Time Series Analysis

UC Berkeley, School of Information: MIDS w271

2023-07-25

Contents

Problem Set 11

This is the final problem set that you will have to work on for this class. Congratulations! (Although there is still a group lab that will be the final assignment in the course.)

You will start with some guided work, and then proceed into less structured work that will let you stretch and demonstrate what you have learned to date.

Notice, in particular, that the last few questions are asking you essentially to "produce a model" using a method. At this point in the course, you should be familiar with many of the model forms that you *might* fit; and, you are familiar with methods that you can use to evaluate models' performances. In these questions, we are asking you to, essentially, fit a good model with a method and then to evaluate how a good model with "this" method is doing compared to another good model with "that" method.

In several of these questions, there isn't a correct answer, *per se*. Instead, there is the process that you will undertake and record as you are producing your argument for the model that you think is best meeting your objectives. This is a **very** applied task that we anticipate you will see many times in your work.

```
knitr::opts_chunk$set(echo=TRUE)
```

We are providing you with an additional challenge, but one that is also very evocative of work that you're likely to come across. This is a well-built repository, that uses a well-documented framework to produce reports, namely `bookdown`.

Once you have done your work, you can render the entire book using the following call in your console:

```
> bookdown::render_book()
```

This will ingest each of the files `01-time_series_...`, `02-cross_validation.Rmd`, `03-ARIMA_model.Rmd` and so on ... and produce a PDF that is stored in `./_book/_main.pdf`. If you would like to read more about this framework, you can do so at the following website: https://bookdown.org/yihui/bookdown/ .

Chapter 1

(10 points) ARIMA model

Consider `fma::sheep`, the sheep population of England and Wales from 1867–1939. :sheep:

```
#install.packages('fma')
library(fma)
sheep_ts <- as_tsibble(fma::sheep)
head(sheep_ts)
```

```
## # A tsibble: 6 x 2 [1Y]
##   index value
##   <dbl> <dbl>
## 1  1867  2203
## 2  1868  2360
## 3  1869  2254
## 4  1870  2165
## 5  1871  2024
## 6  1872  2078
```

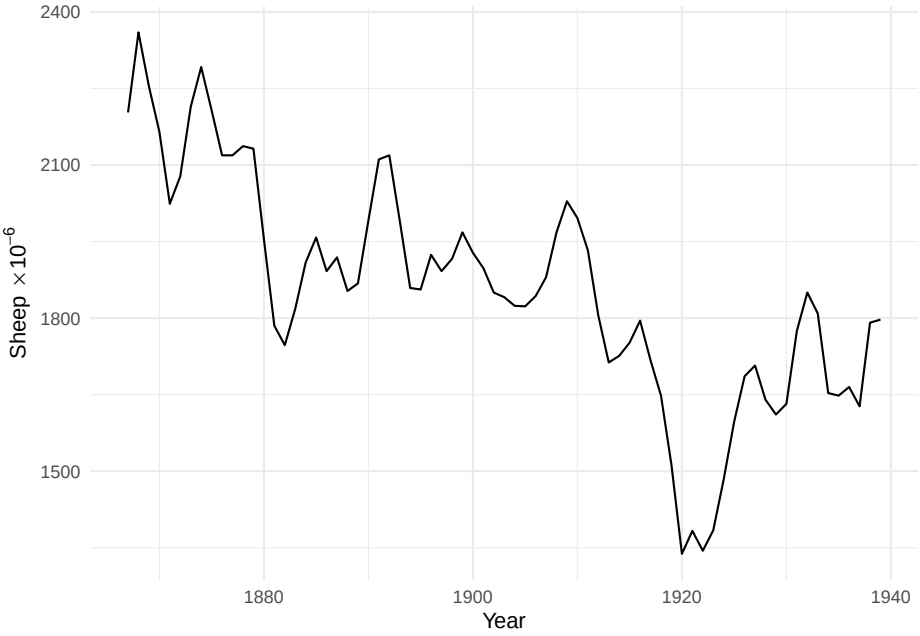
1.1 Time series plot

Produce a time plot of the time series, and comment on what you observe.

```
sheep_plot <- autoplot(sheep_ts) + labs(x = "Year", y = TeX(r'(Sheep $\times 10^{-6}$)'))
```

```
## Plot variable not specified, automatically selected `.vars = value`
```

```
sheep_plot
```



Sheep population is gradually decreasing over this period and appears to stabilize in the 1930's at around 1700×10^6. There is a significant local dip that coincides with the WWI and subsequent period of economic turmoil in Europe.

1.2 Fit a model

Assume you decide to fit the following model:

$$y_t = y_{t-1} + \phi_1(y_{t-1} - y_{t-2}) + \phi_2(y_{t-2} - y_{t-3}) + \phi_3(y_{t-3} - y_{t-4}) + \epsilon_t,$$

where ϵ_t is a white noise series.

1.2.1 Model type

What sort of ARIMA model is this (i.e., what are p, d, and q)?

Thi is a representation of ARIMA(3.1.0). There are no error terms from the past lags, therefore MA term q = 0; there is only one differing step, so d = 1, and the last lag taken into account is y_{t-4} which means p = 3 (given d = 1).

1.2.2 Back to the future

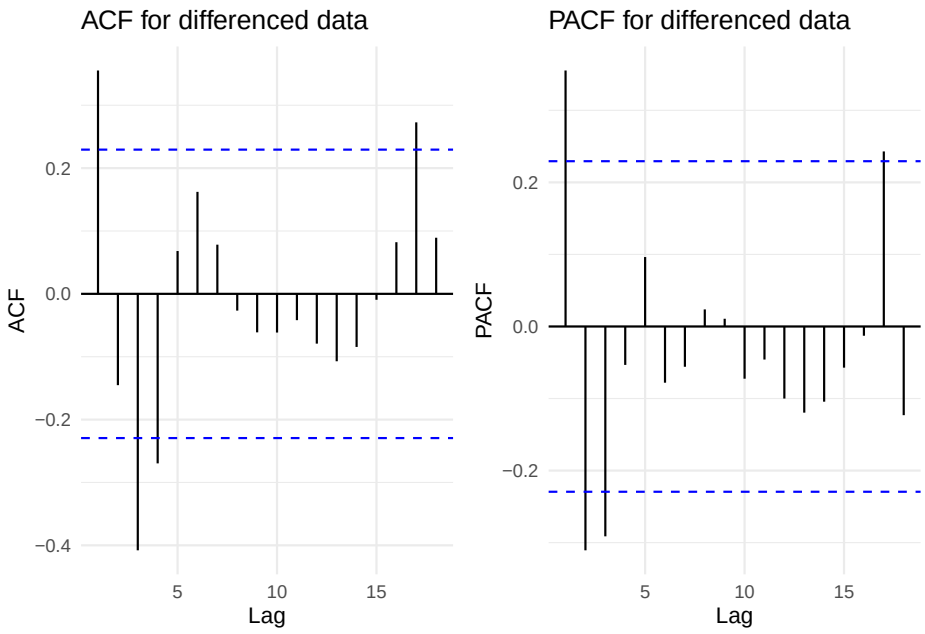
Express this ARIMA model using backshift operator notation.

$$(1-B)y_t = \phi_1(1-B)y_{t-1} + \phi_2(1-B)y_{t-2} + \phi_3(1-B)y_{t-3} + \epsilon_t$$
$$(1-B)y_t = \phi_1(1-B)By_t + \phi_2(1-B)B^2y_t + \phi_3(1-B)B^3y_t + \epsilon_t$$
$$(1-B)y_t = (1-B)(\phi_1By_t + \phi_2B^2y_t + \phi_3B^3y_t) + \epsilon_t$$
$$(1-B)(1-\phi_1B - \phi_2B^2 - \phi_3B^3)y_t = \epsilon_t$$

1.3 Is this model appropriate?

Examine the ACF and PACF of the differenced data. Evaluate whether this model is appropriate.

```
# Add a column with differenced data to the original tsibble
sheep_ts <- sheep_ts %>% mutate(diff = difference(value, lag = 1))
# Generate acf plot
acf_plot <- acf(sheep_ts$diff, na.action = na.pass, plot = FALSE) %>%
  autoplot() + labs(title = "ACF for differenced data")
# Generate pacf plot
pacf_plot <- pacf(sheep_ts$diff, na.action = na.pass, plot = FALSE) %>%
  autoplot() + labs(title = "PACF for differenced data")
acf_plot | pacf_plot # Display plots
```



The model does not capture all the aspects of the underlying data. On one hand, PACF plot shows only 3 signofocant lags, indicating that thsi si indeed AR(3) process. On the other hand, ACF plot shows abrupt decrease of significance after lag 4 and some oscillating

pattern, indicating MA(4) process with some periodicity. Therefore ARIMA(3.1.4) would be more appropriate model.

