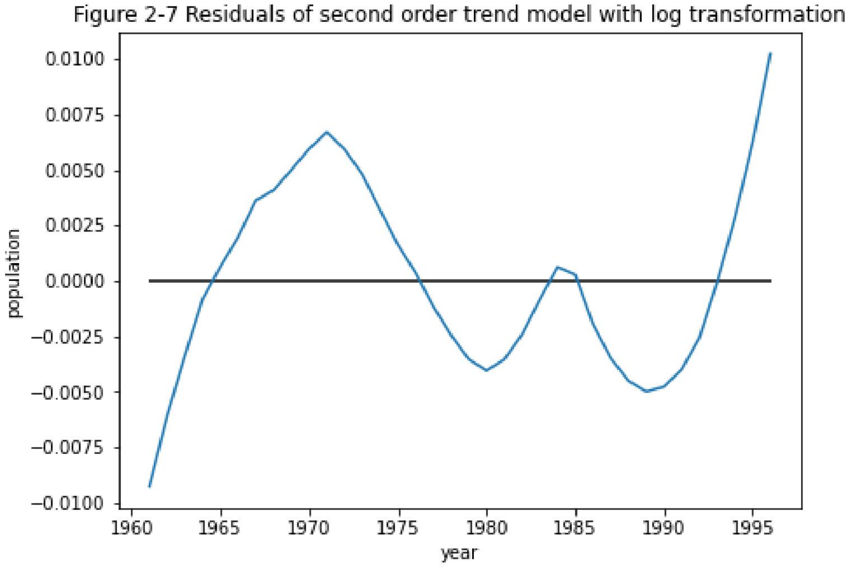
In [4]:
```
fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(z_ts['Zt'] - z_ts["Xt"])
ax.hlines(0, min(z_ts.index), max(z_ts.index), color="black")
ax.xaxis.set_major_formatter(mdates.DateFormatter('%Y'))
ax.xaxis.set_major_locator(mdates.YearLocator(5))
ax.set_xlabel("year")
ax.set_ylabel("population")
ax.set_title("Figure 2-6 Residuals of second order trend model")
plt.show()
```



In [5]:
```
X = np.vstack([t, t2]).T
m2 = linear_model.LinearRegression()
m2.fit(X, ln_pop)

z_ts = pd.DataFrame(np.hstack([ln_pop.reshape(-1, 1), (m2.intercept_ + np.sum(X *
    index=pd.date_range("1960-01-01", periods=len(z), freq="y"),
    columns=["Zt", "Xt"]
)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(z_ts['Zt'], 'b')
ax.plot(z_ts['Xt'], 'r')
ax.xaxis.set_major_formatter(mdates.DateFormatter('%Y'))
ax.xaxis.set_major_locator(mdates.YearLocator(5))
ax.set_xlabel("year")
ax.set_ylabel("population")
ax.set_title("A predcited values of second order trend model with log transformat
plt.show()
```

