Homework Assignment

Jasmine Kwok and Amber Baez

October 23, 2020

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## -- Attaching packages -------------------------------------------------------------------------------

## v ggplot2 3.3.2     v purrr   0.3.3
## v tibble  3.0.3     v stringr 1.4.0
## v tidyr   1.1.2     v forcats 0.5.0
## v readr   1.3.1

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

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

## 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 object is masked from 'package:purrr':
##
##     compact

## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

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

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

## Parsed with column specification:
## cols(
##   season = col_character(),
##   size = col_character(),
##   speed = col_character(),
##   mxPH = col_double(),
##   mn02 = 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()
## )

## 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, ...
## $ mn02   <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> 60.80, 57.75, 40.02, 77.36, 55.35, 65.75, 73.25, 59.07, 21.9...
## $ NO3    <dbl> 6.238, 1.288, 5.330, 2.302, 10.416, 9.248, 1.535, 4.990, 0.8...
## $ NH4    <dbl> 578.00, 370.00, 346.67, 98.18, 233.70, 430.00, 110.00, 205.6...
## $ oPO4   <dbl> 105.00, 428.75, 125.67, 61.18, 58.22, 18.25, 61.25, 44.67, 3...
## $ PO4    <dbl> 170.00, 558.75, 187.06, 138.70, 97.58, 56.67, 111.75, 77.43,...
## $ 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, ...
```

1. **QUESTION 1: Descriptive summary statistics:**

   ```r
   ```

```
# Exploratory analysis
summary(algae)
```

```
##     season              size               speed               mxPH      
##  Length:200         Length:200         Length:200         Min.   :5.60  
##  Class :character   Class :character   Class :character   1st Qu.:7.70  
##  Mode  :character   Mode  :character   Mode  :character   Median :8.06  
##                                                           Mean   :8.01  
##                                                           3rd Qu.:8.40  
##                                                           Max.   :9.70  
##                                                           NA's   :1     
##       mn02             Cl               NO3              NH4        
##  Min.   : 1.50   Min.   :  0.2   Min.   : 0.05   Min.   :    5  
##  1st Qu.: 7.72   1st Qu.: 11.0   1st Qu.: 1.30   1st Qu.:   38  
##  Median : 9.80   Median : 32.7   Median : 2.67   Median :  103  
##  Mean   : 9.12   Mean   : 43.6   Mean   : 3.28   Mean   :  501  
##  3rd Qu.:10.80   3rd Qu.: 57.8   3rd Qu.: 4.45   3rd Qu.:  227  
##  Max.   :13.40   Max.   :391.5   Max.   :45.65   Max.   :24064  
##  NA's   :2       NA's   :10      NA's   :2       NA's   :2      
##       oP04             P04              Chla               a1       
##  Min.   :  1.0   Min.   :  1.0   Min.   :  0.20   Min.   : 0.00  
##  1st Qu.: 15.7   1st Qu.: 41.4   1st Qu.:  2.00   1st Qu.: 1.50  
##  Median : 40.1   Median :103.3   Median :  5.47   Median : 6.95  
##  Mean   : 73.6   Mean   :137.9   Mean   : 13.97   Mean   :16.92  
##  3rd Qu.: 99.3   3rd Qu.:213.8   3rd Qu.: 18.31   3rd Qu.:24.80  
##  Max.   :564.6   Max.   :771.6   Max.   :110.46   Max.   :89.80  
##  NA's   :2       NA's   :2       NA's   :12                      
##        a2               a3               a4               a5        
##  Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00  
##  1st Qu.: 0.00   1st Qu.: 0.00   1st Qu.: 0.00   1st Qu.: 0.00  
##  Median : 3.00   Median : 1.55   Median : 0.00   Median : 1.90  
##  Mean   : 7.46   Mean   : 4.31   Mean   : 1.99   Mean   : 5.06  
##  3rd Qu.:11.38   3rd Qu.: 4.92   3rd Qu.: 2.40   3rd Qu.: 7.50  
##  Max.   :72.60   Max.   :42.80   Max.   :44.60   Max.   :44.40  
##                                                                 
##        a6               a7       
##  Min.   : 0.00   Min.   : 0.0  
##  1st Qu.: 0.00   1st Qu.: 0.0  
##  Median : 0.00   Median : 1.0  
##  Mean   : 5.96   Mean   : 2.5  
##  3rd Qu.: 6.92   3rd Qu.: 2.4  
##  Max.   :77.60   Max.   :31.6  
## 
```

a) count the number of observations in each season using summarise

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

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

```
## # A tibble: 4 x 2
```

```
##   season season_n
##   <chr>     <int>
## 1 autumn       40
## 2 spring       53
## 3 summer       45
## 4 winter       62
```

b) Missing values? Calculate the mean and variance of each chemical.

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

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

```r
mean.var <- na.exclude(algae) %>% summarise_each(funs(mean,var),mn02,Cl,NO3,NH4,
                         oPO4,PO4,Chla)
```

```
## Warning: `summarise_each_()` is deprecated as of dplyr 0.7.0.
## Please use `across()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
## 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.
```

