PSTAT131 LAB1

Zhongyun Zhang 5559158, Zhengyao Lu 6094270

10/20/2020

##Problem1 #a)

```
library(readr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v dplyr   1.0.2
## v tibble  3.0.4     v stringr 1.4.0
## v tidyr   1.1.2     v forcats 0.5.0
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
#install.packages("ISLR")
#install.packages("ggplot2")
#install.packages("plyr")
#install.packages("dplyr")
#install.packages("class")
#Load libraries
library(ISLR)
library(ggplot2)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(plyr)
```

```
## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':
##
```

```
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following object is masked from 'package:purrr':
##
##     compact
```

```r
library(readr)
library(dplyr)
library(class)
algae <- read_table2("algaeBloom.txt", col_names=
                      c('season','size','speed','mxPH','mnO2','Cl','NO3','NH4','oPO4','PO4','Chla',
                     'a1','a2','a3','a4','a5','a6','a7'),
                     na="XXXXXXX")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   season = col_character(),
##   size = col_character(),
##   speed = col_character(),
##   mxPH = col_double(),
##   mnO2 = col_double(),
##   Cl = col_double(),
##   NO3 = col_double(),
##   NH4 = col_double(),
##   oPO4 = col_double(),
##   PO4 = col_double(),
##   Chla = col_double(),
##   a1 = col_double(),
##   a2 = col_double(),
##   a3 = col_double(),
##   a4 = col_double(),
##   a5 = col_double(),
##   a6 = col_double(),
##   a7 = col_double()
## )
```

```r
glimpse(algae)
```

```
## Rows: 200
## Columns: 18
## $ season <chr> "winter", "spring", "autumn", "spring", "autumn", "winter", ...
## $ size   <chr> "small", "small", "small", "small", "small", "small", "small...
## $ speed  <chr> "medium", "medium", "medium", "medium", "medium", "high", "h...
## $ mxPH   <dbl> 8.00, 8.35, 8.10, 8.07, 8.06, 8.25, 8.15, 8.05, 8.70, 7.93, ...
## $ mnO2   <dbl> 9.8, 8.0, 11.4, 4.8, 9.0, 13.1, 10.3, 10.6, 3.4, 9.9, 10.2, ...
## $ Cl     <dbl> 60.800, 57.750, 40.020, 77.364, 55.350, 65.750, 73.250, 59.0...
## $ NO3    <dbl> 6.238, 1.288, 5.330, 2.302, 10.416, 9.248, 1.535, 4.990, 0.8...
## $ NH4    <dbl> 578.000, 370.000, 346.667, 98.182, 233.700, 430.000, 110.000...
## $ oPO4   <dbl> 105.000, 428.750, 125.667, 61.182, 58.222, 18.250, 61.250, 4...
## $ PO4    <dbl> 170.000, 558.750, 187.057, 138.700, 97.580, 56.667, 111.750,...
## $ Chla   <dbl> 50.000, 1.300, 15.600, 1.400, 10.500, 28.400, 3.200, 6.900, ...
## $ a1     <dbl> 0.0, 1.4, 3.3, 3.1, 9.2, 15.1, 2.4, 18.2, 25.4, 17.0, 16.6, ...
## $ a2     <dbl> 0.0, 7.6, 53.6, 41.0, 2.9, 14.6, 1.2, 1.6, 5.4, 0.0, 0.0, 0....
## $ a3     <dbl> 0.0, 4.8, 1.9, 18.9, 7.5, 1.4, 3.2, 0.0, 2.5, 0.0, 0.0, 0.0,...
## $ a4     <dbl> 0.0, 1.9, 0.0, 0.0, 0.0, 0.0, 3.9, 0.0, 0.0, 2.9, 0.0, 0.0, ...
## $ a5     <dbl> 34.2, 6.7, 0.0, 1.4, 7.5, 22.5, 5.8, 5.5, 0.0, 0.0, 1.2, 0.0...
## $ a6     <dbl> 8.3, 0.0, 0.0, 0.0, 4.1, 12.6, 6.8, 8.7, 0.0, 0.0, 0.0, 0.0,...
## $ a7     <dbl> 0.0, 2.1, 9.7, 1.4, 1.0, 2.9, 0.0, 0.0, 0.0, 1.7, 6.0, 1.5, ...
```

```
algae%>%
  dplyr::group_by(season)%>%
  dplyr::summarise(n=n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 4 x 2
##   season     n
##   <chr>  <int>
## 1 autumn    40
## 2 spring    53
## 3 summer    45
## 4 winter    62
```

The number of observations for autumn, spring, summer, winter are 40,53,45, and 62 respectively.

#b

```
sum(is.na(algae))
```

```
## [1] 33
```

```
algae %>%
summarise(avg = mean(mxPH, na.rm=TRUE),var=(sd(mxPH,na.rm=TRUE))^2)
```

```
##        avg       var
## 1 8.011734 0.3579693
```

```
algae %>%
summarise(avg = mean(mnO2, na.rm=TRUE),var=(sd(mnO2,na.rm=TRUE))^2)
```

```
##        avg      var
## 1 9.117778 5.718089
```

```
algae %>%
summarise(avg = mean(Cl, na.rm=TRUE),var=(sd(Cl,na.rm=TRUE))^2)
```

```
##        avg      var
## 1 43.63628 2193.172
```

```
algae %>%
summarise(avg = mean(NO3, na.rm=TRUE),var=(sd(NO3,na.rm=TRUE))^2)
```

```
##        avg      var
## 1 3.282389 14.26176
```

```
algae %>%
summarise(avg = mean(NH4, na.rm=TRUE),var=(sd(NH4,na.rm=TRUE))^2)
```

```
##        avg     var
## 1 501.2958 3851585
```

```
algae %>%
summarise(avg = mean(oPO4, na.rm=TRUE),var=(sd(oPO4,na.rm=TRUE))^2)
```

```
##       avg     var
## 1 73.5906 8305.85
```

```
algae %>%
summarise(avg = mean(PO4, na.rm=TRUE),var=(sd(PO4,na.rm=TRUE))^2)
```

```
##        avg      var
## 1 137.8821 16639.38
```

```
algae %>%
summarise(avg = mean(Chla, na.rm=TRUE),var=(sd(Chla,na.rm=TRUE))^2)
```

```
##       avg      var
## 1 13.9712 420.0827
```

There are missing values. Since I need to use na.rm to remove missing data, otherwise I will get NA for each calculation. As we can see from the output, mean different chemicals differ significantly. This may beacuse of different scales used for chemicals.NH4 and PO4 have greater scale. Also, chemicals with larger means have larger variance. However, NO3 has the smallest mean but a relatively large variance, meaning that the data in NO3 is more scattered.

