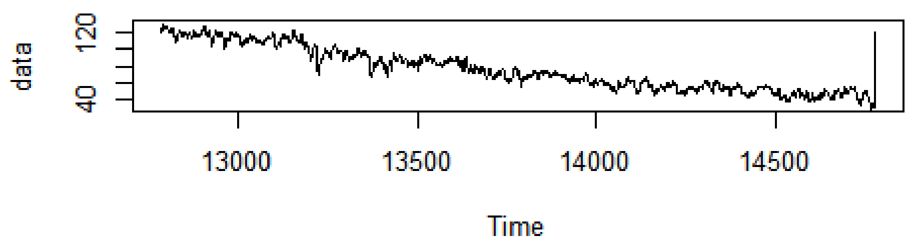
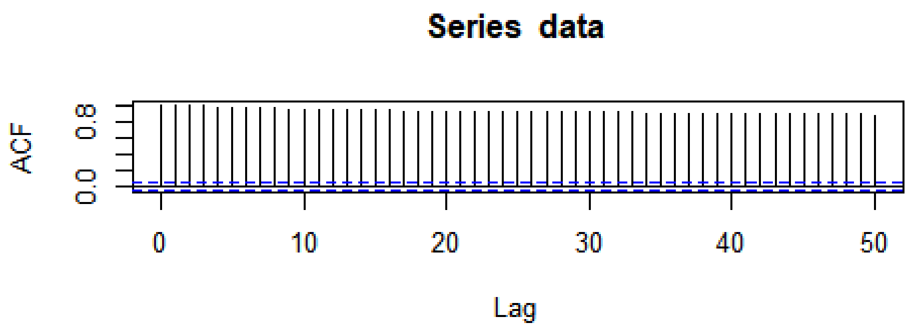
Practice Exam 2

2017311974 진수정

2021 5 25

(a) Time plot, correlograms (ACF) and discuss key features of the data.





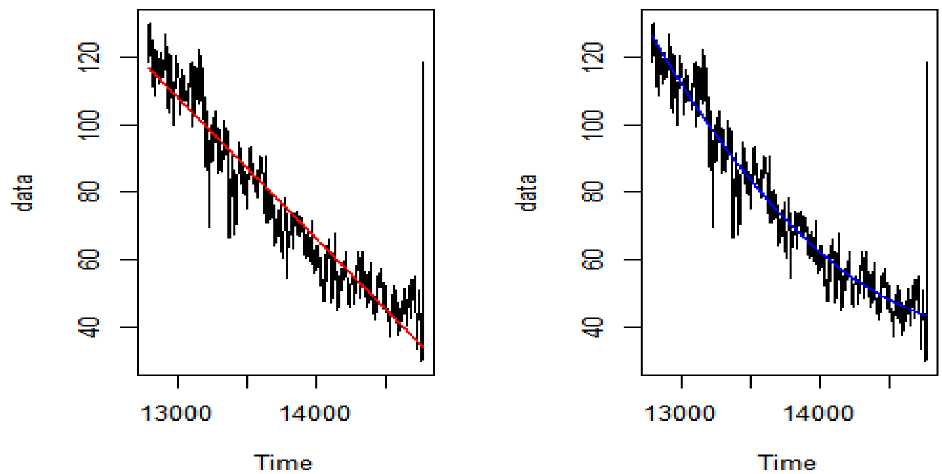
(1) From time plot, we can observe some linear or quadratic decreasing pattern.
(2) From correlograms, SACFs are very slowly decaying.
(3) There are some outliers at the end of the time period.
(4) Variance seems to be constant overt the time period.

(b) Is it stationary? Include your evidence.