```r
mean.var
```

```
## # A tibble: 1 x 14
##   mn02_mean Cl_mean NO3_mean NH4_mean oPO4_mean PO4_mean Chla_mean mn02_var
##       <dbl>   <dbl>    <dbl>    <dbl>     <dbl>    <dbl>     <dbl>    <dbl>
## 1      9.02    44.9     3.38     538.      78.3     147.      13.9     5.79
## # ... with 6 more variables: Cl_var <dbl>, NO3_var <dbl>, NH4_var <dbl>,
## #   oPO4_var <dbl>, PO4_var <dbl>, Chla_var <dbl>
```

Yes, there are 33 missing values in the algae dataset. The mean and variance values for mn02, NO3,and Chla are much smaller than those of Cl, NH4, oPO4, and PO4.

c) finding MAD and median

```r
na.exclude(algae) %>% summarise_each(funs(median, mad),mn02,Cl,NO3,NH4,
                         oPO4,PO4,Chla)
```

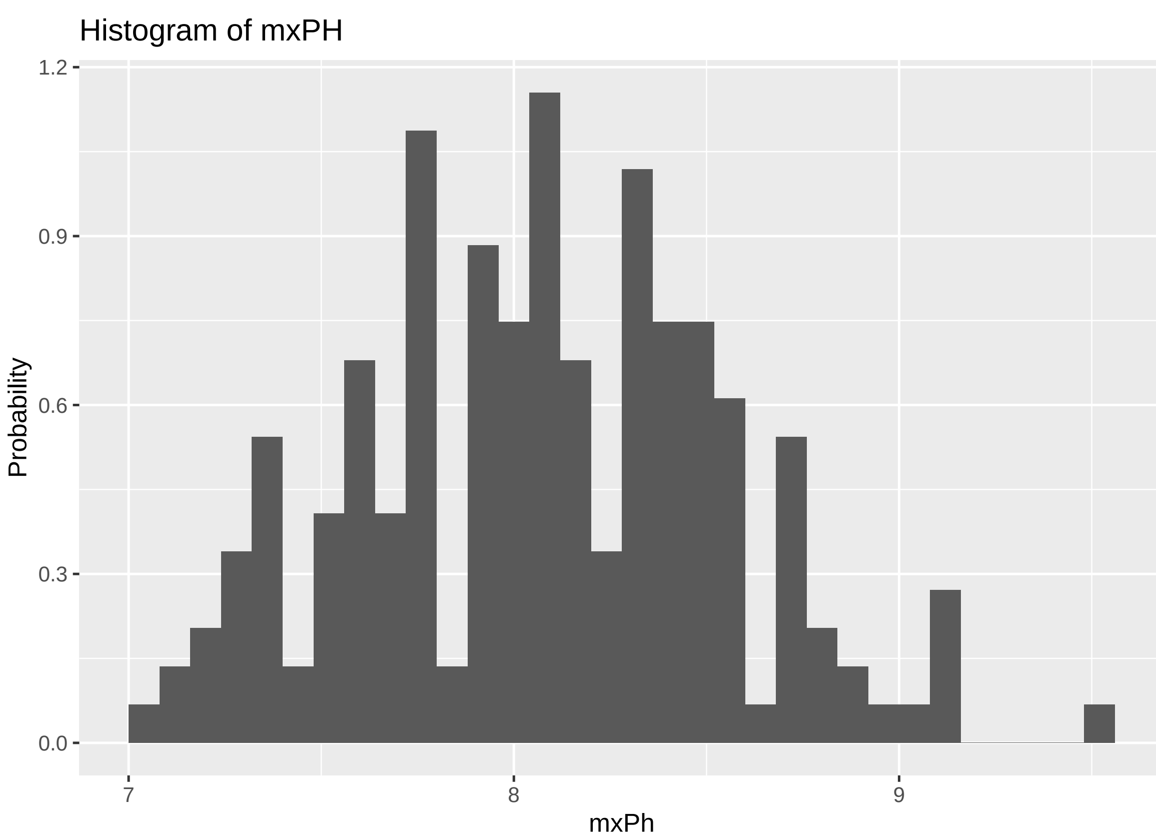
```
## # A tibble: 1 x 14
##   mn02_median Cl_median N03_median NH4_median oP04_median P04_median Chla_median
##         <dbl>     <dbl>      <dbl>      <dbl>       <dbl>      <dbl>       <dbl>
## 1        9.75      35.1       2.82       116.        46.3       116.        5.52
## # ... with 7 more variables: mn02_mad <dbl>, Cl_mad <dbl>, N03_mad <dbl>,
## #   NH4_mad <dbl>, oP04_mad <dbl>, P04_mad <dbl>, Chla_mad <dbl>
```