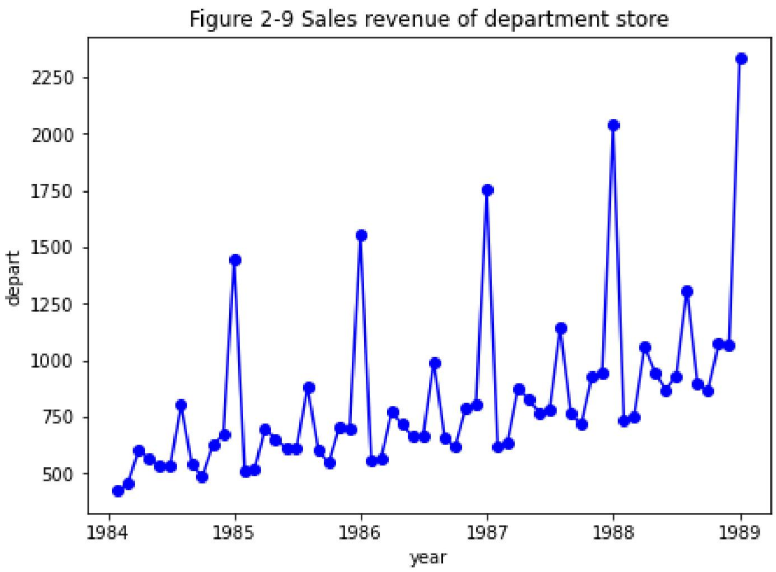


In [6]:
```
fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(z_ts['Zt'] - z_ts["Xt"])
ax.hlines(0, min(z_ts.index), max(z_ts.index), color="black")
ax.xaxis.set_major_formatter(mdates.DateFormatter('%Y'))
ax.xaxis.set_major_locator(mdates.YearLocator(5))
ax.set_xlabel("year")
ax.set_ylabel("population")
ax.set_title("Figure 2-7 Residuals of second order trend model with log transform
plt.show()
```



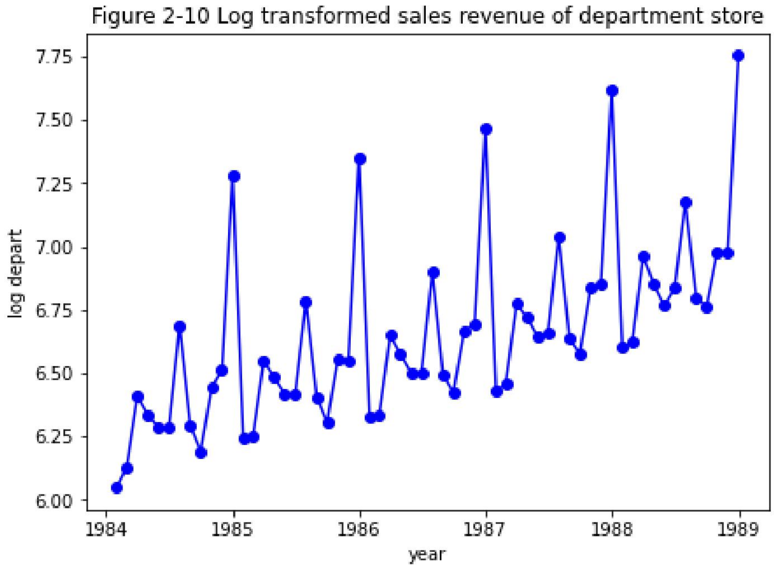
In [7]:

```python
# Example 2.2
z = []

with open('../data/depart.txt') as f:
    for line in f.readlines():
        for elem in line.rstrip().split(" "):
            if len(elem):
                z.append(float(elem))

z_ts = pd.DataFrame(z,
    index=pd.date_range("1984-01-01", periods=len(z), freq="m"),
    columns=["Zt"]
)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(z_ts['Zt'], 'o-b')
ax.set_xlabel("year")
ax.set_ylabel("depart")
ax.set_title("Figure 2-9 Sales revenue of department store")
plt.show()
```

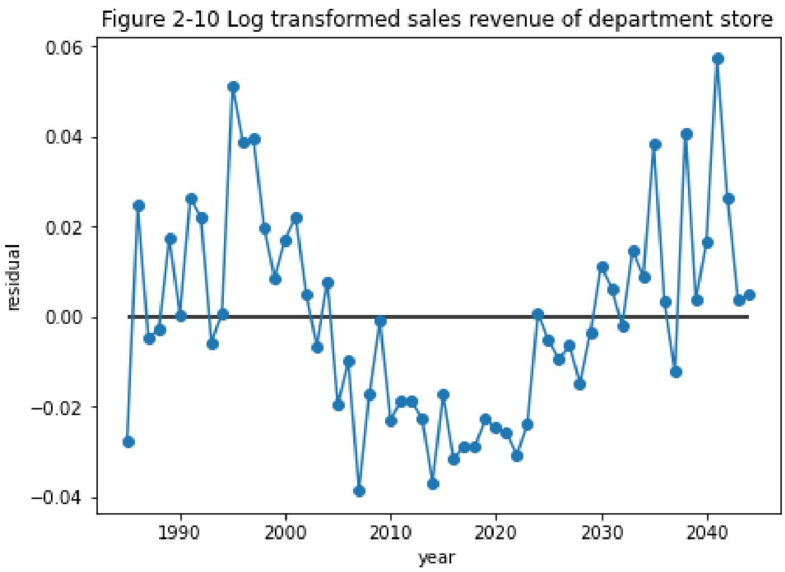


In [8]:

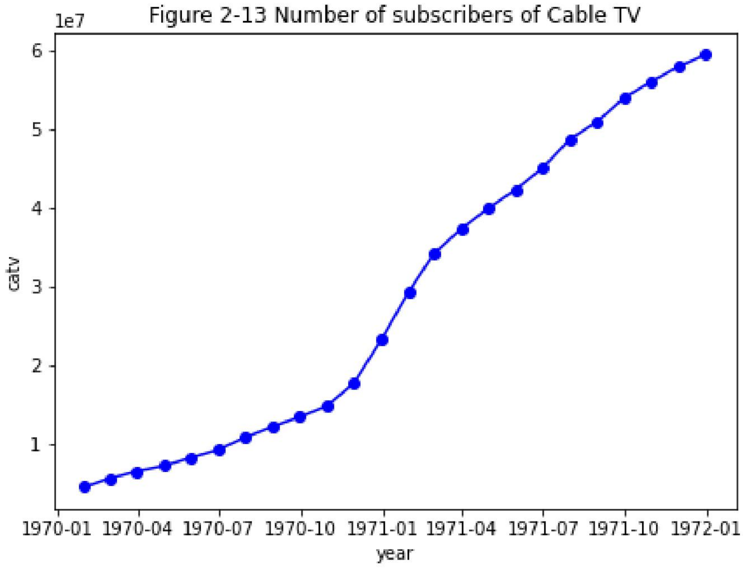
```python
ln_dep = np.log(z)

z_ts = pd.DataFrame(ln_dep,
    index=pd.date_range("1984-01-01", periods=len(z), freq="m"),
    columns=["Zt"]
)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(z_ts['Zt'], 'o-b')
ax.set_xlabel("year")
ax.set_ylabel("log depart")
ax.set_title("Figure 2-10 Log transformed sales revenue of department store")
plt.show()
```



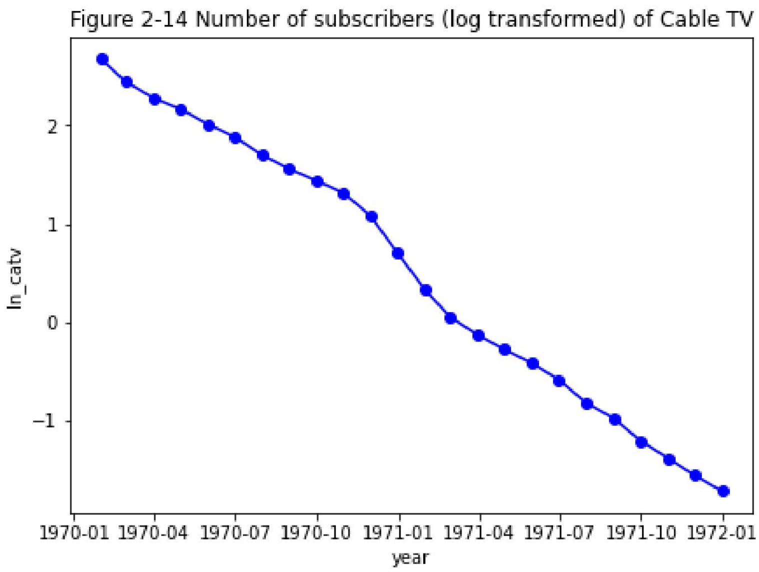
```
In [9]: trend = np.array(list(range(len(z_ts)))).reshape(-1, 1)
        y = z_ts.index.month.values
        y = pd.get_dummies(y).values
        X = np.hstack([trend, y])

        reg = linear_model.LinearRegression(fit_intercept=False)
        reg.fit(X, ln_dep)

        z_ts = pd.DataFrame(np.hstack([ln_dep.reshape(-1, 1), (np.sum(X * reg.coef_, axis
            index=pd.date_range("1984-01-01", periods=len(z), freq="y"),
            columns=["Zt", "Xt"]
        )

        fig, ax = plt.subplots(figsize=(7, 5))
        ax.plot(z_ts['Zt'] - z_ts['Xt'], 'o-')
        ax.set_xlabel("year")
        ax.set_ylabel("residual")
        ax.hlines(0, min(z_ts.index), max(z_ts.index), color="black")
        ax.set_title("Figure 2-10 Log transformed sales revenue of department store")
        plt.show()
```