#c

```
#Medians and MAD
algae %>%
summarise(med = median(mnO2, na.rm=TRUE),MAD=median(abs(mnO2-median(mnO2,na.rm=TRUE)),na.rm=TRUE))
```

```
##   med   MAD
## 1 9.8 1.385
```

```
algae %>%
summarise(med = median(Cl, na.rm=TRUE),MAD=median(abs(Cl-median(Cl,na.rm=TRUE)),na.rm=TRUE))
```

```
##     med     MAD
## 1 32.73 22.4265
```

```
algae %>%
summarise(med = median(NO3, na.rm=TRUE),MAD=median(abs(NO3-median(NO3,na.rm=TRUE)),na.rm=TRUE))
```

```
##     med   MAD
## 1 2.675 1.465
```

```
algae %>%
summarise(med = median(NH4, na.rm=TRUE),MAD=median(abs(NH4-median(NH4,na.rm=TRUE)),na.rm=TRUE))
```

```
##        med    MAD
## 1 103.1665 75.285
```

```
algae %>%
summarise(med = median(oPO4, na.rm=TRUE),MAD=median(abs(oPO4-median(oPO4,na.rm=TRUE)),na.rm=TRUE))
```

```
##     med     MAD
## 1 40.15 29.7085
```

```
algae %>%
summarise(med = median(PO4, na.rm=TRUE),MAD=median(abs(PO4-median(PO4,na.rm=TRUE)),na.rm=TRUE))
```

```
##        med     MAD
## 1 103.2855 82.5045
```

```
algae %>%
summarise(med = median(Chla, na.rm=TRUE),MAD=median(abs(Chla-median(Chla,na.rm=TRUE)),na.rm=TRUE))
```

```
##     med MAD
## 1 5.475 4.5
```

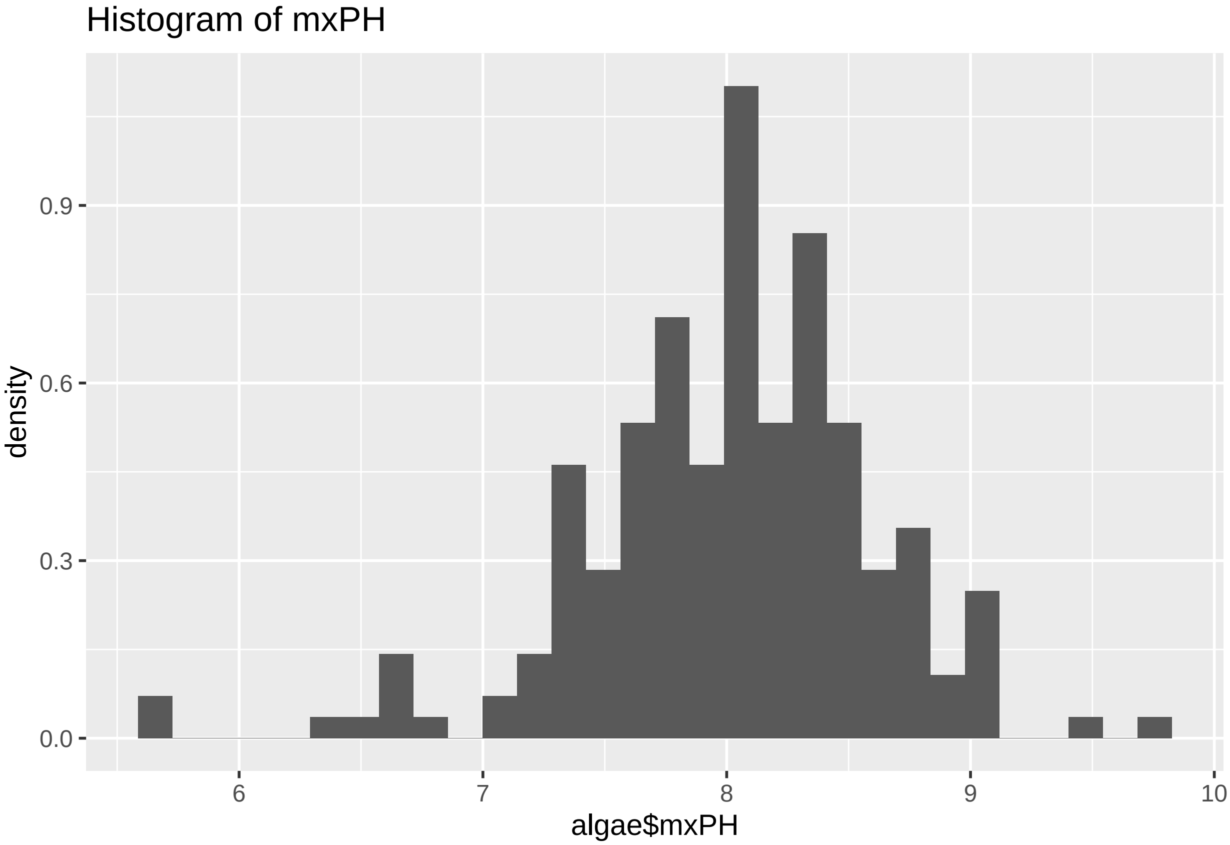
The medians, compared to the means, are similar (little bit smaller than mean); and the MADs are much smaller than the variances.

##Problem2 #a

```
ggplot(algae, aes(algae$mxPH)) + geom_histogram(aes(y = ..density..)) +ggtitle("Histogram of mxPH")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```



As we can see from the graph, the distribution is slightly skewed to the left.

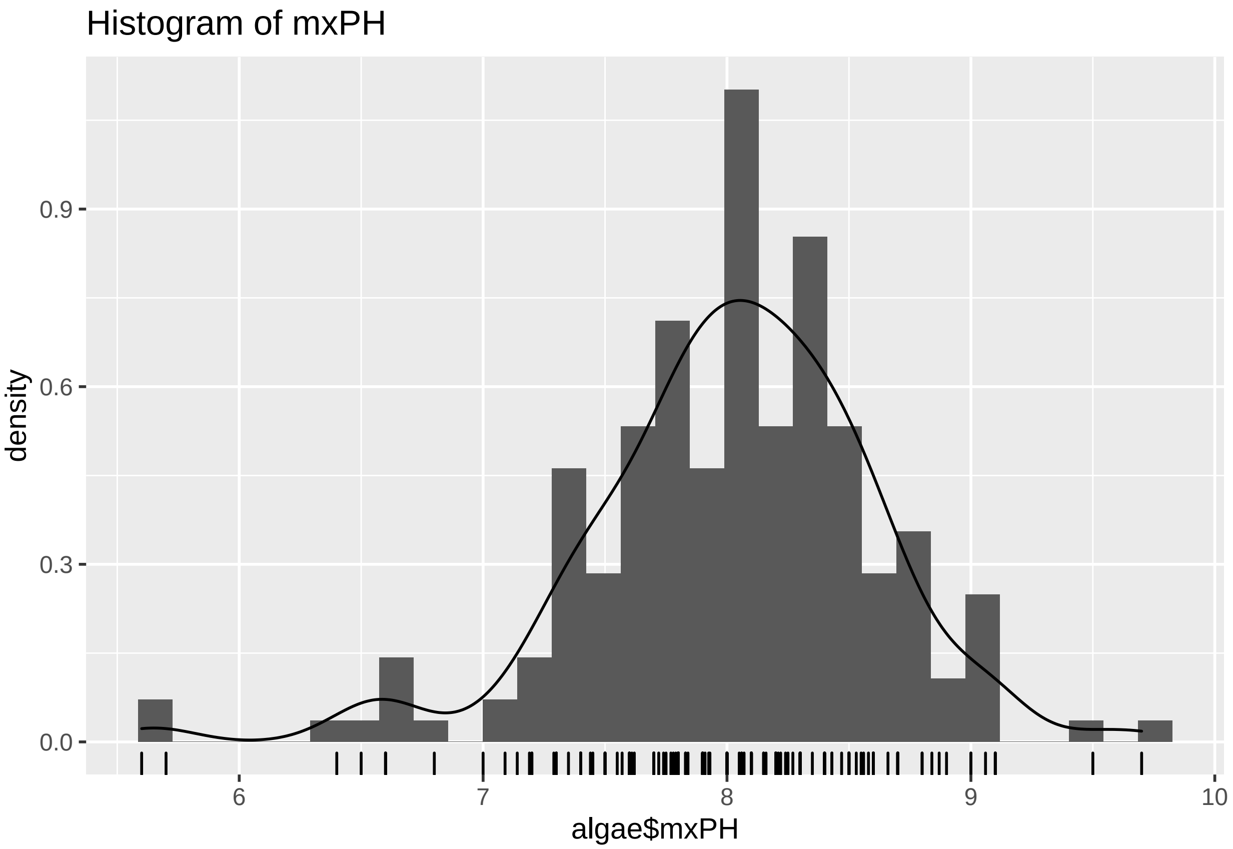
#b