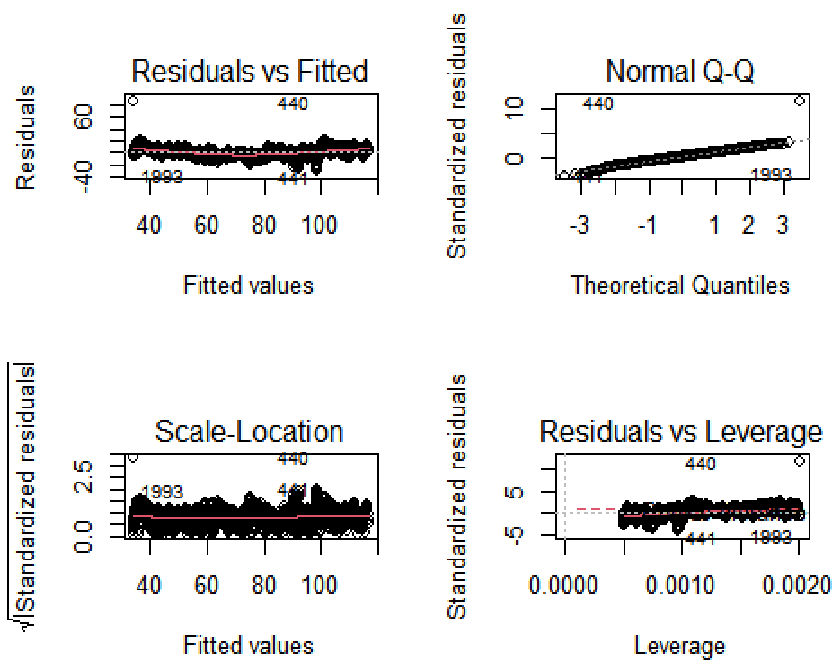
The given data has some deterministic trend. Also, very slowly decaying SACFs can occur because of the existence of trend. Thus, from (1) and (2), the data is not stationary, and we need to detrend.

(c) Find (your) best "regression + stationary errors" model. You need to include reasonings for your selection.

Since there exists some linear or quadratic trend, try using linear model or quadratic model to detrend.

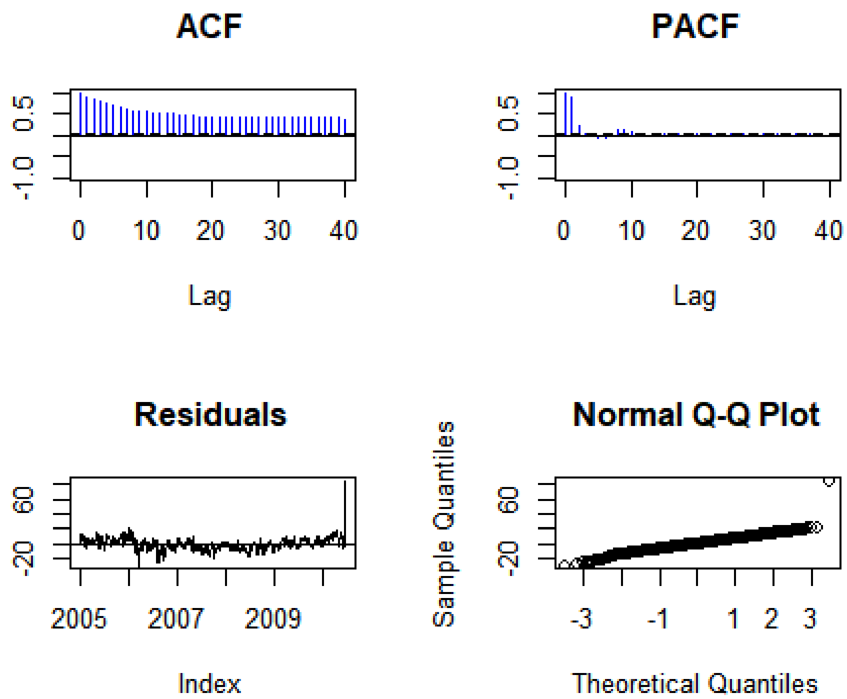


The red line is fitted linear regression line, and the blue line is fitted quadratic line. We can observe both models fit the trend well. Since linear model is enough, proceed with the linear model.



From residual plot and QQplot, normal assumptions such as linearity, constant variance, normality seem to be satisfied. Thus, linear regression model is fine.

```
Null hypothesis: Residuals are iid noise.
Test                        Distribution Statistic   p-value
Ljung-Box Q                Q ~ chisq(20)  34390.58         0 *
McLeod-Li Q                Q ~ chisq(20)   1579.78         0 *
Turning points T(T-1327.3)/18.8 ~ N(0,1)         0         0 *
Diff signs S        (S-996)/12.9 ~ N(0,1)         0         0 *
Rank P        (P-992514)/14834.5 ~ N(0,1)         0         0 *
```