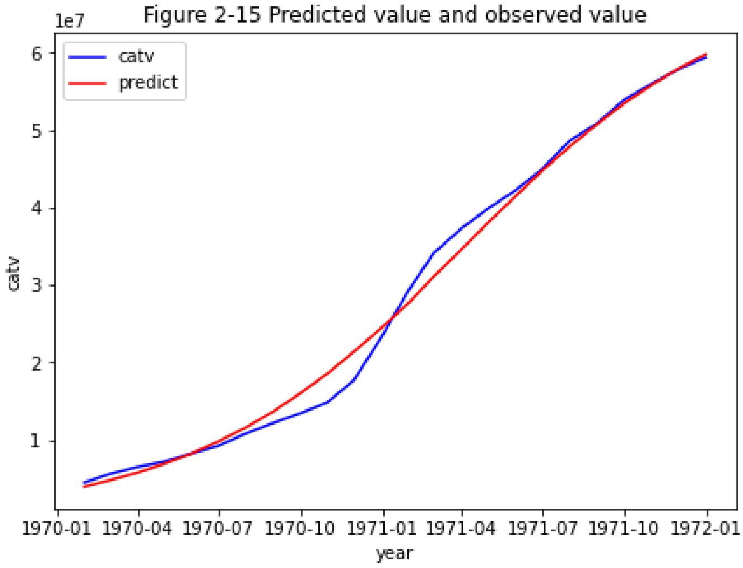
```
In [10]: # Example 2.3
z = []

with open('../data/catv.txt') as f:
    for line in f.readlines():
        for elem in line.rstrip().split(" "):
            if len(elem):
                z.append(float(elem))

k = 70000000
t = np.array(range(len(z)))

catv = np.array(z)
ln_catv = np.log(k / catv - 1)

z_ts = pd.DataFrame(catv,
    index=pd.date_range("1970-01-01", periods=len(z), freq="m"),
    columns=["Zt"]
)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(z_ts['Zt'], 'o-b')
ax.set_xlabel("year")
ax.set_ylabel("catv")
ax.set_title("Figure 2-13 Number of subscribers of Cable TV")
plt.show()
```



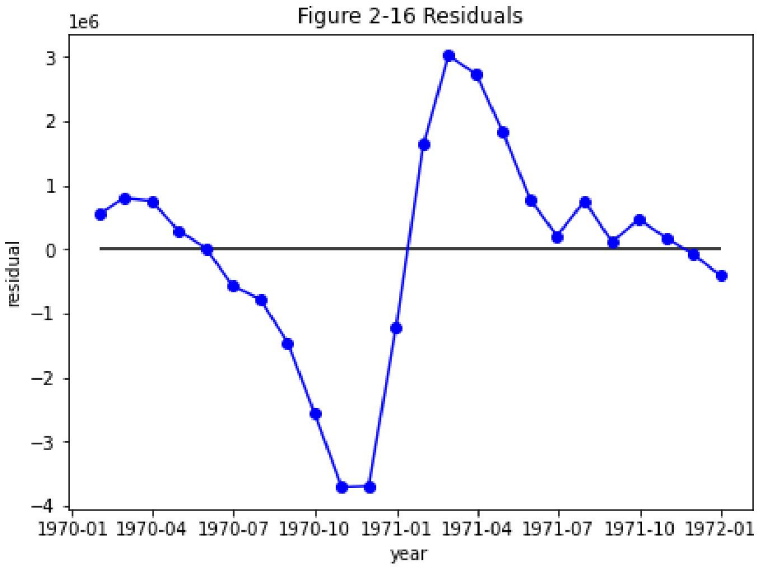
In [11]:
```
z_ts = pd.DataFrame(ln_catv,
    index=pd.date_range("1970-01-01", periods=len(z), freq="m"),
    columns=["Zt"]
)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(z_ts['Zt'], 'o-b')
ax.set_xlabel("year")
ax.set_ylabel("ln_catv")
ax.set_title("Figure 2-14 Number of subscribers (log transformed) of Cable TV")
plt.show()
```