The mean values for the chemicals are close to median values and larger than the median values. The median and MAD values have a much smaller range in comparison to the mean and variance values. The variance values are the largest amongst the 4 groups of values. We noticed that the quantities for mean and variance are larger than the quantities in median and MAD.

QUESTION 2: Data visualization

a) a histogram of mxPH with the title 'Histogram of mxPH' based on algae data set

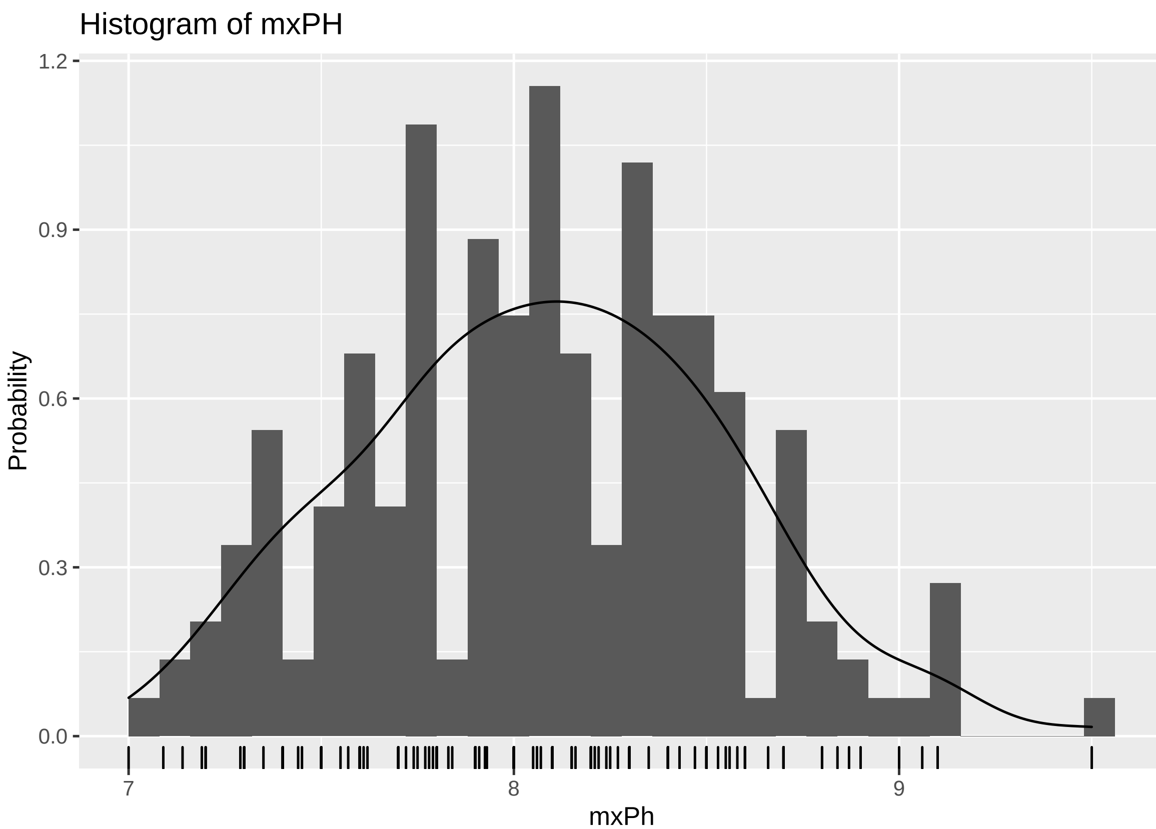
```
# preparing the data; create a dataframe with season and mxPH
mxPH.df <-data.frame(na.exclude(algae))
total<- count(mxPH.df) #199
ggplot(mxPH.df, aes(x=mxPH)) + geom_histogram(binwidth = 0.08,aes(y = ..density..)) + labs(title =
```



The distribution on the histogram seems to be left skewed.

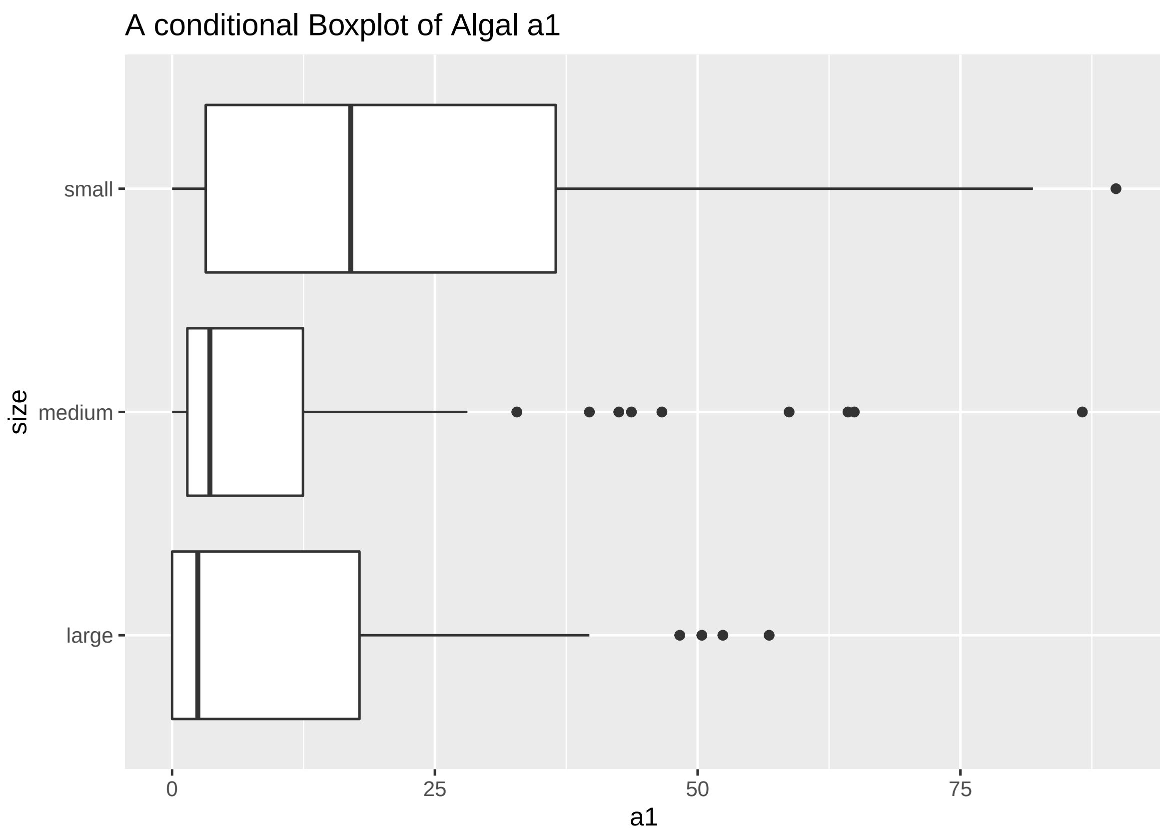
b) add a density curve using geom_density and rug plots using geom_rug