Examining the residuals, trend is disappeared. However, all the five formal tests are rejected. Since the residual is not iid, we need to model the error structure.
From correlograms, we can observe ACFs are decaying and PACF(1), PACF(2), PACF(8), PACF(9) are significant. Thus, sparse AR(9) seems to be fine.

```
Call:
arima(x = data, order = c(9, 0, 0), xreg = xreg, include.mean = F)

Coefficients:
         ar1      ar2     ar3     ar4      ar5
      1.2367  -0.2648  0.0447  0.0003  -0.0341
s.e.  0.0369   0.0560  0.0539  0.0538   0.0539
         ar6      ar7     ar8      ar9     const
      0.0846  -0.5446  0.5423  -0.0998  114.8486
s.e.  0.0539   0.0538  0.0558   0.0369    3.1818
        time
      -0.0383
s.e.   0.0028

sigma^2 estimated as 6.249:  log likelihood = -4656.13,  aic = 9336.26

Training set error measures:
                       ME     RMSE      MAE
Training set -0.004193143 2.499708 1.188822
                    MPE     MAPE      MASE
Training set -0.1061767 1.709914 0.8581809
                     ACF1
Training set 0.0003702706
```

[Linear Regression + AR(9)]

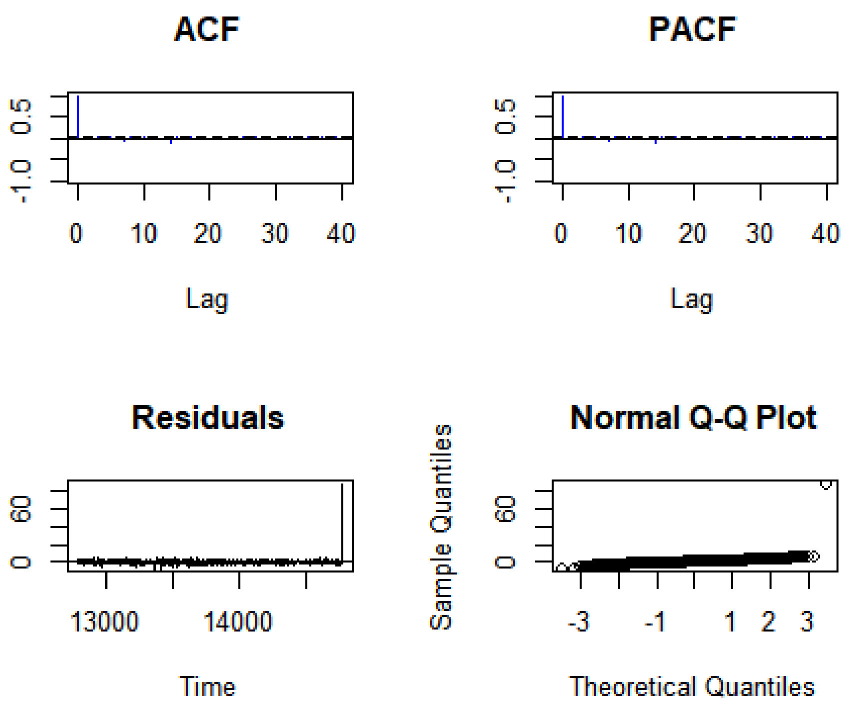
Since some coefficients are not away from zero, I dropped some variables with high p-values if the reduced model has smaller AIC.

```
Call:
arima(x = data, order = c(9, 0, 0), xreg = xreg, include.mean = F, transform.pars = F,
    fixed = c(NA, NA, 0, 0, 0, NA, NA, NA, NA, NA, NA))

Coefficients:
         ar1      ar2  ar3  ar4  ar5     ar6
      1.2295  -0.2301    0    0    0  0.0646
s.e.  0.0363   0.0410    0    0    0  0.0376
          ar7     ar8      ar9     const     time
      -0.5397  0.5392  -0.0980  114.5020  -0.0380
s.e.   0.0530  0.0558   0.0369    3.2284   0.0029

sigma^2 estimated as 6.253:  log likelihood = -4656.79,  aic = 9331.58

Training set error measures:
                       ME     RMSE      MAE
Training set -0.001600881 2.500538 1.190625
                    MPE     MAPE      MASE
Training set -0.1047747 1.712881 0.8594828
                    ACF1
Training set 0.002715064
```

[Linear Regression + AR(9) with constraint optimization]

By removing some unnecessary variables, AIC decreased from 9336.26 to 9331.58. Also, the reduced model would give easier interpretation and stable parameter estimation.