In [12]:
```
lm = linear_model.LinearRegression()
lm.fit(t.reshape(-1, 1), ln_catv)
pred = k / (np.exp(np.sum(lm.intercept_ + t.reshape(-1, 1) * lm.coef_, axis=1)) +

z_ts = pd.DataFrame(np.hstack([catv.reshape(-1, 1), pred.reshape(-1, 1)]),
    index=pd.date_range("1970-01-01", periods=len(z), freq="m"),
    columns=["Zt", "Xt"]
)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(z_ts['Zt'], 'b', label="catv")
ax.plot(z_ts['Xt'], 'r', label="predict")
ax.set_xlabel("year")
ax.set_ylabel("catv")
ax.set_title("Figure 2-15 Predicted value and observed value")
plt.legend()
plt.show()
```



In [13]:
```python
resid = catv - pred

z_ts = pd.DataFrame(resid,
    index=pd.date_range("1970-01-01", periods=len(z), freq="m"),
    columns=["Zt"]
)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(z_ts['Zt'], 'o-b')
ax.set_xlabel("year")
ax.set_ylabel("residual")
ax.set_title("Figure 2-16 Residuals")
ax.hlines(0, min(z_ts.index), max(z_ts.index), "black")
plt.show()
```



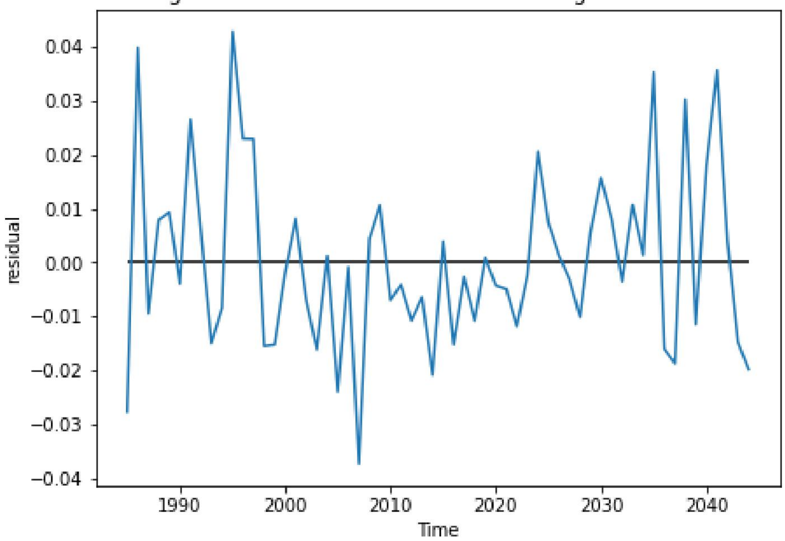
In [14]:

```python
from statsmodels.tsa.arima_model import ARIMA

z = []

with open('../data/depart.txt') as f:
    for line in f.readlines():
        for elem in line.rstrip().split(" "):
            if len(elem):
                z.append(float(elem))

ln_dep = np.log(z)

z_ts = pd.DataFrame(ln_dep,
    index=pd.date_range("1970-01-01", periods=len(z), freq="m"),
    columns=["Zt"]
)

trend = np.array(range(len(z_ts))).reshape(-1, 1)
y = z_ts.index.month.values
y = pd.get_dummies(y).values
X = np.hstack([trend, y])

reg = linear_model.LinearRegression(fit_intercept=False)
reg.fit(X, ln_dep)

z_ts = pd.DataFrame(np.hstack([ln_dep.reshape(-1, 1), (np.sum(X * reg.coef_, axis
    index=pd.date_range("1984-01-01", periods=len(z), freq="y"),
    columns=["Zt", "Xt"]
)

resid = z_ts["Zt"] - z_ts["Xt"]

model = ARIMA(resid, order=(3,0,0))
model_fit = model.fit(trend='c', full_output=True, disp=0)
final_resid = model_fit.resid

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(final_resid, '-')
ax.set_xlabel("Time")
ax.set_ylabel("residual")
ax.set_title("Figure 2-17 Residuals after fitted autoregresive model")
ax.hlines(0, min(z_ts.index), max(z_ts.index), "black")
plt.show()
```

```
/Users/jonghyun/miniforge3/lib/python3.9/site-packages/statsmodels/tsa/arima_mo
del.py:472: FutureWarning:
statsmodels.tsa.arima_model.ARMA and statsmodels.tsa.arima_model.ARIMA have
been deprecated in favor of statsmodels.tsa.arima.model.ARIMA (note the .
between arima and model) and
statsmodels.tsa.SARIMAX. These will be removed after the 0.12 release.

statsmodels.tsa.arima.model.ARIMA makes use of the statespace framework and
is both well tested and maintained.

To silence this warning and continue using ARMA and ARIMA until they are
removed, use:

import warnings
warnings.filterwarnings('ignore', 'statsmodels.tsa.arima_model.ARMA',
                        FutureWarning)
warnings.filterwarnings('ignore', 'statsmodels.tsa.arima_model.ARIMA',
                        FutureWarning)
```

