Pstat 131 HW1

Kalvin Goode (9454554), Patrick Chen (970890)

2019/4/11

```
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")
glimpse(algae)
```

1.

(a)

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

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

(b)

```
!all(!is.na(algae)) #If no missing value, is.na() shows all false, ! makes them all TRUE.
```

```
## [1] TRUE
```

```
#So, all() shows TRUE only if no missing value in data
```

Yes, there are missing values.

```
algae %>%
  dplyr::summarize_each(funs(mean(.,na.rm=TRUE)),c('mxPH':'Chla'))
```

```
## # A tibble: 1 x 8
##    mxPH  mnO2    Cl   NO3   NH4  oPO4   PO4  Chla
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  8.01  9.12  43.6  3.28  501.  73.6  138.  14.0
```

```
algae %>%
  dplyr::summarize_each(funs(var(.,na.rm=TRUE)),c('mxPH':'Chla'))
```

```
## # A tibble: 1 x 8
##    mxPH  mnO2    Cl   NO3      NH4  oPO4    PO4  Chla
##   <dbl> <dbl> <dbl> <dbl>    <dbl> <dbl>  <dbl> <dbl>
## 1 0.358  5.72 2193.  14.3 3851585. 8306. 16639.  420.
```

We notice that among these chemicals, Cl, NH4, oPO4 and PO4 have large mean and variance ratio. Especially for NH4 and PO4, which have variances 100 times greater then their mean.

(c)

```
algae %>%
  dplyr::summarize_each(funs(median(.,na.rm=TRUE)),c('mxPH':'Chla'))
```

```
## # A tibble: 1 x 8
##    mxPH  mn02    Cl   NO3   NH4  oPO4   PO4  Chla
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  8.06   9.8  32.7  2.68  103.  40.2  103.  5.48
```

```
algae %>%
  dplyr::summarize_each(funs(mad(.,constant=1,na.rm=TRUE)),c('mxPH':'Chla'))
```

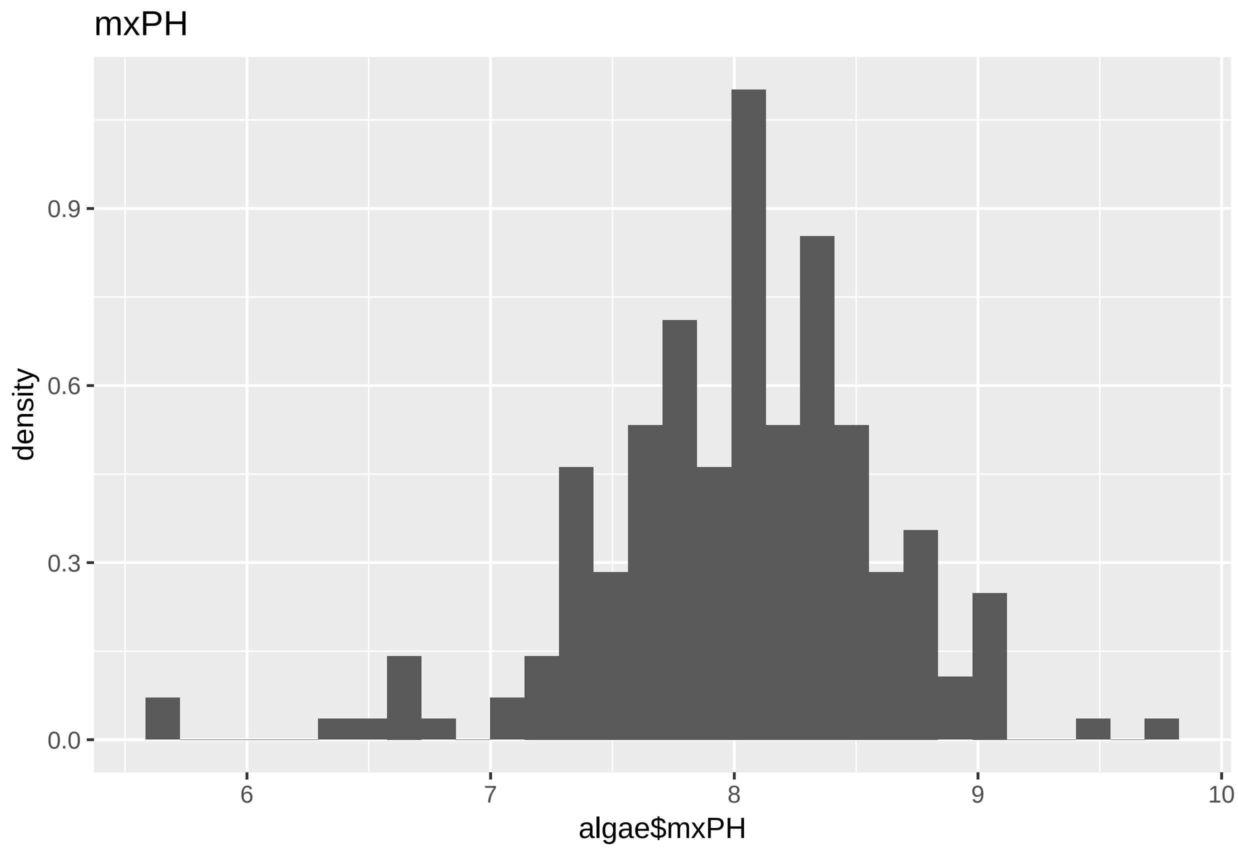
```
## # A tibble: 1 x 8
##    mxPH  mn02    Cl   NO3   NH4  oPO4   PO4  Chla
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 0.340  1.38  22.4  1.46  75.3  29.7  82.5   4.5
```