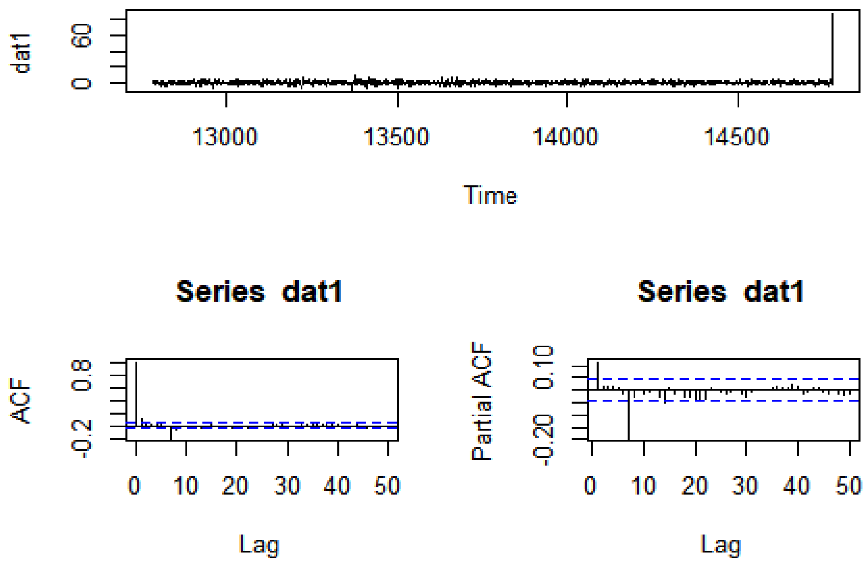
```
Null hypothesis: Residuals are iid noise.
Test                        Distribution Statistic   p-value
Ljung-Box Q                 Q ~ chisq(20)     34.04    0.0259 *
McLeod-Li Q                 Q ~ chisq(20)      0.02         1
Turning points T(T-1327.3)/18.8 ~ N(0,1)      1315    0.5121
Diff signs S        (S-996)/12.9 ~ N(0,1)       994    0.8767
Rank P        (P-992514)/14834.5 ~ N(0,1)    955276    0.0121 *
```



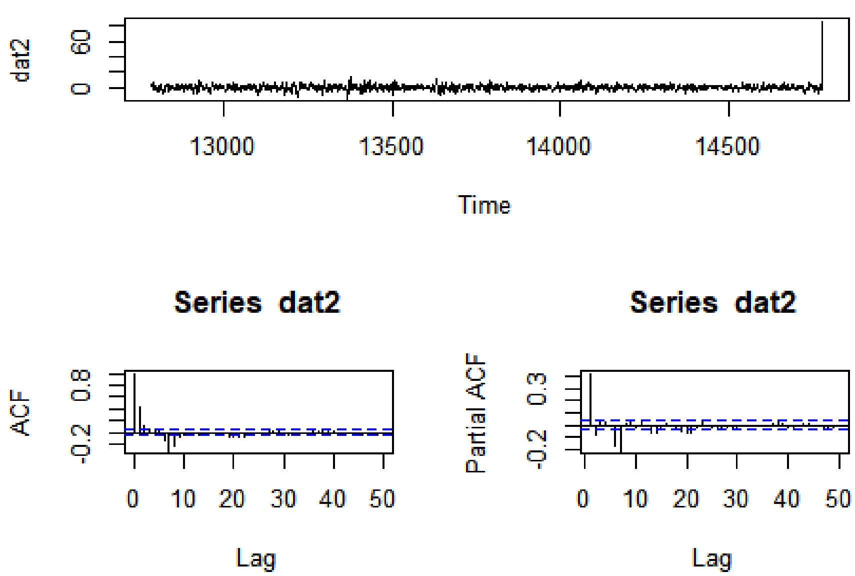
Checking disgnostics, formal test result looks fine. Also, the residuals seem to be iid based on residual plot and correlograms. Thus, "linear model + AR(9) (with constraint optimization)" model is selected as the best regression + stationary error model.

(d) Find (your) best SARIMA model. You need to include reasonings for your selection.

Since the given data has deterministic trend, try differencing first.



[Result of 1st order differencing]



[Result of 2nd order differencing]

We can observe that trend is disappeared. Since there is no significant improvement between order 1 and 2, first order differencing seems enough.