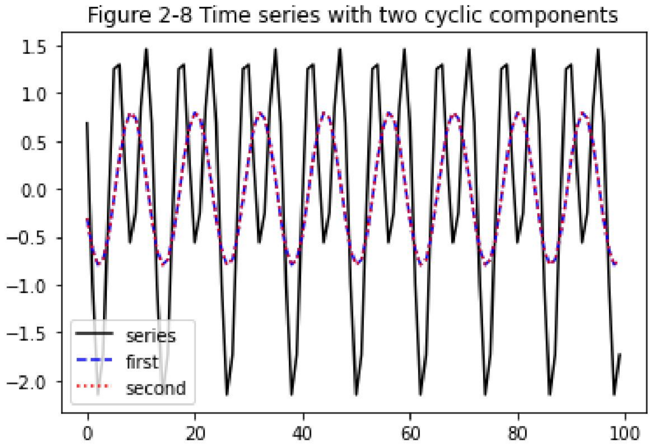
```
warnings.warn(ARIMA_DEPRECATION_WARN, FutureWarning)
```

Figure 2-17 Residuals after fitted autoregresive model



In [15]:

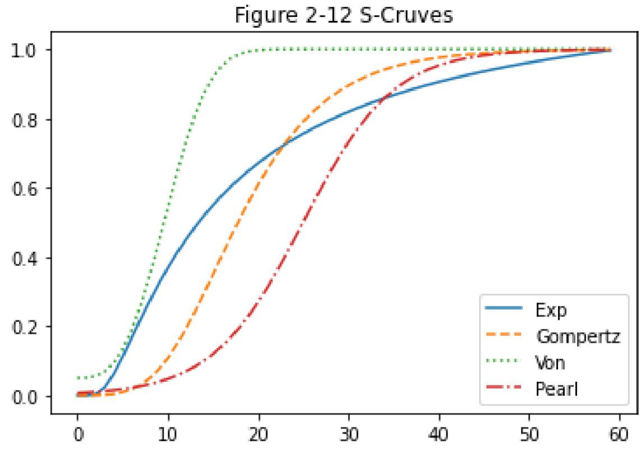
```python
# Figure 2.8 주기 성분을 갖는 시계열

import math

n = 100
t = np.arange(n)
a1 = -0.8
a2 = 1.4
phi_1 = math.pi / 8
phi_2 = 3 * math.pi / 4

first = a1 * np.sin(math.pi * t / 6 + phi_1)
second = a2 * np.sin(math.pi * t / 3 + phi_2)

z = first + second

plt.plot(t, z, color="black", linestyle="-", label="series")
plt.plot(t, first, color="blue", linestyle="--", label="first")
plt.plot(t, first, color="red", linestyle=":", label="second")
plt.title("Figure 2-8 Time series with two cyclic components")
plt.legend()
plt.show()
```