1.4 Forecasts, by hand!

The last five values of the series are given below:

Year	1935	1936	1937	1938	1939
Millions of sheep	1648	1665	1627	1791	1797

The estimated parameters are:

- $\phi_1 = 0.42$;
- $\phi_2 = -0.20$; and,
- $\phi_3 = -0.30$.

Without using the forecast function, calculate forecasts for the next three years (1940–1942).

$y_{1940} = y_{1939} + \phi_1(y_{1939} - y_{1938}) + \phi_2(y_{1938} - y_{1937}) + \phi_3(y_{1937} - y_{1936}) + 0$

$y_{1941} = y_{1940} + \phi_1(y_{1940} - y_{1939}) + \phi_2(y_{1939} - y_{1939}) + \phi_3(y_{1938} - y_{1937}) + 0$

$y_{1942} = y_{1941} + \phi_1(y_{1941} - y_{1940}) + \phi_2(y_{1940} - y_{1939}) + \phi_3(y_{1939} - y_{1938}) + 0$

```
y1936<-1665
y1937<-1627
y1938<-1791
y1939<-1797
phi_1 <-  0.42
phi_2 <- -0.20
phi_3 <- -0.30

y1940 <- y1939 + phi_1*(y1939-y1938) + phi_2*(y1938-y1937) + phi_3*(y1937-y1936)
y1941 <- y1940 + phi_1*(y1940-y1939) + phi_2*(y1939-y1939) + phi_3*(y1938-y1937)
y1942 <- y1941 + phi_1*(y1941-y1940) + phi_2*(y1940-y1939) + phi_3*(y1939-y1938)
```

Forecasted values for 1940, 1941 and 1942 are 1778, 1721 and 1699, respectively.

1.5 Interpret roots

Find the roots of your model's characteristic equation. Is this process stationary?.

```
roots <- polyroot(c(1, -phi_1, -phi_2, -phi_3))
mroot1 <- Mod(roots[1])
```

```
mroot2 <- Mod(roots[2])
mroot3 <- Mod(roots[3])
```

Charachteristic polynomial for the ARMA model is $1-\phi_1 B-\phi_2 B^2-\phi_3 B^3=0$. For the process to be stationary, all of the roots should lle strictly outside of a unit circle on the complex plane, i.e. modules of all roots should be more than 1. We have modules 1.26, 2.09, 1.26, all above 1. That means that the process is stationary.

Chapter 2

(10 points) Seasonal ARIMA model

```
library(fredr)
if (fredr_has_key()){
  ecom_df <- fredr(series_id = "ECOMPCTNSA",
  observation_start = as.Date("1990-01-01"))
  ecom_df <- cbind.data.frame(ecom_df[c(1,3)], index = 1:nrow(ecom_df))
  ecom_ts <- as_tsibble(ecom_df, index = "date")
} else {
  print("Expect FREDR API key as an environment variable")
  quit(save="ask")
}
```

```
split <- as.Date("2020-12-31")
train_ts <- ecom_ts %>% filter(date < split)
test_ts <- ecom_ts %>% filter(date >= split)
```

Download the series of E-Commerce Retail Sales as a Percent of Total Sales here.

(Feel free to explore the `fredr` package and API if interested.)

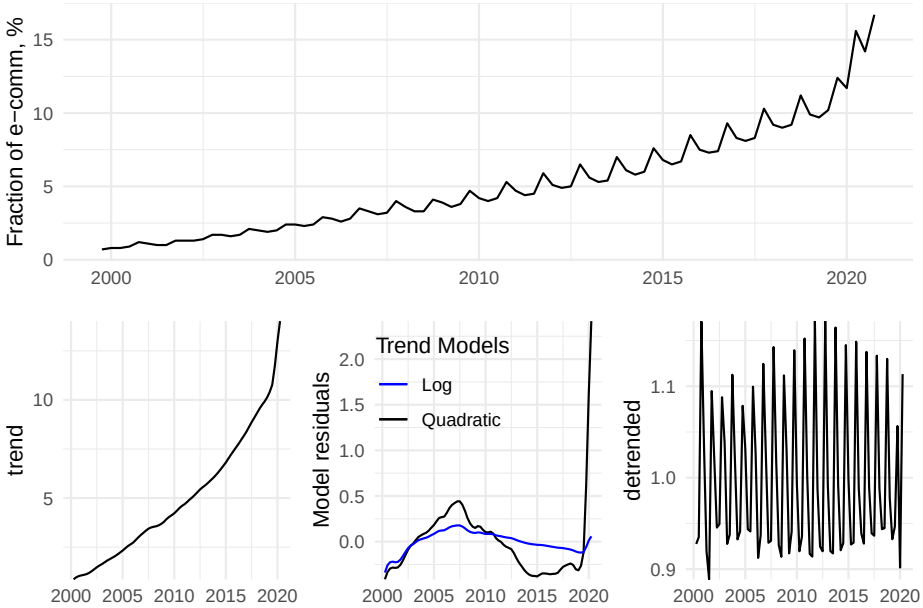
Our goal is to Build a Seasonal ARIMA model, following all appropriate steps for a univariate time series model.

Separate the data set into training and test data. The training data is used to estimate model parameters, and it is for 10/1999-12/2020. The test data is used to evaluate its accuracy, and it is for 01/2021-01/22.

2.1 Time series plot

Plot training data set of Retail Sales. What do you notice? Is there any transformation necessary?

```
## Plot variable not specified, automatically selected `.vars = value`
```



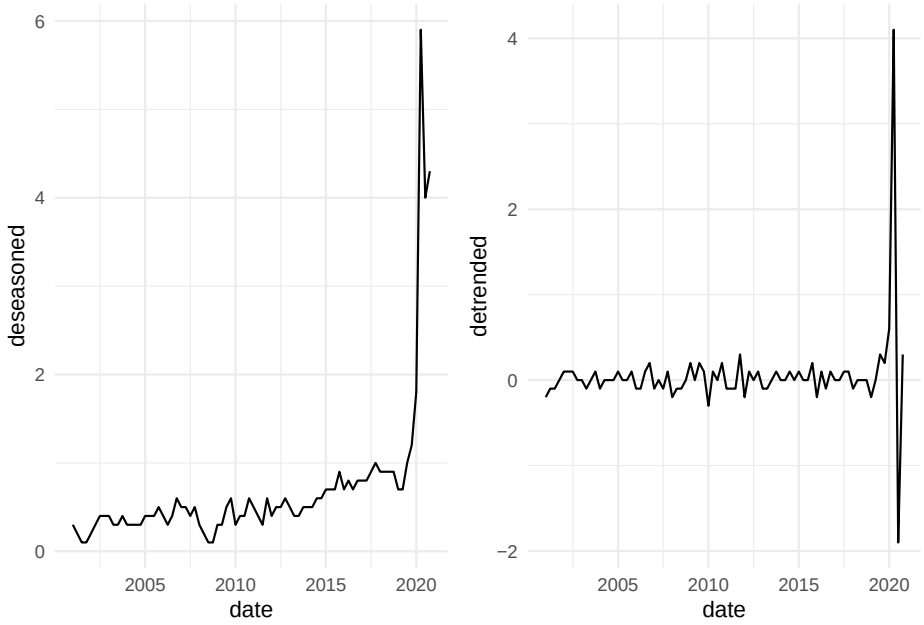
Fraction of e-commerce in the overall retail sales is growing, and its growth is accelerating. It appears that at around the start of the COVID-19 pandemic the acceleration increased. This is evident from abrupt change in the overall trend curvature at the beginning of 2020. The series also has marked seasonal component, and the magnitude of the seasonal component is increasing, suggesting a multiplicative seasoning. The trend has obvious curvature, but neither quadratic, nor logarithmic transformation result in white-noise residuals.