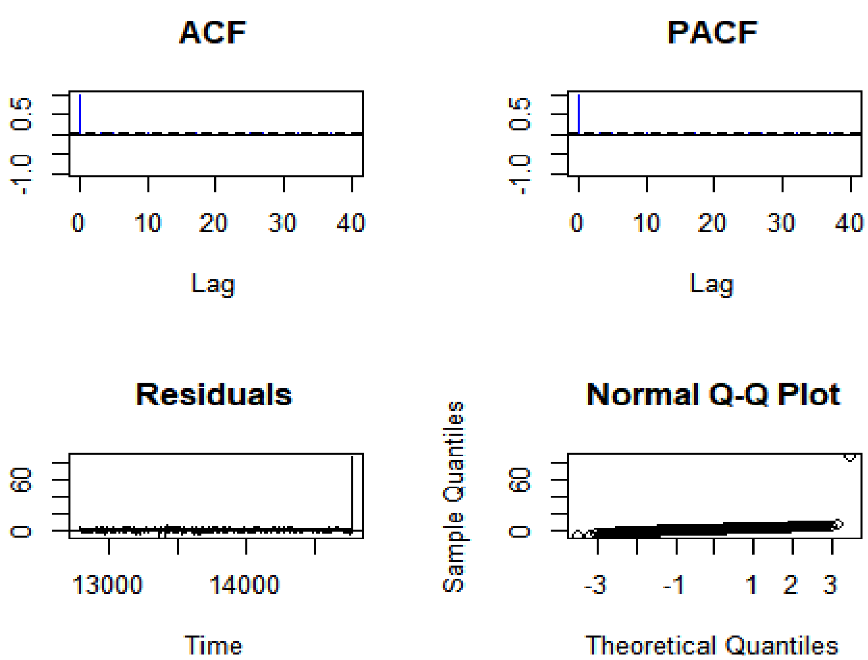
Also, we can observe that PACF(1) and PACF(7) is valid. Thus, simply set period = 7 instead of using sparse model.

```
Call:
arima(x = data, order = c(1, 1, 0), seasonal = list(order = c(0, 0, 1), period = 7))

Coefficients:
         ar1     sma1
      0.2920  -0.8175
s.e.  0.0373   0.0198

sigma^2 estimated as 5.755:  log likelihood = -4573.56,  aic = 9153.12

Training set error measures:
                    ME     RMSE      MAE        MPE
Training set -0.115082 2.398428 1.069037 -0.2385731
                 MAPE      MASE         ACF1
Training set 1.540365 0.7717113 -0.006509658
```

[SARIMA(1,1,0)(0,0,1) with period 7]

According to information criteria result, I tried SARIMA(1,1,0)(0,0,1) model.

```
Null hypothesis: Residuals are iid noise.
Test                        Distribution Statistic   p-value
Ljung-Box Q                Q ~ chisq(20)       6.1    0.9988
McLeod-Li Q                Q ~ chisq(20)      0.02         1
Turning points T(T-1327.3)/18.8 ~ N(0,1)      1344    0.3757
Diff signs S        (S-996)/12.9 ~ N(0,1)      997    0.9382
Rank P       (P-992514)/14834.5 ~ N(0,1)   1002927    0.4827
```



Next step is checking diagnostics. Since all formal tests are not rejected, the residual is iid. There is no pattern left and the residuals seem to be iid in residual plot, correlogram. Normal QQplot also looks fine. Thus, SARIMA(1,1,0)(0,0,1) with period 7 is selected as the best model.

(e) Forecast the next 4 quarters with 95% prediction interval for both models (c) and (d) you selected. Use two decimal places (ex, 1.23) in your report. Report them as the table in the below:

		June 17,2010	June 18,2010	June 19,2010	June 20,2010
Model (c)	Point Forecast	137.50	142.33	143.85	144.78
	95% PI	(132.58,142.41)	(134.55,150.11)	(133.84,153.86)	(132.93,156.64)
Model (d)	Point Forecast	144.81	153.94	158.05	159.79
	95% PI	(140.11,149.51)	(146.26,161.62)	(148.01,168.1)	(147.77,171.8)

(f) Which one do you prefer (c) or (d), and why? If you have better model than models in (c) & (d), you can describe your own model here with your rational.

```
##          model_C    model_D
## MSPE   82.07207   79.69969
## AIC  9331.57772 9153.11954
```

- We can observe that AIC of model (d) is smaller than AIC of model (c). Also, MSPE of model (d) is smaller than MSPE (number of test data: 100, 1 step ahead forecasting) of model (c). Thus, I prefer model (d).
- We can apply smoothing methods like exponential smoothing for forecasting time series data. Also, if we have some relevant variables or generate variables by feature engineering, we may use machine learning models such as random forest and deep learning models like LSTM in time series problem. I think those methods will work quite well and the reason is as follows: Although time series data has some dependency between observations, random forest model is not sensitive to the dependency. Also, since the model is flexible, it will capture some pattern well.

