PSTAT 131 Homework 1

Tanner Berney 7215445

4/15/2021

```
library(tidyverse)
```

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

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

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

```
library(reshape2)
```

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

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

```
algae <- read_table2("algaeBloom.txt", col_names= c('season','size','speed','mxPH','mnO2','Cl','NO3','N
  '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()
## )
```

```
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, ...
```

Question 1 a-c

1a)

```
algae %>%
  group_by(season) %>%
  summarise(length(season))
```

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

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

40 obs in autumn, 53 obs in spring, 45 obs in summer, and 62 obs in winter.

1b)

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

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

```
algae %>%
  select(mxPH,mnO2,Cl,NO3,NH4,oPO4,PO4,Chla) %>%
  filter(!is.na(mxPH),!is.na(mnO2),!is.na(Cl),!is.na(NO3),!is.na(NH4),!is.na(oPO4),!is.na(PO4),!is.na(Cl
  summarise(across(c(mxPH,mnO2,Cl,NO3,NH4,oPO4,PO4,Chla),list(mean=mean,var=var)))
```

```
## # A tibble: 1 x 16
##   mxPH_mean mxPH_var mnO2_mean mnO2_var Cl_mean Cl_var NO3_mean NO3_var NH4_mean
##       <dbl>    <dbl>     <dbl>    <dbl>   <dbl>  <dbl>    <dbl>   <dbl>    <dbl>
## 1      8.08    0.223      9.02     5.79    44.9  2215.     3.38    15.0     538.
## # ... with 7 more variables: NH4_var <dbl>, oPO4_mean <dbl>, oPO4_var <dbl>,
## #   PO4_mean <dbl>, PO4_var <dbl>, Chla_mean <dbl>, Chla_var <dbl>
```

Yes, 33 missing values total. Looking at the magnitude of the two quantities, it is very apparent that the means and standard deviation of each variable differ. For example the mean of NO3 is 3.38 where as NH4 has a mean of 2031.58. The standard deviation of mxPH is 0.47 where as the standard deviation for Cl is 47.06. These values could be due to outliers found in some of the chemical observations.

1c)

```
algae %>%
  select(mxPH,mnO2,Cl,NO3,NH4,oPO4,PO4,Chla) %>%
  filter(!is.na(mxPH),!is.na(mnO2),!is.na(Cl),!is.na(NO3),!is.na(NH4),!is.na(oPO4),!is.na(PO4),!is.na(Cl
  summarise(across(c(mxPH,mnO2,Cl,NO3,NH4,oPO4,PO4,Chla),list(median=median,mad=mad)))
```

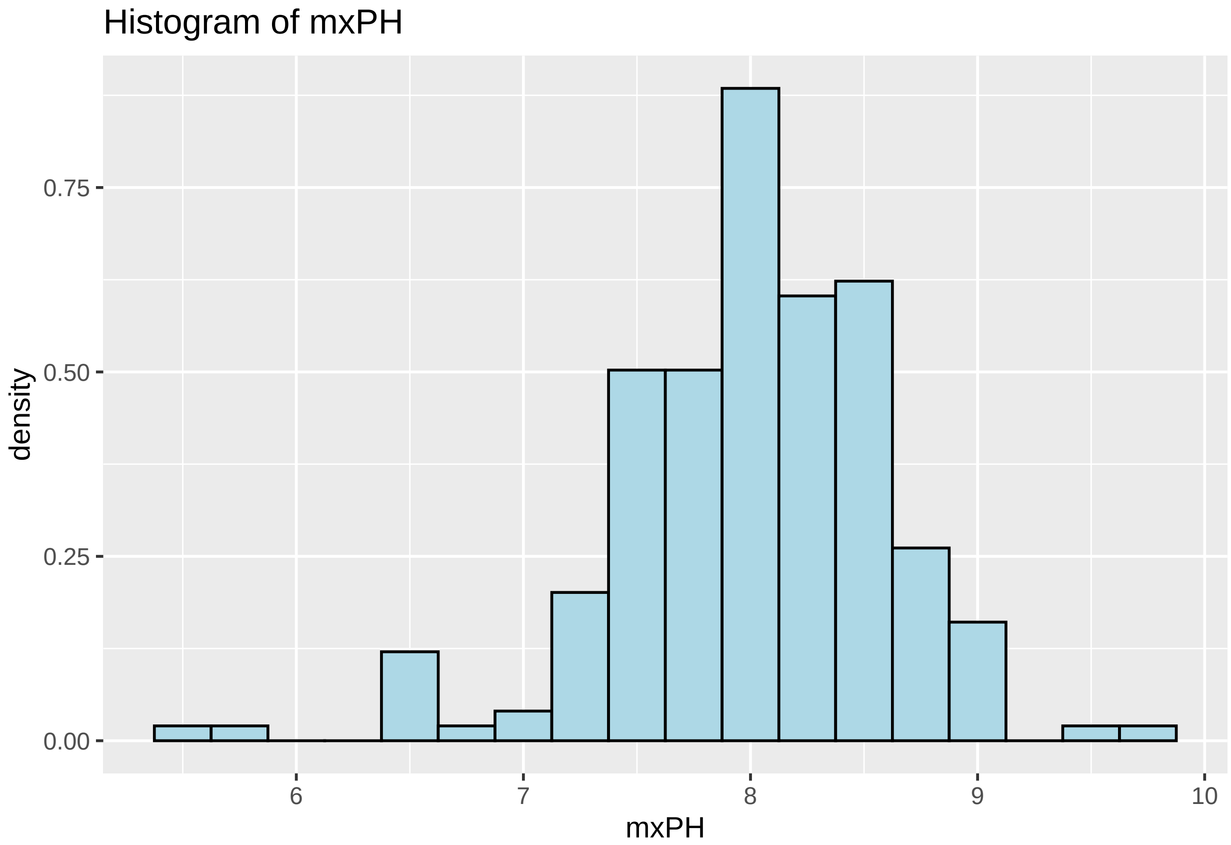
```
## # A tibble: 1 x 16
##   mxPH_median mxPH_mad mnO2_median mnO2_mad Cl_median Cl_mad NO3_median NO3_mad
##         <dbl>    <dbl>       <dbl>    <dbl>     <dbl>  <dbl>      <dbl>   <dbl>
## 1         8.1    0.445        9.75     2.00      35.1   34.5       2.82    2.31
## # ... with 8 more variables: NH4_median <dbl>, NH4_mad <dbl>,
## #   oPO4_median <dbl>, oPO4_mad <dbl>, PO4_median <dbl>, PO4_mad <dbl>,
## #   Chla_median <dbl>, Chla_mad <dbl>
```