```
ggplot(mxPH.df, aes(x=mxPH)) + geom_histogram(binwidth = 0.08,aes(y = ..density..)) + labs(title =
```



c) boxplot with title 'A conditioned Boxplot of Algal a1' for a1 grouped by size

```
ggplot(mxPH.df, aes(x=a1, y=size)) + geom_boxplot() + labs(title = "A conditional Boxplot of Algal
```



d) Are there any outliers for NO3 and NH4? How many observtions would you consider as outliers?

```r
# finding out outliers for NO3
boxplot.stats(mxPH.df$NO3)$out
```

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

```r
# finding out outliers for NH4
boxplot.stats(mxPH.df$NH4)$out
```

```
##  [1]   578.0  8777.6  1729.0  3515.0  6400.0  1911.0   647.6  1386.2  2082.9
## [10]  2167.4   737.5   914.0  5738.3  4073.3   758.8   931.8   723.7  3466.7
## [19]   920.0  1990.2 24064.0  1131.7  1495.0   643.0   627.3  1168.0  1081.7
```

For NO3, there are 4 observations that I would consider outliers based on the boxplot stats function. For NH4, there are a total of 27 observations that are considered outliers according to the function.

e) Compare mean & variance vs median & MAD for NO3 and NH4. What do you notice? Can you conclude which sets of measures is more robust when outliers are present?

```
```r
na.exclude(algae) %>% summarise_each(funs(mean, var, median, mad),NO3,NH4)
```

```
## # A tibble: 1 x 8
```
```

```
##   NO3_mean NH4_mean NO3_var  NH4_var NO3_median NH4_median NO3_mad NH4_mad
##      <dbl>    <dbl>   <dbl>    <dbl>      <dbl>      <dbl>   <dbl>   <dbl>
## 1     3.38     538.    15.0 4127337.       2.82       116.    2.31    121.
```

The mean and variance are affected more by outliers in comparison to median and MAD. The median and MAD values are very similar to one another but the mean and variance may be heavily affected if the outlier is extremely larger or extremely small value. Hence, we can conclude that the median and MAD set of measure is more robust when outliers are present.

QUESTION 3: Dealing with missing values

a) How many observations contain missing values? How many missing values are there in each variable?

```r
# summing the total number of missing values in algae dataset
sum(is.na(algae))
```

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

```r
# summing the missing values by column
colSums(is.na(algae))
```

```
## season   size  speed   mxPH   mn02     Cl    NO3    NH4   oP04    P04   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
```

33 observations contain missing values. There is 1 missing value in mxPH column, 2 missing values in the columns mn02, NO3, NH4, oP04, and PO4, 10 missing values in Cl column, and 12 missing values in Chla column.

b) Removing observations with missing values: use filter() function in dplyr package to observations with any missing value, and save the resulting dataset (without missing values) as algae.del. Report how many observations are in algae.del.

```r
# removing observations with missing values by filtering though the whole dataset algae
algae.del <- algae %>% filter(complete.cases(.))
nrow(algae) # original number of row - 200
```