(g) Write down summary (no longer than 1/2 page) on your data analysis result.

From time plot and correlograms, we can observe the given data has linearly or quadratically decreasing trend, and very slowly decaying SACFs. Also, there are some outliers roughly at the end of the time period. To detrend, regression method was first used. In regression case, the trend is estimated by linear regression model. By examining ACF and PACF plot of the residuals, some PACFs at small lags are valid. Thus, linear regression + AR(9) model with backward constraint optimization model is selected as the best model in (c).

In SARIMA case, 1^{st} order differencing successfully removes the trend. Then, based on information criteria, SARIMA(1,1,0)(0,0,1) with period 7 is selected as the best model in (d). The selected model in (c) and (d) successfully removed some non-stationary factor.

According to the forecasting next 4 quarters results, model (d) gave higher prediction values than model (c). For comparing model (c) and (d), 1step ahead forecasting error of 100 test datasets are computed. Since model (d) showed smaller MSPE as well as smaller AIC, model (d) is better than model (c).

(h) Attach R code

```r
setwd("C:/Users/SJ/OneDrive/바탕 화면/시계열/시험 2")
rm(list = ls())
library(itsmr)
library(forecast)
library(MASS)
library(glmnet)
library(tseries)
library(aTSA)
library(tidyverse)
library(zoo)
source("TS-library.R")
load("Alasso.Rdata")

data = read.csv("practice2-2021sp.csv")
data = zoo(data[,2],seq(from = as.Date("2005-01-01"),
                        to = as.Date("2010-06-16"),by = 1))



layout(matrix(c(1,1,2,2),2,2,byrow = T))
plot.ts(data)
acf(data,lag = 50)
```

```r
n = length(data)
const = rep(1,n)
```

```r
time = 1:n
time2 = time^2

out.lm1 = lm(data ~ time)
summary(out.lm1)

##
## Call:
## lm(formula = data ~ time)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.001  -4.686  -0.045   4.825  84.588
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.169e+02  3.317e-01   352.5   <2e-16 ***
## time        -4.157e-02  2.881e-04  -144.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.401 on 1991 degrees of freedom
## Multiple R-squared:  0.9127, Adjusted R-squared:  0.9126
## F-statistic: 2.081e+04 on 1 and 1991 DF,  p-value: < 2.2e-16

out.lm2 = lm(data ~ time + time2)
summary(out.lm2)

##
## Call:
## lm(formula = data ~ time + time2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.690  -3.628   0.179   3.909  75.188
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.264e+02  4.094e-01  308.60   <2e-16 ***
## time        -6.992e-02  9.484e-04  -73.73   <2e-16 ***
## time2        1.422e-05  4.605e-07   30.88   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.087 on 1990 degrees of freedom
## Multiple R-squared:  0.941,  Adjusted R-squared:  0.9409
## F-statistic: 1.586e+04 on 2 and 1990 DF,  p-value: < 2.2e-16

layout(matrix(c(1,2,1,2),2,2,byrow = T))
plot.ts(data)
```

```r
lines(out.lm1$fitted,col = 'red')
plot.ts(data)
lines(out.lm2$fitted,col = 'blue')

par(mfrow = c(2,2))
plot(out.lm1)

test(residuals(out.lm1))

## Null hypothesis: Residuals are iid noise.
## Test                        Distribution Statistic   p-value
## Ljung-Box Q                Q ~ chisq(20)  34390.58         0 *
## McLeod-Li Q                Q ~ chisq(20)   1579.78         0 *
## Turning points T(T-1327.3)/18.8 ~ N(0,1)         0         0 *
## Diff signs S       (S-996)/12.9 ~ N(0,1)         0         0 *
## Rank P       (P-992514)/14834.5 ~ N(0,1)         0         0 *
```

```r
n = length(data)
const = rep(1,n)
time = 1:n
xreg = cbind(const,time)

# AR(9)
fit.9 = arima(data,order = c(9,0,0),
              xreg = xreg,include.mean = F)
summary(fit.9)

##
## Call:
## arima(x = data, order = c(9, 0, 0), xreg = xreg, include.mean = F)
##
## Coefficients:
##          ar1      ar2     ar3     ar4      ar5     ar6      ar7     ar8
##       1.2367  -0.2648  0.0447  0.0003  -0.0341  0.0846  -0.5446  0.5423
## s.e.  0.0369   0.0560  0.0539  0.0538   0.0539  0.0539   0.0538  0.0558
##           ar9     const     time
##       -0.0998  114.8486  -0.0383
## s.e.   0.0369    3.1818   0.0028
##
## sigma^2 estimated as 6.249:  log likelihood = -4656.13,  aic = 9336.26
##
## Training set error measures:
##                        ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.004193143 2.499708 1.188822 -0.1061767 1.709914 0.8581809
##                      ACF1
## Training set 0.0003702706

2 * (1 - pnorm(abs(fit.9$coef / sqrt(diag(fit.9$var.coef)))))
```

```
##          ar1          ar2          ar3          ar4          ar5
ar6
## 0.000000e+00 2.232143e-06 4.066161e-01 9.953106e-01 5.271217e-01 1.167529e
-01
##          ar7          ar8          ar9        const         time
## 0.000000e+00 0.000000e+00 6.841733e-03 0.000000e+00 0.000000e+00
```