Comparing the two set of quantities we can see that the medians of each chemical is fairly close to the mean of each chemical along with the medians of each having a fairly large difference. We can also see that the MAD and median of each chemical is also fairly close to one another with the exception of mxPH and mmO2. It is also interesting to see that the MAD and standard deviation of each chemical are fairly close to one another as well.

Question 2 a-e

2a)

```
ggplot(algae,aes(x=mxPH))+
  geom_histogram(binwidth=.25,color="black",fill="lightblue",na.rm=T,aes(y=after_stat(density)))+
  labs(title="Histogram of mxPH")
```

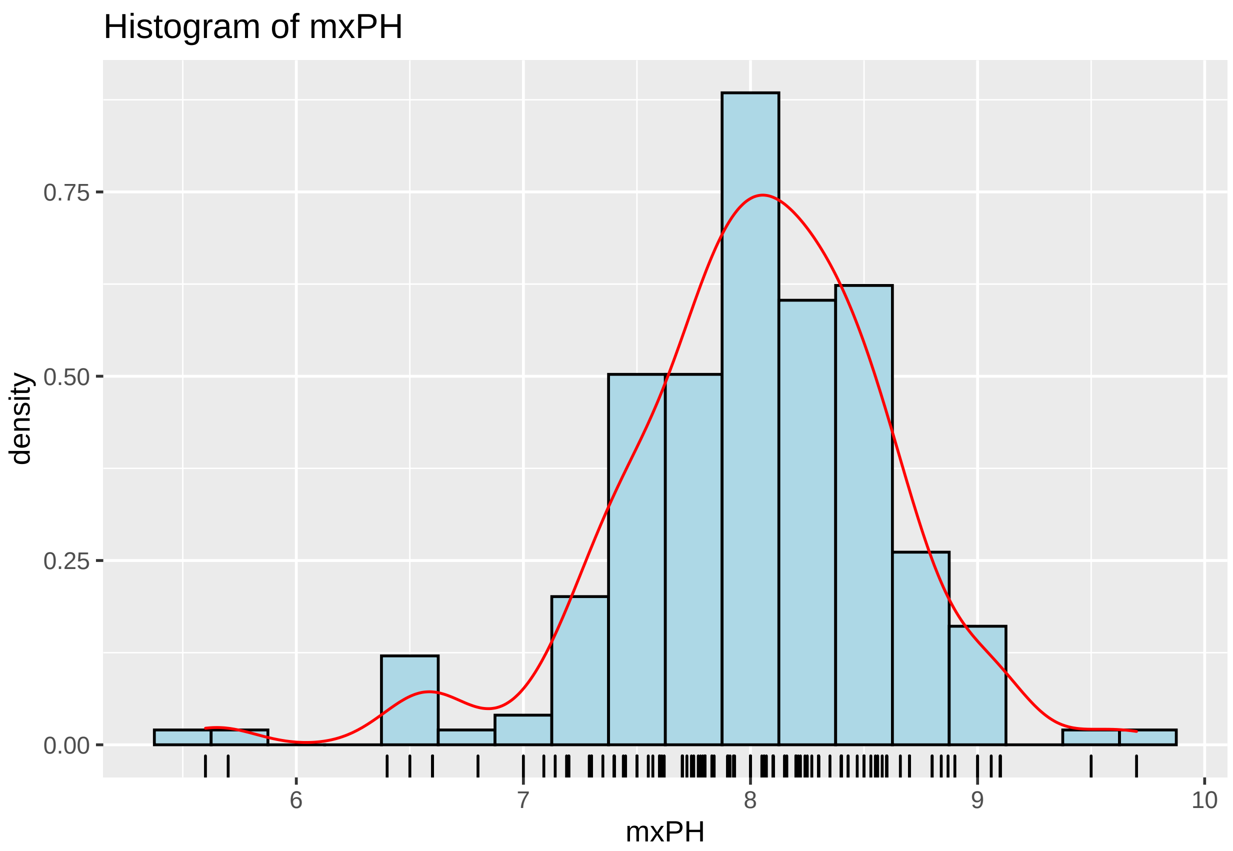


The distribution is NOT skewed, it has a symmetric bell looking shape. If anything, it looks to be SLIGHTLY skewed to the left.

