Libraries

```
library(forecast)
library(ggplot2)
library(seasonal)
library(fma)
library(mlbench)
library(corrplot)
library(caret)
library(e1071)
library(mice)
```

HA 2.1

Question

Use the help function to explore what the series gold, woolyrnq and gas represent.

a) Use autoplot() to plot each of these in separate plots.
b) What is the frequency of each series? Hint: apply the frequency() function.
c) Use which.max() to spot the outlier in the gold series. Which observation was it?

Code

```
# Part A
autoplot(gold)
autoplot(woolyrnq)
autoplot(gas)

# Part B
frequency(gold)
frequency(woolyrnq)
frequency(gas)

# Part C
which.max(gold)
```

Response

(a) Completed in code above. These are time series datasets plotted using autoplot.

(b) Each of the datasets in question are time series. Gold has a frequency of 1 (annual), woolyrnq has a frequency of 4 (quarterly), and gas has a frequency of 12 (monthly).

(c) The outlier for the gold dataset is at 770.

HA 2.3

Question

Download some monthly Australian retail data from the book website. These represent retail sales in various categories for different Australian states, and are stored in a MS-Excel file.

Select one of the time series as follows (but replace the column name with your own chosen column). Explore your chosen retail time series using the following functions:

autoplot(), ggseasonplot(), ggsubseriesplot(), gglagplot(), ggAcf()

Can you spot any seasonality, cyclicity and trend? What do you learn about the series?

Code

```
retaildata <- readxl::read_excel("retail.xlsx", skip=1)
myts <- ts(retaildata[,"A3349335T"], frequency=12, start=c(1982,4))
autoplot(myts)
ggseasonplot(myts)
ggsubseriesplot(myts)
gglagplot(myts)
ggAcf(myts)
```

Response

From looking at the plot, there appears to be a clear positive trend as well as increasing seasonality. The trend is also clear by looking at the autocorrelation plot where you can see large values early on that slowly decrease. By looking at the seasonality plot and the season sub-series plot, it becomes evident that the seasonality I saw at the beginning is in fact a cyclic nature not related to the calendar. This is again evidenced by the lack of scalloping in the autocorrelation plot.

HW 6.2

Question

The plastics data set consists of the monthly sales (in thousands) of product A for a plastics manufacturer for five years.

a) Plot the time series of sales of product A. Can you identify seasonal fluctuations and/or a trend-cycle?
b) Use a classical multiplicative decomposition to calculate the trend-cycle and seasonal indices.
c) Do the results support the graphical interpretation from part a?
d) Compute and plot the seasonally adjusted data.
e) Change one observation to be an outlier (e.g., add 500 to one observation), and recompute the seasonally adjusted data. What is the effect of the outlier?

f) Does it make any difference if the outlier is near the end rather than in the middle of the time series?

Code

```
# Part A
p <- plastics
autoplot(p)

# Part B -- use classical multiplicative decomposition on the
plastics time series
m1 <- decompose(p, type = "multiplicative")
m1$seasonal
m1$trend
m1$random

# Part C
autoplot(m1$seasonal)
autoplot(m1$trend)

# Part D -- Compute and plot the seasonally adjusted data.
adj1 <- p/m1$seasonal
autoplot(adj1)

# Part E
p[30] <- p[30] + 500
m2 <- decompose(p, type = "multiplicative")
adj2 <- p/m2$seasonal
autoplot(adj2)
autoplot(m2$trend)
autoplot(m2$random)

# Part F
p[30] <- p[30] - 500
p[1] <- p[1] + 500
m3 <- decompose(p, type = "multiplicative")
adj3 <- p/m3$seasonal
autoplot(adj3)
autoplot(m3$trend)
autoplot(m3$random)
```

Response

(a) Based on the initial plot, it appears that there is a seasonal nature to the plastics data set. It appears that the seasonality is annual. There also appears to be a strong positive trend-cycle.

(b) Completed in code above.

(c) When just the trend portion of the decomposition is plotted it does show a strong positive graph and when just the seasonal portion of the decomposition is plotted there is a clear annual seasonality. This is in agreement with my graphical interpretation.

(d) Completed in code above.

(e) When a data point in the middle of the data set is changed to be an extreme outlier, the decomposition is impacted. This is because the seasonal adjustment doesn't remove outliers in a classic multiplicative decomposition.

(f) Based on the plots that I have, the location of the outlier does not appear to make a large difference. The largest difference is that an outlier in the middle seems to have a larger impact on the trend-cycle portion of the decomposition.

KJ 3.1

Question

The UC Irvine Machine Learning Repository6 contains a data set related to glass identification. The data consist of 214 glass samples labeled as one of seven class categories. There are nine predictors, including the refractive index and percentages of eight elements: Na, Mg, Al, Si, K, Ca, Ba, and Fe.

a) Using visualizations, explore the predictor variables to understand their distributions as well as the relationships between predictors.
b) Do there appear to be any outliers in the data? Are any predictors skewed?
c) Are there any relevant transformations of one or more predictors that might improve the classification model?

Code

```
# Load the data from the mlbench package
data(Glass)

# Part A -- Create a histogram for each predictor variable
par(mfrow = c(3,3))
for(i in 1:9) {
  hist(Glass[,i], main = names(Glass)[i], breaks = 20)
}

# Part A (continued) -- View correlation between predictor variables
corrplot(cor(Glass[1:9]), order = "hclust")

# Part B -- Check the skew of each predictor variable using the e1071
package
skewValues <- apply(Glass[1:9], 2, skewness)

# Part C -- Center, Scale, and Box-Cox tranform the predictor
variables
```

```
trans <- preProcess(Glass,
                    method = c("BoxCox", "center", "scale"))
```

Response

(a) Histograms were used to view the distribution of each predictor and a correlation plot was used to view the between-predictor correlation.

(b) To check for skewness, the e1071 package has a skewness function. Based on that function, it's easy to see that RI, Mg, K, Ca, Ba, and Fe are all skewed predictors. Additionally, Al and Si are both slightly skewed. Using the histograms from part (a), we can see that for K, Ba, and Fe, most of the data is near-zero with a few outliers at higher ranges, far away from the rest of the data.

(c) There is a 2 order of magnitude difference between Si and Fe/Ba, which indicates centering and scaling would be an appropriate transformation for this dataset. Additionally, RI, Na, Al, Si, and Ca are all suitable for a Box-Cox transformation.

KJ 3.2

Question

The soybean data can also be found at the UC Irvine Machine Learning Repository. Data were collected to predict disease in 683 soybeans. The 35 predictors are mostly categorical and include information on the environmental conditions (e.g., temperature, precipitation) and plant conditions (e.g., left spots, mold growth). The outcome labels consist of 19 distinct classes.

a) Investigate the frequency distributions for the categorical predictors. Are any of the distributions degenerate in the ways discussed earlier in this chapter?
b) Roughly 18% of the data are missing. Are there particular predictors that are more likely to be missing? Is the pattern of missing data related to the classes?
c) Develop a strategy for handling missing data, either by eliminating predictors or imputation.

Code

```
# Load the data from the mlbench package
data(Soybean)

# Part A -- Use caret to find the near zero variance predictors
nzv <- nearZeroVar(Soybean, saveMetrics = TRUE)
nzv[nzv[,"nzv"] == TRUE,]

# Part B -- Check the NA % for the Soybean data
sum(is.na(Soybean))/prod(dim(Soybean))

# Part B (continued) -- Get the NA % for each predictor in the
Soybean data
sort(sapply(Soybean, function(x) sum(is.na(x))/length(x)), decreasing
= TRUE)
```

```
# Part B (continued) -- Get the NA % for each category of soybean
from the Soybean data
na_by_category <- data.frame(matrix(ncol = 3, nrow = 0))
for(category in unique(Soybean$Class)){
  na_perc <- round(sum(is.na(Soybean[which(Soybean$Class ==
category),]))/prod(dim(Soybean[which(Soybean$Class ==
category),])),2)
  observations <- prod(dim(Soybean[which(Soybean$Class ==
category),]))
  new_category <- c(category, na_perc, observations)
  na_by_category <- rbind(na_by_category, new_category)
}
colnames(na_by_category) <- c("Category", "NA Percent",
"Observations")

# Part C --
Soybean <- Soybean[,-19]

set.seed(12345)
init = mice(Soybean, maxit = 0)
imputed = mice(Soybean,
               method = init$method,
               predictorMatrix = init$predictorMatrix,
               m = 5)
Soybean_complete <- complete(imputed
```

Response

(a) There are two conditions for degenerate distributions, listed here:

• The fraction of unique values over the sample
size is low (say 10%).

• The ratio of the frequency of the most prevalent value to the frequency of the second most prevalent value is large (say around 20).

Using the caret package to find any predictors with near zero variance, it appears that leaf.mild, myselium, and sclerotia fit the bill.

(b) The book claims that 18% of the data is missing, however, when I look at the data, I only see about 9% missing. I will use this 9% value for my analysis. With this in mind, you can see that there is a rather clear dichotomy in missing data. About half of the predictors are missing above 12% of thier data while the other half is missing less that 6%. To determine if the NAs were category dependent, I looped through each category and obtained the NA percentage. After looking at each category in the data, it appears that there are 5 categories of soybean that contain all of the NA data. These categories are phytophthora-rot, diaporthe-pod-&-stem-blight, cyst-nematode, 2-4-d-injury, and herbicide-injury.

(c) To deal with missing values in this dataset, I will take a couple of approaches. To start with, I will eliminate the predictor leaf.mild. A significant portion of this predictor (~16%) is missing and it is also a near zero variance predictor. With those two pieces in mind, it makes sense to toss out this predictor as a whole. The rest of the missing values are spread out enough that it doesn't make sense to throw away the data. For these missing values, I will use imputation to predict appropriate values. In order to do this imputation, I use the mice package...

HA 7.1

Question

Consider the pigs series — the number of pigs slaughtered in Victoria each month.

a) Use the ses() function in R to find the optimal values of α and ℓ, and generate forecasts for the next four months.
b) Compute a 95% prediction interval for the first forecast using y ± 1.96s where s is the standard deviation of the residuals. Compare your interval with the interval produced by R.

Code

```
# Part A
fc <- ses(pigs, h = 4)
fc$model

# Part B
my_ub <- fc$mean[1]+(1.96*sd(fc$residuals))
my_lb <- fc$mean[1]-(1.96*sd(fc$residuals))

model_ub <- fc$upper[5]
model_lb <- fc$lower[5]
```

Response

(a) The ses model foudn the optimal value for α and ℓ to be 0.2971 and 77260.0561 respectively.

(b) When I generate a 95% confidence interval for the first predicted month from the ses model, I get [78679.97, 118952.8]. The interval produced by the ses model itself is [78611.97, 119020.8]. The interval I calculated is slightly narrower than the one calculated by the ses model itself.

HA 7.2

Question

Write your own function to implement simple exponential smoothing. The function should take arguments y (the time series), alpha (the smoothing parameter α) and level (the initial level ℓ). It

should return the forecast of the next observation in the series. Does it give the same forecast as ses()?

Code

```
simple_es <- function(y, alpha, level){
  T <- length(y)
  fc <- 0
  for(j in 1:T-1){
    x <- (alpha*((1-alpha)^j)*y[T-j])+(((1-alpha)^T)*level)
    print(j)
    fc <- fc+x
  }
  return(fc)
}

simple_es(pigs,0.2971,77260.0561)
```

Response

HA 7.3

Question

Modify your function from the previous exercise to return the sum of squared errors rather than the forecast of the next observation. Then use the optim() function to find the optimal values of α and ℓ. Do you get the same values as the ses() function?

Code

Response

HA 8.1

Question

Code

Response

HA 8.2

Question

A classic example of a non-stationary series is the daily closing IBM stock price series (data set ibmclose). Use R to plot the daily closing prices for IBM stock and the ACF and PACF. Explain how each plot shows that the series is non-stationary and should be differenced.

Code

```
ggAcf(ibmclose)
ggPacf(ibmclose)
```

Response

HA 8.6

Question

Code

Response

HA 8.8

Question

Code

Response