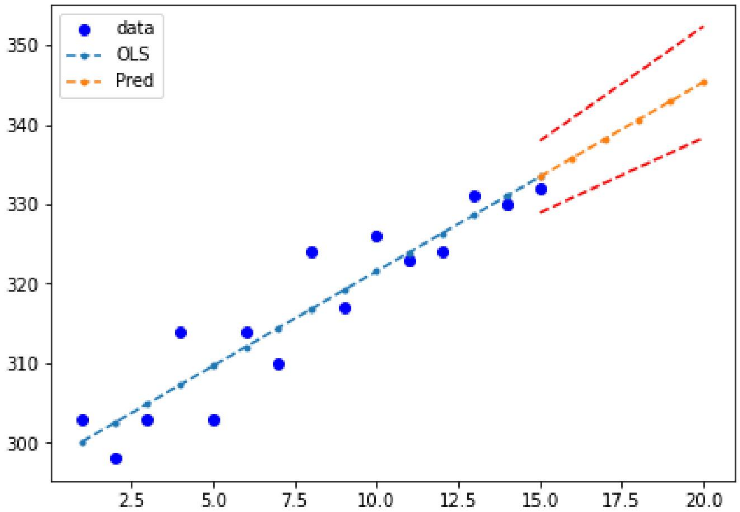
In [16]:

```
# Figure 2.12

b0 = 0.2
b1 = -12
t = np.arange(60)
z1 = np.exp(b0 + b1 / t)

# Gompertz
b0 = 10
b1 = 0.15
k = 1
z2 = k * np.exp(-b0 * np.exp(-b1 * t))

# Von
b0 = 0.95
b1 = 0.09
z3 = 1 - b0 * np.exp((-b1*t)**3)

# Perl
b0 = 5
b1 = -0.2
k = 1
z4 = k / (1 + np.exp(b0 + b1 * t))

plt.plot(t, z1, linestyle="-", label="Exp")
plt.plot(t, z2, linestyle="--", label="Gompertz")
plt.plot(t, z3, linestyle=":", label="Von")
plt.plot(t, z4, linestyle="-.", label="Pearl")
plt.title("Figure 2-12 S-Cruves")
plt.legend()
plt.show()
```

```
/var/folders/nw/x1lqw0rx1mj698bqcgqxhk880000gn/T/ipykernel_94681/512967855.py:
6: RuntimeWarning: divide by zero encountered in true_divide
  z1 = np.exp(b0 + b1 / t)
```

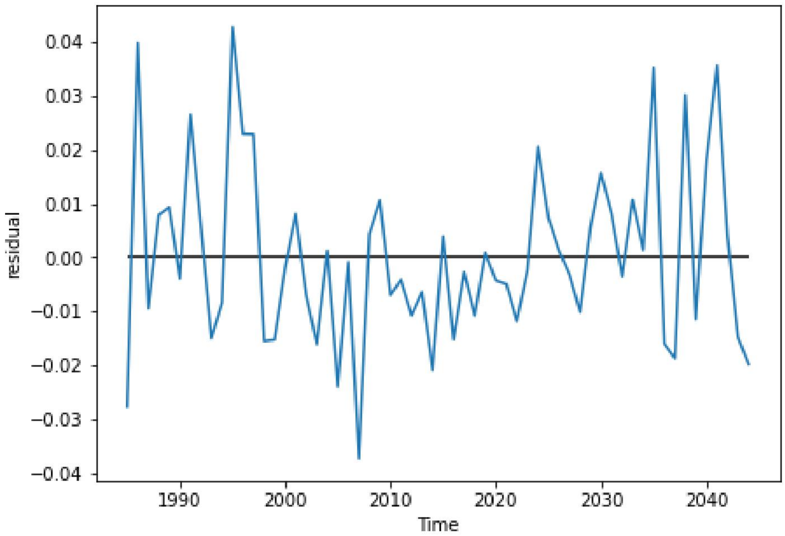


In [17]:
```python
import statsmodels.api as sm

z = np.array([303, 298, 303, 314, 303, 314, 310, 324, 317, 326, 323, 324, 331, 33
t = np.arange(1, len(z) + 1).reshape(-1, 1)
t_ = sm.add_constant(t)
model = sm.OLS(z, t_)
res = model.fit()

X_test = np.array([
    [1, 15],
    [1, 16],
    [1, 17],
    [1, 18],
    [1, 19],
    [1, 20],
])
new_pred = res.get_prediction(X_test).summary_frame(alpha=0.05)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(t, z, "bo", label="data")
ax.plot(t, res.fittedvalues, "--.", label="OLS")
ax.plot(np.arange(15, 21), new_pred["mean"], "--.", label="Pred")
ax.plot(np.arange(15, 21), new_pred["mean_ci_lower"], "r--")
ax.plot(np.arange(15, 21), new_pred["mean_ci_upper"], "r--")
ax.legend(loc="best")
plt.show()
```