```
fit.9 = arima(data,order = c(9,0,0),
              xreg = xreg,include.mean = F,
              fixed = c(NA,NA,0,0,0,NA,NA,NA,NA,NA,NA),
              transform.pars = F)
summary(fit.9)
```

```
##
## Call:
## arima(x = data, order = c(9, 0, 0), xreg = xreg, include.mean = F, transfo
rm.pars = F,
##     fixed = c(NA, NA, 0, 0, 0, NA, NA, NA, NA, NA, NA))
##
## Coefficients:
##          ar1      ar2  ar3  ar4  ar5     ar6      ar7     ar8      ar9
##       1.2295  -0.2301    0    0    0  0.0646  -0.5397  0.5392  -0.0980
## s.e.  0.0363   0.0410    0    0    0  0.0376   0.0530  0.0558   0.0369
##          const     time
##       114.5020  -0.0380
## s.e.    3.2284   0.0029
##
## sigma^2 estimated as 6.253:  log likelihood = -4656.79,  aic = 9331.58
##
## Training set error measures:
##                        ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.001600881 2.500538 1.190625 -0.1047747 1.712881 0.8594828
##                     ACF1
## Training set 0.002715064
```

```
test(residuals(fit.9))
```

```
## Null hypothesis: Residuals are iid noise.
## Test                        Distribution Statistic   p-value
## Ljung-Box Q                Q ~ chisq(20)     34.04    0.0259 *
## McLeod-Li Q                Q ~ chisq(20)      0.02         1
## Turning points T(T-1327.3)/18.8 ~ N(0,1)      1315    0.5121
## Diff signs S       (S-996)/12.9 ~ N(0,1)       994    0.8767
## Rank P       (P-992514)/14834.5 ~ N(0,1)    955276    0.0121 *
```

```
# 1st order differencing
dat1 = diff(data,1)
layout(matrix(c(1,1,2,3),2,2,byrow = T))
```

```r
plot.ts(dat1)
acf(dat1,lag = 50)
pacf(dat1,lag = 50)

# 2nd order differencing
dat2 = diff(data,2)
layout(matrix(c(1,1,2,3),2,2,byrow = T))
plot.ts(dat2)
acf(dat2,lag = 50)
pacf(dat2,lag = 50)

fit.s = arima(data,order = c(1,1,0),
              seasonal = list(order = c(0,0,1),period = 7))
summary(fit.s)

##
## Call:
## arima(x = data, order = c(1, 1, 0), seasonal = list(order = c(0, 0, 1), pe
riod = 7))
##
## Coefficients:
##          ar1     sma1
##       0.2920  -0.8175
## s.e.  0.0373   0.0198
##
## sigma^2 estimated as 5.755:  log likelihood = -4573.56,  aic = 9153.12
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.115082 2.398428 1.069037 -0.2385731 1.540365 0.7717113
##                      ACF1
## Training set -0.006509658

2 * (1 - pnorm(abs(fit.s$coef / sqrt(diag(fit.s$var.coef)))))

##          ar1         sma1
## 4.884981e-15 0.000000e+00

test(residuals(fit.s))

## Null hypothesis: Residuals are iid noise.
## Test                        Distribution Statistic   p-value
## Ljung-Box Q                Q ~ chisq(20)       6.1    0.9988
## McLeod-Li Q                Q ~ chisq(20)      0.02         1
## Turning points T(T-1327.3)/18.8 ~ N(0,1)      1344    0.3757
## Diff signs S       (S-996)/12.9 ~ N(0,1)       997    0.9382
## Rank P      (P-992514)/14834.5 ~ N(0,1)   1002927    0.4827
```

```r
# Final model
fit.9 = Arima(data,order = c(9,0,0),
              xreg = xreg,include.mean = F,
              fixed = c(NA,NA,0,0,0,NA,NA,NA,NA,NA,NA),
              transform.pars = F)

fit.s = arima(data,order = c(1,1,0),
              seasonal = list(order = c(0,0,1),
                              period = 7))

# newx
h = 4
const = rep(1,h)
time = (n+1):(n+h)
newx = cbind(const,time)

# prediction
prediction = data.frame(
  model.c = rep(NA,4),
  model.c.lower = rep(NA,4),
  model.c.upper = rep(NA,4),
  model.d = rep(NA,4),
  model.d.lower = rep(NA,4),
  model.d.upper = rep(NA,4)
)

# Model (c)
pred.c = forecast::forecast(fit.9,xreg = newx,h = 4)
prediction$model.c = pred.c$mean
prediction$model.c.lower = pred.c$lower[,2]
prediction$model.c.upper = pred.c$upper[,2]

# Model (d)
pred.d = forecast::forecast(fit.s,h = 4)
prediction$model.d = pred.d$mean
prediction$model.d.lower = pred.d$lower[,2]
prediction$model.d.upper = pred.d$upper[,2]

prediction

##    model.c model.c.lower model.c.upper  model.d model.d.lower model.d.uppe
r
## 1 137.4958      132.5850      142.4066 144.8083      140.1063      149.510
3
## 2 142.3307      134.5477      150.1137 153.9421      146.2600      161.624
1
## 3 143.8501      133.8406      153.8596 158.0549      148.0075      168.102
3
```

```
## 4 144.7817       132.9282       156.6352 159.7853       147.7680       171.802
7

model_C_PI = paste0('(',round(prediction$model.c.lower,2),',',round(predictio
n$model.c.upper,0),')')
model_D_PI = paste0('(',round(prediction$model.d.lower,2),',',round(predictio
n$model.d.upper,0),')')

tab = data.frame(
  model_C_point = prediction$model.c,
  model_C_PI = model_C_PI,
  model_D_point = prediction$model.d,
  model_D_PI = model_D_PI
)
rownames(tab) = c('Q1','Q2',"Q3","Q4")
tab = as.data.frame(t(tab))
tab
```