```
## [1] 200
```

```r
nrow(algae.del) # after filter, the new dataset has 184 rows
```

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

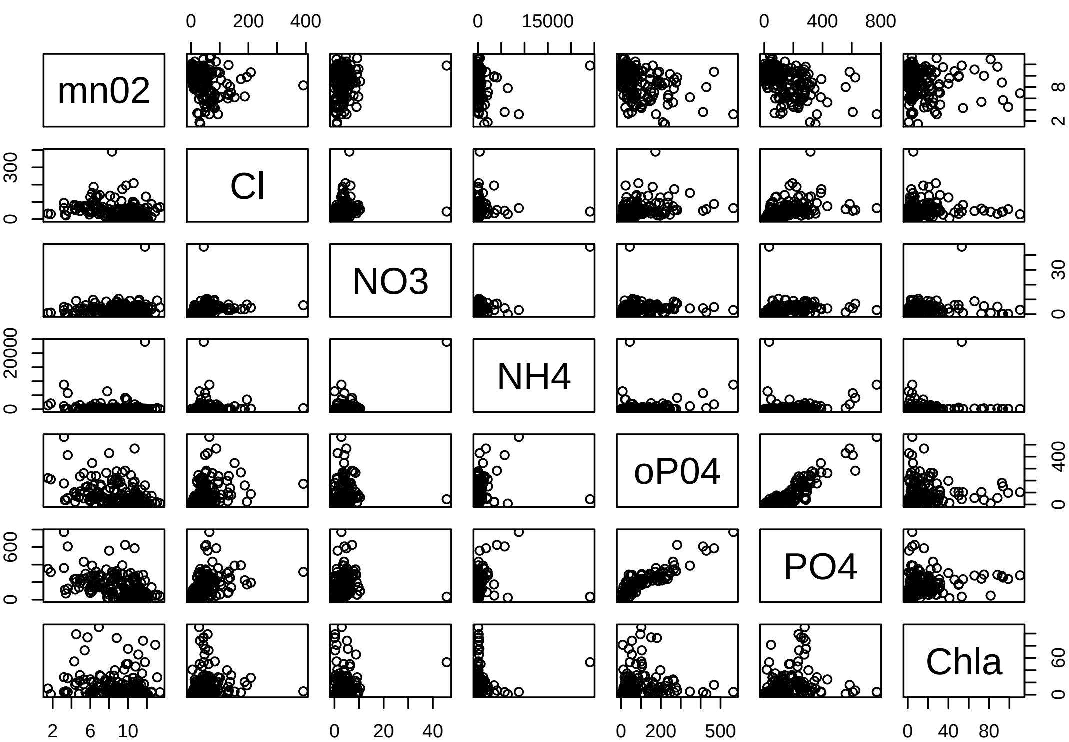
c) Imputing unknowns with measures of central tendency

```r
# save imputed dataset as algae.med; impute the NAs with median data
algae.med <- algae %>% mutate_at(vars(c('mxPH','mn02','Cl','NO3','NH4','oP04','P04','Chla')),funs(i
#algae.med
```

d) Imputing unknowns using correlations

```r
# selecting all the columns with chemicals
chemicals <- na.exclude(algae) %>% select('mn02','Cl','NO3','NH4','oP04','P04','Chla')

# compute pairwise correlation between continuous chemical variables
pairs(chemicals) # visuals
```




```r
cor(chemicals) # pariwise correlation values
```

```
##          mn02       Cl    NO3      NH4    oP04    P04    Chla
## mn02  1.00000 -0.26325 0.1179 -0.07827 -0.3938 -0.4640 -0.1312
## Cl   -0.26325  1.00000 0.2110  0.06598  0.3793  0.4452  0.1430
## NO3   0.11791  0.21096 1.0000  0.72468  0.1330  0.1570  0.1455
## NH4  -0.07827  0.06598 0.7247  1.00000  0.2193  0.1994  0.0912
## oP04 -0.39375  0.37926 0.1330  0.21931  1.0000  0.9120  0.1069
## P04  -0.46396  0.44519 0.1570  0.19940  0.9120  1.0000  0.2485
## Chla -0.13122  0.14296 0.1455  0.09120  0.1069  0.2485  1.0000
```

```r
# fill in missing value for PO4 based on oPO4 in the 28th observation.
lm.PO4 <- lm(data=chemicals, PO4 ~ oPO4)
# predict using the value of oPO4 in the 28th observation
oPO4val <- algae[28,9] # 4
pred.PO4 <- predict(lm.PO4, oPO4val) # 48.07
# fill in missing values for PO4
algae.cor <- algae %>% mutate_at(vars(c('PO4')),funs(ifelse(is.na(.),pred.PO4,.)))
algae.cor
```

```
## # A tibble: 200 x 18
##    season size  speed  mxPH  mnO2    Cl    NO3   NH4  oPO4   PO4  Chla    a1
##    <chr>  <chr> <chr> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 winter small medi~  8      9.8  60.8  6.24  578    105   170   50      0
##  2 spring small medi~  8.35   8    57.8  1.29  370    429.  559.   1.3    1.4
##  3 autumn small medi~  8.1   11.4  40.0  5.33  347.   126.  187.  15.6    3.3
##  4 spring small medi~  8.07   4.8  77.4  2.30   98.2   61.2 139.   1.4    3.1
##  5 autumn small medi~  8.06   9    55.4 10.4   234.    58.2  97.6 10.5    9.2
##  6 winter small high   8.25  13.1  65.8  9.25  430     18.2  56.7 28.4   15.1
##  7 summer small high   8.15  10.3  73.2  1.54  110     61.2 112.   3.2    2.4
##  8 autumn small high   8.05  10.6  59.1  4.99  206.    44.7  77.4  6.9   18.2
##  9 winter small medi~  8.7    3.4  22.0  0.886 103.    36.3  71    5.54  25.4
## 10 winter small high   7.93   9.9   8    1.39    5.8   27.2  46.6  0.8   17
## # ... with 190 more rows, and 6 more variables: a2 <dbl>, a3 <dbl>, a4 <dbl>,
## #   a5 <dbl>, a6 <dbl>, a7 <dbl>
```

e) Questioning missing data assumptions Imputation using only the observed data might lead to incorrect conclusions when there is survivorship bias present in the data. It is possible that water samples are collected from a certain area where the levels of chemicals are within a certain range Data of water samples which have much large chemical samples may not be collected as the algae may have died or decompose given the higher quantities of chemical present in the water.

QUESTION 4: Cross validation using algae.med dataset

a) Randomly partition data into 5 equal sized chunks

```r
# specify we want a 5 fold cross validation
nfold = 5

# dividing all training observations into 5 intervals
set.seed(72)
folds = cut(1:nrow(algae.med), breaks = nfold, labels = FALSE) %>% sample()
folds
```

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

b) perform 5-fold cross-validation with training error and validation errors of each chunck determined from 4a.