2b)

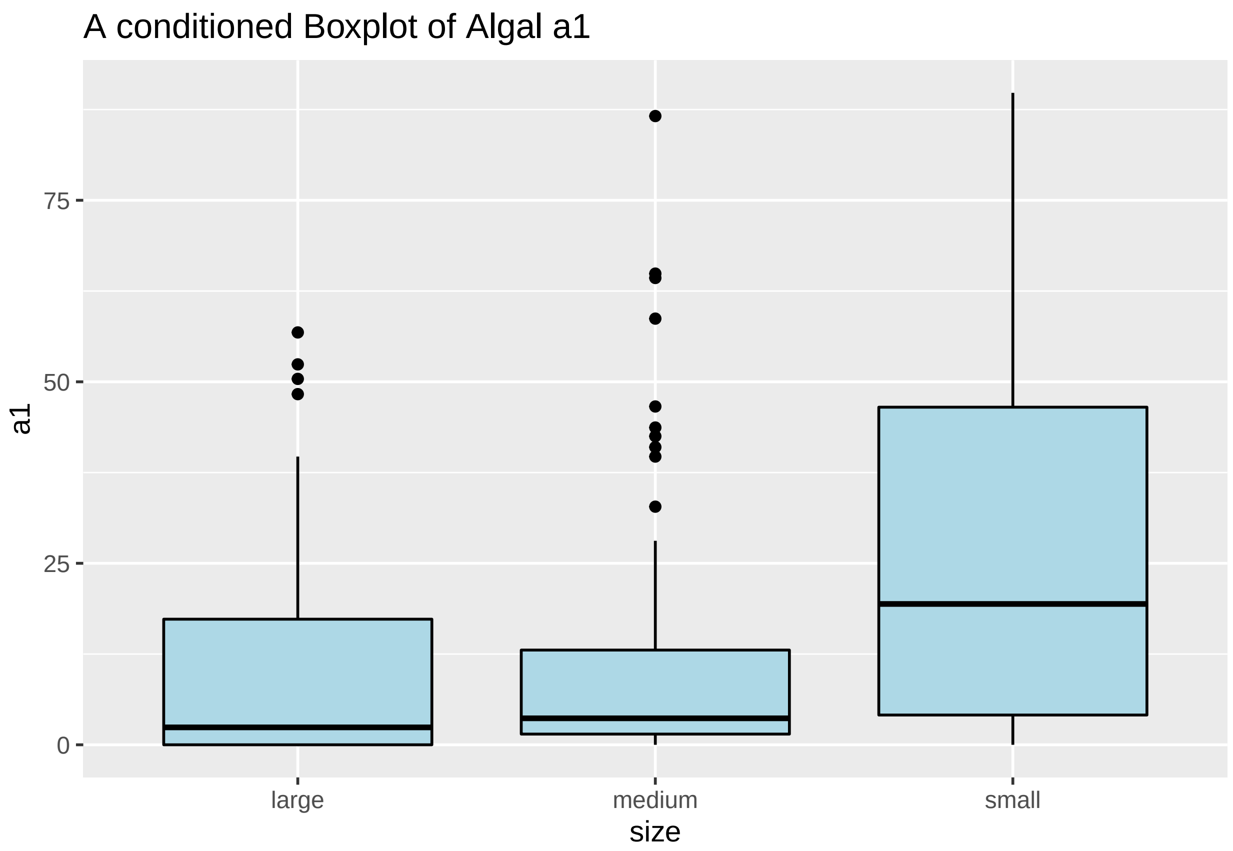
```
ggplot(algae,aes(x=mxPH))+
  geom_histogram(binwidth=.25,color="black",fill="lightblue",na.rm=T,aes(y=after_stat(density)))+
  labs(title="Histogram of mxPH")+geom_density(color="red")+
  geom_rug()
```

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



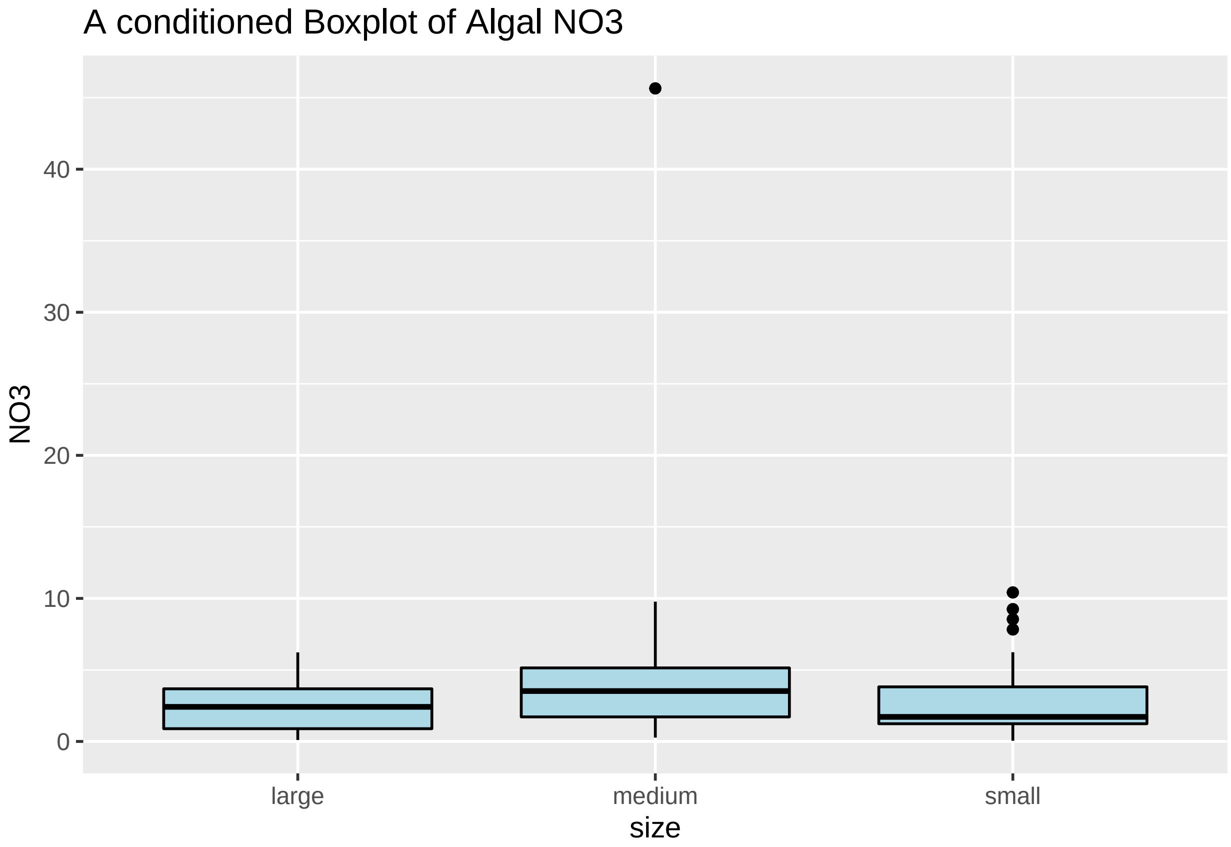
2c)

```
ggplot(algae,aes(x=size,y=a1))+
  labs(title="A conditioned Boxplot of Algal a1")+
  geom_boxplot(color="black",fill="lightblue")
```