2.2 Check for Stationary

Use ACF/PACF and a unit root test to check if Retail Sales is stationary. If data is not stationary, difference the data, and apply the test again until it becomes stationary? How many differences are needed to make data stationary?

```
my_lag <- 4
train_ts <- mutate(train_ts,
                   deseasoned = difference(value, lag = my_lag),
                   detrended = difference(deseasoned, lag = 1))
train_ts <- na.omit(train_ts)
```

```
deseasoned_plot <- ggplot(data = train_ts) + aes(x = date, y = deseasoned) +
                   geom_line()
detrended_plot <- ggplot(data = train_ts) + aes(x = date, y = detrended) +
                   geom_line()
deseasoned_plot | detrended_plot
```




```
value_p <- PP.test(train_ts$value)
deseasoned_p <- PP.test(train_ts$deseasoned)
detrended_p <- PP.test(train_ts$detrended)
```

Given strong trend and seasonality in the data it is obvious that the original series is not stationary. Deseasonng via differencing with lag 4 (1 year), does not eliminate the trend. Further de-trending with lag = 1 eliminates the trend, but the resulting trend is clearly heteroschedastic. Results of Phillips-Perron test reflect these observations. The test fails to reject the hypothesis of non-stationarity for the original series (p-value = 0.983) and de-seasoned series (p-value = 0.535), but strongly rejects this hypothesis for the de-trended series with p-value = 0.01)

2.3 Model identification and estimation

Use ACF/PACF to identify an appropriate SARIMA model. Estimate both select model and model chosen by ARIMA()

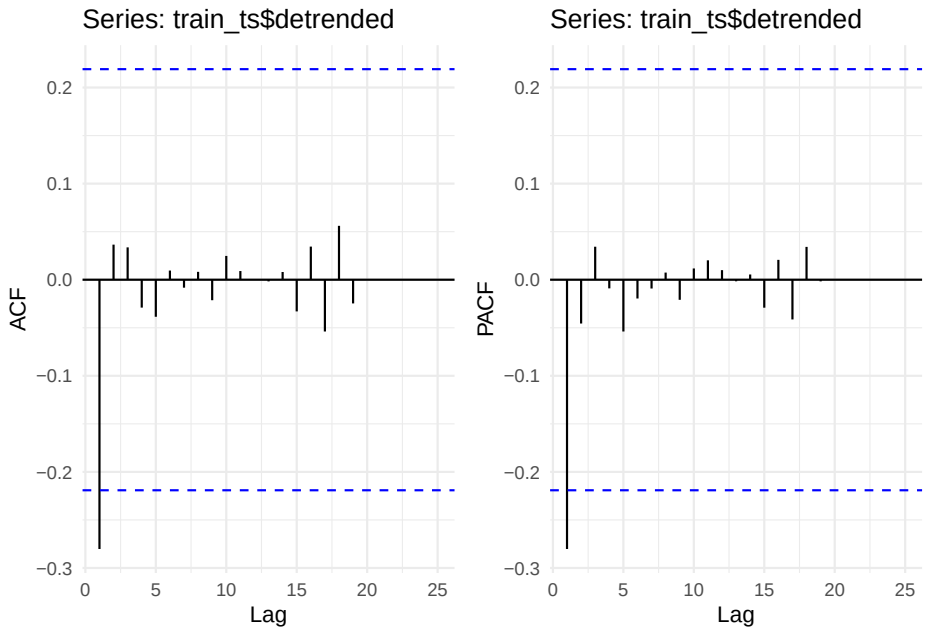
```
acf_plot <- acf(train_ts$detrended, plot = F) %>% autoplot() + xlim(1,25)
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

```
pacf_plot <- pacf(train_ts$detrended, plot = F) %>% autoplot() + xlim(1,25)
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

```
acf_plot | pacf_plot
```




```
model.manual<- arima(train_ts$value, order = c(1, 1, 1),
  seasonal = list(order = c(0, 1, 0), period = 4))
model.manual
```

```
##
## Call:
## arima(x = train_ts$value, order = c(1, 1, 1), seasonal = list(order = c(0, 1,
##     0), period = 4))
##
## Coefficients:
##           ar1      ma1
##       -0.2077  -0.0673
## s.e.   0.3814   0.3856
##
## sigma^2 estimated as 0.2697:  log likelihood = -57.32,  aic = 120.64
```

```
model.auto <- train_ts$value %>% ts(frequency = 4) %>% auto.arima(d = 1, D = 1,
              max.p = 5, max.q = 5, max.P = 2, max.Q = 2, max.d = 2, max.D = 2,
              max.order = 10,
              start.p = 0, start.q = 0, start.P = 0, start.Q = 0,
  ic="aic", seasonal = TRUE, stepwise = FALSE, approximation = FALSE, trace = FALSE)
model.auto
```

```
## Series: .
## ARIMA(1,1,0)(0,1,0)[4]
##
## Coefficients:
##           ar1
##       -0.2702
## s.e.   0.1105
##
## sigma^2 = 0.2735:  log likelihood = -57.34
## AIC=118.67   AICc=118.84   BIC=123.31
```

```
train_short <- train_ts %>% filter(date < as.Date("2020-01-01"))
model.short <- train_short$value %>% ts(frequency = 4) %>% auto.arima(d = 1, D = 1,
              max.p = 5, max.q = 5, max.P = 2, max.Q = 2, max.d = 2, max.D = 2,
              max.order = 10,
              start.p = 0, start.q = 0, start.P = 0, start.Q = 0,
  ic="aic", seasonal = TRUE, stepwise = FALSE, approximation = FALSE, trace = FALSE)
model.short
```

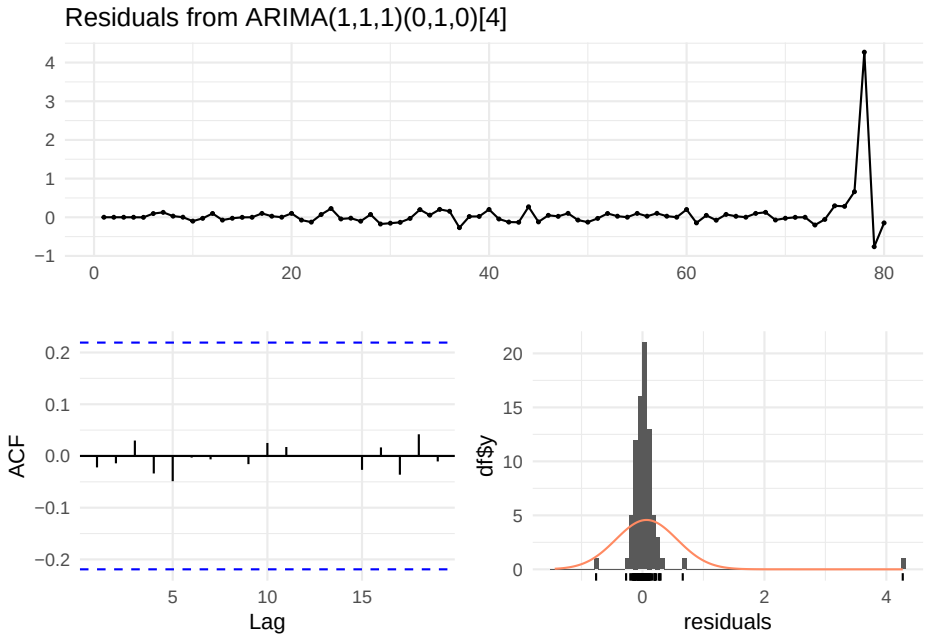
```
## Series: .
## ARIMA(0,1,3)(2,1,0)[4]
##
## Coefficients:
##           ma1      ma2      ma3    sar1    sar2
##       -0.2811  -0.0158  -0.4032  0.0990  0.3872
## s.e.   0.1236   0.1477   0.1312  0.1379  0.1206
##
## sigma^2 = 0.0114:  log likelihood = 59.74
## AIC=-107.47   AICc=-106.16   BIC=-93.9
```