```
ggplot(algae, aes(algae$mxPH)) + geom_histogram(aes(y = ..density..)) +ggtitle("Histogram of mxPH")+geom
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

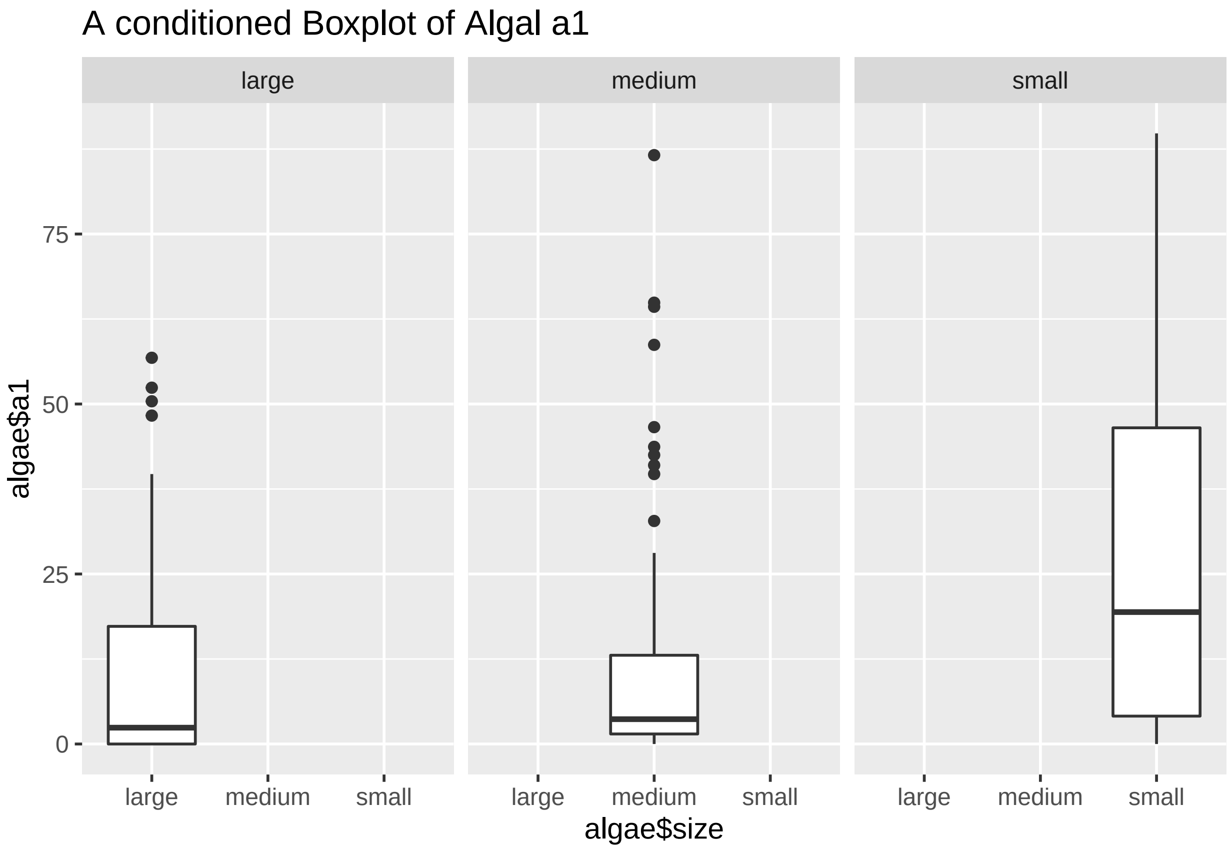
```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 1 rows containing non-finite values (stat_density).
```



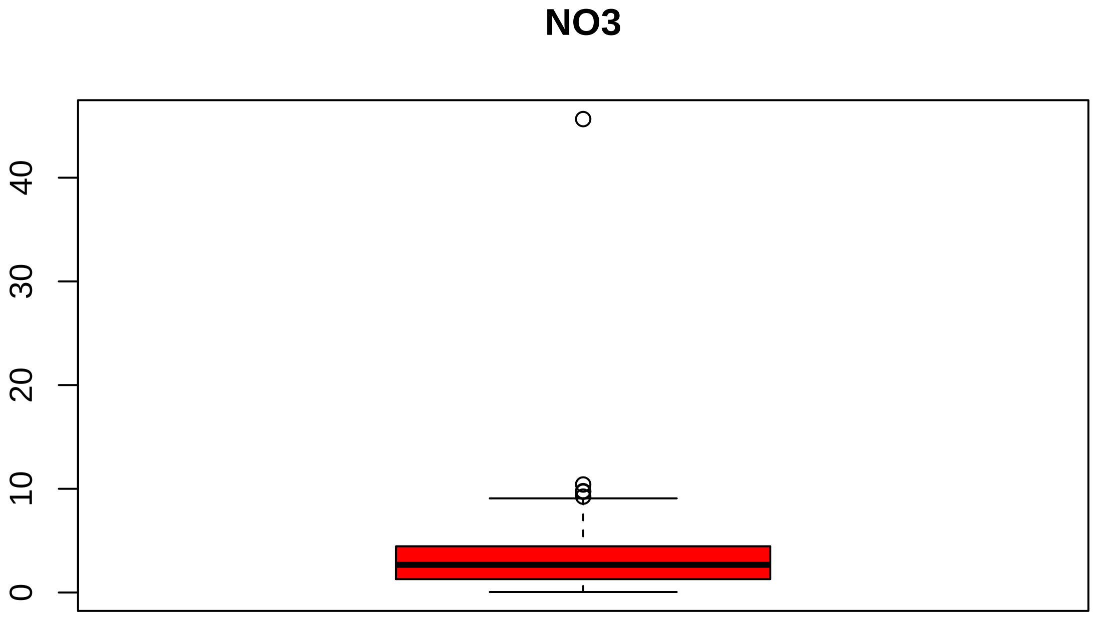
#c

```
#boxplot(algae$a1~algae$size,main=" ???A conditioned Boxplot of Algal a1??? ")
#ggplot(algae,aes(algae$size))+geom_boxplot() +ggtitle("A conditioned Boxplot of Algal a1")
ggplot(algae, aes(algae$size,algae$a1)) + geom_boxplot() +ggtitle("A conditioned Boxplot of Algal a1")
```