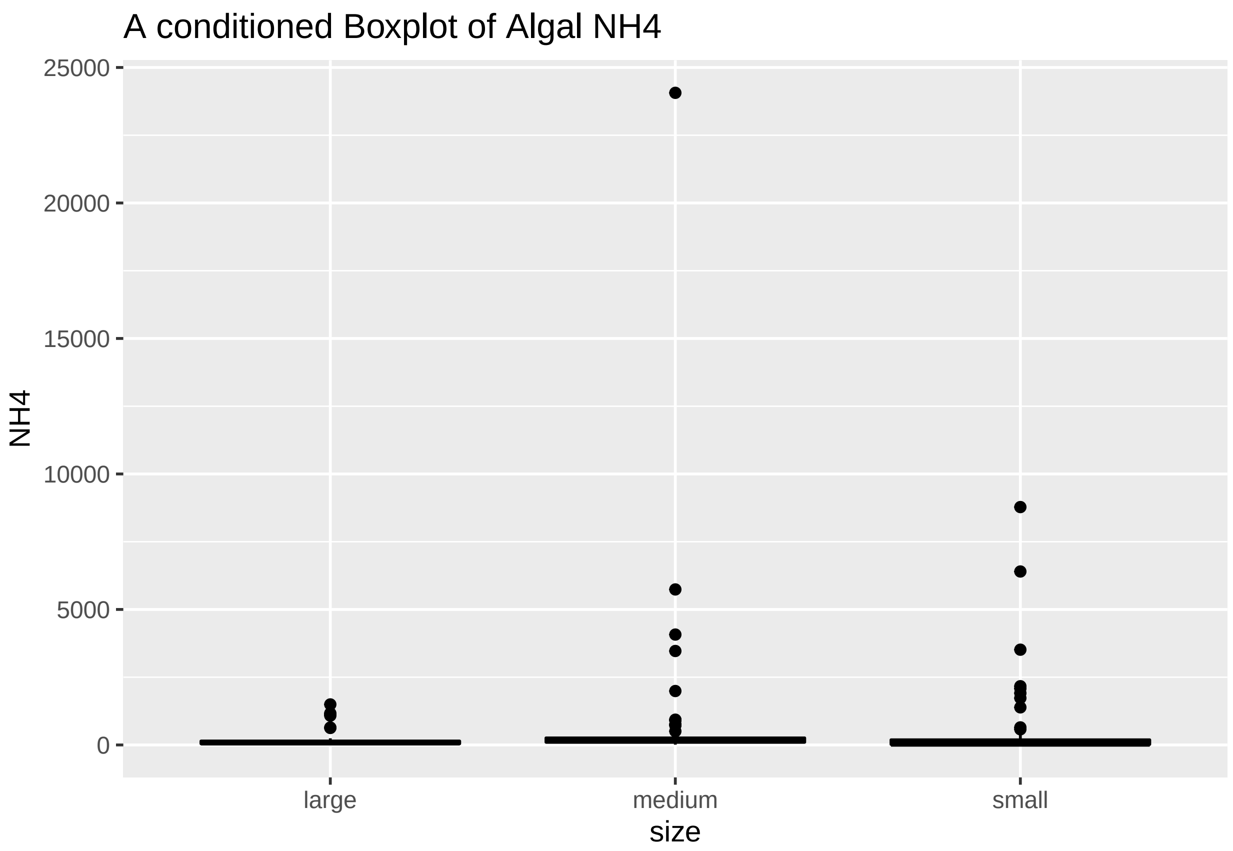


2d)

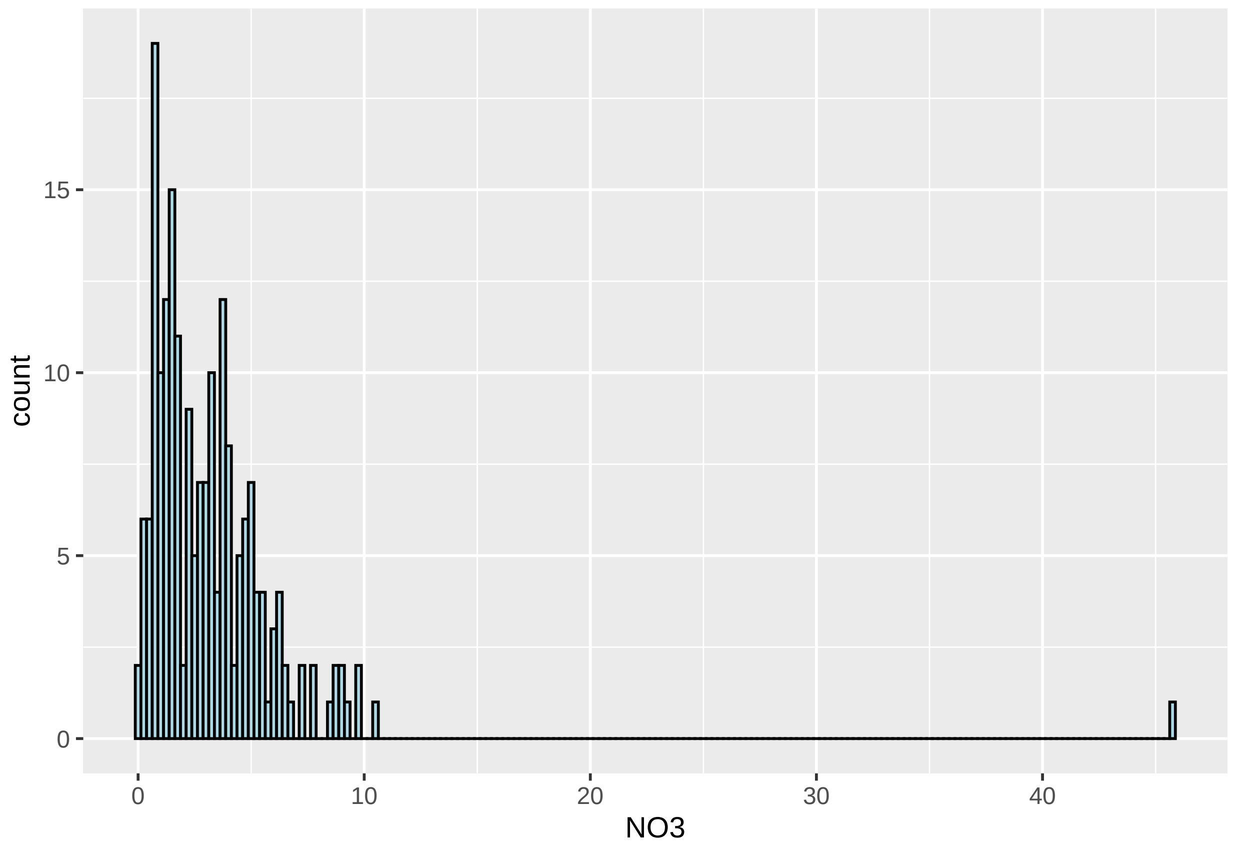
```
ggplot(algae,aes(x=size,y=NO3))+
  labs(title="A conditioned Boxplot of Algal NO3")+
  geom_boxplot(color="black",fill="lightblue",na.rm = T)
```



```
ggplot(algae,aes(x=size,y=NH4))+
  labs(title="A conditioned Boxplot of Algal NH4")+
  geom_boxplot(color="black",fill="lightblue",na.rm = T)
```



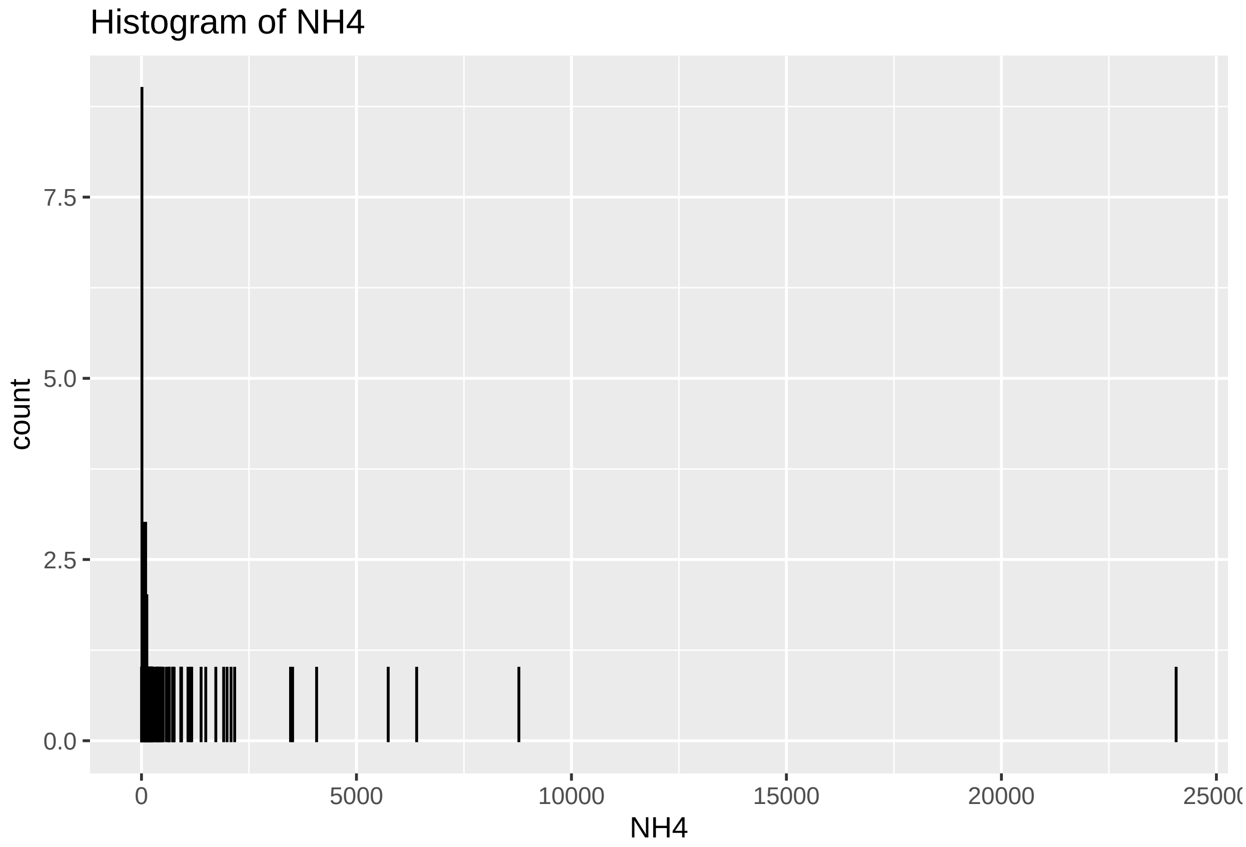
```
ggplot(algae,aes(x=NO3))+
  geom_histogram(binwidth=.25,color="black",fill="lightblue",na.rm=T)
```



```
labs(title="Histogram of NO3")
```

```
## $title
## [1] "Histogram of NO3"
##
## attr(,"class")
## [1] "labels"
```

```
ggplot(algae,aes(x=NH4))+
  geom_histogram(binwidth=.25,color="black",fill="lightblue",na.rm=T)+
  labs(title="Histogram of NH4")
```



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

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

```
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
```

```
max(algae$NH4,na.rm=T)
```

```
## [1] 24064
```

```
max(algae$NO3,na.rm=T)
```

```
## [1] 45.65
```

Yes there is an outlier for both NH4 and NO3, but I would only consider one observation to be considered the outlier, which can be found to be the 153rd observation in NH4 and NO3. I came to this conclusion by plotting box plots of each along with histograms to get a better

sense of the outliers. Along with using boxplot.stats() to see what values seemed to be deemed outliers by R software. After close inspection, the only big outlier was found to be the 153rd observation where the other values were close enough in the histogram/boxplots to not be considered outliers to me.

2e)

From the results of 1c) we have N03 with mean 3.38, variance 15.01, median 2.82, and MAD 2.31. NH4 has mean 537.6, variance 4127337, median 115.7, and MAD 120.9. Since mean and variance take outliers into account, it makes them sensitive when there is an outlier present. Because of this, those two estimators are not the best choice in this case. Using Median and MAD are less sensitive to any outliers so they are more robust when outliers are present. This also helps to show why the mean/variance differ so much from the median and MAD.

Question 3 a-e

3a)

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

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

```
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
```

There are 33 NA values. Number of missing values per chemical: mxph=1, mnO2=2, Cl=10, NO3=2, NH4=2, oPO4=2, PO4=2, Chla=12, all other chemicals have 0 missing values.

3b)

```
algae.del <- algae %>%
  filter(complete.cases(.))
nrow(algae.del)
```

```
## [1] 184
```

There are 184 observations in the algae.del data set.

3c)

```
algae.med <- algae %>%
  mutate_at(.vars=vars(mxPH,mnO2,Cl,NO3,NH4,oPO4,PO4,Chla),.funs=list(~ifelse(is.na(.), median(., na.rm
algae.med[c(48,62,199),]
```

```
## # A tibble: 3 x 18
##   season size  speed  mxPH  mnO2    Cl   NO3   NH4  oPO4   PO4  Chla    a1    a2
##   <chr>  <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 winter small low    8.06  12.6   9    0.23   10    5      6   1.1   35.5   0
## 2 summer small medi~  6.4    9.8  32.7  2.68  103.  40.2   14   5.48  19.4   0
## 3 winter large medi~  8      7.6  32.7  2.68  103.  40.2  103.  5.48   0    12.5
## # ... with 5 more variables: a3 <dbl>, a4 <dbl>, a5 <dbl>, a6 <dbl>, a7 <dbl>
```

3d)

```
cor(x=algae.med[4:13], use= "pairwise.complete.obs")
```

```
##              mxPH        mnO2          Cl         NO3         NH4       oPO4
## mxPH  1.00000000 -0.16793588  0.13348318 -0.12637570 -0.08905891  0.1604940
## mnO2 -0.16793588  1.00000000 -0.27790470  0.09853221 -0.08731331 -0.4150941
## Cl    0.13348318 -0.27790470  1.00000000  0.22532102  0.07450448  0.3927796
## NO3  -0.12637570  0.09853221  0.22532102  1.00000000  0.72152844  0.1450640
## NH4  -0.08905891 -0.08731331  0.07450448  0.72152844  1.00000000  0.2277842
## oPO4  0.16049404 -0.41509407  0.39277958  0.14506398  0.22778417  1.0000000
## PO4   0.18976104 -0.48641358  0.45668016  0.16988077  0.20913887  0.9132424
## Chla  0.38915072 -0.16571514  0.15158609  0.14342461  0.09447493  0.1307048
## a1   -0.26427002  0.28581138 -0.35932387 -0.24023942 -0.13172053 -0.4151466
## a2    0.32840046 -0.10150105  0.08957723  0.02382355 -0.02940305  0.1477486
##              PO4        Chla         a1          a2
## mxPH  0.1897610  0.38915072 -0.2642700  0.32840046
## mnO2 -0.4864136 -0.16571514  0.2858114 -0.10150105
## Cl    0.4566802  0.15158609 -0.3593239  0.08957723
## NO3   0.1698808  0.14342461 -0.2402394  0.02382355
## NH4   0.2091389  0.09447493 -0.1317205 -0.02940305
## oPO4  0.9132424  0.13070484 -0.4151466  0.14774857
## PO4   1.0000000  0.26920346 -0.4847729  0.16446431
## Chla  0.2692035  1.00000000 -0.2817370  0.38141781
## a1   -0.4847729 -0.28173702  1.0000000 -0.29376781
## a2    0.1644643  0.38141781 -0.2937678  1.00000000
```

```
PO4_predict <- predict(lm(PO4~oPO4, data = algae.med))
PO4_predict[28]
```

```
##       28
## 48.04407
```

We get a value of 48.04407.

3e)

Using the correlation of other predictor variables can leave us with missing values which is a poor substitution attempt. If given a data set with a large amount of missing values in it, this method will not be useful to us. It will instead leave the values that have NA unchanged which does not help us.

Question 4 a-b

4a)

```
set.seed(50)
chunks <- cut((1:nrow(algae)), breaks=5, labels= FALSE) %>%
  sample()
```

4b)

```
set.seed(333)
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 training 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(as.matrix((predYtr - Ytr)^2)), # compute and store training error
val.error = mean(as.matrix((predYvl - Yvl)^2))) # compute and store test error
}
print(lapply(1:5,FUN=do.chunk,chunkdef=chunks,dat=algae.med))
```

```
## [[1]]
##   fold train.error val.error
## 1    1    275.2421  357.3675
##
## [[2]]
##   fold train.error val.error
## 1    2    282.7523  320.1005
##
## [[3]]
##   fold train.error val.error
## 1    3    310.5641  208.2703
##
## [[4]]
##   fold train.error val.error
## 1    4    282.0022  325.2386
##
## [[5]]
##   fold train.error val.error
## 1    5    257.2879  444.9756
```

Question 5a

```
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()
## )
```

```
a1_predict <- predict(lm(a1~season+size+speed+mxPH+mnO2+Cl+NO3+NH4+oPO4+PO4+Chla,data=algae.med),data=a
a1_true <- algae.Test[,12]
mean(as.matrix((a1_predict - a1_true)^2))
```

```
## [1] 596.3176
```

Looking at the CV test error from part 4 and comparing to question 5 CV test error, we see that the difference is about 200 which seems large to me. I expected the values to be a little closer to one another however.

Question 6 a-c

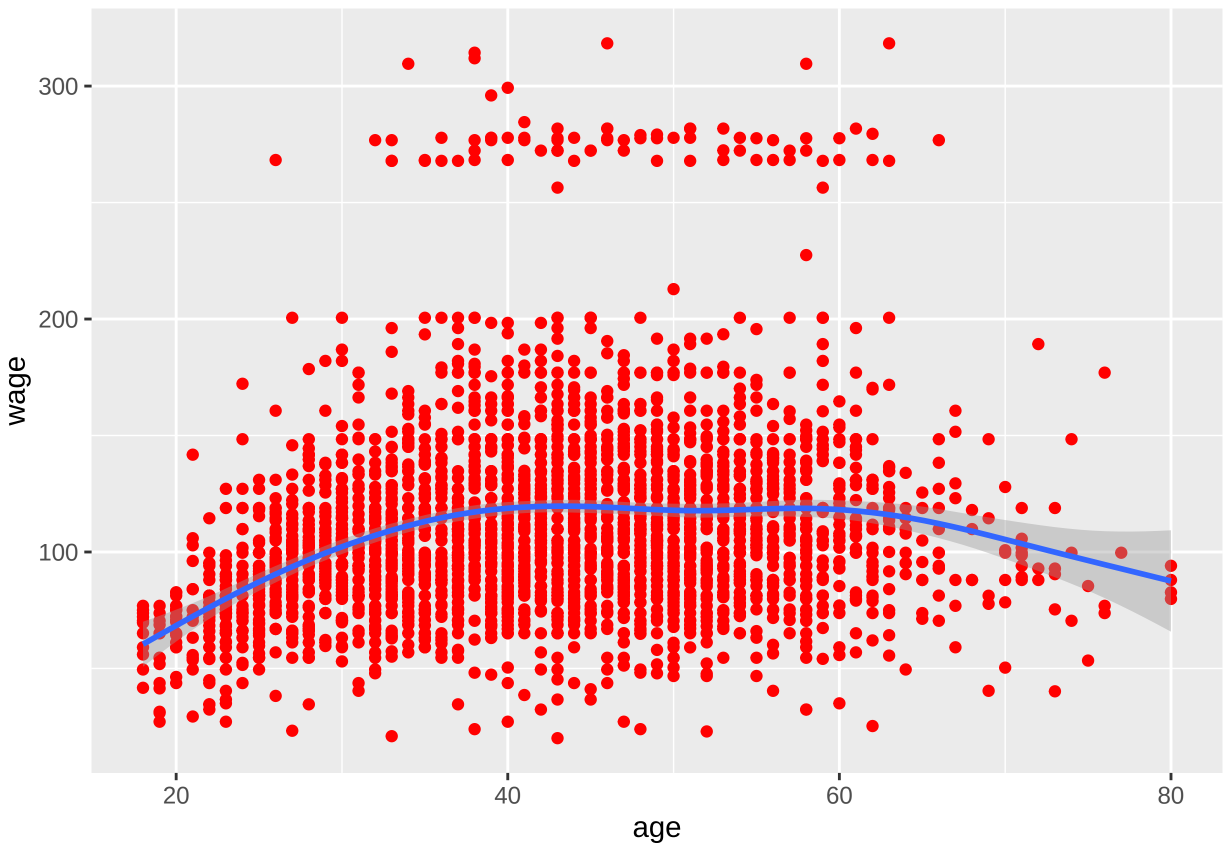
6a)

```
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(x=age,y=wage))+geom_point(color="red")+geom_smooth()
```