```r
# given function
do.chunk <- function(chunkid, chunkdef, dat){ # Function arguments
  train = (chunkdef!=chunkid) # Get training index

  Xtr = dat[train,1:11] # Get training set by the above index
  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 labels

  data.frame(fold = chunkid, # k folds
             train.error = mean((predYtr - Ytr$a1)^2), # compute and store training error
             val.error = mean((predYvl - Yvl$a1)^2)) # compute and store test error
}

# set error.folds to save validation errors in future
error.folds = NULL

# give a possible number of nearest neighbors to be considered
allK = 1:50

# set seed since do.chunk() contains a random component induced by knn()
set.seed(999)

# Apply do.chunk() function to each fold
tmp = ldply(1:nfold, do.chunk,chunkdef=folds, dat=algae.med)
error.folds = rbind(error.folds, tmp)
error.folds
```

```
##   fold train.error val.error
## 1    1       251.1     545.1
## 2    2       239.0     526.5
## 3    3       297.2     366.4
## 4    4       294.9     279.4
## 5    5       309.0     213.9
```

QUESTION 5: Test error on additional data

```
```r
algae.Test <- read_table2('algaeTest.txt', col_names = c('season','size','speed','mxPH', 'mnO2','Cl
```
```

```
```
Parsed with column specification:
cols(
season = col_character(),
size = col_character(),
speed = col_character(),
mxPH = col_double(),
```
```

```
##   mn02 = col_double(),
##   Cl = col_double(),
##   NO3 = col_double(),
##   NH4 = col_double(),
##   oPO4 = col_double(),
##   PO4 = col_double(),
##   Chla = col_double(),
##   a1 = col_double()
## )
```

```r
# Build the model & predict
set.seed(6)
model.a1 <- lm(a1 ~., data = algae.med[,1:12])
predictions <- model.a1 %>% predict(algae.Test[,1:12])

#value of true test error
true.error = mean((predictions -  algae.Test$a1)^2)
true.error
```

```
## [1] 250.2
```

Yes, based on the cross validation estimated test error from part 4, this is roughly the amount of "true" test error we expected.

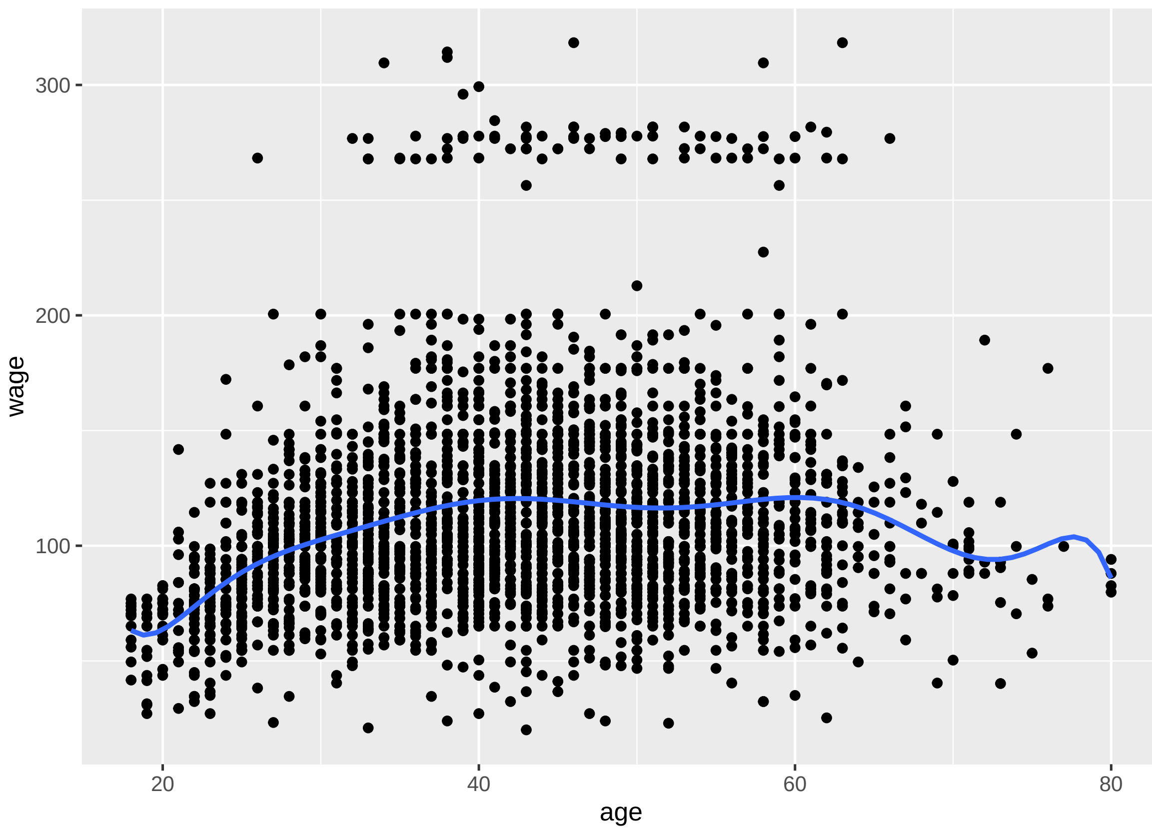
QUESTION 6: Cross Validation(CV) for Model Selection

a) plot wages as a function of age using ggplot