We see that the chemicals mxPH and mn02 have the largerst difference in term of percentage between MAD and median. Other than this, we also see that MAD and median of that chemical is always smaller than mean and variance of that chemical. We believe that the reason is that MAD and median mitigate the effect on outliers for each chemical.

2.

(a)

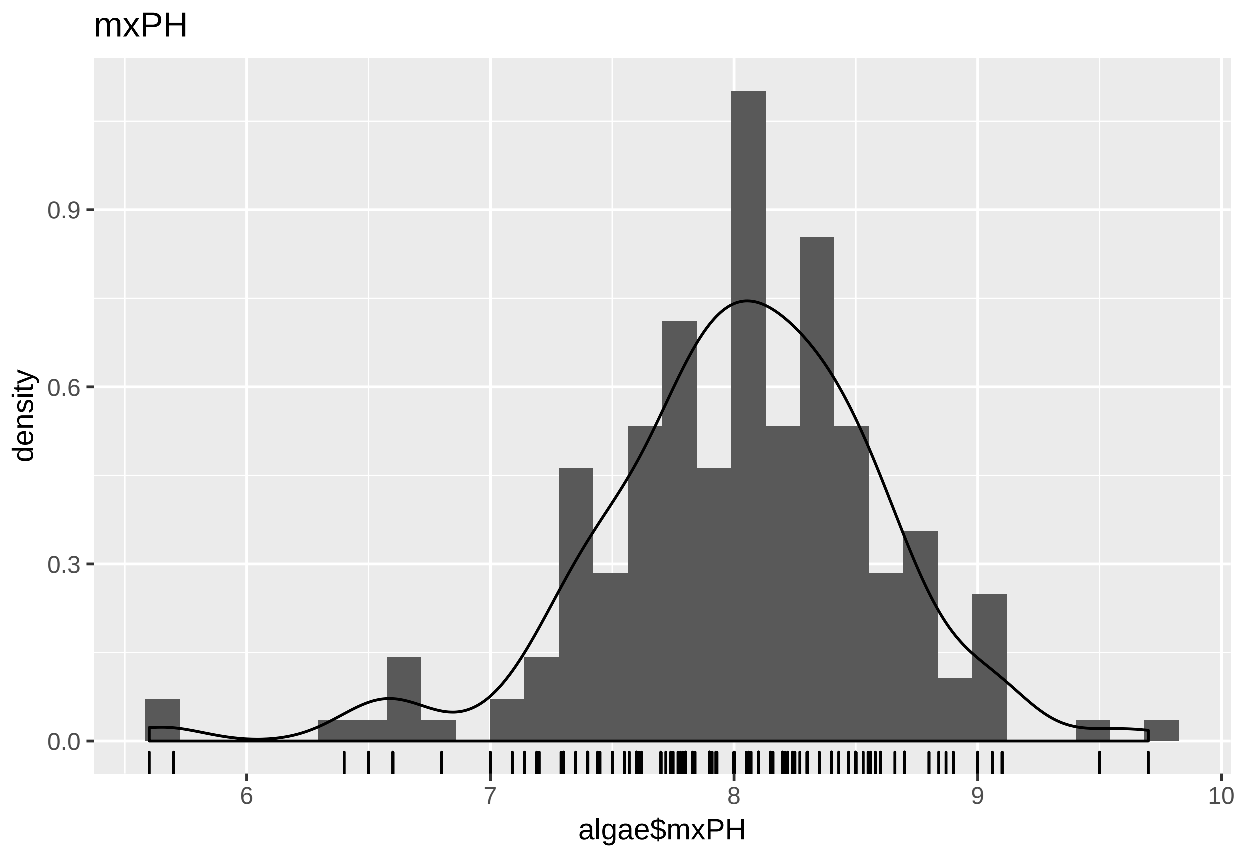
```
ggplot(data=algae,aes(x=algae$mxPH,stat(density)))+
              geom_histogram()+ggtitle("mxPH")
```



From the graph above, we see that the graph is roughly normal and is not skewed.

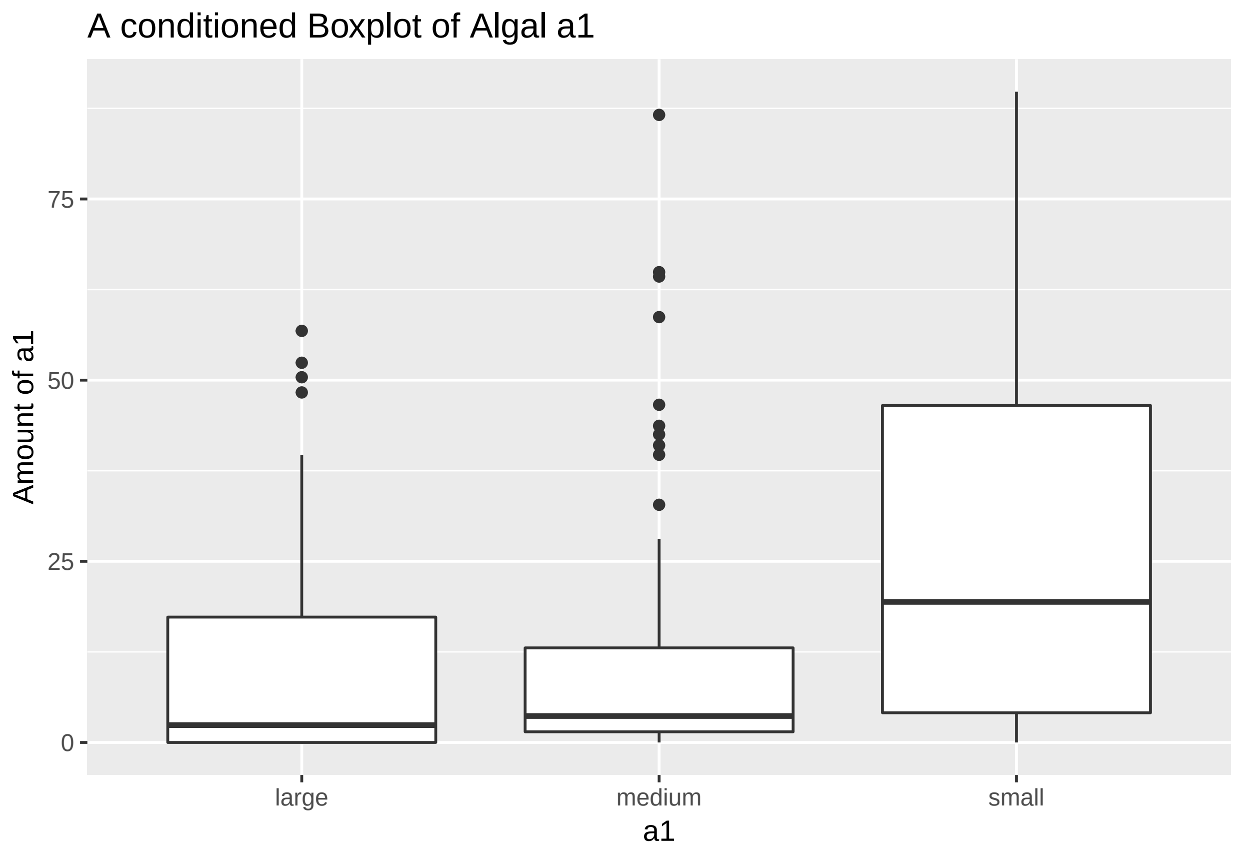
(b)

```r
ggplot(data=algae,aes(x=algae$mxPH))+geom_histogram(aes(y=stat(density)))+
        ggtitle("mxPH")+geom_density()+geom_rug()
```



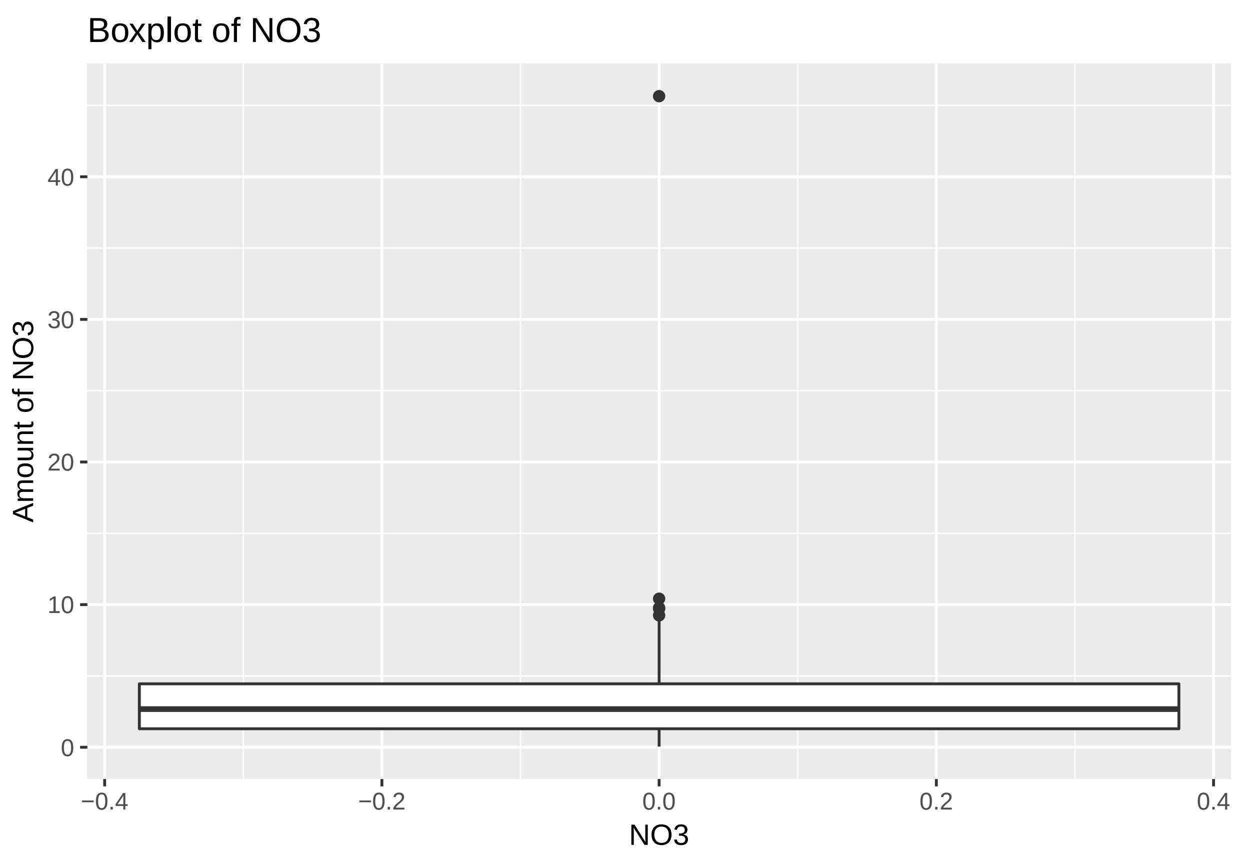
(c)

```r
ggplot(data=algae,aes(x=algae$size,y=a1))+geom_boxplot()+
          xlab("a1")+ylab("Amount of a1")+
          ggtitle("A conditioned Boxplot of Algal a1")
```

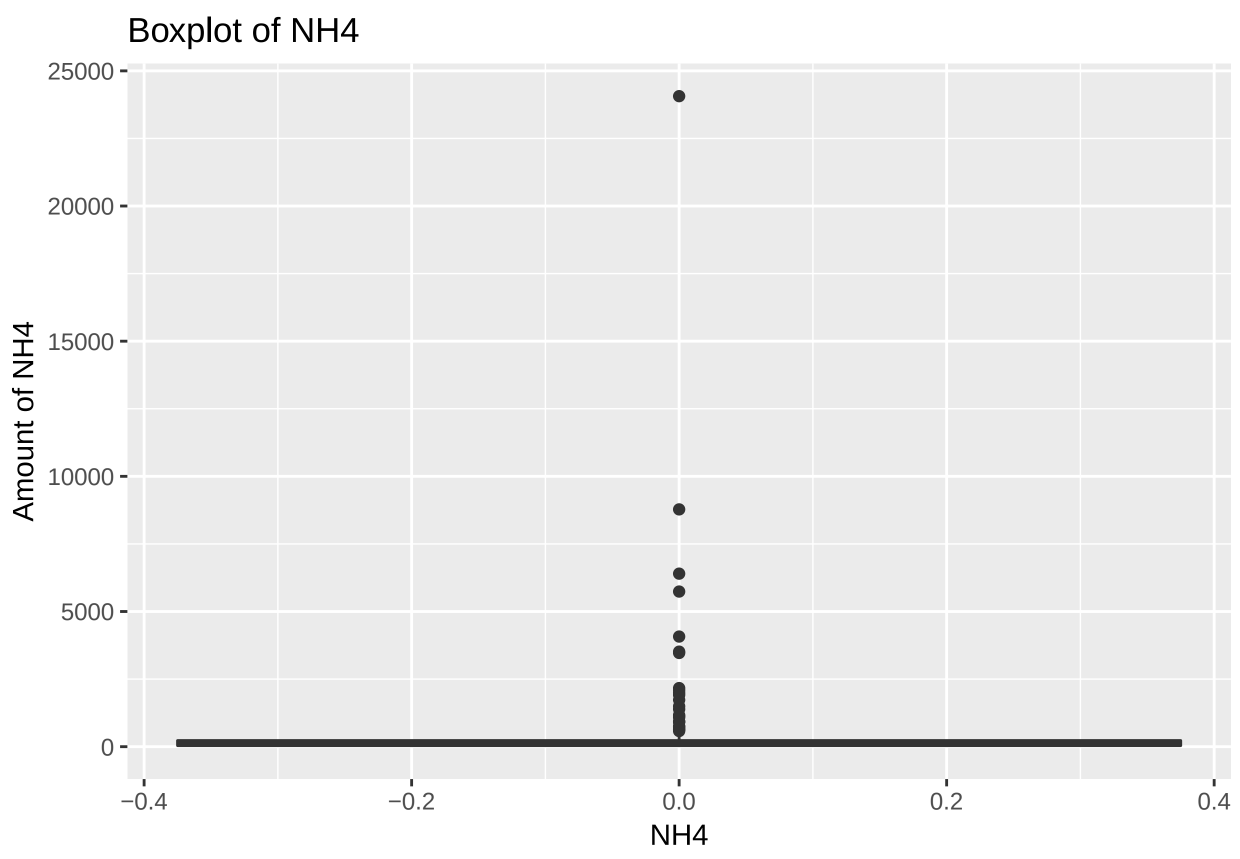


(d)

```
ggplot(data=algae,aes(,y=algae$NO3))+geom_boxplot()+
            xlab("NO3")+ylab("Amount of NO3")+
            ggtitle("Boxplot of NO3")
```



```
ggplot(data=algae,aes(,y=algae$NH4))+geom_boxplot()+
            xlab("NH4")+ylab("Amount of NH4")+
            ggtitle("Boxplot of NH4")
```



```
out=function(x)
{
  lower=quantile(x,0.25,na.rm=TRUE)-1.5*IQR(x,na.rm=TRUE)
  upper=quantile(x,0.75,na.rm=TRUE)+1.5*IQR(x,na.rm=TRUE)
  outliers=(x<lower)|(x>upper)
  count=length(na.omit(x[outliers]))
  return(count)
}

sprintf("The amount of outliers of NO3 and NH4 are %i and %i, respectively.",
        out(algae$NO3), out(algae$NH4))
```

```
## [1] "The amount of outliers of NO3 and NH4 are 5 and 27, respectively."
```

We show boxplot of NO3 and NH4 to check if outliers exists, then we wrote a function to count the amount of outlier for each elements. The defintion of an outlier is a number that is more than 1.5 times the length of data from lower or upper quartiles.

(e)

```
algae %>%
  dplyr::summarize_each(funs(mean(.,na.rm=TRUE),
                             var(.,na.rm=TRUE)), c("NO3","NH4"))
```

```
## # A tibble: 1 x 4
##   NO3_mean NH4_mean NO3_var  NH4_var
##      <dbl>    <dbl>   <dbl>    <dbl>
## 1     3.28     501.    14.3 3851585.
```

```
algae %>%
  dplyr::summarize_each(funs(median(.,na.rm=TRUE),
                             mad(.,constant=1,na.rm=TRUE)), c("NO3","NH4"))
```

```
## # A tibble: 1 x 4
##   NO3_median NH4_median NO3_mad NH4_mad
##        <dbl>      <dbl>   <dbl>   <dbl>
## 1       2.68       103.    1.46    75.3
```

For these two chemicals, we see that the ratio of mean and variance is significantly larger than the ratio of median and MAD. Therefore, we conclude that using median and MAD for measures is much more robust than using mean and variance when outliers are present. The outliers would dramatically increase the value of variance while having little effect on median and MAD.

3.

(a)

```
sprintf("There are %i variables and %i obervations that have missing values.", sum(colSums(is.na(algae)
```

```
## [1] "There are 8 variables and 16 obervations that have missing values."
```

```
sprintf("Below is the summary chart for amount of missing value for each predictors.")
```

```
## [1] "Below is the summary chart for amount of missing value for each predictors."
```

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

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

(b)

```
algae.del=algae%>%
  filter(complete.cases(algae))
sprintf("Now, there are %i observation in algae.del.",  nrow(algae.del))
```

```
## [1] "Now, there are 184 observation in algae.del."
```

(c)

```
algae.med=algae%>%
  mutate_at(vars(c("mxPH":"Chla")),
        .~ifelse(is.na(.),median(.,na.rm=TRUE),.))
sprintf("Now, there are %i observation in algae.med.",nrow(algae.med))
```

```
## [1] "Now, there are 200 observation in algae.med."
```

```
for (x in c(48,62,199))
{
  print(algae.med[x,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
## # 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
## # 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
```

(d)

```r
cor(algae[,4:11],use="pairwise.complete.obs")
```

```
##             mxPH        mnO2          Cl         NO3         NH4
## mxPH  1.00000000 -0.16861234  0.13610778 -0.13098054 -0.09353577
## mnO2 -0.16861234  1.00000000 -0.27833251  0.09944373 -0.08747825
## Cl    0.13610778 -0.27833251  1.00000000  0.22504091  0.07191298
## NO3  -0.13098054  0.09944373  0.22504091  1.00000000  0.72144352
## NH4  -0.09353577 -0.08747825  0.07191298  0.72144352  1.00000000
## oPO4  0.15899936 -0.41616295  0.39105351  0.14458782  0.22723723
## PO4   0.18990814 -0.48748615  0.45744903  0.16860096  0.20818040
## Chla  0.44596182 -0.15326480  0.14985650  0.13967921  0.08894652
##            oPO4        PO4        Chla
## mxPH  0.1589994  0.1899081  0.44596182
## mnO2 -0.4161629 -0.4874862 -0.15326480
## Cl    0.3910535  0.4574490  0.14985650
## NO3   0.1445878  0.1686010  0.13967921
## NH4   0.2272372  0.2081804  0.08894652
## oPO4  1.0000000  0.9143652  0.11562132
## PO4   0.9143652  1.0000000  0.25362130
## Chla  0.1156213  0.2536213  1.00000000
```

```r
relation=lm(algae$PO4~algae$oPO4) #make a linear regression on y=PO4 and x=oPO4
predict(relation)[29]
```

```
##       30
## 58.41378
```

```r
algae$PO4[28]=predict(relation)[28]
algae$PO4[28]
```

```
## [1] 76.51663
```

(e)

If we impute missing values with mean or medians, we may have a biased result because it would have smaller variance and does not reflect the uncertainty on prediction of unknown missing values. Such process would cause survivorship bias. ALso, for small sample size, the estimation can be dramatically different compare with and without impution.

4.

(a)

```r
nfold=5
set.seed(10)
folds=cut(1:nrow(algae.med), breaks=nfold, labels=FALSE) %>% sample()
folds
```

```
##   [1] 3 2 3 4 1 2 2 2 3 3 4 3 1 3 2 2 1 2 2 4 4 3 4 2 2 5 4 2 4 2 3 1 1 4 2
##  [36] 5 4 5 5 3 2 1 1 3 1 1 4 2 5 4 5 4 4 2 1 5 5 4 2 5 1 5 2 2 3 3 2 2 1 5
##  [71] 5 2 5 4 4 4 2 5 1 1 3 2 1 5 1 4 3 1 1 5 4 2 2 4 2 3 4 3 2 1 1 1 4 1 3
## [106] 4 3 4 5 5 5 4 5 2 5 3 5 2 2 4 1 3 3 5 4 3 5 3 5 2 5 3 5 5 1 5 3 5 3 1
## [141] 4 1 4 3 5 5 1 1 4 2 4 4 2 1 4 5 4 1 3 2 4 4 4 1 5 5 3 5 3 5 3 3 5 1 3
## [176] 4 4 1 1 3 2 3 1 3 1 3 2 3 2 3 1 1 4 2 3 5 5 1 1 2
```

(b)

```
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$a1)^2),
             # compute and store training error
             val.error = mean((predYvl - Yvl$a1)^2))
             # compute and store test error
}

ldply(1:nfold, do.chunk, folds, algae.med)
```

```
##   fold train.error val.error
## 1    1    313.8769  183.6954
## 2    2    238.0351  552.8065
## 3    3    306.6797  229.1438
## 4    4    284.6941  325.9966
## 5    5    263.2112  410.4312
```

5.

(a)

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

```
lm.a1=lm(a1~., data=algae.med[,1:12])
Xtr = algae.med[,1:11] # get training set
Ytr = algae.med[,12] # get true response values in trainig set
Xvl = algae.Test[,1:11] # get validation set
Yvl = algae.Test[,12] # get true response values in validation set
lm.a1 <- lm(a1~., data = algae.med[,1:12])
predYtr = predict(lm.a1) # predict training values
predYvl = predict(lm.a1,Xvl) # predict validation values
train.error = mean((predYtr - Ytr$a1)^2) # compute and store training error
val.error = mean((predYvl - Yvl$a1)^2) # compute and store test error

train.error
```

```
## [1] 286.2661
```

```
val.error
```

```
## [1] 250.1794
```

We see that, compare with problem 4, the training error is roughly the same, but this new test error is smaller than most of the previous, test errors. This shows that this new model is fit for new dataset.

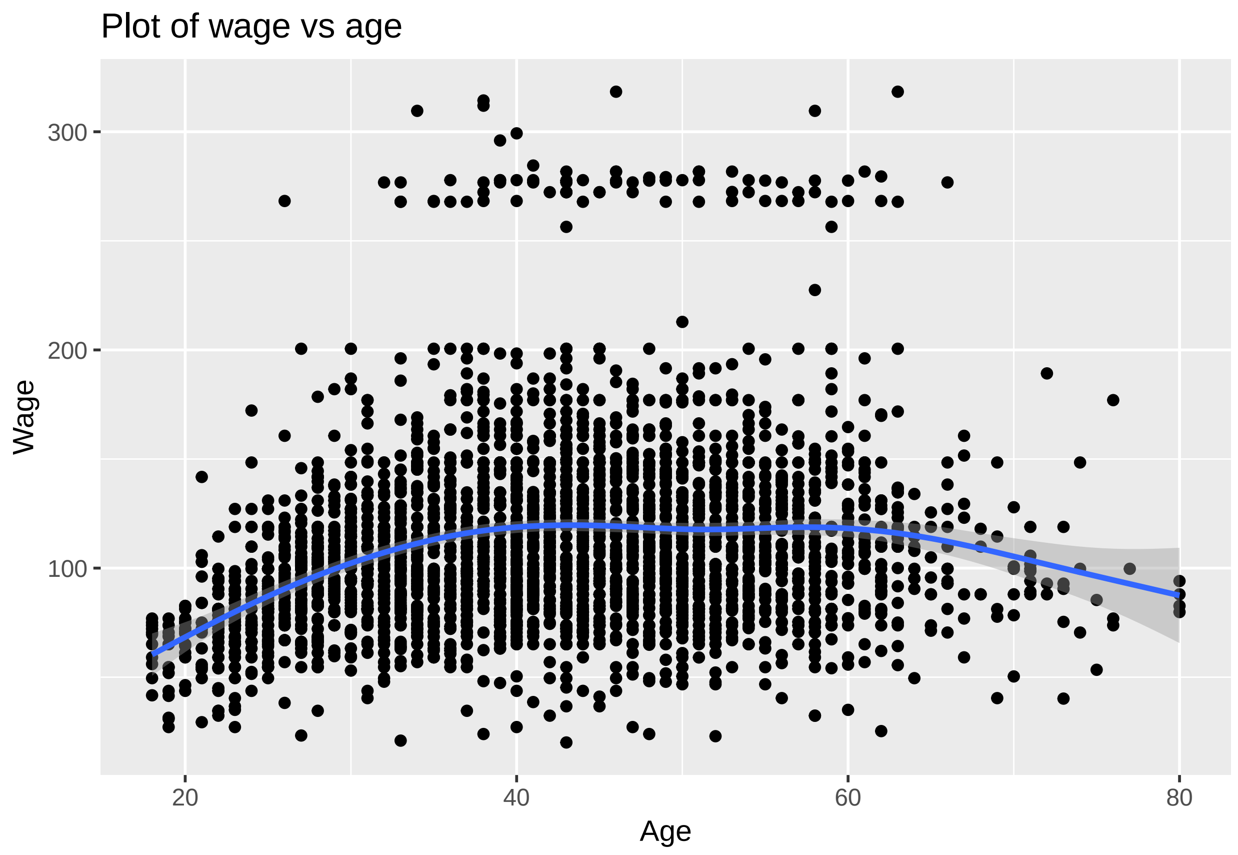
6.

(a)

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

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

```r
data(Wage)
ggplot(data=Wage, aes(x=age, y=wage))+
  geom_point()+geom_smooth()+
  xlab("Age")+ylab("Wage")+
  ggtitle("Plot of wage vs age")
```



We see that the relationship between Age and Wage is not really linear. Notably between age 30 and 65, we see that there are outliers that earn above 250 while most of the wages is below 200. Also, as age increase, we see that there are less and less data points. These match our expectation because people tend to retire after 65 years old.

(b).

```
nfold=5
set.seed(11)
folds=cut(1:nrow(Wage), breaks=nfold, labels=FALSE) %>% sample()
do.chunk1=function(chunkid,folddef,dat) #constant polynomial
{
  train=(folddef!=chunkid)
  Xtr=dat[train,2]
  Ytr=dat[train,11]
  Xvl=dat[!train,2]
  Yvl=dat[!train,11]

  poly1=lm(wage~1,data=dat[train,c(2,11)])
  pdYtr=predict(poly1)
  pdYvl=predict(poly1,newdata=dat[!train,c(2,11)])
  data.frame(fold=chunkid,
    train.error=mean((pdYtr-Ytr)^2),
    # compute and store training error
    val.error=mean((pdYvl-Yvl)^2))
  # compute and store test error
}
```

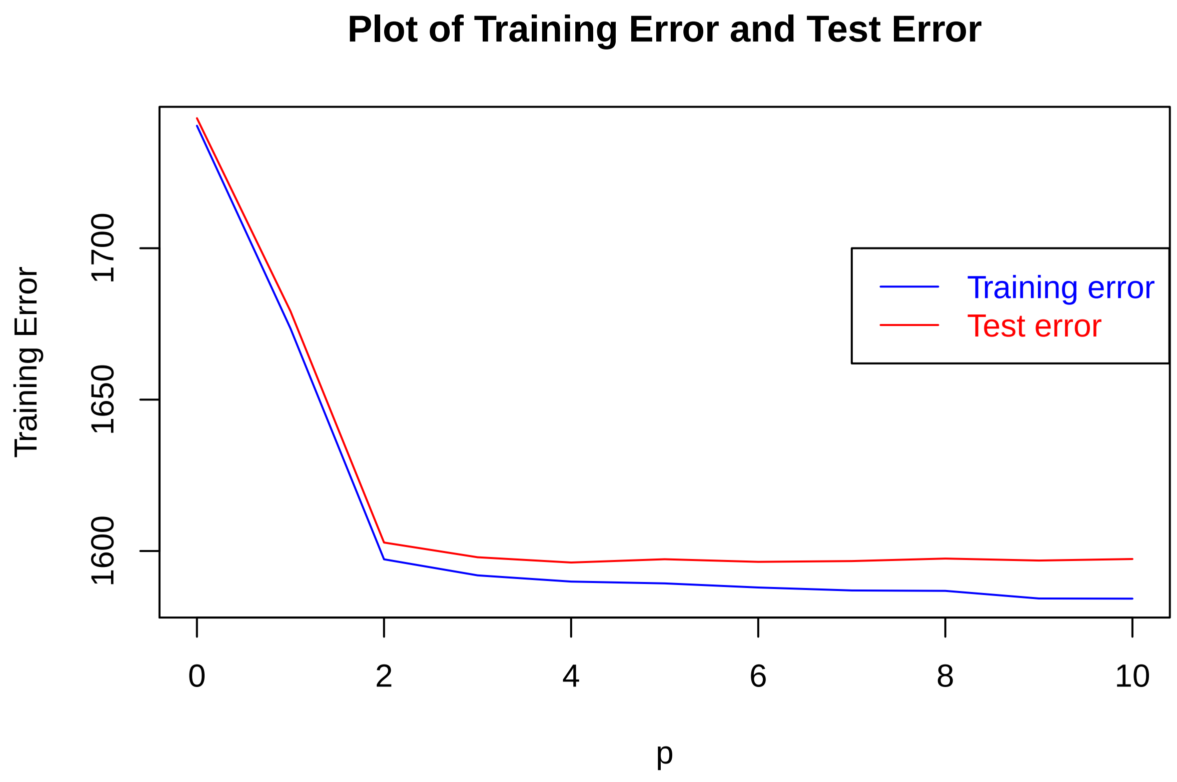
```r
table=NULL
table=rbind(table,ldply(1:nfold, do.chunk1,folds, Wage))
#combine result


do.chunk2=function(chunkid,folddef,dat,deg) #polynomial
{
  train=(folddef!=chunkid)
  Xtr=dat[train,2]
  Ytr=dat[train,11]
  Xvl=dat[!train,2]
  Yvl=dat[!train,11]

  poly=lm(wage~poly(age,degree=deg,raw=FALSE),
          data=dat[train,c(2,11)])
  pdYtr=predict(poly)
  pdYvl=predict(poly,newdata=dat[!train,c(2,11)])
  data.frame(fold=chunkid,
    train.error=mean((pdYtr-Ytr)^2),
    # compute and store training error
    val.error=mean((pdYvl-Yvl)^2))
  # compute and store test error
}

for(deg in 1:10)
  table=rbind(table,ldply(1:nfold, do.chunk2,folds, Wage, deg))

sequence=c(0,0,0,0,0,1,1,1,1,1,2,2,2,2,2,3,3,3,3,3,4,4,4,4,4,5,5,5,5,5,6,6,6,6,6,7,7,7,7,7,8,8,8,8,8,9,
DataFrame=cbind(table,sequence)
colnames(DataFrame)=c("fold","train.error","val.error","degree")
mean=DataFrame %>%
  group_by(degree) %>%
  summarize_at(.funs=funs(mean),.var=vars(train.error, val.error))

mean
```

```
## # A tibble: 11 x 3
##    degree train.error val.error
##     <dbl>       <dbl>     <dbl>
##  1      0       1740.     1743.
##  2      1       1674.     1679.
##  3      2       1597.     1603.
##  4      3       1592.     1598.
##  5      4       1590.     1596.
##  6      5       1589.     1597.
##  7      6       1588.     1596.
##  8      7       1587.     1597.
##  9      8       1587.     1598.
## 10      9       1584.     1597.
## 11     10       1584.     1597.
```

c.

```r
plot(train.error~degree, data=mean, xlab="p", ylab="Training Error",
main="Plot of Training Error and Test Error",col="blue", type="l")
```

```
lines(mean$degree,mean$val.error,col="Red")
legend(7,1700,c("Training error", "Test error"), col=c("Blue","Red"), text.col=c("Blue","Red"),lty=c(1)
```



From the graph above, we see that the errors decrease dramatically between p=0 and p=2. After p=2, the test errors remain consistent. When choosing model, we want to have the most simplest model while having low test errors. Therefore, we decide that the model for this relationship is $wage = \beta_0 + \beta_1 age + \beta_2 age^2 + \epsilon$