#d

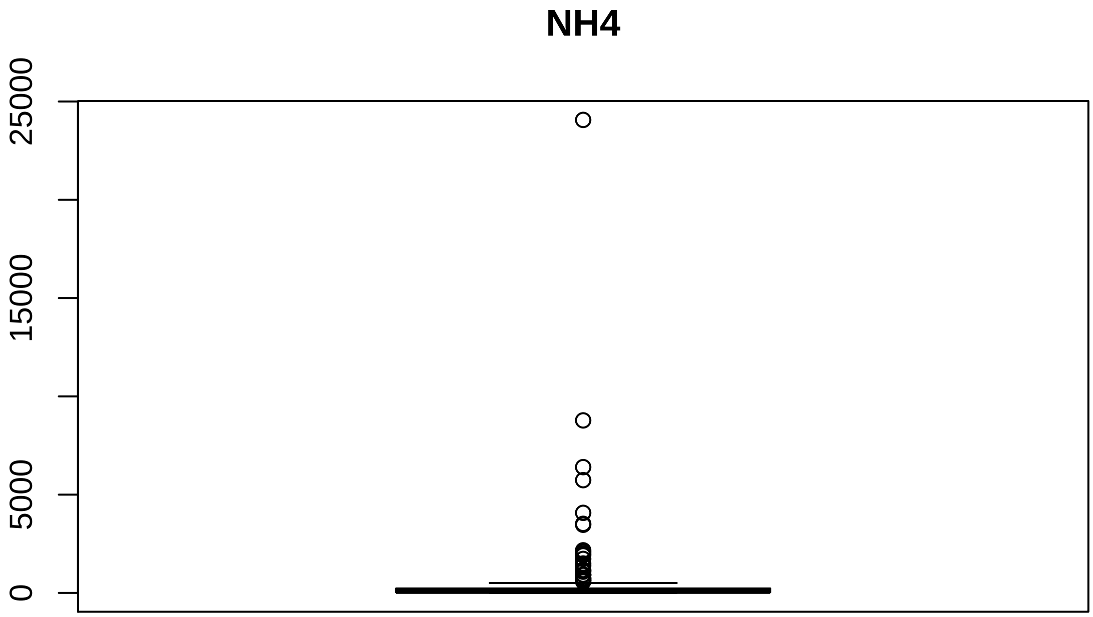
```
#boxplot(algae$NO3 ~ algae$size, main = "A conditional BoxPlot of Algae NO3")
#boxplot(algae$NH4 ~ algae$size, main = "A conditional BoxPlot of Algae NH4")
plot_NO3 <- boxplot(algae$NO3, main = "NO3",col="red")
```



```r
length(plot_NO3$out)
```

```
## [1] 5
```

```r
plot_NH4 <- boxplot(algae$NH4, main = "NH4",col="red")
```



```r
length(plot_NH4$out)
```

```
## [1] 27
```

```r
boxplot.stats(algae$NO3)$out
```

```
## [1] 10.416  9.248  9.773  9.715 45.650
```

```r
boxplot.stats(algae$NH4)$out
```

```
##  [1]   578.000  8777.600  1729.000  3515.000  6400.000  1911.000   647.570
##  [8]  1386.250  2082.850  2167.370   737.500   914.000  5738.330  4073.330
## [15]   758.750   931.833   723.667  3466.660   920.000  1990.160 24064.000
## [22]  1131.660  1495.000   643.000   627.273  1168.000  1081.660
```

There are outlier's for both since we see them in the boxplot, they are 1.5 distance away from the quantiles. It is hard to visualize in the boxplot graph. Therefore, from the outlier check, we can see that there are 5 outliers in NO3, and 27 outliers in NH4.

#e mean of NO3 was 3.28 and NH4 is 501.3 and variance was 14.26 and 3851585 Medians are 2.67 and 103.16 and MAD are 2.17 and 111.675

Outliers are defined as having higher variance from the rest of the data points. The computation of the mean and variance take outliers into account, thus it is possible to take them as outliers. Therefore, the mean and variance have weak resistance to outliers and are not sufficient estimators. In addition, MAD and median are better estimators because they are less sensitive to outliers, and MAD is a more robust estimator than the sample variance and mean in the presence of outliers. In this way it explain for the means and variances of NO3 and NH4 are so different from their respective medians and MADs. The chemicals' means and variances values are affected by outlier, and their medians and MADs are better estimators for these chemicals.

##Problem 3 #a

```
sum(complete.cases(algae)==FALSE)
```

```
## [1] 16
```

```
colSums(is.na(algae))
```

```
## season   size  speed   mxPH   mnO2     Cl    NO3    NH4   oPO4    PO4   Chla 
##      0      0      0      1      2     10      2      2      2      2     12 
##     a1     a2     a3     a4     a5     a6     a7 
##      0      0      0      0      0      0      0
```

a)There are 16 total observations that has missing values b)mxPH contains 1 missing value, mn02 contains 2 missing values, Cl contains 10 missing values, NO3 contains 2 missing values, NH4 contains 2 missing values, oPo4 contains 2 missing values, Po4 contains 2 missing values, Chla contains 12 missing values.

#b

```
algae.del <- algae%>%filter(complete.cases(.))
print(paste('There are', count(algae.del),'observations in algae.del.'))
```

```
##  [1] "There are c(\"winter\", \"spring\", \"autumn\", \"spring\", \"autumn\", \"winter\", \"summer\"
##  [2] "There are c(\"small\", \"small\", \"small\", \"small\", \"small\", \"small\", \"small\", \"smal
##  [3] "There are c(\"medium\", \"medium\", \"medium\", \"medium\", \"medium\", \"high\", \"high\", \"
##  [4] "There are c(8, 8.35, 8.1, 8.07, 8.06, 8.25, 8.15, 8.05, 8.7, 7.93, 7.7, 7.45, 7.74, 7.72, 7.9,
##  [5] "There are c(9.8, 8, 11.4, 4.8, 9, 13.1, 10.3, 10.6, 3.4, 9.9, 10.2, 11.7, 9.6, 11.8, 9.6, 11.5
##  [6] "There are c(60.8, 57.75, 40.02, 77.364, 55.35, 65.75, 73.25, 59.067, 21.95, 8, 8, 8.69, 5, 6.3
##  [7] "There are c(6.238, 1.288, 5.33, 2.302, 10.416, 9.248, 1.535, 4.99, 0.886, 1.39, 1.527, 1.588, 
##  [8] "There are c(578, 370, 346.66699, 98.182, 233.7, 430, 110, 205.66701, 102.75, 5.8, 21.571, 18.4
##  [9] "There are c(105, 428.75, 125.667, 61.182, 58.222, 18.25, 61.25, 44.667, 36.3, 27.25, 12.75, 10
## [10] "There are c(170, 558.75, 187.05701, 138.7, 97.58, 56.667, 111.75, 77.434, 71, 46.6, 20.75, 19,
## [11] "There are c(50, 1.3, 15.6, 1.4, 10.5, 28.4, 3.2, 6.9, 5.544, 0.8, 0.8, 0.6, 41, 0.5, 0.3, 1.1,
## [12] "There are c(0, 1.4, 3.3, 3.1, 9.2, 15.1, 2.4, 18.2, 25.4, 17, 16.6, 32.1, 43.5, 31.1, 52.2, 69
## [13] "There are c(0, 7.6, 53.6, 41, 2.9, 14.6, 1.2, 1.6, 5.4, 0, 0, 0, 0, 1, 5, 0, 0, 0, 0, 0, 0, 0,
## [14] "There are c(0, 4.8, 1.9, 18.9, 7.5, 1.4, 3.2, 0, 2.5, 0, 0, 0, 2.1, 3.4, 7.8, 1.7, 0, 3.1, 9.9
## [15] "There are c(0, 1.9, 0, 0, 0, 0, 3.9, 0, 0, 2.9, 0, 0, 0, 0, 0, 0, 1.2, 4.8, 4.3, 44.6, 6.8, 2.
## [16] "There are c(34.2, 6.7, 0, 1.4, 7.5, 22.5, 5.8, 5.5, 0, 0, 1.2, 0, 1.2, 1.9, 4, 0, 0, 7.7, 3.6,
## [17] "There are c(8.3, 0, 0, 0, 4.1, 12.6, 6.8, 8.7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1.4, 8.2, 0, 0, 0, 1
## [18] "There are c(0, 2.1, 9.7, 1.4, 1, 2.9, 0, 0, 0, 1.7, 6, 1.5, 2.1, 4.1, 0, 0, 0, 7.2, 2.2, 1.4, 
## [19] "There are c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
```

There are 184 observations in algae.del.

#c)

```
algae.med<-algae%>% mutate_at(vars(mxPH,mnO2,Cl,NO3,NH4,oPO4,PO4,Chla),funs(ifelse(is.na(.),median(., na
```

```
## Warning: `funs()` is deprecated as of dplyr 0.8.0.
## Please use a list of either functions or lambdas: 
## 
##   # Simple named list: 
##   list(mean = mean, median = median)
## 
##   # Auto named with `tibble::lst()`: 
##   tibble::lst(mean, median)
## 
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
print(paste('There are', count(algae.med),'observations in algae.med.'))
```

```
##  [1] "There are c(\"winter\", \"spring\", \"autumn\", \"spring\", \"autumn\", \"winter\", \"summer\"
##  [2] "There are c(\"small\", \"small\", \"small\", \"small\", \"small\", \"small\", \"small\", \"sma
##  [3] "There are c(\"medium\", \"medium\", \"medium\", \"medium\", \"medium\", \"high\", \"high\", \"
##  [4] "There are c(8, 8.35, 8.1, 8.07, 8.06, 8.25, 8.15, 8.05, 8.7, 7.93, 7.7, 7.45, 7.74, 7.72, 7.9,
##  [5] "There are c(9.8, 8, 11.4, 4.8, 9, 13.1, 10.3, 10.6, 3.4, 9.9, 10.2, 11.7, 9.6, 11.8, 9.6, 11.5
##  [6] "There are c(60.8, 57.75, 40.02, 77.364, 55.35, 65.75, 73.25, 59.067, 21.95, 8, 8, 8.69, 5, 6.3
##  [7] "There are c(6.238, 1.288, 5.33, 2.302, 10.416, 9.248, 1.535, 4.99, 0.886, 1.39, 1.527, 1.588,
##  [8] "There are c(578, 370, 346.66699, 98.182, 233.7, 430, 110, 205.66701, 102.75, 5.8, 21.571, 18.4
##  [9] "There are c(105, 428.75, 125.667, 61.182, 58.222, 18.25, 61.25, 44.667, 36.3, 27.25, 12.75, 10
## [10] "There are c(170, 558.75, 187.05701, 138.7, 97.58, 56.667, 111.75, 77.434, 71, 46.6, 20.75, 19,
## [11] "There are c(50, 1.3, 15.6, 1.4, 10.5, 28.4, 3.2, 6.9, 5.544, 0.8, 0.8, 0.6, 41, 0.5, 0.3, 1.1,
## [12] "There are c(0, 1.4, 3.3, 3.1, 9.2, 15.1, 2.4, 18.2, 25.4, 17, 16.6, 32.1, 43.5, 31.1, 52.2, 69
## [13] "There are c(0, 7.6, 53.6, 41, 2.9, 14.6, 1.2, 1.6, 5.4, 0, 0, 0, 0, 1, 5, 0, 0, 0, 0, 0, 0, 0,
## [14] "There are c(0, 4.8, 1.9, 18.9, 7.5, 1.4, 3.2, 0, 2.5, 0, 0, 0, 2.1, 3.4, 7.8, 1.7, 0, 3.1, 9.9
## [15] "There are c(0, 1.9, 0, 0, 0, 0, 3.9, 0, 0, 2.9, 0, 0, 0, 0, 0, 0, 1.2, 4.8, 4.3, 44.6, 6.8, 2.
## [16] "There are c(34.2, 6.7, 0, 1.4, 7.5, 22.5, 5.8, 5.5, 0, 0, 1.2, 0, 1.2, 1.9, 4, 0, 0, 7.7, 3.6,
## [17] "There are c(8.3, 0, 0, 0, 4.1, 12.6, 6.8, 8.7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1.4, 8.2, 0, 0, 0, 1
## [18] "There are c(0, 2.1, 9.7, 1.4, 1, 2.9, 0, 0, 0, 1.7, 6, 1.5, 2.1, 4.1, 0, 0, 0, 7.2, 2.2, 1.4,
## [19] "There are c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
```

```
algae.med[48,4:11]
```

```
## # A tibble: 1 x 8
##    mxPH  mnO2    Cl   NO3   NH4  oPO4   PO4  Chla
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  8.06  12.6     9  0.23    10     5     6   1.1
```

```
algae.med[62,4:11]
```

```
## # A tibble: 1 x 8
##    mxPH  mnO2    Cl   NO3   NH4  oPO4   PO4  Chla
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1   6.4   9.8  32.7  2.68  103.  40.2    14  5.48
```

```
algae.med[199,4:11]
```

```
## # A tibble: 1 x 8
##    mxPH  mnO2    Cl   NO3   NH4  oPO4   PO4  Chla
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     8   7.6  32.7  2.68  103.  40.2  103.  5.48
```

There are 200 observations

#d

```
#algae.med1<-algae.med %>% select(-season)%>% select(-size)%>% select(-speed)
#algae.med1
#cor_algae=cor((algae.med1),use = "pairwise.complete.obs")
#prediction <- predict(lm(PO4~oPO4, data = algae.med))
#prediction[28]
cor(algae.del%>%select(mxPH:Chla))
```

```
##             mxPH        mnO2          Cl        NO3         NH4        oPO4
## mxPH  1.00000000 -0.10269374  0.14709539 -0.1721302 -0.15429757  0.09022909
## mnO2 -0.10269374  1.00000000 -0.26324536  0.1179077 -0.07826816 -0.39375269
## Cl    0.14709539 -0.26324536  1.00000000  0.2109583  0.06598336  0.37925596
## NO3  -0.17213024  0.11790769  0.21095831  1.0000000  0.72467766  0.13301452
## NH4  -0.15429757 -0.07826816  0.06598336  0.7246777  1.00000000  0.21931121
## oPO4  0.09022909 -0.39375269  0.37925596  0.1330145  0.21931121  1.00000000
## PO4   0.10132957 -0.46396073  0.44519118  0.1570297  0.19939575  0.91196460
## Chla  0.43182377 -0.13121671  0.14295776  0.1454929  0.09120406  0.10691478
##            PO4        Chla
## mxPH  0.1013296  0.43182377
## mnO2 -0.4639607 -0.13121671
## Cl    0.4451912  0.14295776
## NO3   0.1570297  0.14549290
## NH4   0.1993958  0.09120406
## oPO4  0.9119646  0.10691478
## PO4   1.0000000  0.24849223
## Chla  0.2484922  1.00000000
```

```
model<-lm(algae$PO4~algae$oPO4)
x<-predict(model,algae[28,9])[28]
```

```
## Warning: 'newdata' had 1 row but variables found have 200 rows
```

```
x
```

```
##       28
## 48.06929
```

```
algae[28,10]<-x
```

48.06929 is our value for the 28th observation

#e It is possible that chemical abundance profile is related to the missing of some algae, and the chemical abundance profile of algae that survive in samples is possible to be different than the chemical abundance profile of missing algae. This difference in chemical abundance profile between missing and non-missing algae may contributes to survivorship bias, which imputed values might be a poor substitude, also we may loss information about algae boom.

##Problem4 #a

```
set.seed(666)
folds = sample(cut(1:nrow(algae.med), breaks=5, labels=FALSE))
folds
```

```
##   [1] 2 4 3 4 4 1 4 4 3 4 5 2 4 3 1 5 3 2 5 3 2 3 2 5 4 2 2 4 4 5 1 4 2 1 4 2 1
##  [38] 2 2 1 4 1 3 4 5 1 5 2 1 4 1 4 1 2 5 4 1 1 3 5 3 5 1 3 3 4 3 1 5 4 3 5 2 5
##  [75] 3 1 2 4 4 4 1 2 2 4 2 3 2 5 3 3 1 5 3 5 2 2 1 2 3 3 4 3 4 1 5 5 5 4 2 5 3
## [112] 5 3 1 1 2 3 2 1 3 3 2 1 5 5 4 4 2 4 3 5 1 1 1 4 3 5 3 1 3 2 2 2 1 5 2 3 2
## [149] 4 4 2 5 1 5 1 4 5 5 4 1 3 4 1 5 2 5 2 3 3 1 4 1 3 2 5 5 5 3 1 2 1 5 5 4 3
## [186] 3 1 3 4 1 5 2 5 2 1 5 4 2 3 4
```

```
do.chunk <- function(chunkid, chunkdef, dat){ # function argument
  train = (chunkdef != chunkid)
  Xtr = dat[train,1:11] # get training set
  Ytr = dat[train,12] # get true response values in trainig set
  Xvl = dat[!train,1:11] # get validation set
  Yvl = dat[!train,12] # get true response values in validation set
  lm.a1 <- lm(a1~., data = dat[train,1:12])
predYtr = predict(lm.a1) # predict training values
predYvl = predict(lm.a1,Xvl) # predict validation values
data.frame(fold = chunkid,
           train.error = mean((predYtr - Ytr$a1)^2),
           val.error = mean((predYvl - Yvl$a1)^2))
 }
lapply(c(1:5),do.chunk,chunkdef=folds,dat=algae.med)
```

```
## [[1]]
##   fold train.error val.error
## 1    1    254.9655  467.8999
##
## [[2]]
##   fold train.error val.error
## 1    2    267.0268  424.9731
##
## [[3]]
##   fold train.error val.error
## 1    3    307.1619  210.3142
##
## [[4]]
##   fold train.error val.error
## 1    4    285.3865  341.7519
##
## [[5]]
##   fold train.error val.error
## 1    5    282.6613  331.4836
```

##Problem5

```
algae.Test <- read_table2('algaeTest.txt', col_names=c('season','size','speed','mxPH','mnO2','Cl','NO3'
'NH4','oPO4','PO4','Chla','a1'), na=c('XXXXXXX'))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   season = col_character(),
##   size = col_character(),
##   speed = col_character(),
##   mxPH = col_double(),
##   mnO2 = col_double(),
##   Cl = col_double(),
##   NO3 = col_double(),
##   NH4 = col_double(),
##   oPO4 = col_double(),
##   PO4 = col_double(),
##   Chla = col_double(),
##   a1 = col_double()
## )
```

```
Xtrain=algae.med[,1:11]
Ytrain=algae.med[,12]
Xval=algae.Test[,1:11]
Yval=algae.Test[,12]
fit<-lm(a1~.,data=algae.med[,1:12])
predYtrain=predict(fit)
predYval=predict(fit,Xval)
train.error=mean(((predYtrain - Ytrain)^2)$a1)
```

```
test.error=mean(((predYval - Yval)^2)$a1)
data.frame(train.error, test.error)
```

```
##   train.error test.error
## 1    286.2661   250.1794
```

```
a=(284.9137+290.9481+274.9146+253.3843+296.4739)/5
a
```

```
## [1] 280.1269
```

```
b=(310.8238+288.1278+389.9608+453.7588+289.3328)/5
b
```

```
## [1] 346.4008
```

The test error from problem 4 is 453.7588 , which is larger than the ???true??? test error in problem 5. This is not what i expected for most cases, the ???true??? test error should be larger.

##Problem6 #a

```
library(ISLR)
head(Wage)
```

```
##        year  age          maritl     race       education             region
## 231655 2006  18 1. Never Married 1. White    1. < HS Grad 2. Middle Atlantic
## 86582  2004  24 1. Never Married 1. White 4. College Grad 2. Middle Atlantic
## 161300 2003  45       2. Married 1. White 3. Some College 2. Middle Atlantic
## 155159 2003  43       2. Married 3. Asian 4. College Grad 2. Middle Atlantic
## 11443  2005  50      4. Divorced 1. White      2. HS Grad 2. Middle Atlantic
## 376662 2008  54       2. Married 1. White 4. College Grad 2. Middle Atlantic
##              jobclass         health health_ins  logwage      wage
## 231655  1. Industrial      1. <=Good      2. No 4.318063  75.04315
## 86582  2. Information 2. >=Very Good      2. No 4.255273  70.47602
## 161300  1. Industrial      1. <=Good     1. Yes 4.875061 130.98218
## 155159 2. Information 2. >=Very Good     1. Yes 5.041393 154.68529
## 11443  2. Information      1. <=Good     1. Yes 4.318063  75.04315
## 376662 2. Information 2. >=Very Good     1. Yes 4.845098 127.11574
```

```
ggplot(Wage, aes(Wage$age,Wage$wage)) + geom_point()+geom_smooth()
```

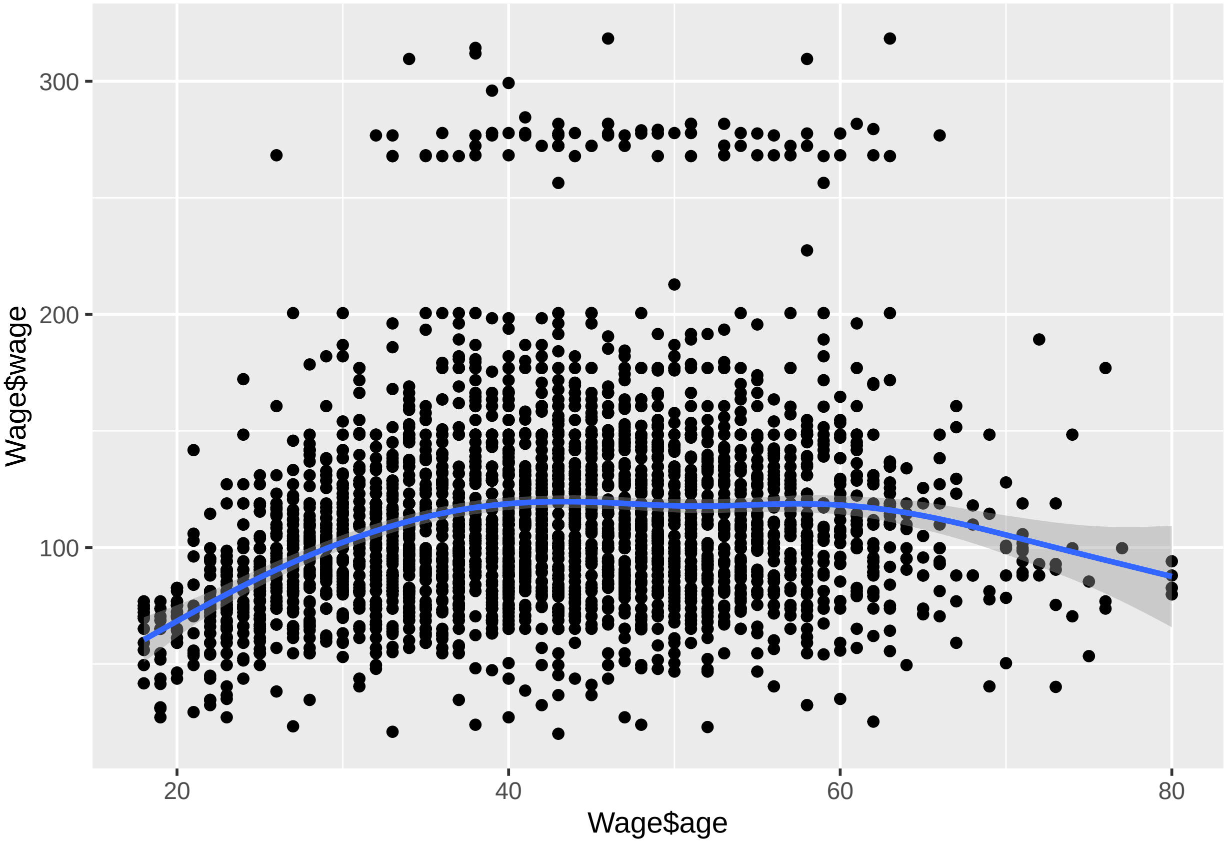
```
## Warning: Use of `Wage$age` is discouraged. Use `age` instead.

## Warning: Use of `Wage$wage` is discouraged. Use `wage` instead.

## Warning: Use of `Wage$age` is discouraged. Use `age` instead.

## Warning: Use of `Wage$wage` is discouraged. Use `wage` instead.

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



This pattern looks like an inverse parabola. The wage is lowest smaller than 20, it increases as age is increasing until 40, then it becomes relatively steady. From 60 to 80, it???s decreasing, and i believe it???s because most people are retired around that age. As we can see, the wage is lowest at smaller than 20, and it increases as age until it reaches 40. Then, it becomes quite steady from 40 to 60. After 60, it gradually decreases until age of 80. I belive most people retired around 65.

#b i)

```
for(p in 10){ if(p==0){
m1=lm(wage~1,data=Wage)
print(m1) }
else{
mp = lm(wage~poly(age,degree=p,raw=FALSE),data=Wage)
print(mp)
} }
```

```
##
## Call:
## lm(formula = wage ~ poly(age, degree = p, raw = FALSE), data = Wage)
##
## Coefficients:
##                          (Intercept)   poly(age, degree = p, raw = FALSE)1
##                              111.704                               447.068
##  poly(age, degree = p, raw = FALSE)2   poly(age, degree = p, raw = FALSE)3
##                             -478.316                               125.522
##  poly(age, degree = p, raw = FALSE)4   poly(age, degree = p, raw = FALSE)5
##                              -77.911                               -35.813
##  poly(age, degree = p, raw = FALSE)6   poly(age, degree = p, raw = FALSE)7
##                               62.708                                50.550
##  poly(age, degree = p, raw = FALSE)8  poly(age, degree = p, raw = FALSE)9
##                              -11.255                              -83.692
## poly(age, degree = p, raw = FALSE)10
##                                1.624
```

ii)

```
set.seed(333)
folds = sample(cut(1:nrow(Wage), breaks=5, labels=FALSE))
do.chunk2 <- function(chunkid, chunkdef, dat,p){ # function argument
train = (chunkdef != chunkid)
Xtr = dat[train,2] # get training set
Ytr = dat[train,11] # get true response values in trainig set
Xvl = dat[!train,2] # get validation set
Yvl = dat[!train,11] # get true response values in validation set
if(p==0){ fit<-lm(wage~1,data=dat[train,c(2,11)])} else{
fit<-lm(wage~poly(age,degree=p,raw=FALSE),data=dat[train,c(2,11)]) }
predYtr = predict(fit) # predict training values
predYvl = predict(fit,data.frame(age=Xvl)) # predict validation values
data.frame(fold = chunkid,
train.error = mean((predYtr - Ytr)^2), val.error = mean((predYvl - Yvl)^2))
}
r1<-ldply(1:5,do.chunk2,folds,Wage,0)
r1
```

```
##   fold train.error val.error
## 1    1    1686.239  1959.814
## 2    2    1714.821  1844.475
## 3    3    1783.122  1572.594
## 4    4    1772.133  1616.190
## 5    5    1745.643  1724.066
```