```r
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   4.318  75.04
## 86582  2. Information 2. >=Very Good      2. No   4.255  70.48
## 161300  1. Industrial      1. <=Good     1. Yes   4.875 130.98
## 155159 2. Information 2. >=Very Good     1. Yes   5.041 154.69
## 11443  2. Information      1. <=Good     1. Yes   4.318  75.04
## 376662 2. Information 2. >=Very Good     1. Yes   4.845 127.12
```

```r
ggplot(data = Wage, aes(x=age, y=wage)) + geom_point() +
  geom_smooth(method="lm", formula = y~poly(x,10), se = FALSE)
```



The general pattern of wages as a function of age is an inverted parabola. Yes, it matches our expectations because individuals would usually make more in their late 30s to around 60 years old as they gain more experience and expertise over time. They would earn less when they are younger due to the lack of experience and expertise and they would earn less when they are older due to the lack productivity, working for less hours, or retirement.

b)

i. fit a linear regression to predict wages as a function of age^p where p=0,1,. . . ,10.

```
age = Wage$age
wage = Wage$wage
# using lm to find linear regression
fitreg <-function(p){
  if (p==0){
    lm.wage<-lm(wage~1,data=Wage)}
  else{
    lm.wage<- lm(wage~poly(age,p), data = Wage)}
}

print(fitreg(10))
```

```
##
## Call:
## lm(formula = wage ~ poly(age, p), data = Wage)
##
## Coefficients:
```

```
##    (Intercept)   poly(age, p)1   poly(age, p)2   poly(age, p)3   poly(age, p)4
##          111.70          447.07         -478.32          125.52          -77.91
##  poly(age, p)5   poly(age, p)6   poly(age, p)7   poly(age, p)8   poly(age, p)9
##          -35.81           62.71           50.55          -11.25          -83.69
## poly(age, p)10
##            1.62
```

ii.

```r
# specify we want a 5 fold cross validation
nfold = 5
# dividing all training observations into 5 intervals
set.seed(72)
folds = cut(1:nrow(Wage), breaks = nfold, labels = FALSE) %>% sample()
age = Wage$age
wage = Wage$wage
do.chunk <- function(chunkid, chunkdef, dat, p){ # function argument
  train = (chunkdef != chunkid)
  Xtr = dat[train, ]$age # get training set
  Ytr = dat[train, ]$wage # get true response values in trainig set
  Xvl = dat[!train, ]$age # get validation set
  Yvl = dat[!train, ]$wage # get true response values in validation set
if (p == 0){
  lm.wage <- lm(wage~1, data= dat[train,])
}else{
  lm.wage <- lm(wage~poly(age,p), data= dat[train,])
}
predYtr = predict(lm.wage) # predict training values
predYvl = predict(lm.wage,dat[!train,]) # predict validation values
data.frame(fold = chunkid,
train.error = mean((predYtr - Ytr)^2), # compute and store training error
val.error = mean((predYvl - Yvl)^2),d) # compute and store test error
}

error.folds = NULL
for (d in 0:10){
tmp = ldply(1:nfold, do.chunk, chunkdef=folds, dat=Wage, d)
error.folds = rbind(error.folds, tmp)
}
# printing out the test error and training error
error.folds
```

```
##    fold train.error val.error  d
## 1     1        1708      1873  0
## 2     2        1780      1582  0
## 3     3        1718      1832  0
## 4     4        1719      1826  0
## 5     5        1777      1595  0
## 6     1        1642      1801  1
## 7     2        1707      1543  1
## 8     3        1642      1804  1
## 9     4        1656      1748  1
## 10    5        1722      1486  1
## 11    1        1569      1713  2
## 12    2        1633      1458  2
## 13    3        1577      1683  2
```

```
## 14    4         1573      1699  2
## 15    5         1636      1449  2
## 16    1         1564      1709  3
## 17    2         1629      1449  3
## 18    3         1573      1676  3
## 19    4         1568      1693  3
## 20    5         1628      1455  3
## 21    1         1562      1706  4
## 22    2         1626      1448  4
## 23    3         1569      1681  4
## 24    4         1567      1688  4
## 25    5         1626      1453  4
## 26    1         1562      1705  5
## 27    2         1626      1447  5
## 28    3         1569      1680  5
## 29    4         1566      1688  5
## 30    5         1625      1456  5
## 31    1         1559      1710  6
## 32    2         1625      1445  6
## 33    3         1567      1680  6
## 34    4         1565      1685  6
## 35    5         1624      1453  6
## 36    1         1559      1708  7
## 37    2         1625      1443  7
## 38    3         1566      1680  7
## 39    4         1564      1687  7
## 40    5         1623      1455  7
## 41    1         1558      1715  8
## 42    2         1625      1443  8
## 43    3         1566      1680  8
## 44    4         1564      1687  8
## 45    5         1622      1458  8
## 46    1         1555      1716  9
## 47    2         1623      1438  9
## 48    3         1563      1680  9
## 49    4         1559      1695  9
## 50    5         1621      1452  9
## 51    1         1555      1716 10
## 52    2         1623      1438 10
## 53    3         1563      1680 10
## 54    4         1559      1696 10
## 55    5         1621      1454 10
```

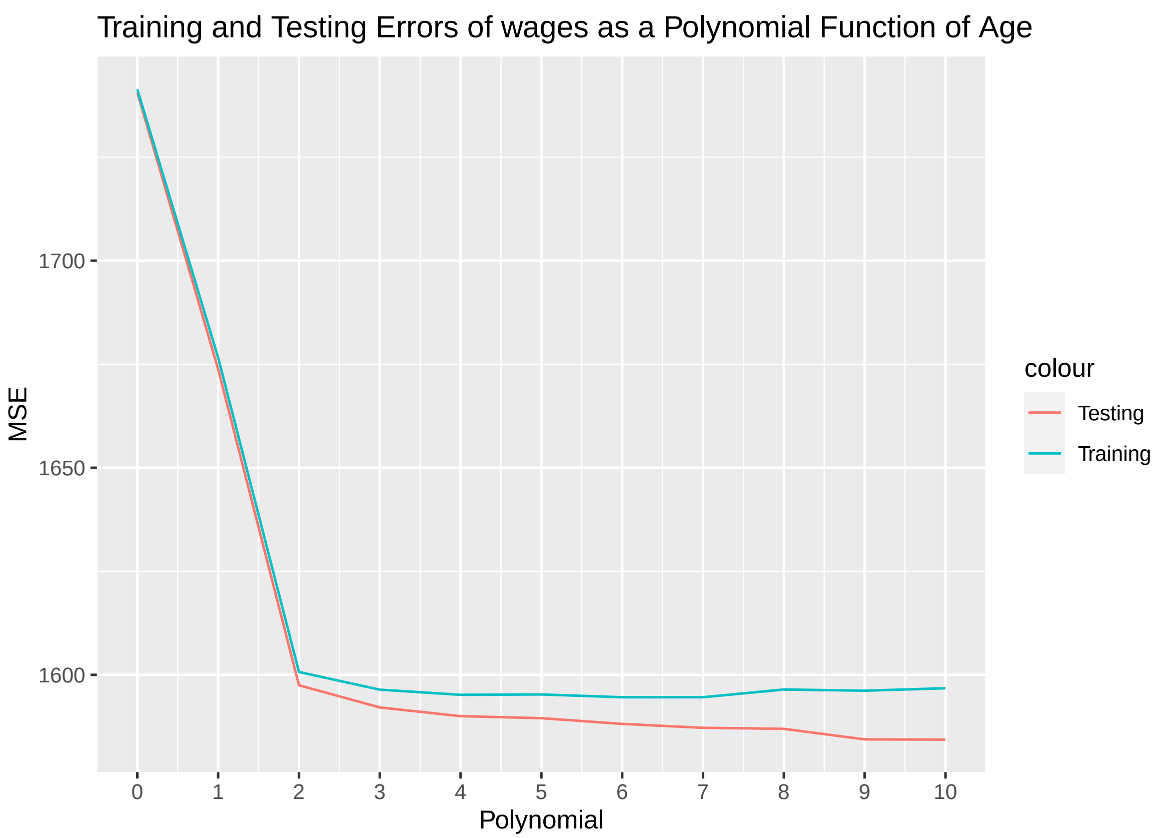
c) Plotting both the test error and training error for each of the models

```r
# mean for all chunks
final.error1 <- select(error.folds,-fold) %>% group_by(d)%>% summarise_each(funs(mean),train.error,
final.error1
```

```
## # A tibble: 11 x 3
##         d train.error val.error
##     <int>       <dbl>     <dbl>
##   1     0       1741.     1741.
##   2     1       1674.     1677.
```

```
## 3      2        1597.      1601.
## 4      3        1592.      1596.
## 5      4        1590.      1595.
## 6      5        1590.      1595.
## 7      6        1588.      1595.
## 8      7        1587.      1595.
## 9      8        1587.      1596.
## 10     9        1584.      1596.
## 11    10        1584.      1597.
```

```
train1 = final.error1$train.error
val1 = final.error1$val.error
#plotting the values
ggplot() + geom_line(aes(x=d,y=train1,colour =
                       "blue"), data =final.error1)+ geom_line(aes(x=abs(d),y=val1, colour =
                       "orange"), data =final.error1) + labs(title = "Training and Testing Errors of
    scale_color_discrete(labels = c("Testing", "Training")) + scale_x_continuous(breaks=c(0:10))
```



As p increases, the training error decreases sharply until the model with polynomial 2 and decreases at a much slower rate as the value of p increases. Similar to the training error, the test errors decreases sharply as p increases until polynomial model 2 and decreases at a much lower rate as p continues to increase. The values for training and test error are close in the beginning until model with polynomial 2 and as p increases, the value of test error is lower in comparison to training error. Based on the results, we would select the model with polynomial 2 as it is the point where the sharp decrease for testing and training error subsides. Although this is not the model with the lowest testing error, the testing error values are similar hence we would choose a simpler model which is model with polynomial 2 rather than a complex model with regression model 10.