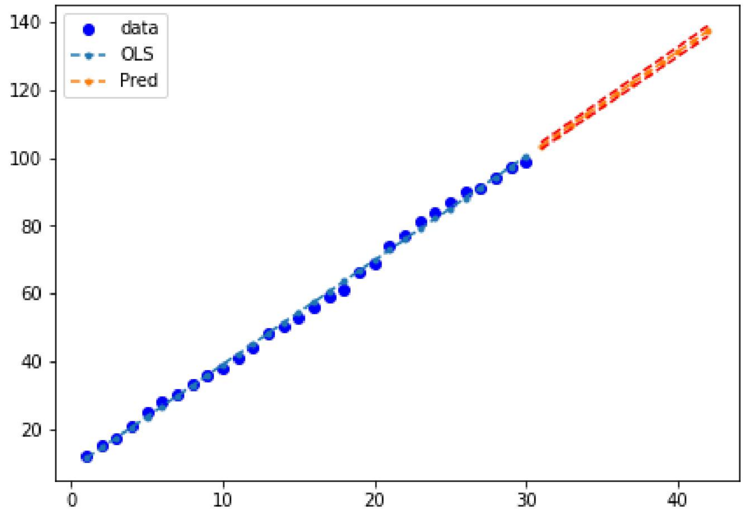
In [18]:

```python
# Exercise 2.6

z = []

with open('../data/depart.txt') as f:
    for line in f.readlines():
        for elem in line.rstrip().split(" "):
            if len(elem):
                z.append(float(elem))

t = np.arange(len(z))
ln_dept = np.log(z)

x1 = np.sin(2 * np.pi * t / 12).reshape(-1, 1)
x2 = np.cos(2 * np.pi * t / 12).reshape(-1, 1)
x3 = np.sin(2 * np.pi * t / 12*2).reshape(-1, 1)
x4 = np.cos(2 * np.pi * t / 12*2).reshape(-1, 1)
x5 = np.sin(2 * np.pi * t / 12*3).reshape(-1, 1)
x6 = np.cos(2 * np.pi * t / 12*3).reshape(-1, 1)
x7 = np.sin(2 * np.pi * t / 12*4).reshape(-1, 1)
x8 = np.cos(2 * np.pi * t / 12*4).reshape(-1, 1)
x9 = np.sin(2 * np.pi * t / 12*6).reshape(-1, 1)
x10 = np.cos(2 * np.pi * t / 12*6).reshape(-1, 1)

X = np.hstack([t.reshape(-1, 1), x1, x2, x3, x4, x5, x6, x7, x8, x9, x10])
model = linear_model.LinearRegression()
model.fit(X, ln_dep)

z_ts = pd.DataFrame(np.hstack([ln_dep.reshape(-1, 1), (model.intercept_ + np.sum(
    index=pd.date_range("1984-01-01", periods=len(z), freq="y"),
    columns=["Zt", "Xt"]
)

resid = z_ts["Zt"] - z_ts["Xt"]

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(final_resid, '-')
ax.set_xlabel("Time")
ax.set_ylabel("residual")
ax.hlines(0, min(z_ts.index), max(z_ts.index), "black")
plt.show()
```



In [ ]:

In [19]:
```python
# Exercise 2.6

z = []

with open('../data/book.txt') as f:
    for line in f.readlines():
        for elem in line.rstrip().split(" "):
            if len(elem):
                z.append(float(elem))

t = np.arange(1, len(z) + 1).reshape(-1, 1)
t_ = sm.add_constant(t)
model = sm.OLS(z, t_)
res = model.fit()

X_test = sm.add_constant(np.arange(31, 43))
new_pred = res.get_prediction(X_test).summary_frame(alpha=0.05)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(t, z, "bo", label="data")
ax.plot(t, res.fittedvalues, "--.", label="OLS")
ax.plot(np.arange(31, 43), new_pred["mean"], "--.", label="Pred")
ax.plot(np.arange(31, 43), new_pred["mean_ci_lower"], "r--")
ax.plot(np.arange(31, 43), new_pred["mean_ci_upper"], "r--")
ax.legend(loc="best")
plt.show()
```



In [ ]: