Data 624 Home work 2

…

11/09/2019

Read HA #6

# HW 6.2))….

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

# load the libraries  
library(fma)

## Loading required package: forecast

## Registered S3 method overwritten by 'xts':  
## method from  
## as.zoo.xts zoo

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

## Registered S3 methods overwritten by 'forecast':  
## method from   
## fitted.fracdiff fracdiff  
## residuals.fracdiff fracdiff

library(tidyverse)

## ── Attaching packages ────────────────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 3.2.0 ✔ purrr 0.3.2  
## ✔ tibble 2.1.3 ✔ dplyr 0.8.3  
## ✔ tidyr 0.8.3 ✔ stringr 1.4.0  
## ✔ readr 1.3.1 ✔ forcats 0.4.0

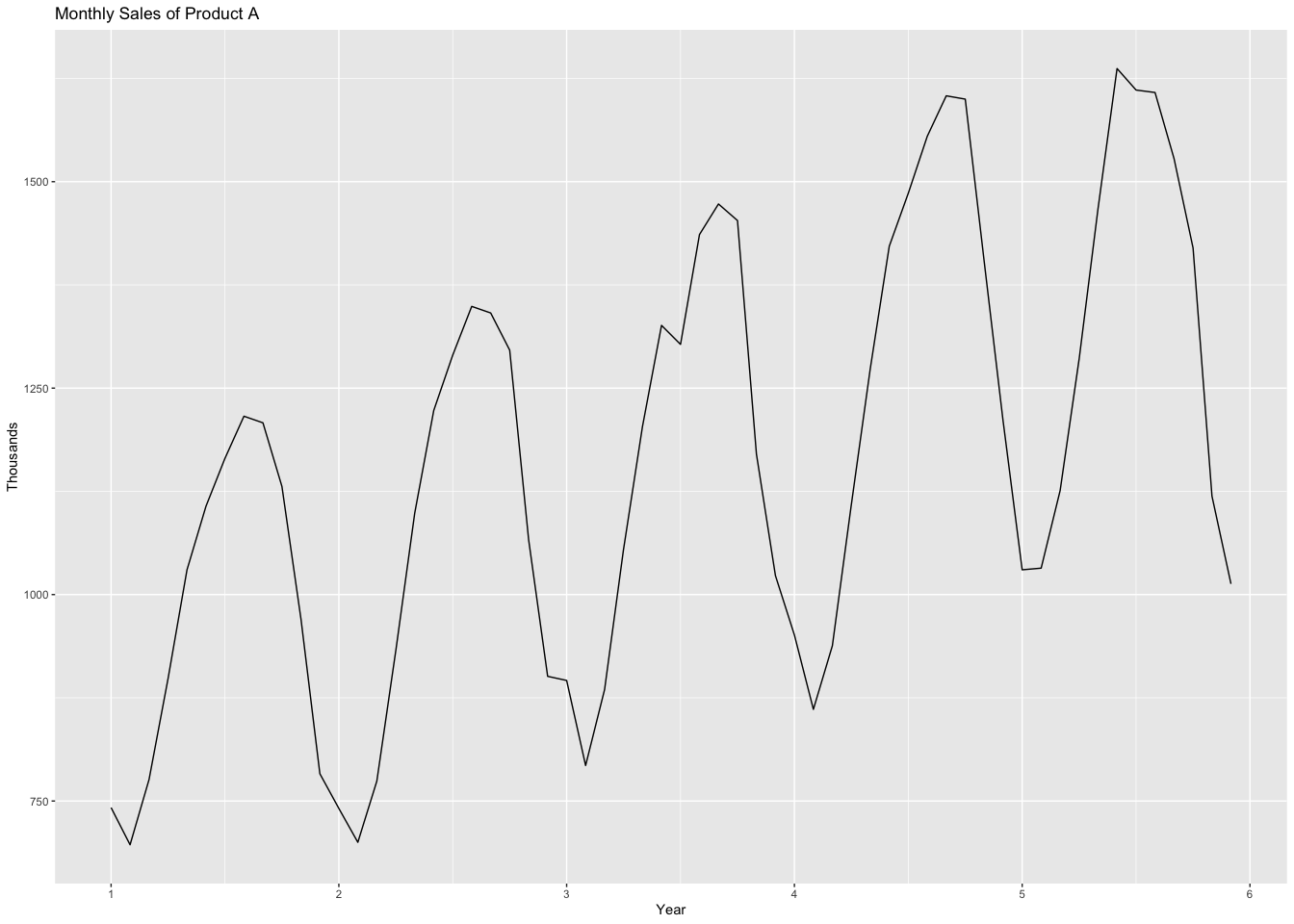
## ── Conflicts ───────────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

# load the data  
plastics <- fma::plastics  
glimpse(plastics)

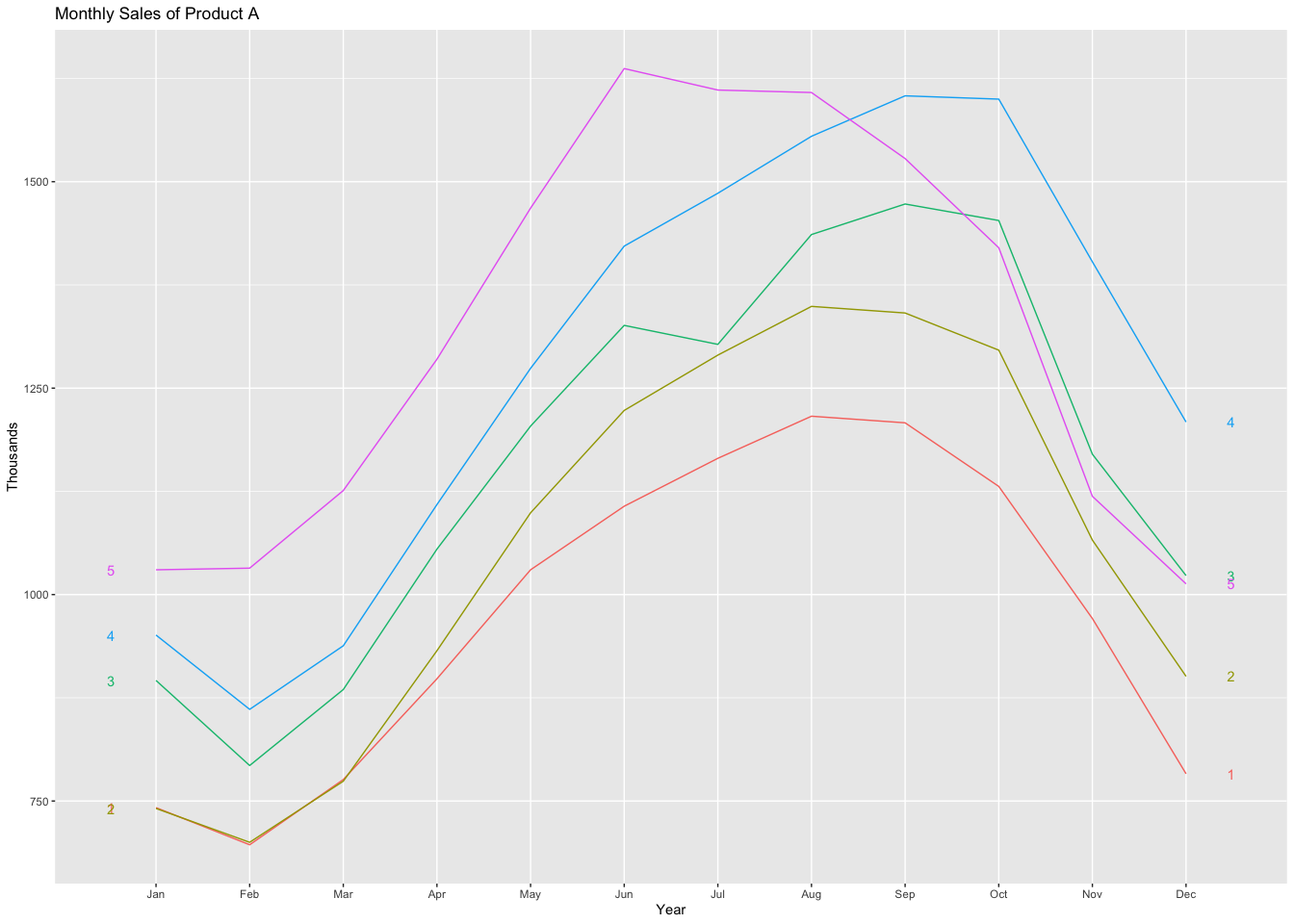
## Time-Series [1:60] from 1 to 5.92: 742 697 776 898 1030 ...

1. Plot the time series of sales of product A. Can you identify seasonal fluctuations and/or a trend-cycle?

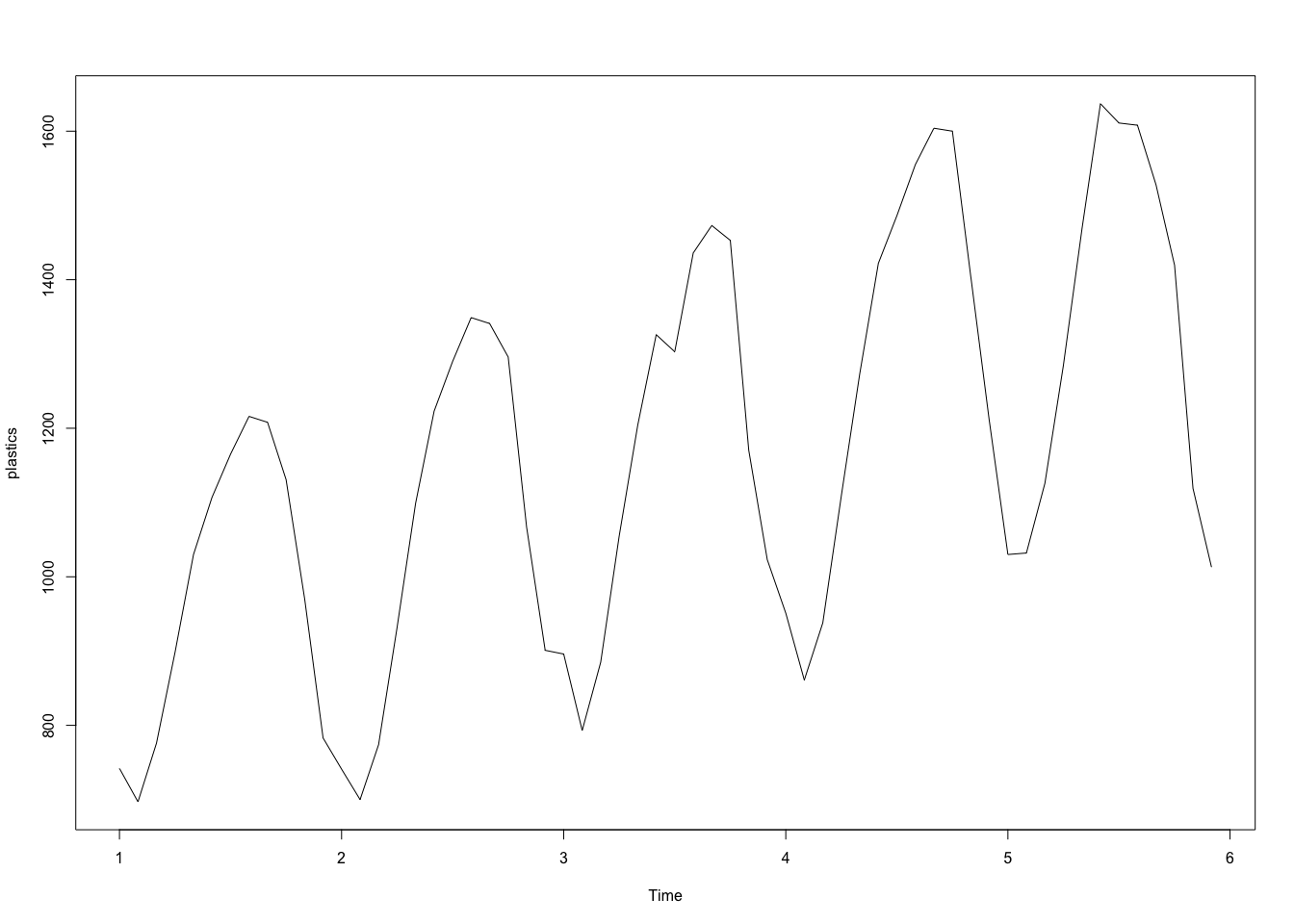
autoplot(plastics) +  
 ggtitle("Monthly Sales of Product A") +  
 xlab("Year") +  
 ylab("Thousands")

 seasonal plots

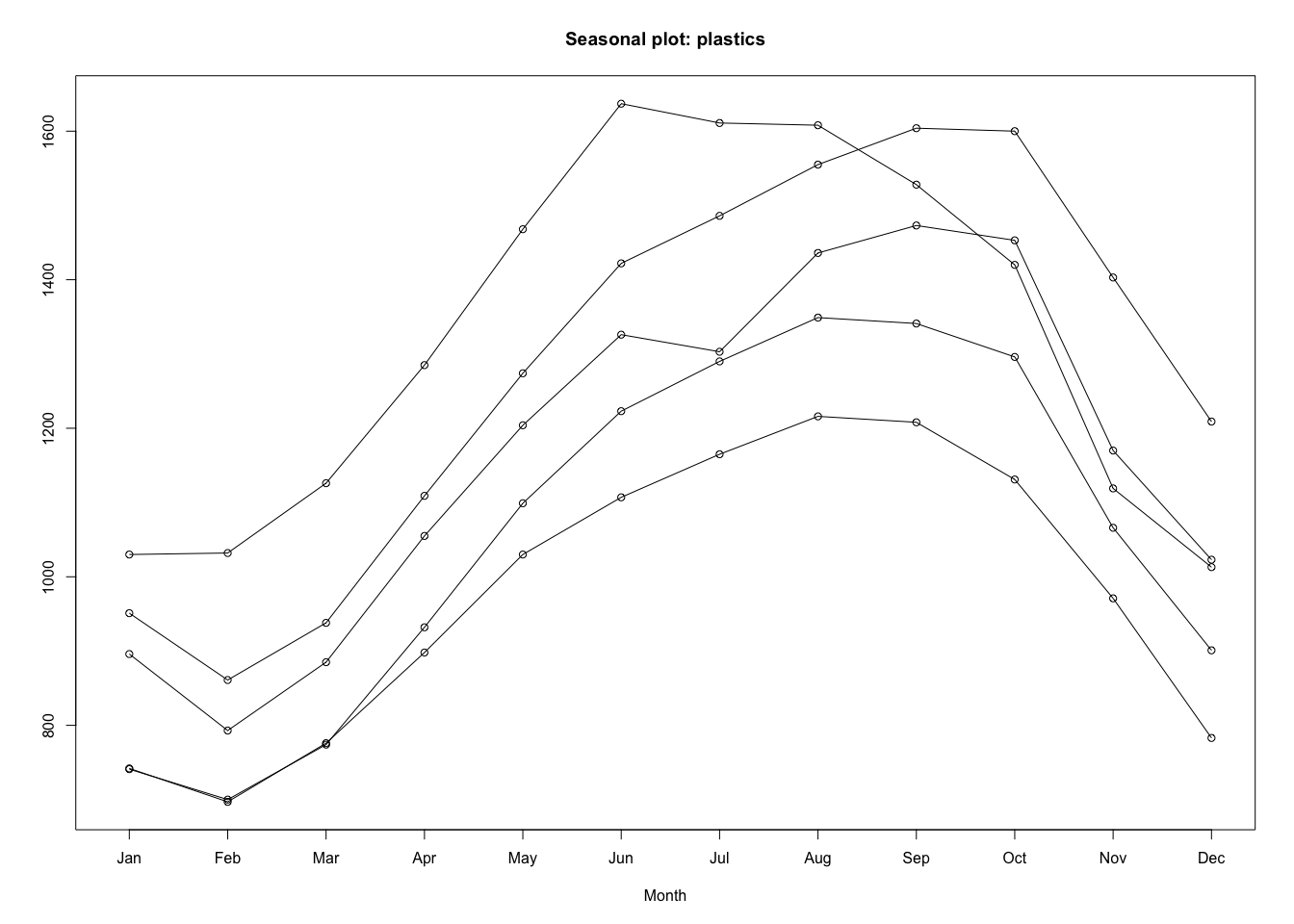
ggseasonplot(plastics, year.labels=TRUE, year.labels.left=TRUE) +  
 ggtitle("Monthly Sales of Product A") +  
 xlab("Year") +  
 ylab("Thousands")



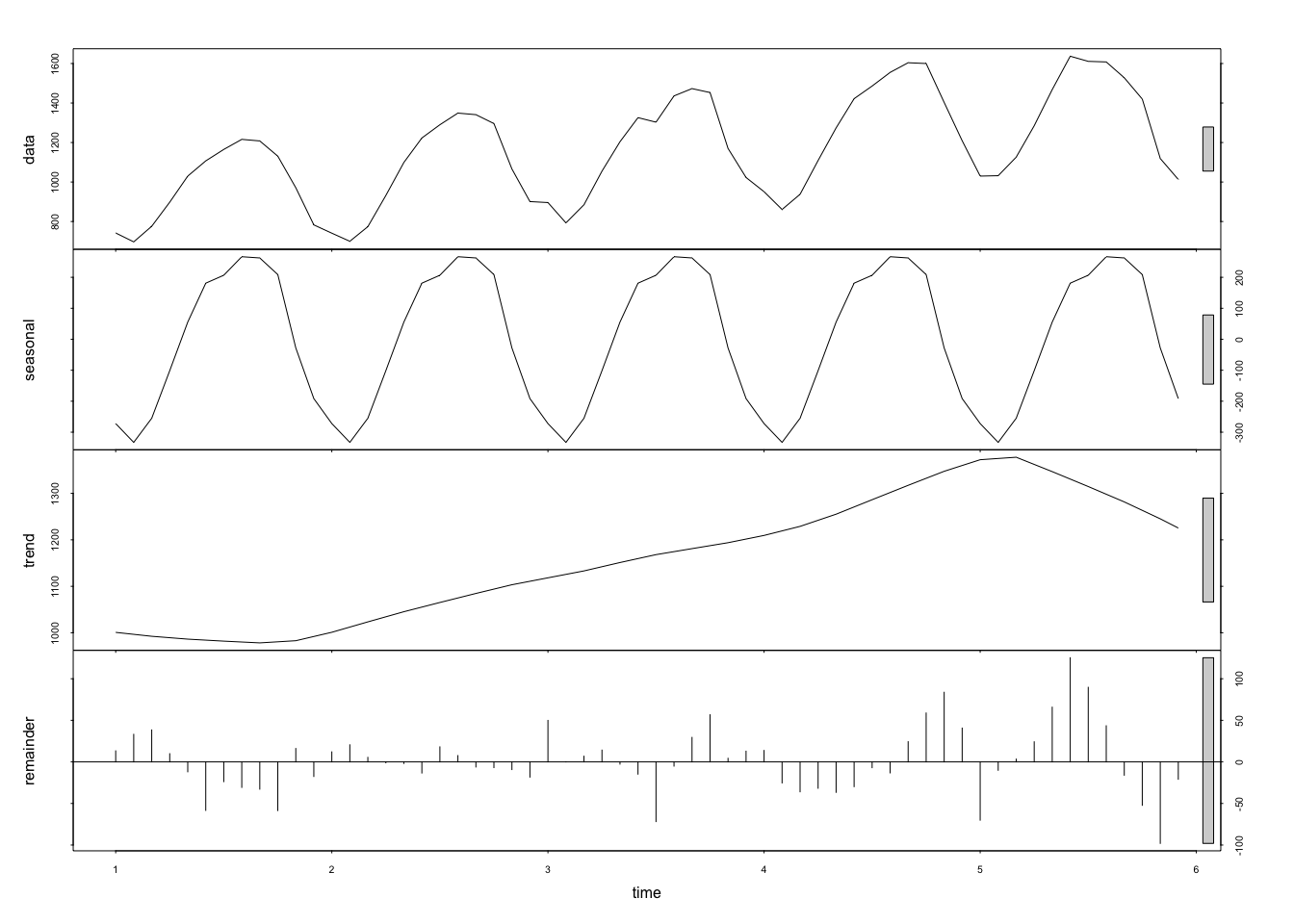
plot(plastics)



seasonplot(plastics)



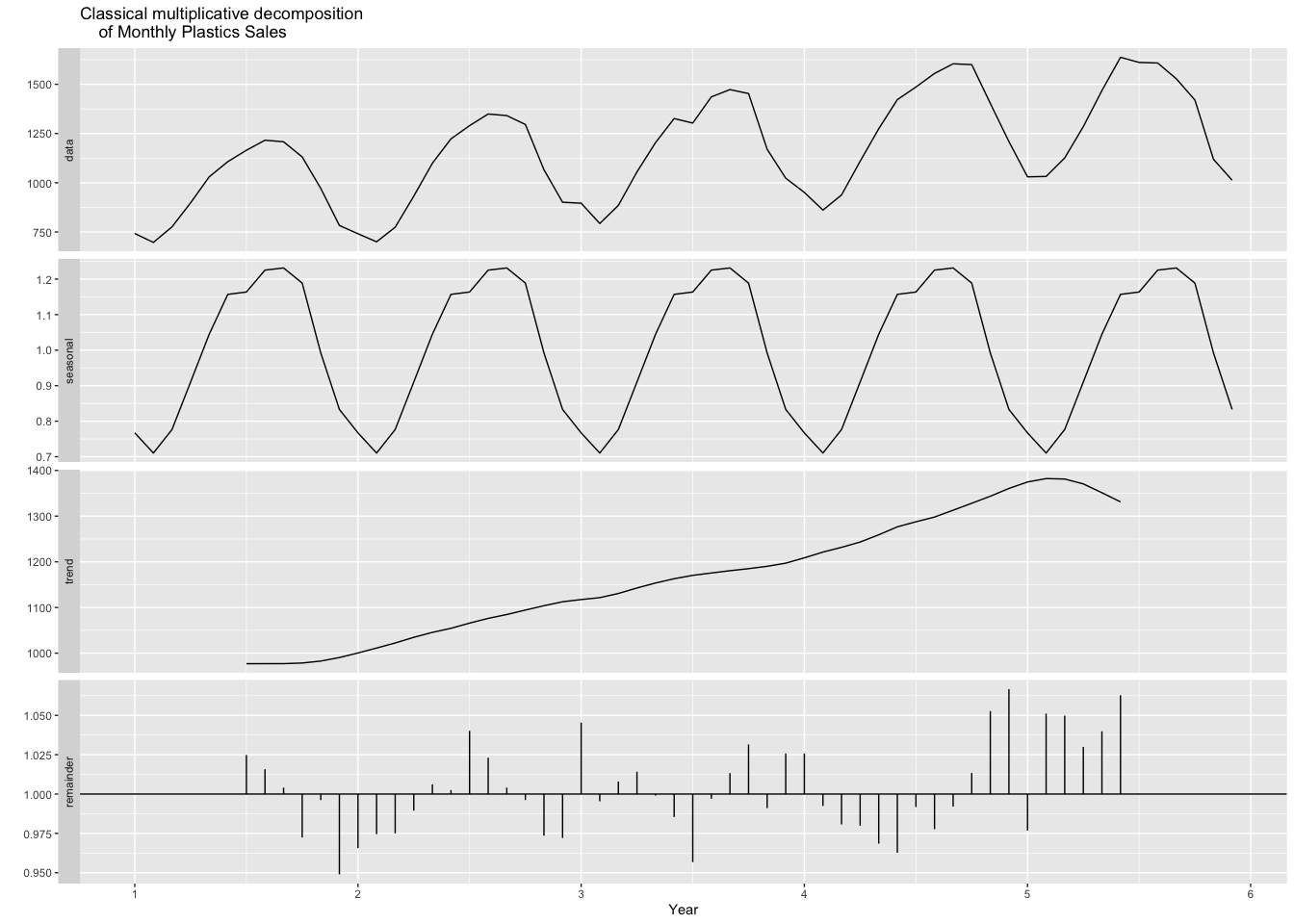
plot(stl(plastics,"periodic"))



From the plots we can identify seasonal fluctuations and increasing trend.

1. Use a classical multiplicative decomposition to calculate the trend-cycle and seasonal indices.

plastics %>% decompose(type="multiplicative") %>%  
 autoplot() + xlab("Year") +  
 ggtitle("Classical multiplicative decomposition  
 of Monthly Plastics Sales")

 Here seasonal indices m = 12

ts\_plastics = ts(plastics, frequency = 12)  
decompose\_plastics = decompose(ts\_plastics, "multiplicative")  
print("Trend and seasonal components")

## [1] "Trend and seasonal components"

decompose\_plastics$trend

## Jan Feb Mar Apr May Jun Jul  
## 1 NA NA NA NA NA NA 976.9583  
## 2 1000.4583 1011.2083 1022.2917 1034.7083 1045.5417 1054.4167 1065.7917  
## 3 1117.3750 1121.5417 1130.6667 1142.7083 1153.5833 1163.0000 1170.3750  
## 4 1208.7083 1221.2917 1231.7083 1243.2917 1259.1250 1276.5833 1287.6250  
## 5 1374.7917 1382.2083 1381.2500 1370.5833 1351.2500 1331.2500 NA  
## Aug Sep Oct Nov Dec  
## 1 977.0417 977.0833 978.4167 982.7083 990.4167  
## 2 1076.1250 1084.6250 1094.3750 1103.8750 1112.5417  
## 3 1175.5000 1180.5417 1185.0000 1190.1667 1197.0833  
## 4 1298.0417 1313.0000 1328.1667 1343.5833 1360.6250  
## 5 NA NA NA NA NA

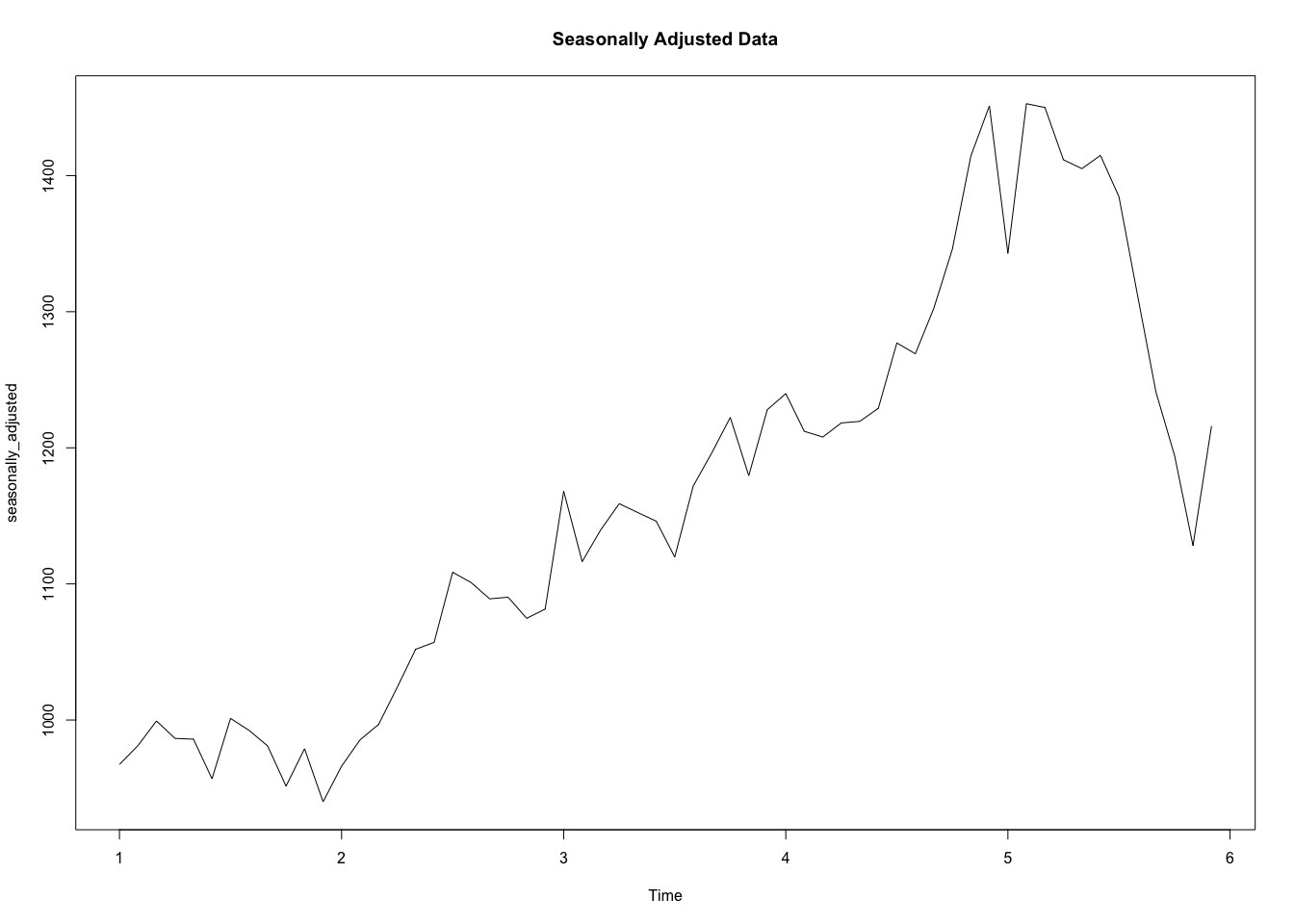
decompose\_plastics$seasonal

## Jan Feb Mar Apr May Jun Jul  
## 1 0.7670466 0.7103357 0.7765294 0.9103112 1.0447386 1.1570026 1.1636317  
## 2 0.7670466 0.7103357 0.7765294 0.9103112 1.0447386 1.1570026 1.1636317  
## 3 0.7670466 0.7103357 0.7765294 0.9103112 1.0447386 1.1570026 1.1636317  
## 4 0.7670466 0.7103357 0.7765294 0.9103112 1.0447386 1.1570026 1.1636317  
## 5 0.7670466 0.7103357 0.7765294 0.9103112 1.0447386 1.1570026 1.1636317  
## Aug Sep Oct Nov Dec  
## 1 1.2252952 1.2313635 1.1887444 0.9919176 0.8330834  
## 2 1.2252952 1.2313635 1.1887444 0.9919176 0.8330834  
## 3 1.2252952 1.2313635 1.1887444 0.9919176 0.8330834  
## 4 1.2252952 1.2313635 1.1887444 0.9919176 0.8330834  
## 5 1.2252952 1.2313635 1.1887444 0.9919176 0.8330834

1. Do the results support the graphical interpretation from part a?

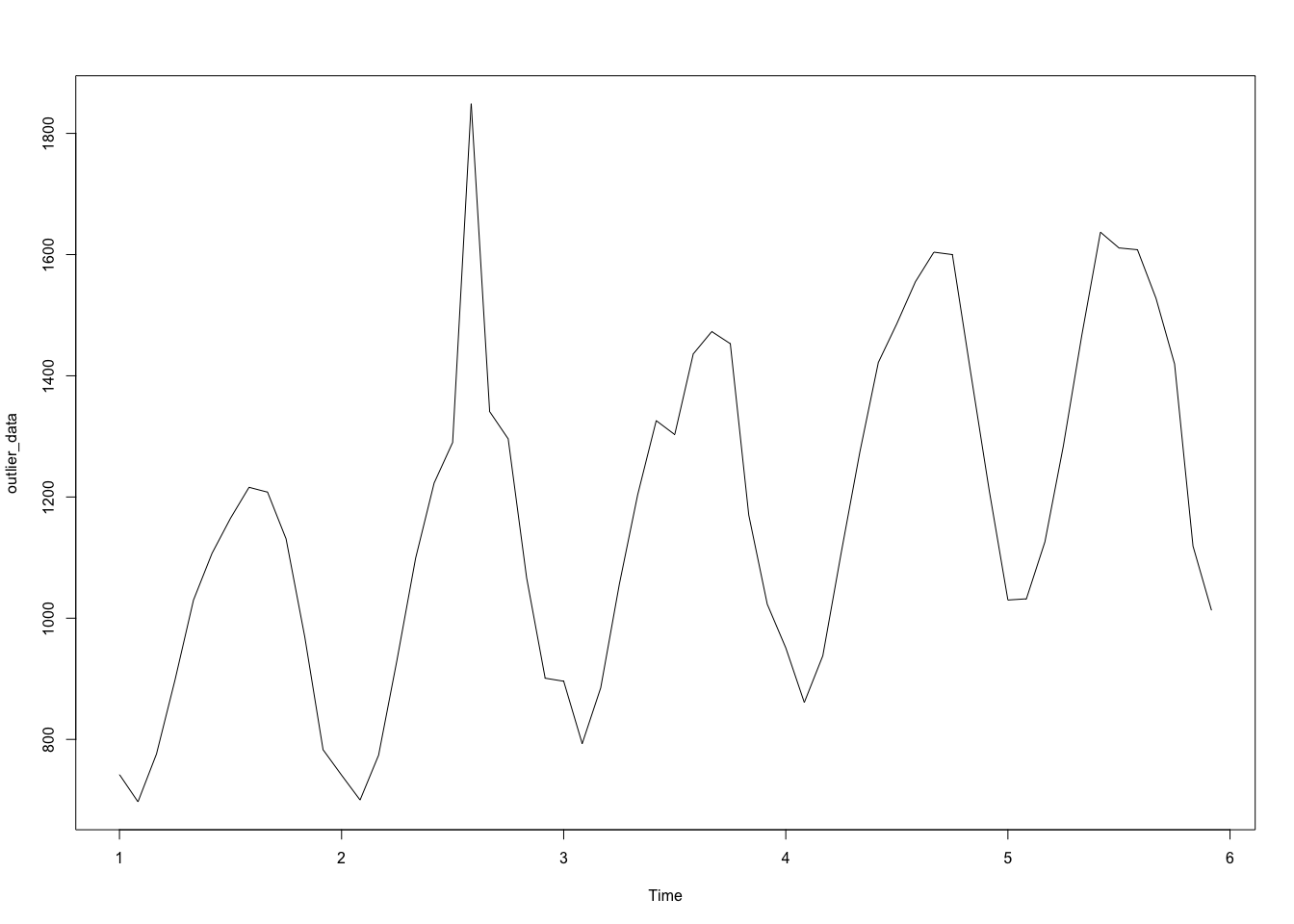
Answer: Yes. d. Compute and plot the seasonally adjusted data.

seasonally\_adjusted <- plastics/decompose\_plastics$seasonal  
plot(seasonally\_adjusted,main="Seasonally Adjusted Data")

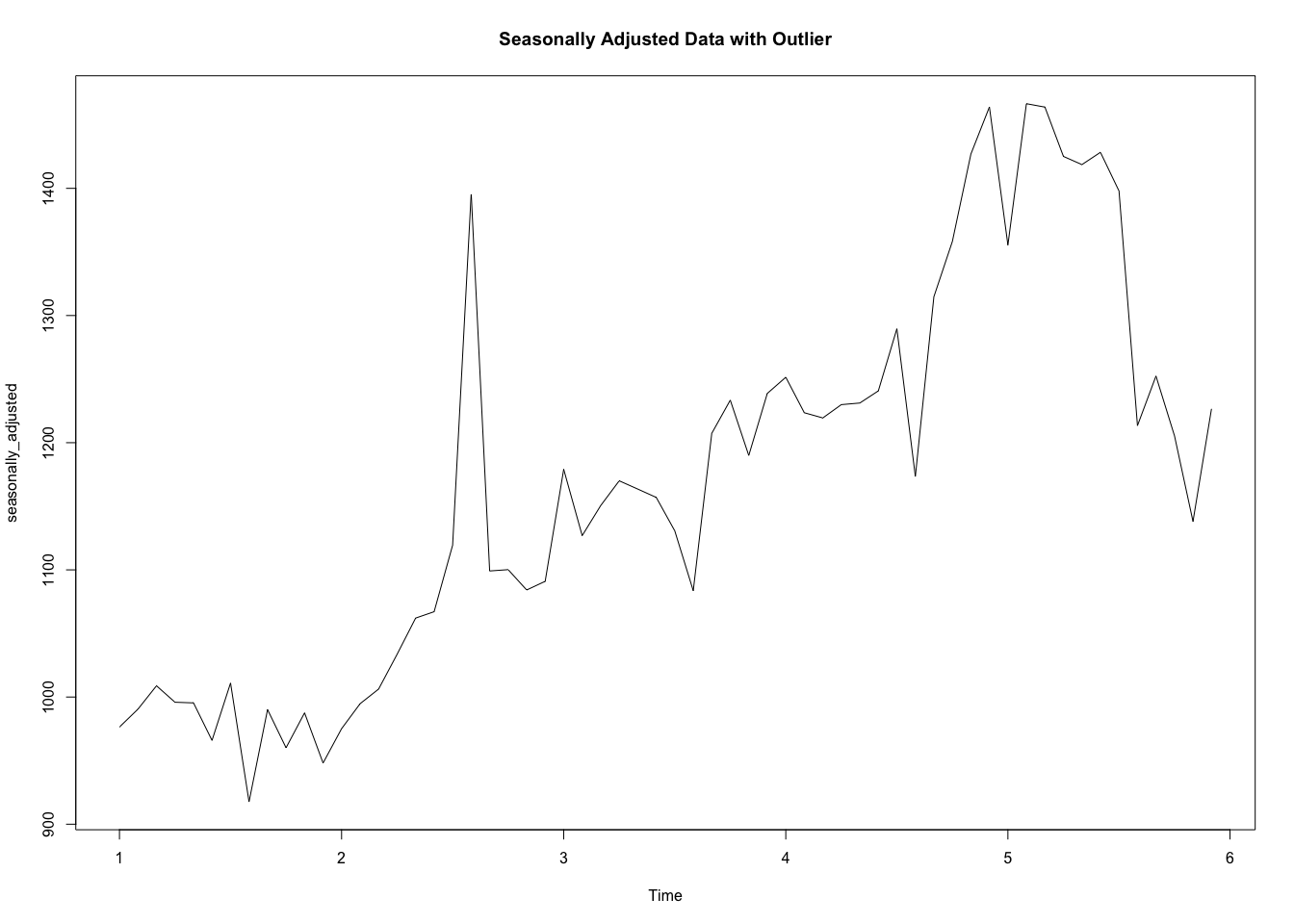


1. Change one observation to be an outlier (e.g., add 500 to one observation), and recompute the seasonally adjusted data. What is the effect of the outlier?

outlier\_data <- plastics  
# add 500 to it  
outlier\_data[20] <- outlier\_data[20] + 500  
plot(outlier\_data)



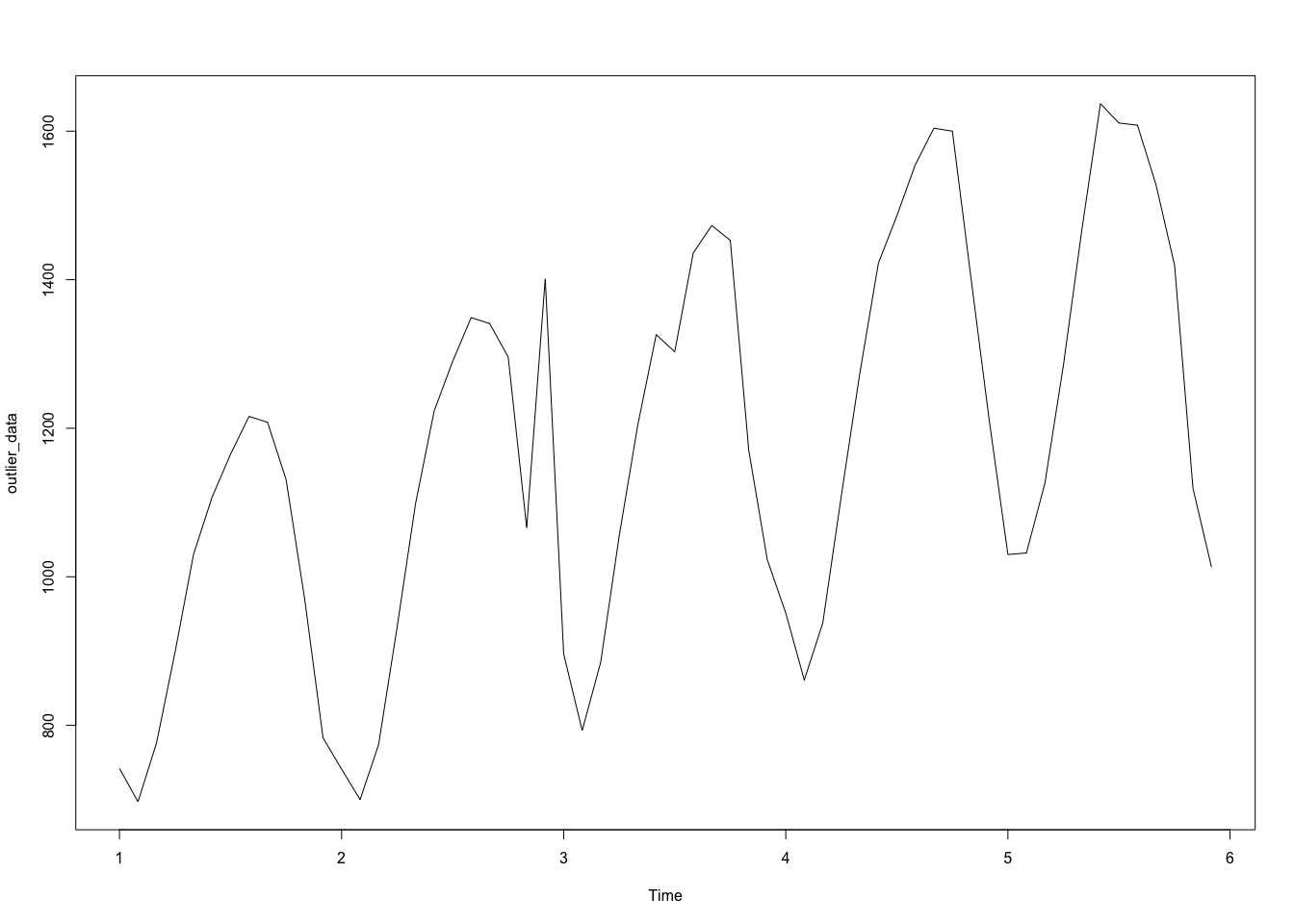
outlier\_data = ts(outlier\_data, frequency = 12)  
decompose\_outlier = decompose(outlier\_data, "multiplicative")  
seasonally\_adjusted <- outlier\_data/decompose\_outlier$seasonal  
plot(seasonally\_adjusted,main="Seasonally Adjusted Data with Outlier")



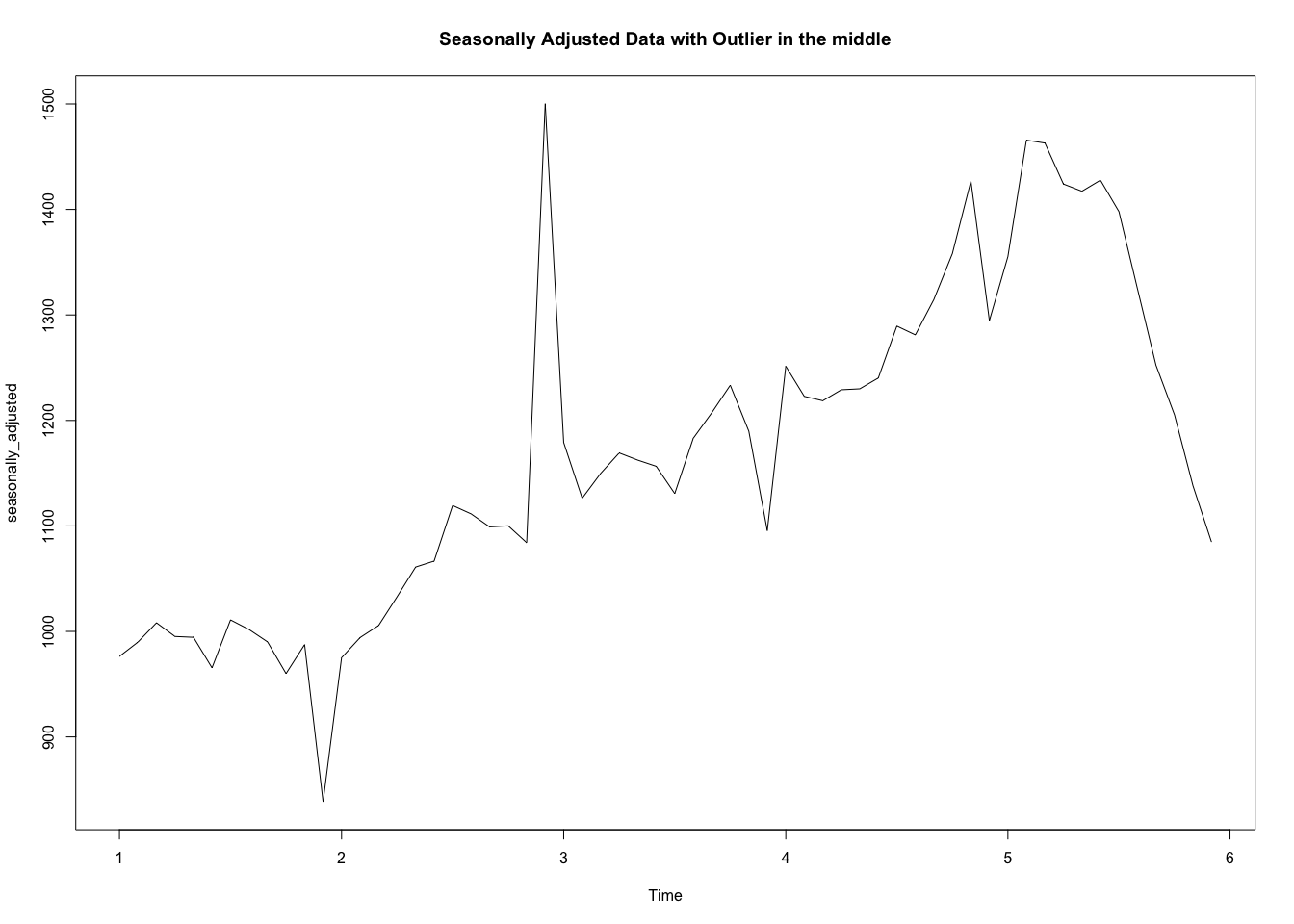
The outlier affects the trend and seasonally adjusted data.

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

outlier\_data <- plastics  
# add 500 to it  
outlier\_data[24] <- outlier\_data[24] + 500  
plot(outlier\_data)



outlier\_data = ts(outlier\_data, frequency = 12)  
decompose\_outlier = decompose(outlier\_data, "multiplicative")  
seasonally\_adjusted <- outlier\_data/decompose\_outlier$seasonal  
plot(seasonally\_adjusted,main="Seasonally Adjusted Data with Outlier in the middle")

 Yes. It makes difference if the outlier is near the end rather than in the middle of the time series.

# Read KJ #3

# HW 3.1

3.1. 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. The data can be accessed via:

library(mlbench)  
data(Glass)  
str(Glass)

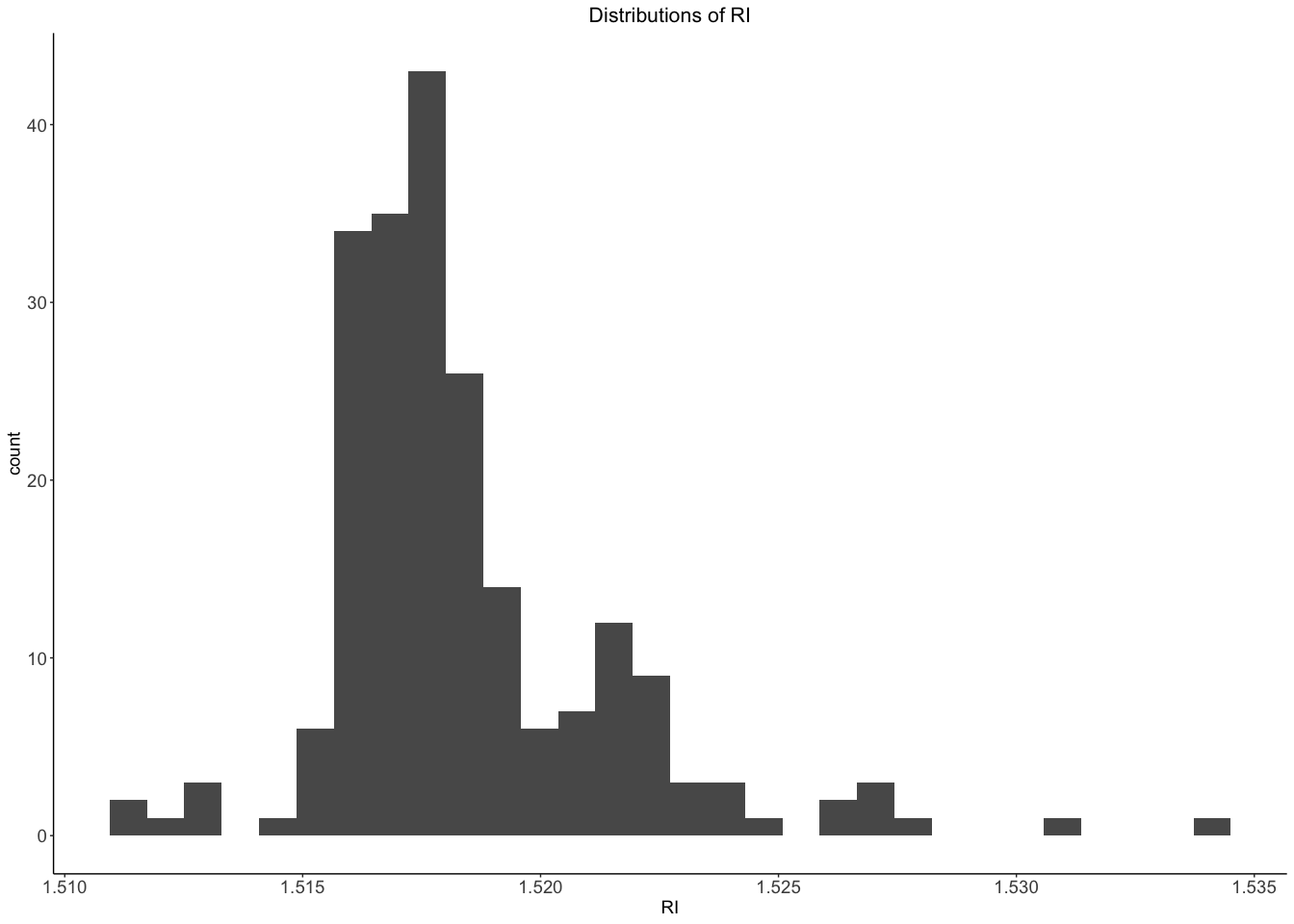
## 'data.frame': 214 obs. of 10 variables:  
## $ RI : num 1.52 1.52 1.52 1.52 1.52 ...  
## $ Na : num 13.6 13.9 13.5 13.2 13.3 ...  
## $ Mg : num 4.49 3.6 3.55 3.69 3.62 3.61 3.6 3.61 3.58 3.6 ...  
## $ Al : num 1.1 1.36 1.54 1.29 1.24 1.62 1.14 1.05 1.37 1.36 ...  
## $ Si : num 71.8 72.7 73 72.6 73.1 ...  
## $ K : num 0.06 0.48 0.39 0.57 0.55 0.64 0.58 0.57 0.56 0.57 ...  
## $ Ca : num 8.75 7.83 7.78 8.22 8.07 8.07 8.17 8.24 8.3 8.4 ...  
## $ Ba : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ Fe : num 0 0 0 0 0 0.26 0 0 0 0.11 ...  
## $ Type: Factor w/ 6 levels "1","2","3","5",..: 1 1 1 1 1 1 1 1 1 1 ...

1. Using visualizations, explore the predictor variables to understand their distributions as well as the relationships between predictors.

### Histogram of predictors

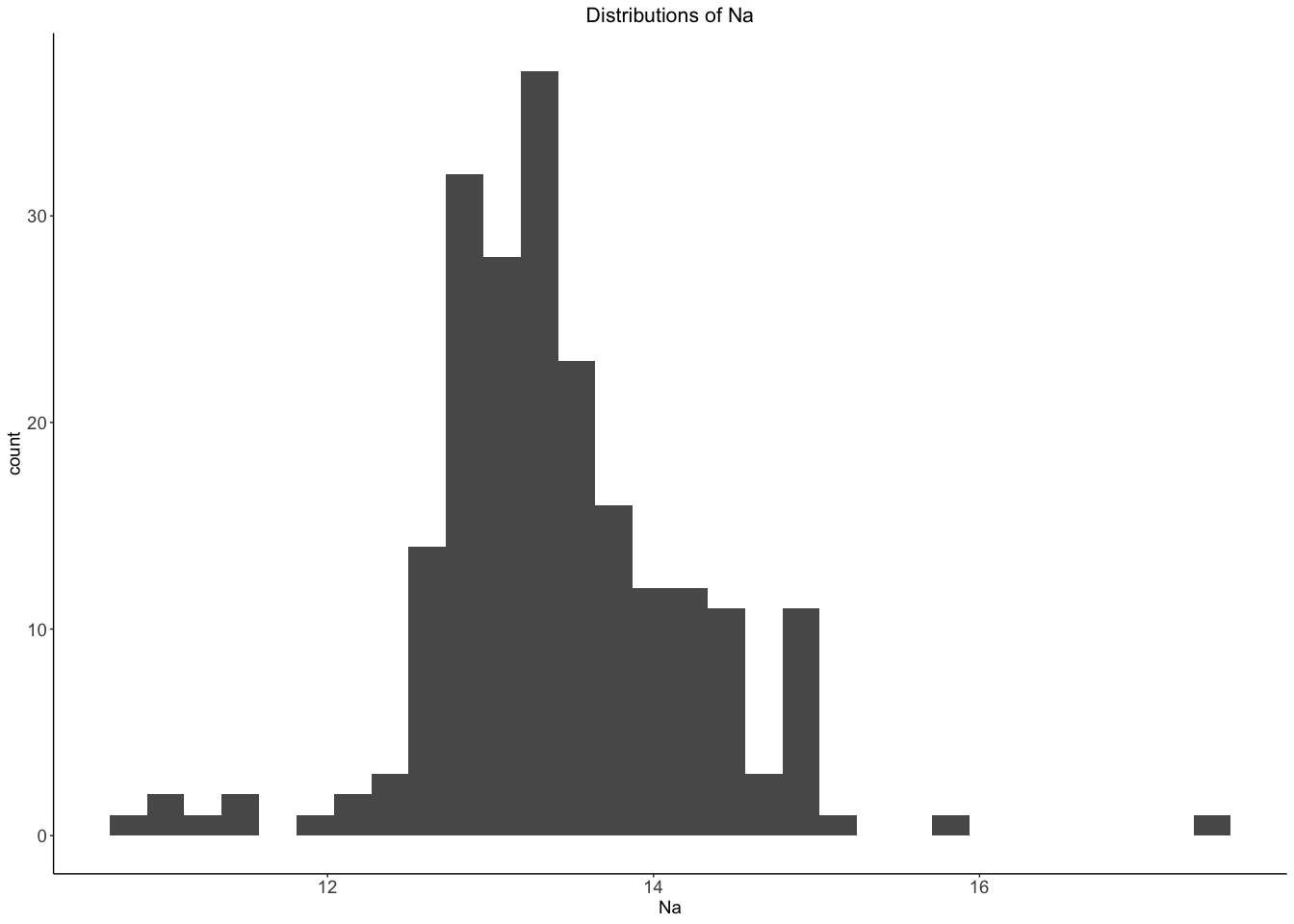
attach(Glass)  
ggplot(data = Glass)+geom\_histogram(aes(RI))+ggtitle("Distributions of RI")+theme\_classic()+theme(  
 plot.title = element\_text(size = 16,hjust = 0.5),  
 axis.text = element\_text(size =14),  
 axis.title = element\_text(size = 14)  
)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



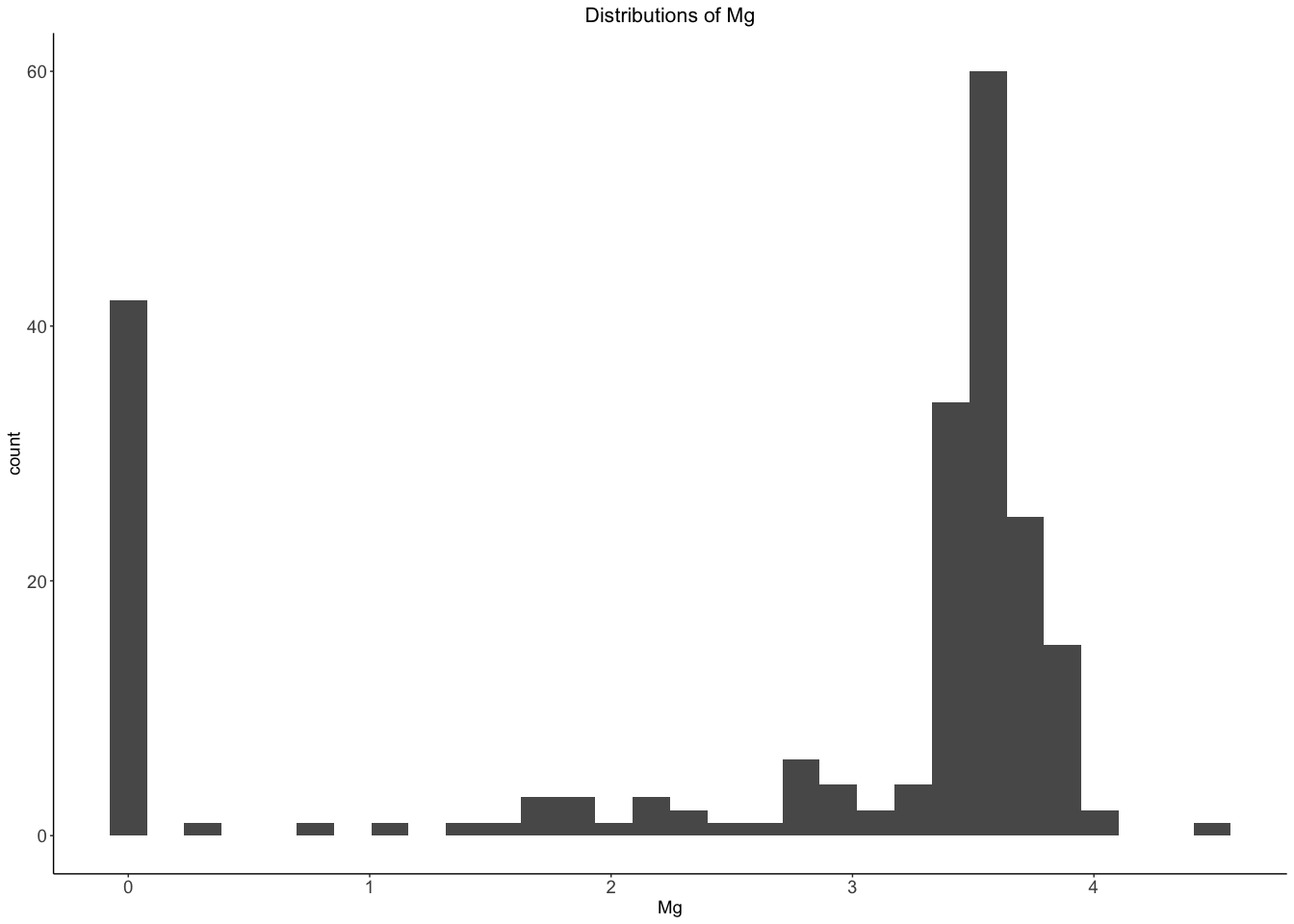
ggplot(data = Glass)+geom\_histogram(aes(Na))+ggtitle("Distributions of Na")+theme\_classic()+theme(  
 plot.title = element\_text(size = 16,hjust = 0.5),  
 axis.text = element\_text(size =14),  
 axis.title = element\_text(size = 14)  
)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



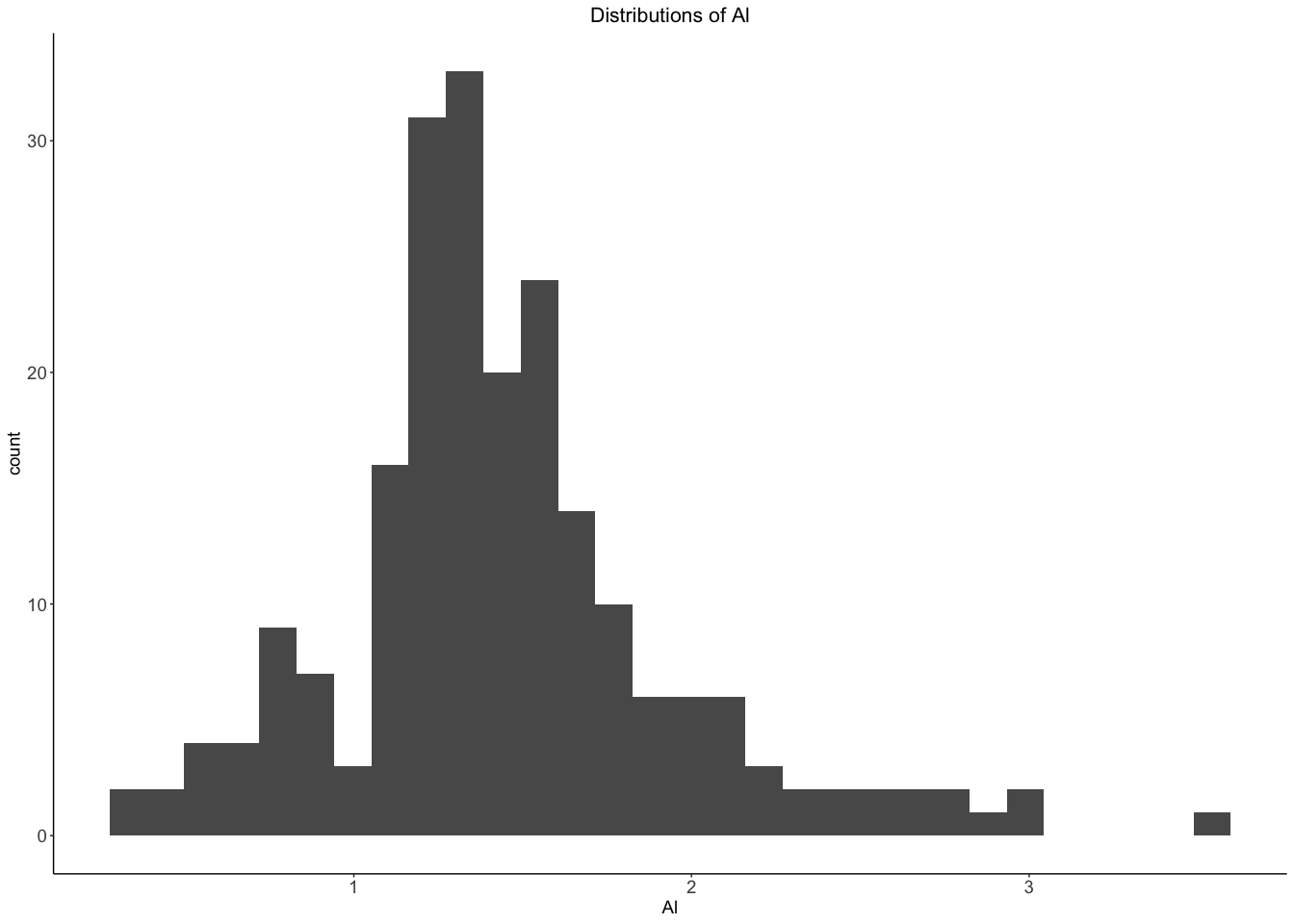
ggplot(data = Glass)+geom\_histogram(aes(Mg))+ggtitle("Distributions of Mg")+theme\_classic()+theme(  
 plot.title = element\_text(size = 16,hjust = 0.5),  
 axis.text = element\_text(size =14),  
 axis.title = element\_text(size = 14)  
)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



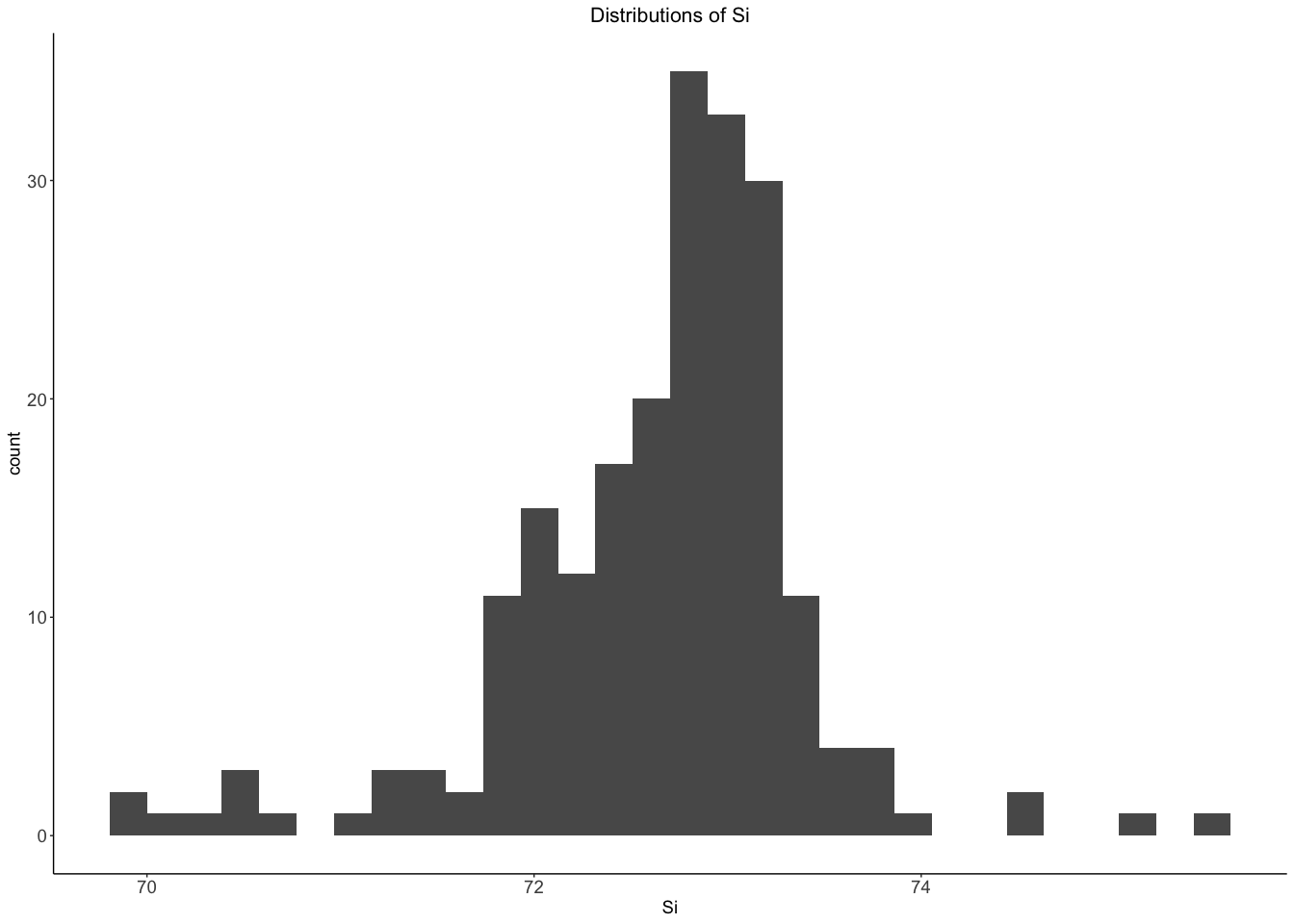
ggplot(data = Glass)+geom\_histogram(aes(Al))+ggtitle("Distributions of Al")+theme\_classic()+theme(  
 plot.title = element\_text(size = 16,hjust = 0.5),  
 axis.text = element\_text(size =14),  
 axis.title = element\_text(size = 14)  
)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



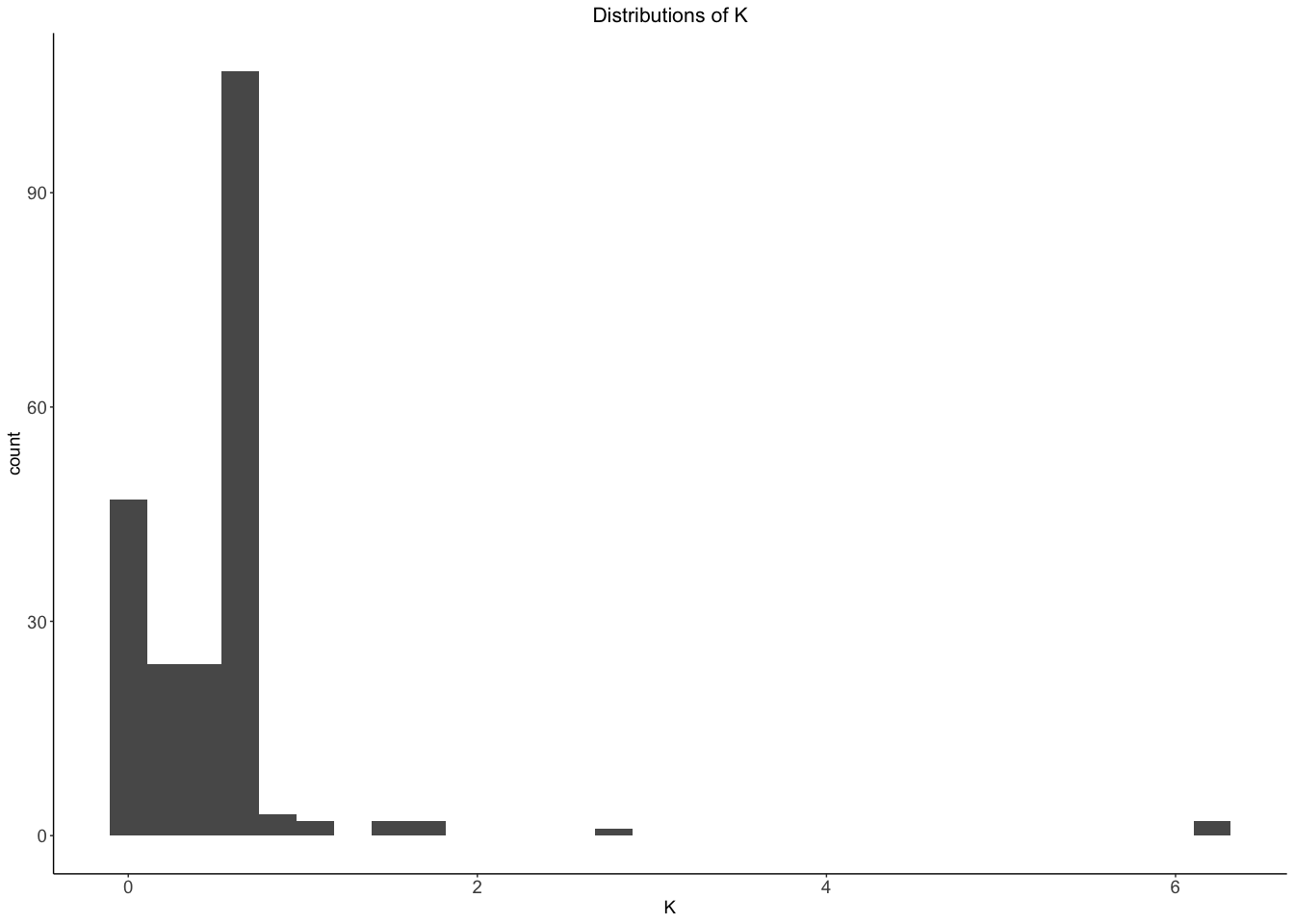
ggplot(data = Glass)+geom\_histogram(aes(Si))+ggtitle("Distributions of Si")+theme\_classic()+theme(  
 plot.title = element\_text(size = 16,hjust = 0.5),  
 axis.text = element\_text(size =14),  
 axis.title = element\_text(size = 14)  
)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



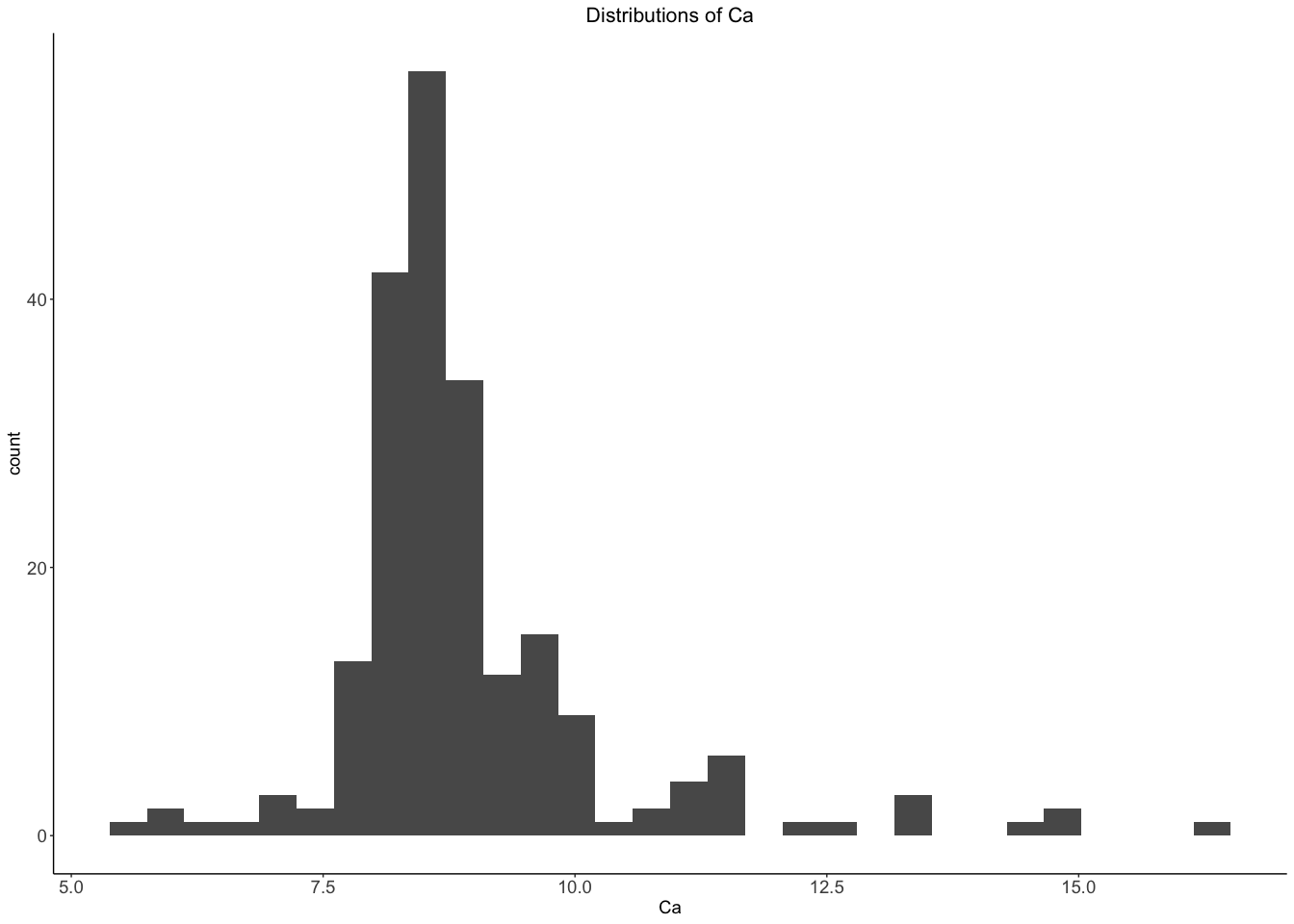
ggplot(data = Glass)+geom\_histogram(aes(K))+ggtitle("Distributions of K")+theme\_classic()+theme(  
 plot.title = element\_text(size = 16,hjust = 0.5),  
 axis.text = element\_text(size =14),  
 axis.title = element\_text(size = 14)  
)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



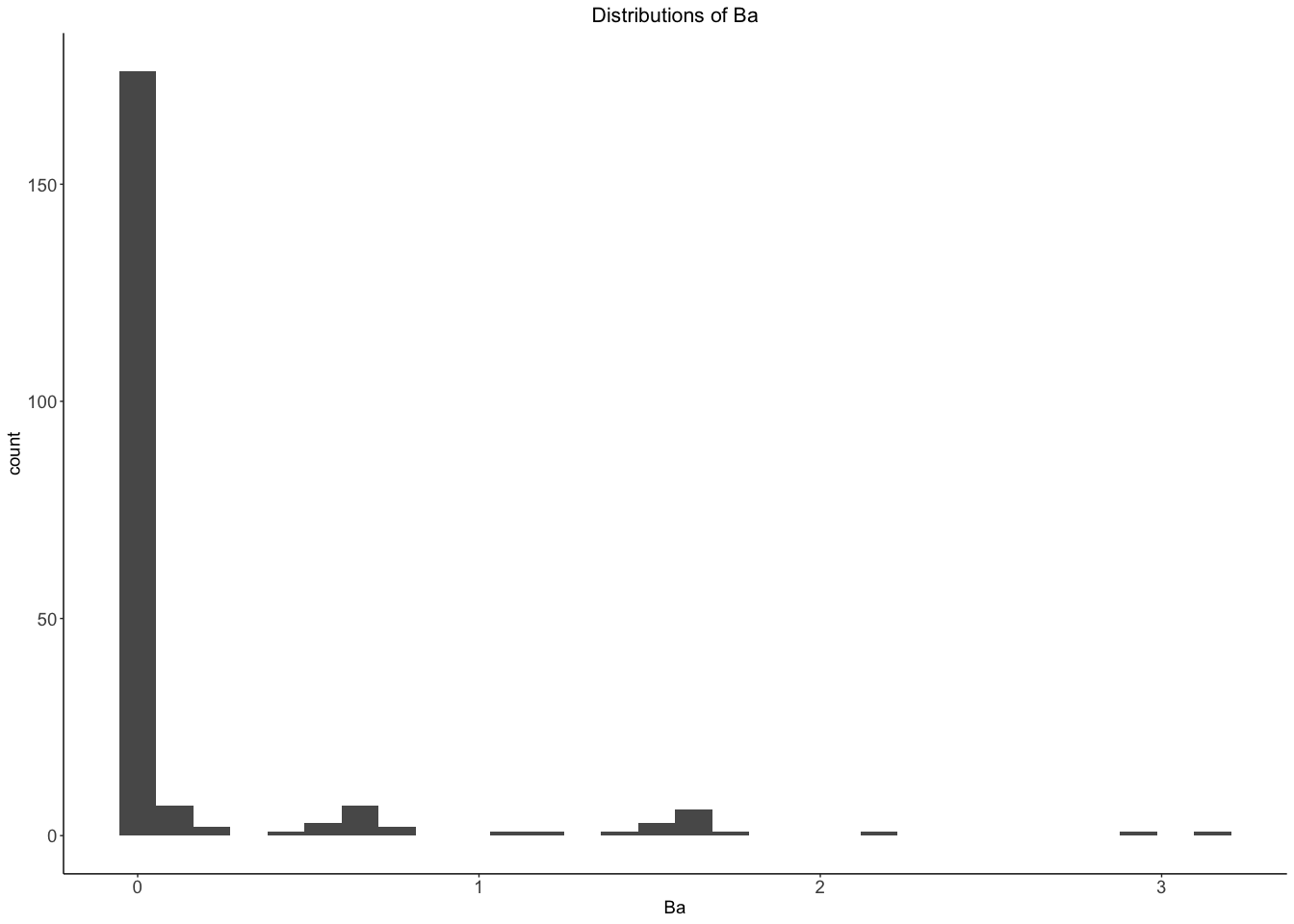
ggplot(data = Glass)+geom\_histogram(aes(Ca))+ggtitle("Distributions of Ca")+theme\_classic()+theme(  
 plot.title = element\_text(size = 16,hjust = 0.5),  
 axis.text = element\_text(size =14),  
 axis.title = element\_text(size = 14)  
)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



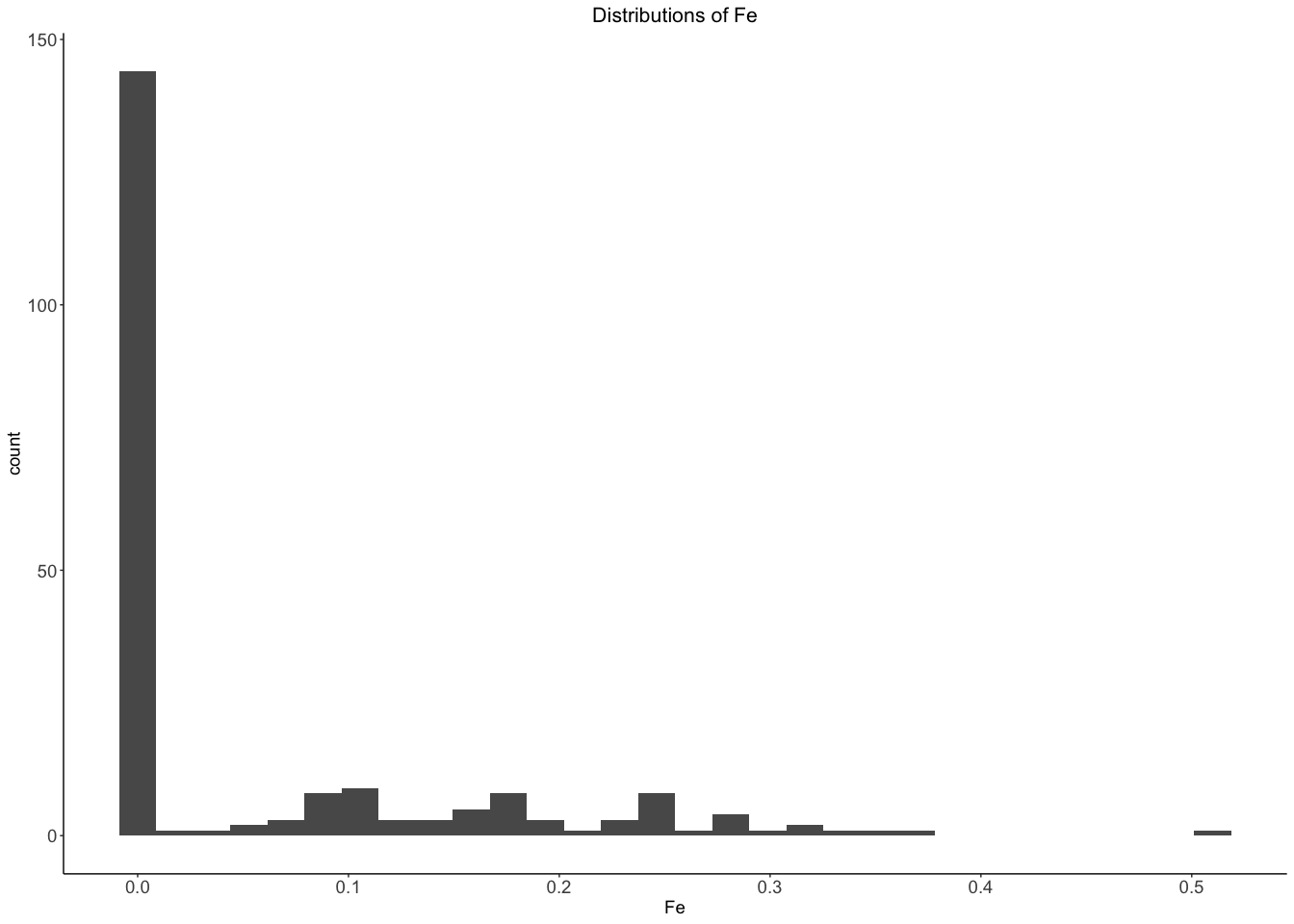
ggplot(data = Glass)+geom\_histogram(aes(Ba))+ggtitle("Distributions of Ba")+theme\_classic()+theme(  
 plot.title = element\_text(size = 16,hjust = 0.5),  
 axis.text = element\_text(size =14),  
 axis.title = element\_text(size = 14)  
)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



ggplot(data = Glass)+geom\_histogram(aes(Fe))+ggtitle("Distributions of Fe")+theme\_classic()+theme(  
 plot.title = element\_text(size = 16,hjust = 0.5),  
 axis.text = element\_text(size =14),  
 axis.title = element\_text(size = 14)  
)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

 RI ,Na,Mg,Al,Si,Ca is approximately normally distributed

K,Ba,Fe is not normally distributed

Correlation of the predictions

library("PerformanceAnalytics")

## Loading required package: xts

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

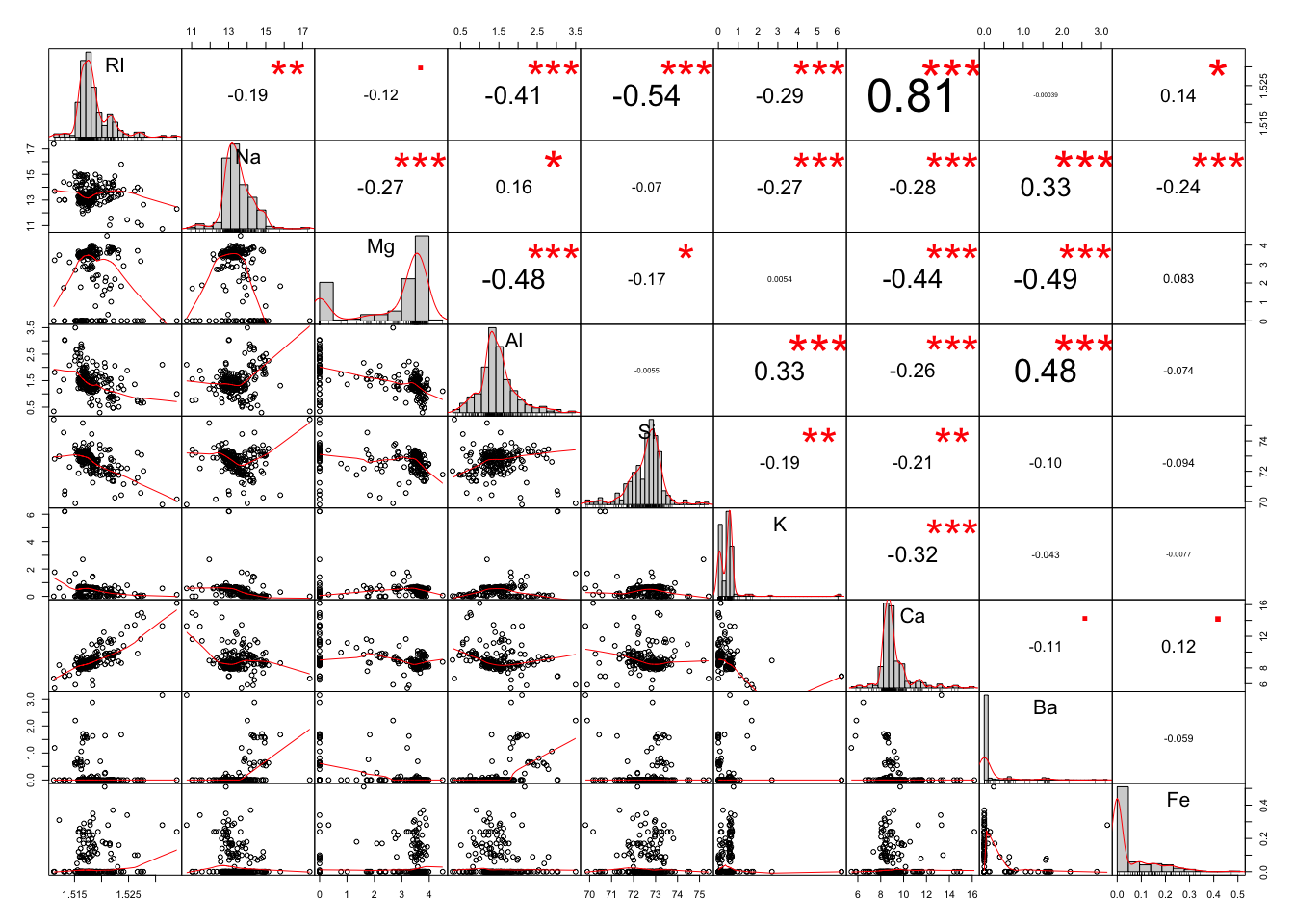
##   
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':  
##   
## first, last

##   
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':  
##   
## legend

chart.Correlation(Glass[,1:9], histogram=TRUE, pch=19)



From correlation we can see that

RI is significantly positively correlated with CA and negatively correlated with AL,Si,K. Na is Significantly positively correlated with Ba and negatively correlated with Mg,Al,K,Ca,Fe. Mg is significantly negatively correlated with Ca,Ba ,Al. Al is significantly positively correlated with K,Ba and negatively correlated with Ca. Si is weakly negatively correlated with K and Cal.

1. Do there appear to be any outliers in the data? Are any predictors skewed?

From the above plot of histograms we can see that Mg,Si,K,Ca,Ba and Fe has outliers. Fe,Ba,Ca,K,Na,RI are positively skewed and Mg,Si are negatively skewed.

1. Are there any relevant transformations of one or more predictors that might improve the classification model?

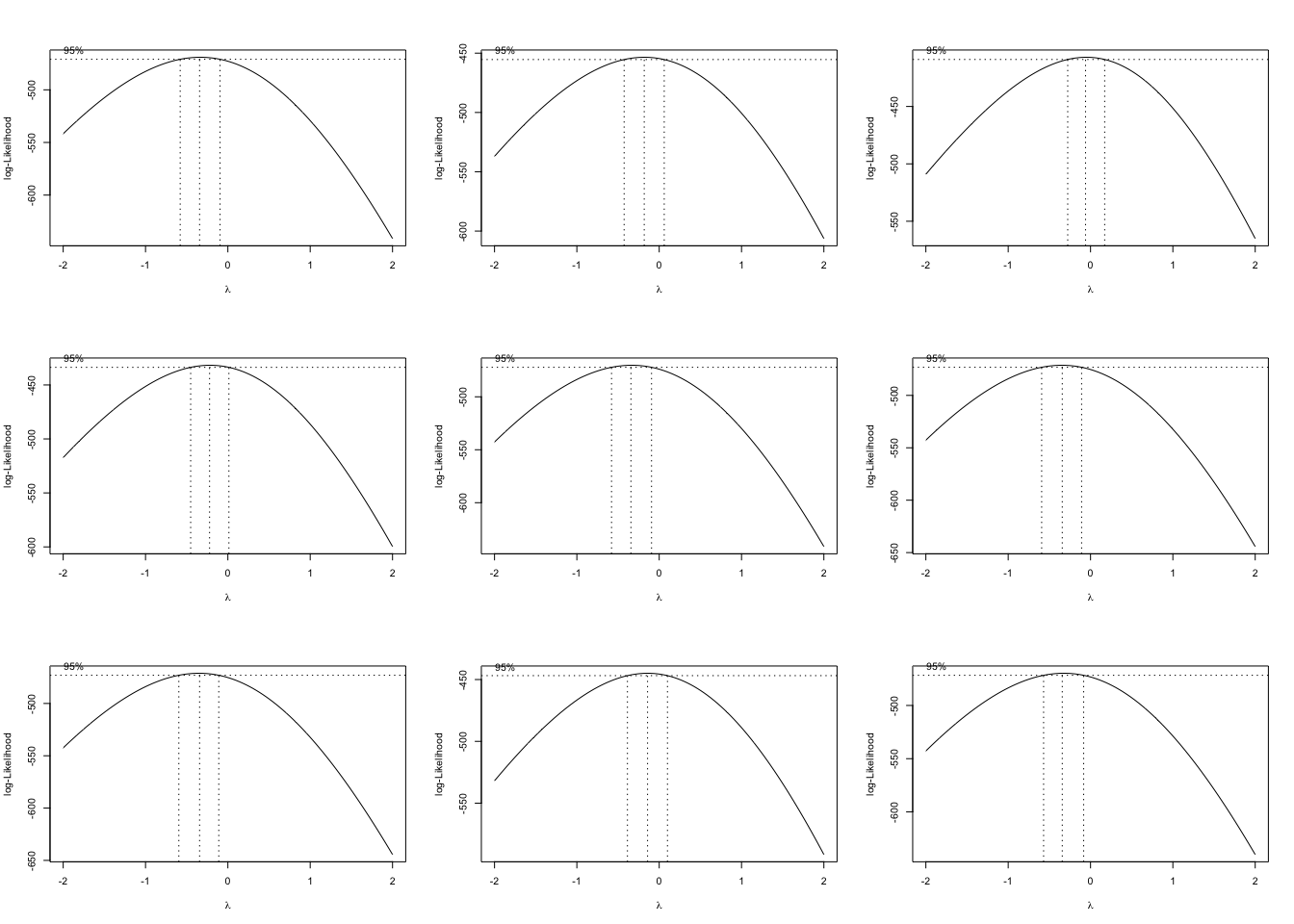
library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

## The following objects are masked from 'package:fma':  
##   
## cement, housing, petrol

Glass$Type <- as.numeric(Glass$Type)  
par(mfrow = c(3,3))  
boxcox(Type~RI, data = Glass)  
boxcox(Type~Na, data = Glass)  
boxcox(Type~Mg, data = Glass)  
boxcox(Type~Al, data = Glass)  
boxcox(Type~Si, data = Glass)  
boxcox(Type~K, data = Glass)  
boxcox(Type~Ca, data = Glass)  
boxcox(Type~Ba, data = Glass)  
boxcox(Type~Fe, data = Glass)

 From the box cox transformation plot we can see that log transformation of Na, Mg and Ba will improve the model.

# 3.2

3.2. 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 environmen- tal conditions (e.g., temperature, precipitation) and plant conditions (e.g., left spots, mold growth). The outcome labels consist of 19 distinct classes.

The data can be loaded via:

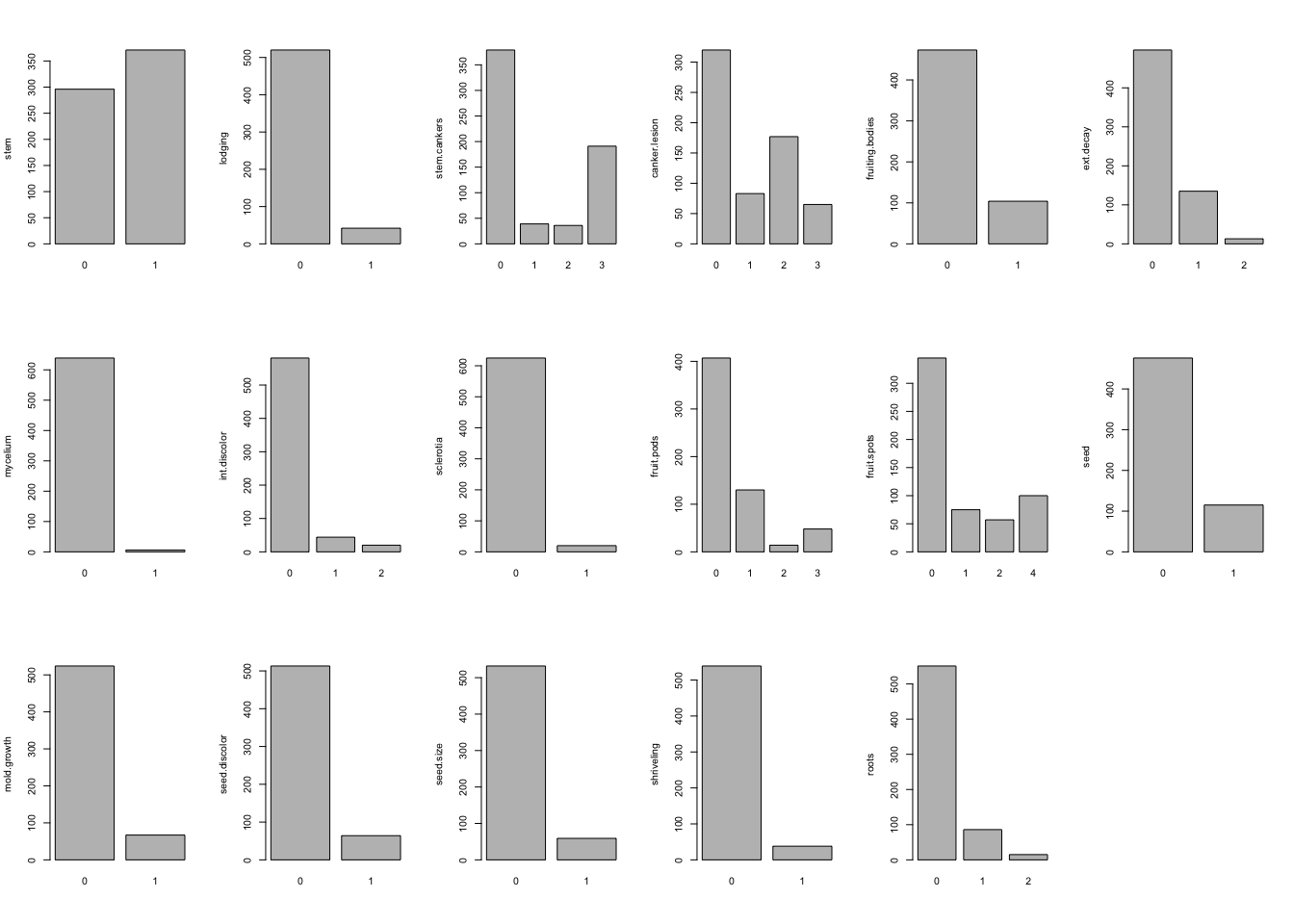
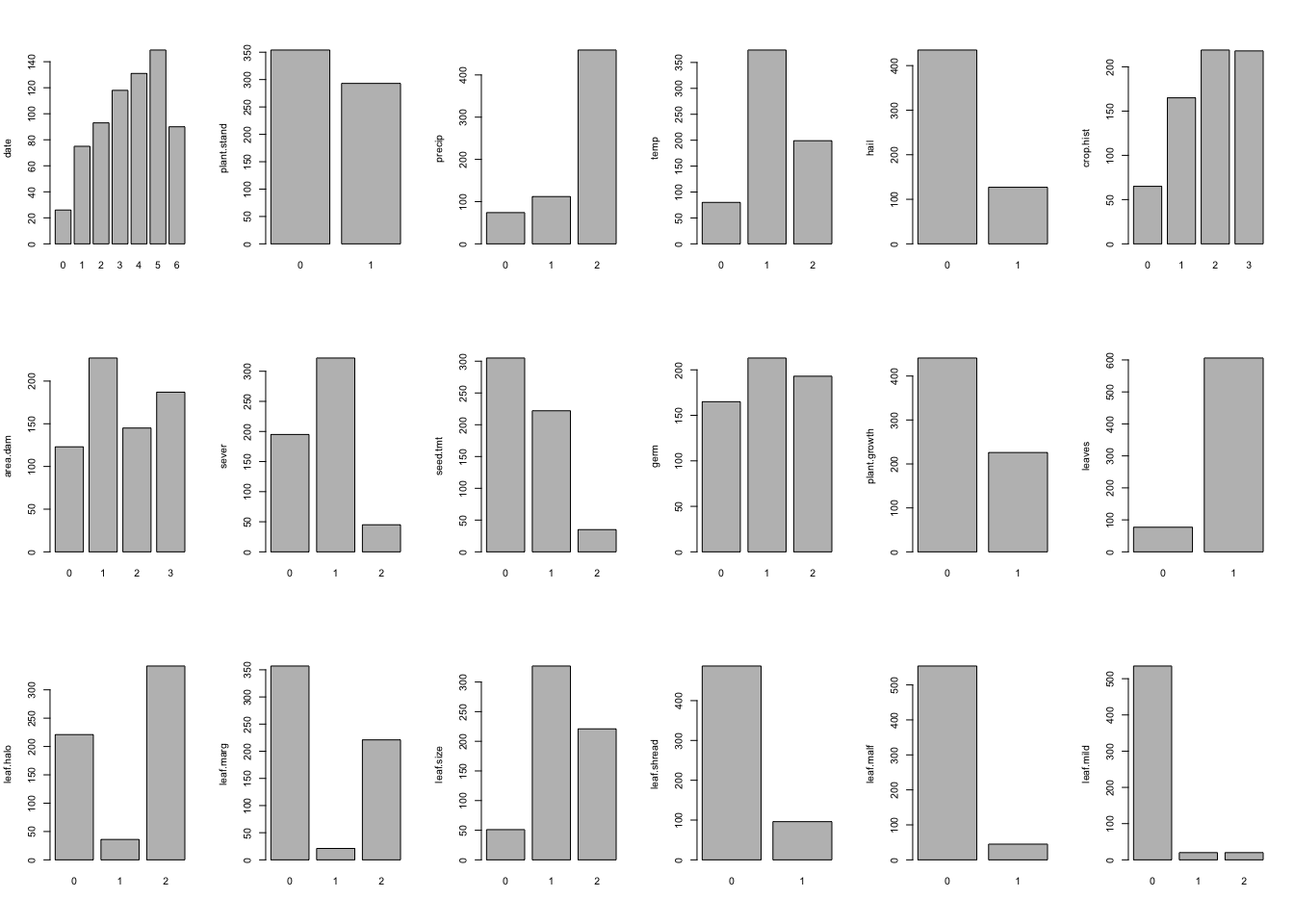
library(mlbench)  
data(Soybean)  
glimpse(Soybean)

## Observations: 683  
## Variables: 36  
## $ Class <fct> diaporthe-stem-canker, diaporthe-stem-canker, di…  
## $ date <fct> 6, 4, 3, 3, 6, 5, 5, 4, 6, 4, 6, 4, 3, 6, 6, 5, …  
## $ plant.stand <ord> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ precip <ord> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, …  
## $ temp <ord> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 2, 2, …  
## $ hail <fct> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, …  
## $ crop.hist <fct> 1, 2, 1, 1, 2, 3, 2, 1, 3, 2, 1, 1, 1, 3, 1, 3, …  
## $ area.dam <fct> 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 3, 2, 3, 3, 3, …  
## $ sever <fct> 1, 2, 2, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, …  
## $ seed.tmt <fct> 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, …  
## $ germ <ord> 0, 1, 2, 1, 2, 1, 0, 2, 1, 2, 0, 1, 0, 0, 1, 2, …  
## $ plant.growth <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …  
## $ leaves <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …  
## $ leaf.halo <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ leaf.marg <fct> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, …  
## $ leaf.size <ord> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, …  
## $ leaf.shread <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ leaf.malf <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ leaf.mild <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ stem <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …  
## $ lodging <fct> 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, …  
## $ stem.cankers <fct> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 0, 0, 0, 0, 0, 0, …  
## $ canker.lesion <fct> 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 3, 3, 3, 3, 3, 3, …  
## $ fruiting.bodies <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, …  
## $ ext.decay <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, …  
## $ mycelium <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ int.discolor <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, …  
## $ sclerotia <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, …  
## $ fruit.pods <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ fruit.spots <fct> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, …  
## $ seed <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ mold.growth <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ seed.discolor <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ seed.size <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ shriveling <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ roots <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …

1. Investigate the frequency distributions for the categorical predictors. Are any of the distributions degenerate in the ways discussed earlier in this chapter?

We’ll first plot make barplot

df <- Soybean[,2:36]  
par(mfrow = c(3, 6))  
for (i in 1:ncol(df)) {  
 barplot(table(df[,i]),ylab = names(df[i]))  
}



nearZeroVar in R for the categorical variables

library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

nearZeroVar(df,names = TRUE, saveMetrics=T)

## freqRatio percentUnique zeroVar nzv  
## date 1.137405 1.0248902 FALSE FALSE  
## plant.stand 1.208191 0.2928258 FALSE FALSE  
## precip 4.098214 0.4392387 FALSE FALSE  
## temp 1.879397 0.4392387 FALSE FALSE  
## hail 3.425197 0.2928258 FALSE FALSE  
## crop.hist 1.004587 0.5856515 FALSE FALSE  
## area.dam 1.213904 0.5856515 FALSE FALSE  
## sever 1.651282 0.4392387 FALSE FALSE  
## seed.tmt 1.373874 0.4392387 FALSE FALSE  
## germ 1.103627 0.4392387 FALSE FALSE  
## plant.growth 1.951327 0.2928258 FALSE FALSE  
## leaves 7.870130 0.2928258 FALSE FALSE  
## leaf.halo 1.547511 0.4392387 FALSE FALSE  
## leaf.marg 1.615385 0.4392387 FALSE FALSE  
## leaf.size 1.479638 0.4392387 FALSE FALSE  
## leaf.shread 5.072917 0.2928258 FALSE FALSE  
## leaf.malf 12.311111 0.2928258 FALSE FALSE  
## leaf.mild 26.750000 0.4392387 FALSE TRUE  
## stem 1.253378 0.2928258 FALSE FALSE  
## lodging 12.380952 0.2928258 FALSE FALSE  
## stem.cankers 1.984293 0.5856515 FALSE FALSE  
## canker.lesion 1.807910 0.5856515 FALSE FALSE  
## fruiting.bodies 4.548077 0.2928258 FALSE FALSE  
## ext.decay 3.681481 0.4392387 FALSE FALSE  
## mycelium 106.500000 0.2928258 FALSE TRUE  
## int.discolor 13.204545 0.4392387 FALSE FALSE  
## sclerotia 31.250000 0.2928258 FALSE TRUE  
## fruit.pods 3.130769 0.5856515 FALSE FALSE  
## fruit.spots 3.450000 0.5856515 FALSE FALSE  
## seed 4.139130 0.2928258 FALSE FALSE  
## mold.growth 7.820896 0.2928258 FALSE FALSE  
## seed.discolor 8.015625 0.2928258 FALSE FALSE  
## seed.size 9.016949 0.2928258 FALSE FALSE  
## shriveling 14.184211 0.2928258 FALSE FALSE  
## roots 6.406977 0.4392387 FALSE FALSE

There are few distributions degenerate . Specifically leaf.mild,mycelium and sclerotia.

1. 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?

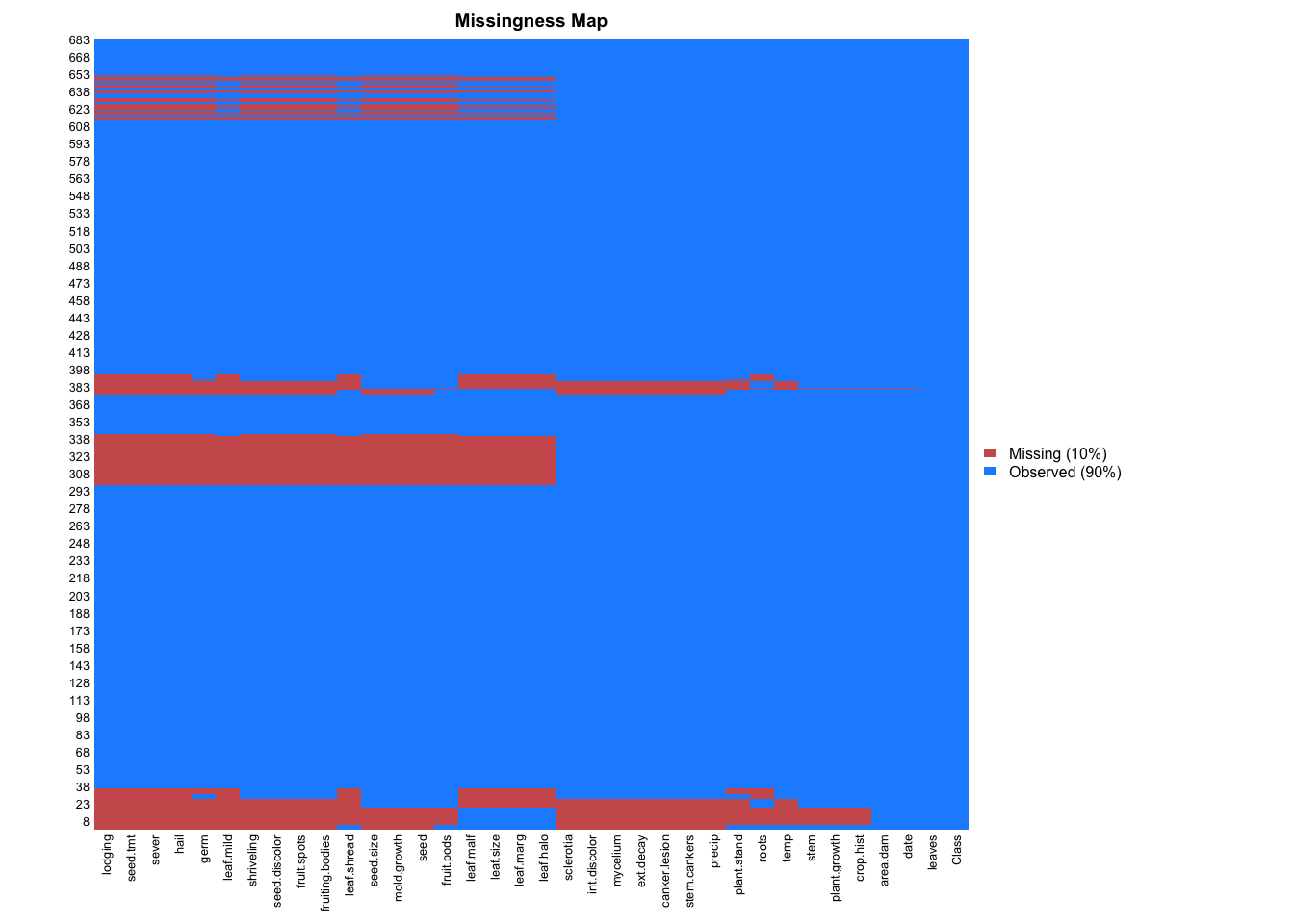
Now we’ll plot map of the missing data

library(Amelia)

## Loading required package: Rcpp

## ##   
## ## Amelia II: Multiple Imputation  
## ## (Version 1.7.5, built: 2018-05-07)  
## ## Copyright (C) 2005-2019 James Honaker, Gary King and Matthew Blackwell  
## ## Refer to http://gking.harvard.edu/amelia/ for more information  
## ##

missmap(Soybean)



Amount of missing in each variable

sort(colMeans(is.na(Soybean)),decreasing = T)

## hail sever seed.tmt lodging   
## 0.177159590 0.177159590 0.177159590 0.177159590   
## germ leaf.mild fruiting.bodies fruit.spots   
## 0.163982430 0.158125915 0.155197657 0.155197657   
## seed.discolor shriveling leaf.shread seed   
## 0.155197657 0.155197657 0.146412884 0.134699854   
## mold.growth seed.size leaf.halo leaf.marg   
## 0.134699854 0.134699854 0.122986823 0.122986823   
## leaf.size leaf.malf fruit.pods precip   
## 0.122986823 0.122986823 0.122986823 0.055636896   
## stem.cankers canker.lesion ext.decay mycelium   
## 0.055636896 0.055636896 0.055636896 0.055636896   
## int.discolor sclerotia plant.stand roots   
## 0.055636896 0.055636896 0.052708638 0.045387994   
## temp crop.hist plant.growth stem   
## 0.043923865 0.023426061 0.023426061 0.023426061   
## date area.dam Class leaves   
## 0.001464129 0.001464129 0.000000000 0.000000000

Particularly hail , sever , seed.tmt , lodging , germ , leaf.mild fruiting.bodies , fruit.spots , seed.discolor , shriveling , leaf.shread seed , mold.growth , seed.size , leaf.halo , are more likelly to be missing.

Missing values by group

Soybean %>%  
mutate(total = n()) %>%   
group\_by(Class) %>%  
mutate(Missing = n(), Proportion=Missing/total) %>%  
dplyr::select(Class, Missing, Proportion) %>%  
unique() %>%   
 arrange(-Proportion)

## # A tibble: 19 x 3  
## # Groups: Class [19]  
## Class Missing Proportion  
## <fct> <int> <dbl>  
## 1 brown-spot 92 0.135   
## 2 alternarialeaf-spot 91 0.133   
## 3 frog-eye-leaf-spot 91 0.133   
## 4 phytophthora-rot 88 0.129   
## 5 brown-stem-rot 44 0.0644  
## 6 anthracnose 44 0.0644  
## 7 diaporthe-stem-canker 20 0.0293  
## 8 charcoal-rot 20 0.0293  
## 9 rhizoctonia-root-rot 20 0.0293  
## 10 powdery-mildew 20 0.0293  
## 11 downy-mildew 20 0.0293  
## 12 bacterial-blight 20 0.0293  
## 13 bacterial-pustule 20 0.0293  
## 14 purple-seed-stain 20 0.0293  
## 15 phyllosticta-leaf-spot 20 0.0293  
## 16 2-4-d-injury 16 0.0234  
## 17 diaporthe-pod-&-stem-blight 15 0.0220  
## 18 cyst-nematode 14 0.0205  
## 19 herbicide-injury 8 0.0117

Most of the missing values are in the following classes brown-spot,alternarialeaf-spot,frog-eye-leaf-spot,phytophthora-rot.

1. Develop a strategy for handling missing data, either by eliminating predictors or imputation.

We can simply drop the rows having missing values. We’ll now drop them.

Soybean\_complete <- na.omit(Soybean)  
head(Soybean\_complete)

## Class date plant.stand precip temp hail crop.hist  
## 1 diaporthe-stem-canker 6 0 2 1 0 1  
## 2 diaporthe-stem-canker 4 0 2 1 0 2  
## 3 diaporthe-stem-canker 3 0 2 1 0 1  
## 4 diaporthe-stem-canker 3 0 2 1 0 1  
## 5 diaporthe-stem-canker 6 0 2 1 0 2  
## 6 diaporthe-stem-canker 5 0 2 1 0 3  
## area.dam sever seed.tmt germ plant.growth leaves leaf.halo leaf.marg  
## 1 1 1 0 0 1 1 0 2  
## 2 0 2 1 1 1 1 0 2  
## 3 0 2 1 2 1 1 0 2  
## 4 0 2 0 1 1 1 0 2  
## 5 0 1 0 2 1 1 0 2  
## 6 0 1 0 1 1 1 0 2  
## leaf.size leaf.shread leaf.malf leaf.mild stem lodging stem.cankers  
## 1 2 0 0 0 1 1 3  
## 2 2 0 0 0 1 0 3  
## 3 2 0 0 0 1 0 3  
## 4 2 0 0 0 1 0 3  
## 5 2 0 0 0 1 0 3  
## 6 2 0 0 0 1 0 3  
## canker.lesion fruiting.bodies ext.decay mycelium int.discolor sclerotia  
## 1 1 1 1 0 0 0  
## 2 1 1 1 0 0 0  
## 3 0 1 1 0 0 0  
## 4 0 1 1 0 0 0  
## 5 1 1 1 0 0 0  
## 6 0 1 1 0 0 0  
## fruit.pods fruit.spots seed mold.growth seed.discolor seed.size  
## 1 0 4 0 0 0 0  
## 2 0 4 0 0 0 0  
## 3 0 4 0 0 0 0  
## 4 0 4 0 0 0 0  
## 5 0 4 0 0 0 0  
## 6 0 4 0 0 0 0  
## shriveling roots  
## 1 0 0  
## 2 0 0  
## 3 0 0  
## 4 0 0  
## 5 0 0  
## 6 0 0

dim(Soybean\_complete)

## [1] 562 36

After dropping 562 observations remains.