	Q1	Q2	Q3	Q4
model_C_point	137.50	142.33	143.85	144.78
model_C_PI	(132.58,142.41)	(134.55,150.11)	(133.84,153.86)	(132.93,156.64)
model_D_point	144.81	153.94	158.05	159.79
model_D_PI	(140.11,149.51)	(146.26,161.62)	(148.01,168.1)	(147.77,171.8)

```
m = 100; n = length(data)
N = n - m
testindex = (N+1):n

# model (c)
err.c = numeric(m)
for (i in 1:m) {
  trainindex = time = 1:(N+i-1)
  const = rep(1,N+i-1)
  xreg = cbind(const,time)
  fit.c = Arima(data[trainindex],order = c(9,0,0),
               xreg = xreg,include.mean = F,
               fixed = c(NA,NA,0,0,0,NA,NA,NA,NA,NA,NA),
               transform.pars = F)
  time = N+i
  Xhat = forecast::forecast(fit.c,h = 1,
                 xreg = cbind(1,time))$mean
  err.c[i] = (data[N+i] - Xhat)^2
}
```

```r
# model (d)
err.d = numeric(m)
for (i in 1:m) {
  trainindex = 1:(N+i-1)
  fit.d = arima(data[trainindex],order = c(1,1,0),
              seasonal = list(order = c(0,0,1),
                              period = 7))
  Xhat = forecast::forecast(fit.d,h = 1)$mean
  err.d[i] = (data[N+i] - Xhat)^2
}

comp = data.frame(
  model_C = c(mean(err.c),fit.9$aic),
  model_D = c(mean(err.d),fit.s$aic)
)
rownames(comp) = c('MSPE','AIC')
comp

##          model_C    model_D
## MSPE   82.07207   79.69969
## AIC  9331.57772 9153.11954
```