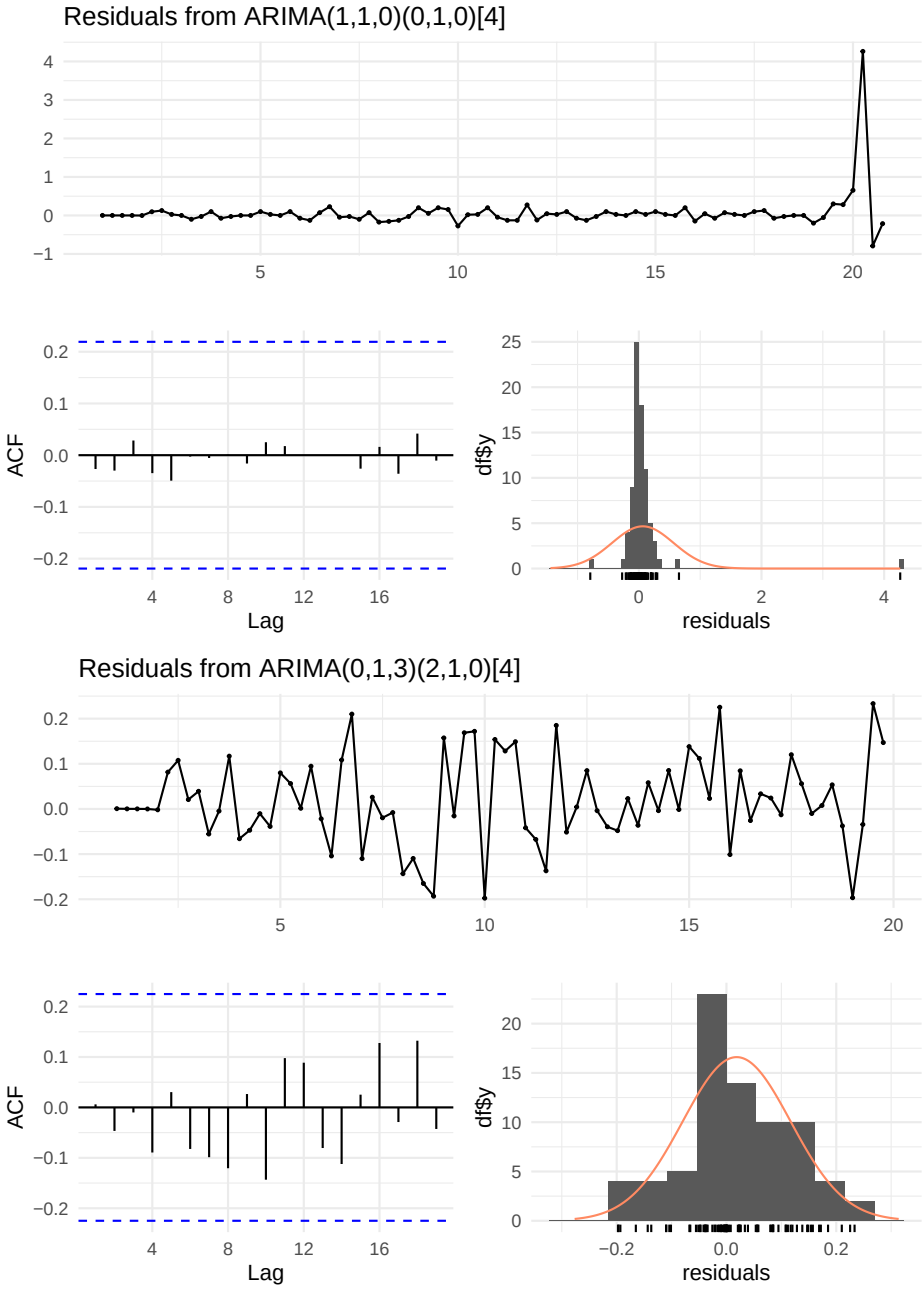
ACF plot for de-seasoned and de-trended series contains one strongly significant lag = 1, same as PACF plot. Formally, this is typical for ARMA(1.1) process, however, there is also a chance that this is an artifact caused by the few lags at the end of the series that are affected by the pandemic abnormality. Having abnormally large values at the tail of the series might make short lags a lot more significant. This considerations aside, ACF and PACF plot suggest SARIMA(1.1.1)(0.1.0)[4] model. Estimating this model results in AIC = 120.6. Grid search with `auto.arima` function yields very similar result. The function find s the lowest AIC 118.7 for SARIMA(1.1.0)(0.1.0)[4]. The fact that MA component of ARIMA does not improve the model, despite the fact that PACF

plot goes down abruptly after lag 1, supports the hypothesis that observed plots are just artifacts. Repeating this grid search on the shortned training set that excludes the pandemic data results in a much lower AIC of -107.5.

2.4 Model diagnostic

Do residual diagnostic checking of both models. Are the residuals white noise? Use the Ljung-box test to check if the residuals are white noise.





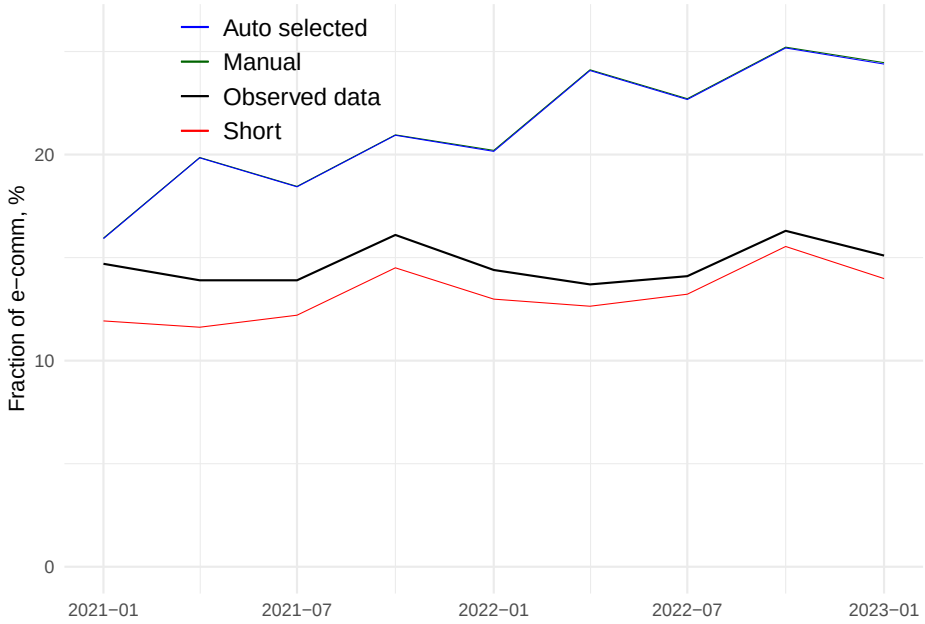
Both manually estimated model as well as `auto.arima` result fail visual tests for model quality: in both cases residual distribution has a significant outlier and residual plots contain strong spikes. However, Ljung-Box Test fails to reject the null hypothesis with p-values 0.84 and 0.81 respectively. The model estimated on the shorter data does have all the charachteristics of good fit: normally

distributed residuals with no pattern in acf plot and p-value 0.96 on Ljung-Box Test, indicating no autocorrelation in residuals.

2.5 Forcasting

Use the both models to forecast the next 12 months and evaluate the forecast accuracy of these models.

```
frcst_length = length(test_ts$value)
frcst_manual <- forecast::forecast(model.manual, h=frcst_length)
frcst_auto <- forecast::forecast(model.auto, h=frcst_length)
frcst_short <- forecast::forecast(model.short, h=frcst_length+4)
```



Both manually selected and auto-selected models make essentially the same predictions, both significantly overestimate the growth rate for the e-commerce sales. This is because the models were trained on the data that contained a large shock and they implicitly expect more of the similar shocks to come. The model that was trained on truncated data, on the other hand, captures the underlying market forces without compounding effect of once-in-a-lifetime anomaly. As a result, once the anomaly passes, this model predicts the actual observed data much better than the previous two.

Chapter 3

(10 points) Time Series Linear Model and Cointegration

Daily electricity demand and temperature (in degrees Celsius) is recorded in `./data/temperature_demand.csv`. Please work through the following questions to build a time series linear model against this data.

```
temperature <- read_csv('./data/temperature_demand.csv') %>%
  rename(
     'index'       = '...1',
     'demand'      = 'Demand',
     'work_day'    = 'WorkDay',
     'temperature' = 'Temperature'
  )
temp_ts <- temperature %>% filter(work_day == 1) %>%
                           as_tsibble(index = index)
names(temp_ts) <- c("my_day", "demand", "work_day", "temps")
```

3.1 Plot electricity

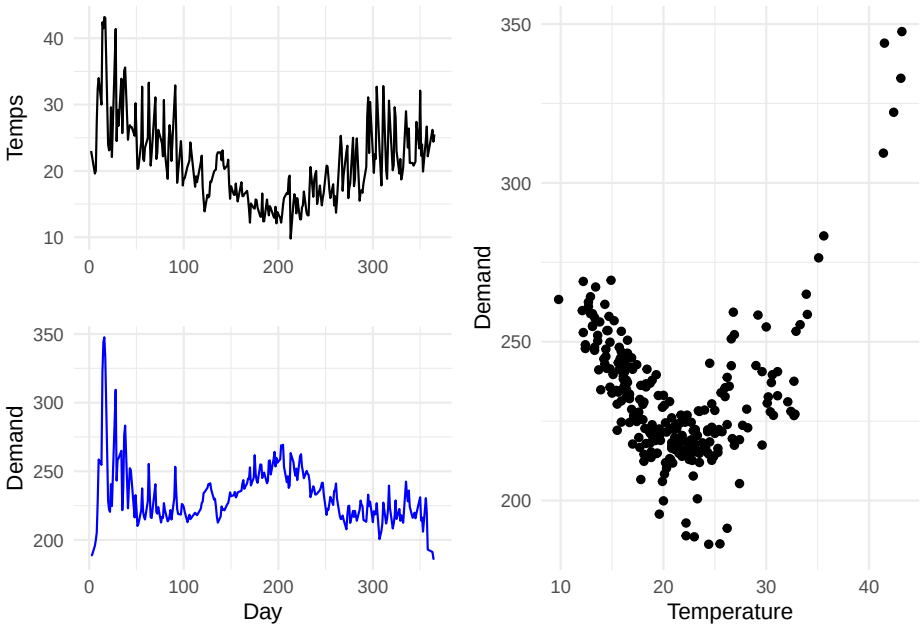
Plot electricity demand and temperature as time series. Is there any correlation between these to variables? If yes, Do you think is it a spurious correlation?

```
temp_plot <- ggplot(data = temp_ts) +
  geom_line(aes(x = my_day, y = temps), color = "black") +
  labs(x = "", y = "Temps")
```

```
demand_plot <-  ggplot(data = temp_ts) +
  geom_line(aes(x = my_day, y = demand), color = "blue") +
  labs(x = "Day", y = "Demand")

corr_plot <- ggplot(data = temp_ts) +
  geom_point(aes(x = temps, y = demand)) +
  labs(x = "Temperature", y = "Demand")

(temp_plot/demand_plot)|corr_plot
```



First of all, I decided to filter out the non-working days. Electricity demand is dramatically different between working and non-working days, so mixing the two together in the same model is equvalent to naive pooling of panel data - a sure way to cloudy conclusions. This transformation revealed reasonable correlation between the two series. This is likely a real correlation: energy is required to heat and cool houses, so the further away temperature from the comfortable 20-25^{o}C, the higher energy demand. From that perspective, re-framing temperature time series as an absolute value of deviation from 23^{o}C shows even higher correlation. It appears that cooling is a lot more energy expensive, so it would be even more evident if the deviation was weighted by the average amount of energy needed to cool or heat by 1 degree.

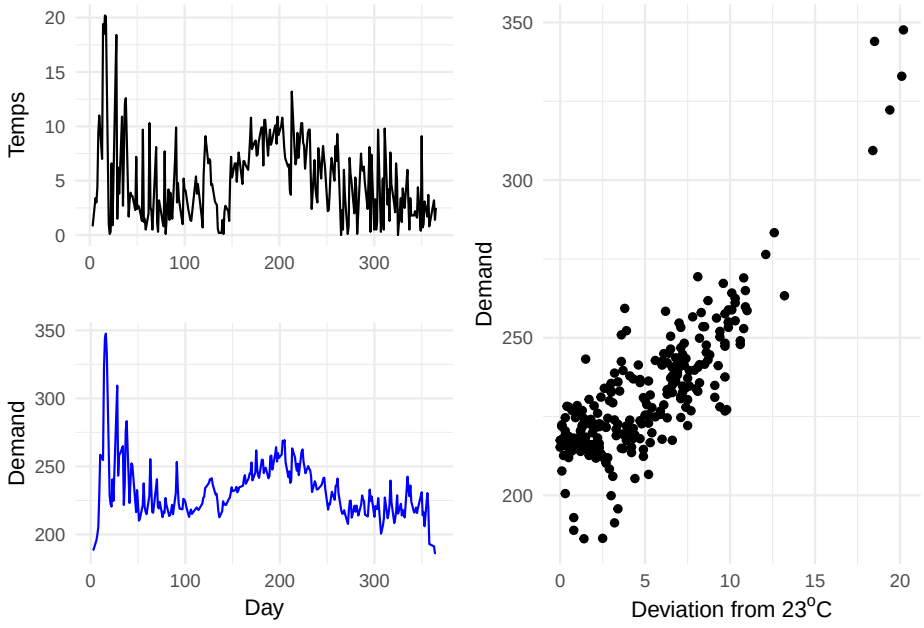
```
temp_ts <- temp_ts %>% mutate(
                        dev = abs(temp_ts$temps-23),
                        dev_sign = as.factor(ifelse(temp_ts$temps > 23, 1, 0)),
                        temp_diff = difference(temp_ts$temps, lag = 1),
```

```
                        demand_diff = difference(temp_ts$demand, lag = 1)
                      )
temp_ts <- na.omit(temp_ts)

temp_dev_plot <- ggplot(data = temp_ts) +
  geom_line(aes(x = my_day, y = dev), color = "black") +
  labs(x = "", y = "Temps")

corr_plot <- ggplot(data = temp_ts) +
  geom_point(aes(x = dev, y = demand)) +
  labs(x = TeX("Deviation from 23$^{o}$C"), y = "Demand")

(temp_dev_plot/demand_plot)|corr_plot
```



3.2 Cointegration test

Use the Engle-Granger test to check for cointegration. What do you conclude?

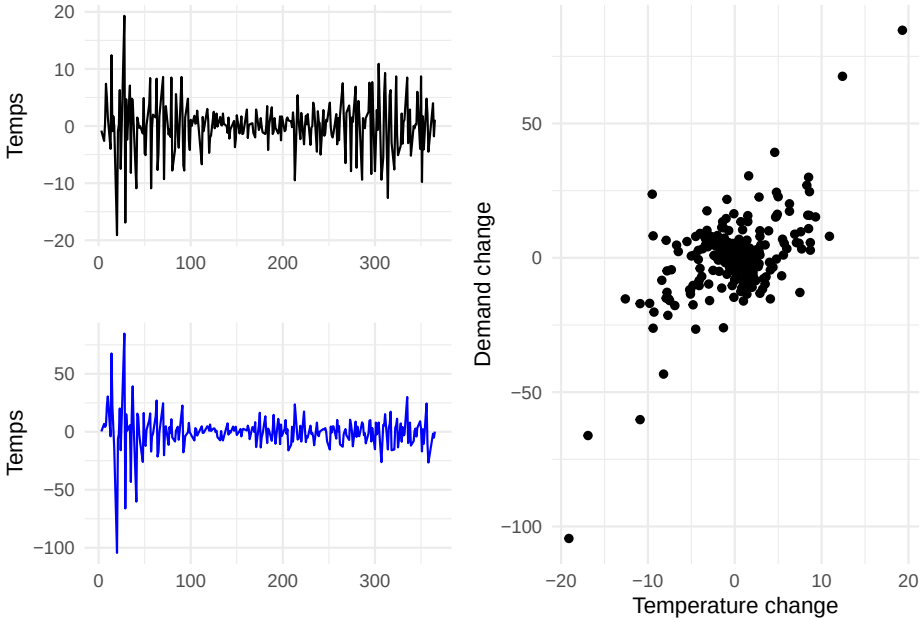
```
no_diff_test <- coint.test(temp_ts$demand, temp_ts$dev, d = 0, output = F)
diff_test <- coint.test(temp_ts$demand, temp_ts$dev, d = 1, output = F)
```

```
demand_diff_plot <- ggplot(data = temp_ts) +
  geom_line(aes(x = my_day, y = demand_diff), color = "blue") +
  labs(x = "", y = "Temps")
```

```
temp_diff_plot <- ggplot(data = temp_ts) +
  geom_line(aes(x = my_day, y = temp_diff), color = "black") +
  labs(x = "", y = "Temps")

diff_corr_plot <- ggplot(data = temp_ts) +
  geom_point(aes(x = temp_diff, y = demand_diff)) +
  labs(x = TeX("Temperature change"), y = "Demand change")

(temp_diff_plot/demand_diff_plot)|diff_corr_plot
```



In case of non-differenced series, Engle-Granger test rejects null-hypothesis of no co-integration with p-value < 0.01. The same holds even after differencing of both series with lag 1 (p-value < 0.01). This indicates that the series are likely related to each other and the correlation is real. At the same time, the differenced time series only appear correlated because of a few extreme points, within the bounds of normal temparature variability correlation is very weak.

3.3 Fit Model

Based on cointegration test,fit a regression model for demand with temperature as an explanatory variable(or their first difference).

```
nodiff.model <- lm(formula = demand ~ dev + dev_sign, data = temp_ts)
summary(nodiff.model)
```

```
##
```

```
## Call:
## lm(formula = demand ~ dev + dev_sign, data = temp_ts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.873  -6.820   0.737   8.047  44.777
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 206.4793     1.3848 149.109   <2e-16 ***
## dev           4.8108     0.2011  23.925   <2e-16 ***
## dev_sign1     3.7365     1.6296   2.293   0.0227 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.02 on 247 degrees of freedom
## Multiple R-squared:  0.7007, Adjusted R-squared:  0.6983
## F-statistic: 289.1 on 2 and 247 DF,  p-value: < 2.2e-16
```

```
diff.model <- lm(formula = demand_diff ~ temp_diff, data = temp_ts)
diff.model
```

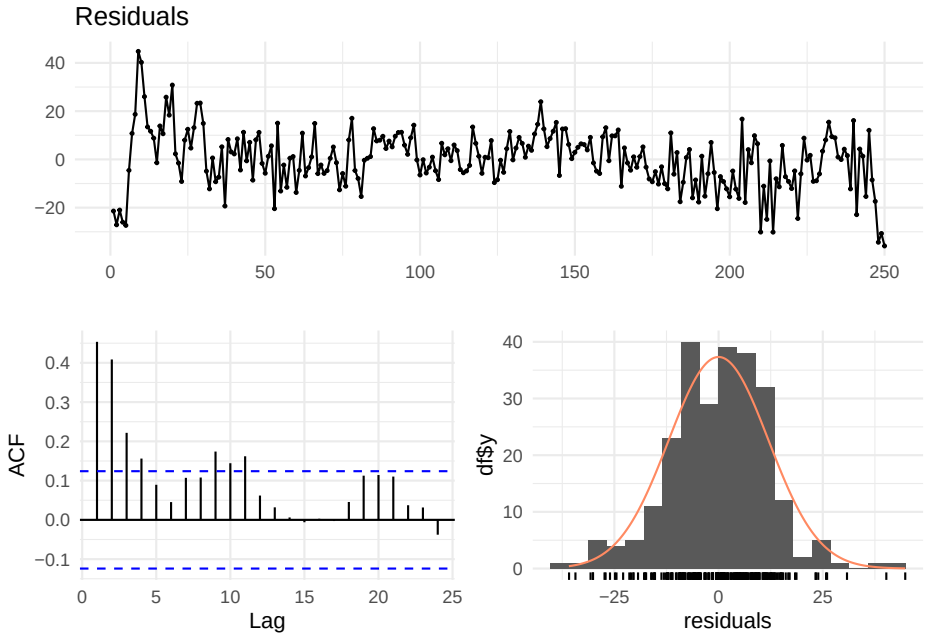
```
##
## Call:
## lm(formula = demand_diff ~ temp_diff, data = temp_ts)
##
## Coefficients:
## (Intercept)    temp_diff
##    -0.02865      1.97675
```

The model that is the most realistic takes into account how far temperature is out of the comfortable zone around 23°C and which way this deviation is. Introducing a categorical variable for sign of deviation from 23°C allowed us to take into account the fact that cooling is more energy intensive than heating.

3.4 Residuals Plot

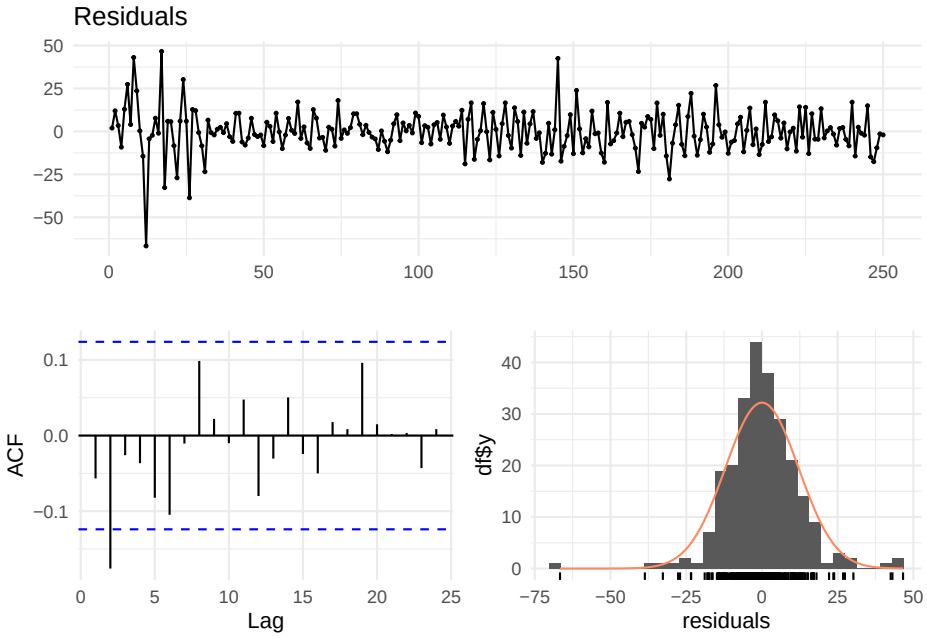
Produce a residual plot of the estimated model in previous part. Is the model adequate? Describe any outlines or influential observations, and discuss how the model could be improved.

```
checkresiduals(nodiff.model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 77.864, df = 10, p-value = 1.315e-12
```

```
checkresiduals(diff.model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 21.503, df = 10, p-value = 0.01785
```

Residual plot for the model with no differencing has a few features indicating bad fit: significant lags in the ACF plot, visible pattern in the residual graph. Diagnostic plots for the model built on differenced data are free from these issues, but the residuals have few very strong outliers. Explanatory power of the latter model is questionable, because the bulk of the data shoes no correlation and the model hinges on just 6 points. It should be possible to improve the model by smoothing the data over 2 or 3 days and/or introducing ARMA terms in the regression.

3.5 Forcasting model

Use a model to forecast the electricity demand (with **prediction** intervals) that you would expect for the next day if the maximum temperature was 15°. Compare this with the forecast if the with maximum temperature was 35°. Do you believe these forecasts? Why or why not?

```
# Make df with the future data
dt.15 <- data.frame(dev = 8, dev_sign = as.factor(0), my_day = 366)
dt.35 <- data.frame(dev = 12, dev_sign = as.factor(1), my_day = 366)
```

```
# Do forecasts with two models
pred_15 <- predict.lm(nodiff.model, dt.15,
                      interval = "prediction", level = 0.95)

pred_35 <- predict.lm(nodiff.model, dt.35,
                      interval = "prediction", level = 0.95)
```

The model taking into account sign and the magnitude of deviation from the comfortable temperature makes reasonable predictions in both cases. It suggests that if the next working day (that is, day 366 on the current series) is going to be 15C, electricity demand is estimated to be 245 with 95% CI from 221 to 269. In case the temperature jumps up to 35C, the demand is estimated to be 268 with 95% CI from 244 to 292. These estimates are well within the range of the values observed in the past.

Chapter 4

(12 points) Vector autoregression

```
# For estimation, lag selection, diagnostic testing, forecasting, and impulse response functions
library(vars)
```

Annual values for real mortgage credit (RMC), real consumer credit (RCC) and real disposable personal income (RDPI) for the period 1946-2006 are recorded in `./data/mortgage_credit.csv`.

All of the observations are measured in billions of dollars, after adjustment by the Consumer Price Index (CPI).

Our goal is to develop a VAR model for these data for the period 1946-2003, and then forecast the last three years, 2004-2006.

```
credit <- read_csv('./data/mortgage_credit.csv')
cred_ts <- as_tsibble(credit, index = Year)
#glimpse(credit)
cred_ts <- credit %>% as_tsibble(index = "Year")
```

4.1 Time series plot

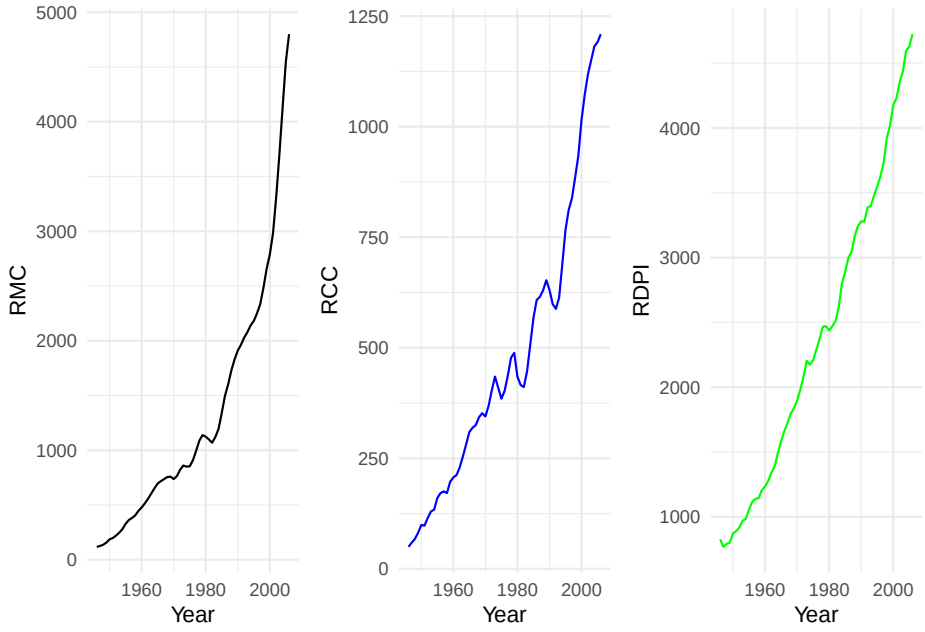
Plot the time-series of real mortgage credit (RMC), real consumer credit (RCC) and real disposable personal income (RDPI)? Do they look stationary?

```
rmc_plot <- ggplot(data = cred_ts) +
  geom_line(aes(x = Year, y = RMC), color = "black")

rcc_plot <-  ggplot(data = cred_ts) +
  geom_line(aes(x = Year, y = RCC), color = "blue")
```

```
rdpi_plot <- ggplot(data = cred_ts) +
  geom_line(aes(x = Year, y = RDPI), color = "green")

rmc_plot|rcc_plot|rdpi_plot
```

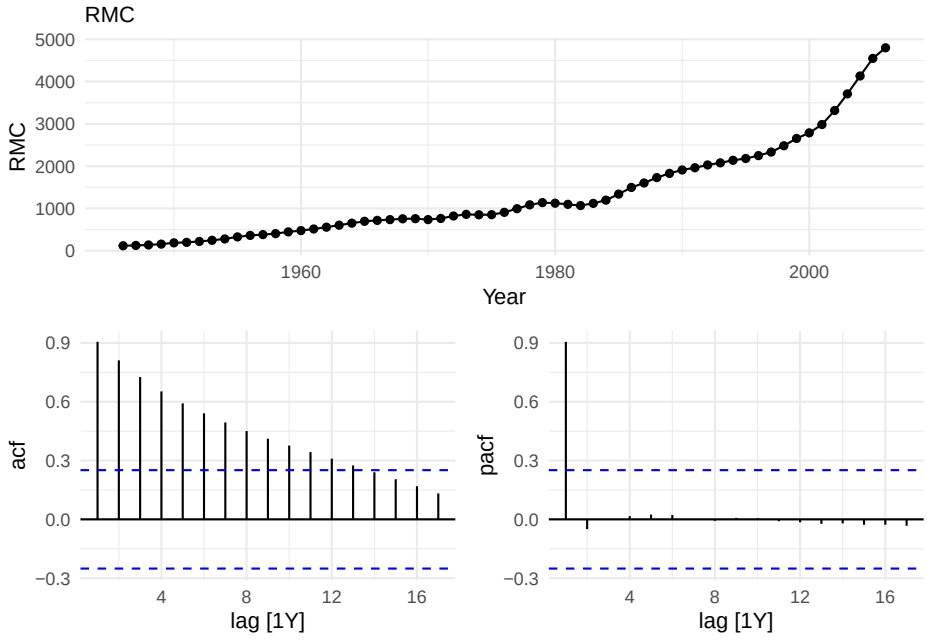


The series have clear trend and variable variance - they do not look stationary at all.

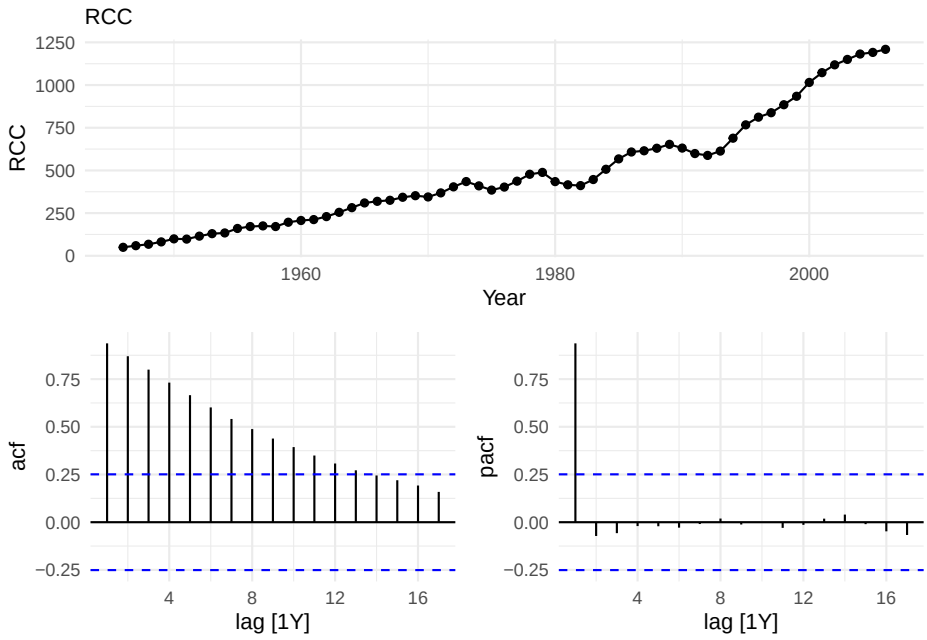
4.2 Check for the unit root

Plot ACF/PACF and Perform the unit root test on these variables and report the results. Do you reject the null of unit root for them? Is the first differencing necessary?

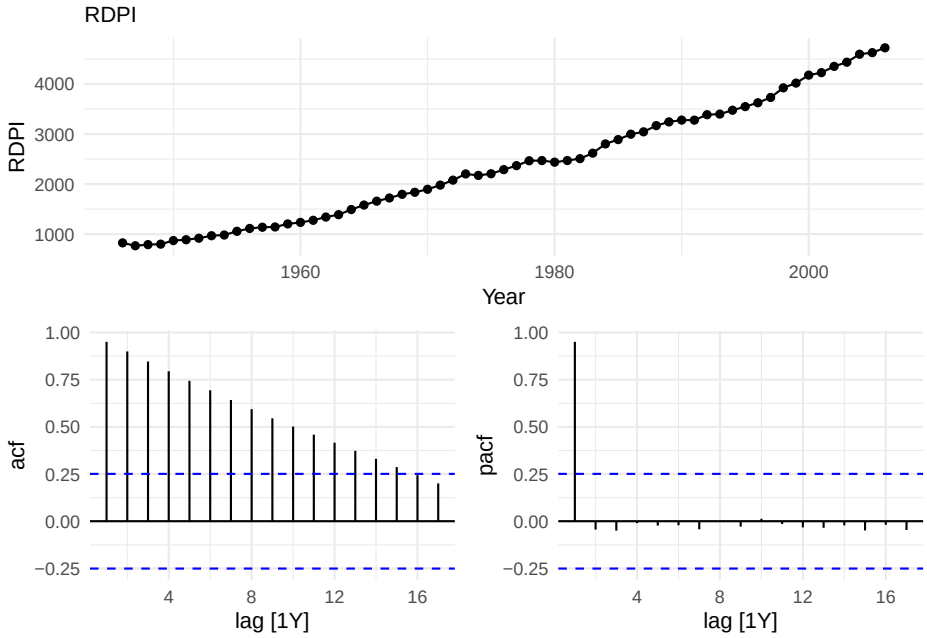
```
cred_ts %>% gg_tsdisplay(RMC, plot_type="partial") +labs(subtitle = "RMC")
```




```
cred_ts %>% gg_tsdisplay(RCC, plot_type="partial") +labs(subtitle = "RCC")
```




```
cred_ts %>% gg_tsdisplay(RDPI, plot_type="partial") +labs(subtitle = "RDPI")
```




```
RMC_kpss <- cred_ts %>% features(RMC, unitroot_kpss)
RCC_kpss <- cred_ts %>% features(RCC, unitroot_kpss)
RDPI_kpss <- cred_ts %>% features(RDPI, unitroot_kpss)
```

The ACF of all three series decay slowly, and The first lag of their PACF is 1 in all three cases, which suggests non-stationarity. The KPSS tests for RMC, RCC and RDPI reject the null hypothesis with the p values of 0.01, es should have a unit root. The KPSS tests for RMC, RCC and RDPI reject the null hypothesis with the p values of 0.01,0.01 and 0.01 respectively, suggesting that the series are not stationary.

d) Now calculate the first differences of the series

```
cred_diff <- cred_ts %>%
mutate(diff_RMC = difference(RMC), diff_RCC = difference(RCC), diff_RDPI = difference(
RMC_kpss_diff <- cred_diff %>% features(diff_RMC, unitroot_kpss)
RCC_kpss_diff <- cred_diff %>% features(diff_RCC, unitroot_kpss)
RDPI_kpss_diff <- cred_diff %>% features(diff_RDPI, unitroot_kpss)
```

4.3 Determine VAR model

Based on the unit root results transform the variables and determine the lag length of the VAR using the information criteria.

```
df_to_test <- cred_diff %>%
dplyr::select(diff_RMC,diff_RCC,diff_RDPI) %>% na.omit()
```

```
VARselect(df_to_test, lag.max = 4, type="none")
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      2      2      2      2
##
## $criteria
##                  1             2             3             4
## AIC(n)    10.50908 -3.560311e+01 -3.532259e+01 -3.501325e+01
## HQ(n)     10.73343 -3.515441e+01 -3.464954e+01 -3.411585e+01
## SC(n)     11.08775 -3.444576e+01 -3.358657e+01 -3.269856e+01
## FPE(n) 36682.44332  3.476850e-16  4.691545e-16  6.642861e-16
```

All criteria agree on VAR(3)

4.4 Estimation

Estimate the selected VAR in previous part and comment on the results.

```
# var_diff = vars::VAR(df_to_test, p = 3, type = "none")
# summary(var_diff)
```

In the first equation, none of the coefficients are statistically significant in 5% for change in bitcoin regression. R-squared is small, and only 6.9 percent of the variations of change in bitcoin prices can be explained by the lagged change in google search volume and lagged change in bitcoin prices. In the second equation, for change in google trend, The estimated VAR model suggests that the past values of change in bitcoin prices have explanatory power for current values of change in google trend. However, we find that only lag one and three of the bitcoin prices is significant at a 10% percent significance level. So apparently, large fluctuations in bitcoin prices lead to higher attention to the bitcoin and higher google search volume. Also, 21 percent of the variations of change in google search volume can be explained by the lag of change in google search volume and the lag of change in bitcoin prices. So, based on these results, we can conclude that higher bitcoin prices could predict higher google search volume, but not the other way around.

4.5 Model diagnostic

Do diagnostic checking of the VAR model.

```
# roots(var_diff)

# var_diff_test<- serial.test(var_diff, lags.pt = 12)
# var_diff_test
```

First, we need to check if the estimated VAR(3) is a stable process, and we will need to check if the eigenvalues of the companion matrix are all less than one. Here, since here all eigenvalues are less than 1, VAR(3) is a stable process. Then, we need to check for tests for autocorrelation in residuals. The serial.test() computes the multivariate Portmanteau- for serial correlation. Based on the test results, the null hypothesis of no autocorrelation is not rejected since the p-value is 0.4868.

4.6 Forecasting

forecast the last three years, 2004-2006.

```
#var_diff = vars::VAR(as.ts(df1), p = 3, type = "none")
# forecast(var_diff) %>% autoplot() + xlab("Weeks")
```

Both changes in log of bitcoin prices and google trend search revert to their means, which is onsistent with stationarity.