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



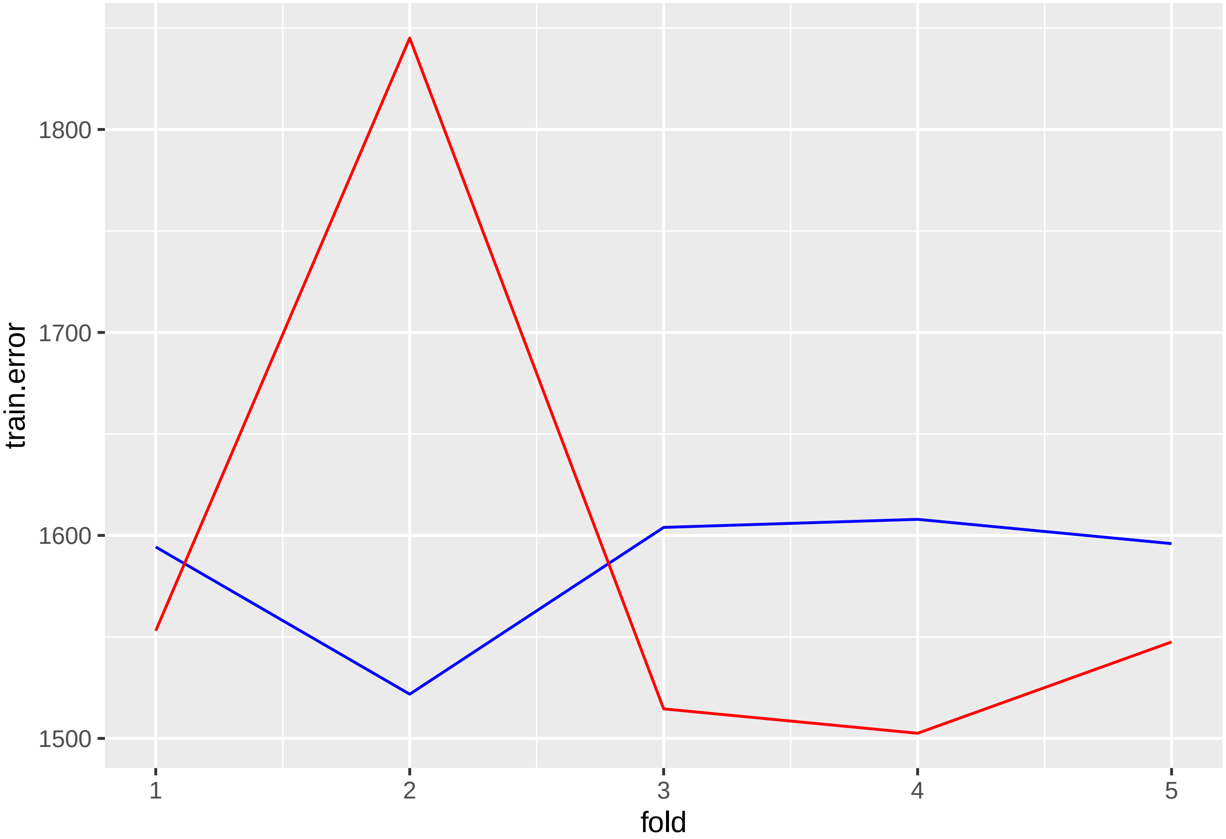
Looking at the visualization, it seems to me that as age increases so does the wages. However, at about age 65 the wages start to decrease and be similar to those of younger people. This is what I expected as people get older their wages increase. However, I thought wages would be higher for people 65 and older.

6b)

```
set.seed(211)
do.chunk.2 <- function(chunkid, chunkdef, dat){ # function argument
train = (chunkdef != chunkid)
Xtr = dat[train,1:10] # get training set
Ytr = dat[train,11] # get true response values in training set
Xvl = dat[!train,1:10] # get validation set
Yvl = dat[!train,11] # get true response values in validation set
lm.age <- lm(wage~poly(age,degree=10,raw=F),data=dat[train,1:11])
predYtr = predict(lm.age) # predict training values
predYvl = predict(lm.age,Xvl) # predict validation values
data.frame(fold = chunkid,
train.error = mean(as.matrix((predYtr - Ytr)^2)), # compute and store training error
val.error = mean(as.matrix((predYvl - Yvl)^2))) # compute and store test error
}
set.seed(111)
chunks.wage <- cut((1:nrow(Wage)), breaks=5, labels= FALSE) %>%
  sample()
errors <- lapply(1:5,FUN=do.chunk.2,chunkdef=chunks.wage,dat=Wage)
errors.1 = melt(errors, id.vars=c(’fold’, ’train.error’,"val.error"), value.name=’error’)
```

6c)

```
ggplot()+
    geom_line(errors.1,mapping= aes(x=fold,y=train.error),color="blue")+
  geom_line(errors.1,mapping=aes(x=fold,y=val.error),color="red")
```



As p increases the training error decreases and then increases to follow a steady line. The test error starts off by increasing a lot and then after drops down to nearing zero. We should select model 2 since the test error is much larger than the training error.