```
r2<-ldply(1:5,do.chunk2,folds,Wage,1)
r2
```

```
##   fold train.error val.error
## 1    1    1617.593  1901.817
## 2    2    1659.761  1734.209
## 3    3    1711.219  1527.514
## 4    4    1705.630  1549.463
## 5    5    1673.924  1677.445
```

```
r3<-ldply(1:5,do.chunk2,folds,Wage,2)
r3
```

```
##   fold train.error val.error
## 1    1    1545.088  1811.942
## 2    2    1584.143  1654.128
## 3    3    1627.820  1480.676
## 4    4    1632.939  1458.542
## 5    5    1596.884  1603.329
```

```
r4<-ldply(1:5,do.chunk2,folds,Wage,3)
r4
```

```
##   fold train.error val.error
## 1    1    1541.868  1799.690
## 2    2    1577.704  1653.887
## 3    3    1622.452  1475.960
## 4    4    1623.989  1471.716
## 5    5    1593.487  1591.589
```

```
r5<-ldply(1:5,do.chunk2,folds,Wage,4)
r5
```

```
##   fold train.error val.error
## 1    1    1538.451  1803.601
## 2    2    1575.983  1650.840
## 3    3    1621.457  1470.675
## 4    4    1621.761  1469.663
## 5    5    1591.517  1589.586
```

```
r6<-ldply(1:5,do.chunk2,folds,Wage,5)
r6
```

```
##   fold train.error val.error
## 1    1    1538.170  1802.506
## 2    2    1574.532  1655.893
## 3    3    1620.964  1470.499
## 4    4    1621.189  1469.654
## 5    5    1591.517  1589.503
```

```
r7<-ldply(1:5,do.chunk2,folds,Wage,6)
r7
```

```
##   fold train.error val.error
## 1    1    1535.725  1806.819
## 2    2    1573.636  1653.198
## 3    3    1620.572  1466.826
## 4    4    1619.606  1469.730
## 5    5    1589.665  1590.637
```

```
r8<-ldply(1:5,do.chunk2,folds,Wage,7)
r8
```

```
##   fold train.error val.error
## 1    1    1534.967  1805.623
## 2    2    1570.750  1663.980
## 3    3    1620.019  1464.995
## 4    4    1618.492  1470.785
## 5    5    1589.553  1588.541
```

```
r9<-ldply(1:5,do.chunk2,folds,Wage,8)
r9
```

```
##   fold train.error val.error
## 1    1    1534.933  1806.321
## 2    2    1570.745  1664.182
## 3    3    1620.012  1464.866
## 4    4    1617.768  1475.367
## 5    5    1589.464  1588.743
```

```
r10<-ldply(1:5,do.chunk2,folds,Wage,9)
r10
```

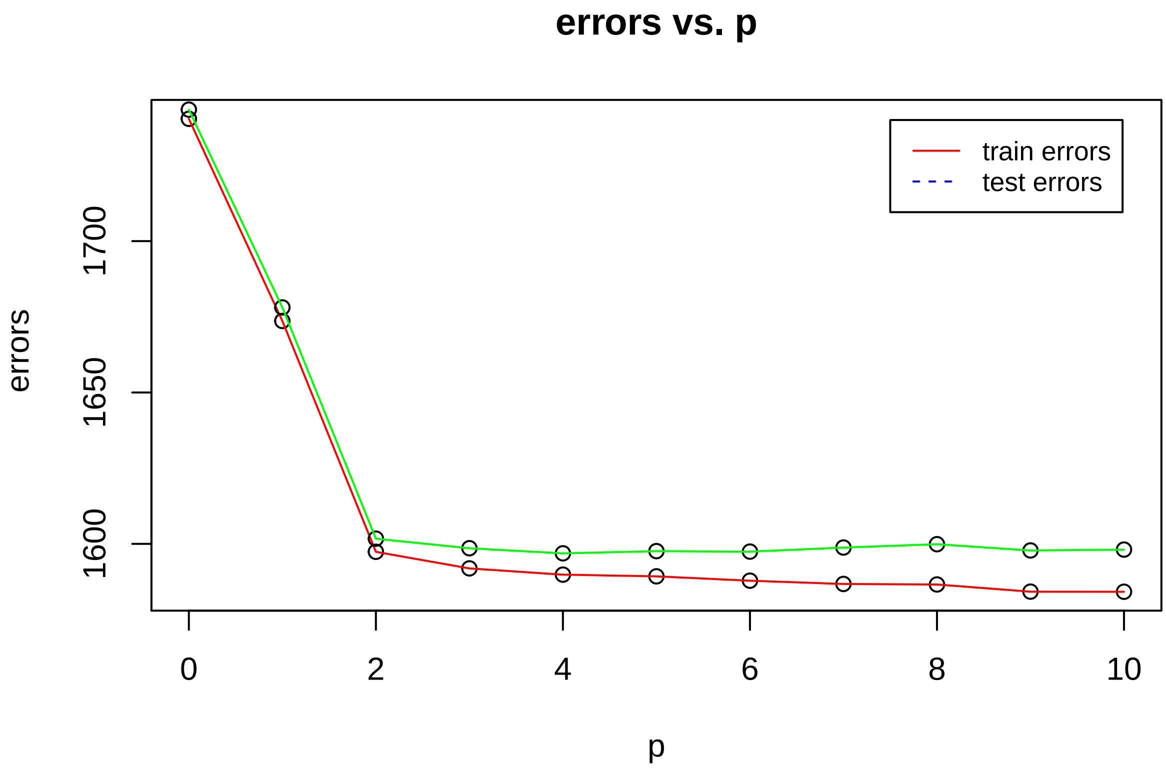
```
##   fold train.error val.error
## 1    1    1532.757  1803.889
## 2    2    1568.553  1661.147
## 3    3    1617.401  1463.593
## 4    4    1614.435  1477.071
## 5    5    1587.939  1583.432
```

```
r11<-ldply(1:5,do.chunk2,folds,Wage,10)
r11
```

```
##   fold train.error val.error
## 1    1    1532.716  1804.223
## 2    2    1568.504  1661.516
## 3    3    1617.401  1463.595
## 4    4    1614.431  1477.163
## 5    5    1587.877  1584.041
```

#c

```
x<-1:11
y1<- c(mean(r1$train.error),mean(r2$train.error),mean(r3$train.error),mean(r4$train.error),mean(r5$trai
y2<- c(mean(r1$val.error),mean(r2$val.error),mean(r3$val.error),mean(r4$val.error) ,mean(r5$val.error),m
plot(0:10, y1,main = "errors vs. p", xlab = "p", ylab=" errors")
lines(0:10, y1,col="red")
points(0:10,y2,pch=1)
lines(0:10,y2,col="green")
legend(7.5, 1740, legend=c("train errors", "test errors"),
col=c("red", "blue"), lty=1:2, cex=0.8)
```



From the graph, we can see after p=2 the test error change slightly, so we should choose p=2, it means meaning wage~1+age+I(age??2).