PSTAT 131 Homework 1

Luis Aragon and Ronak Parikh

4/10/2019

PROBLEM 1

```
# Need Packages
library(readr)
library(dplyr)
```

```
##
## 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
```

```
library(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
# Load Data
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")
```

```
## Parsed with 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()
## )
```

```
# Summary of dataset
glimpse(algae)
```

```
## Observations: 200
## Variables: 18
## $ season <chr> "winter", "spring", "autumn", "spring", "autumn", "wint...
## $ size   <chr> "small", "small", "small", "small", "small", "small", "...
## $ speed  <chr> "medium", "medium", "medium", "medium", "medium", "high...
## $ mxPH   <dbl> 8.00, 8.35, 8.10, 8.07, 8.06, 8.25, 8.15, 8.05, 8.70, 7...
## $ mnO2   <dbl> 9.8, 8.0, 11.4, 4.8, 9.0, 13.1, 10.3, 10.6, 3.4, 9.9, 1...
## $ Cl     <dbl> 60.800, 57.750, 40.020, 77.364, 55.350, 65.750, 73.250,...
## $ NO3    <dbl> 6.238, 1.288, 5.330, 2.302, 10.416, 9.248, 1.535, 4.990...
## $ NH4    <dbl> 578.000, 370.000, 346.667, 98.182, 233.700, 430.000, 11...
## $ oPO4   <dbl> 105.000, 428.750, 125.667, 61.182, 58.222, 18.250, 61.2...
## $ PO4    <dbl> 170.000, 558.750, 187.057, 138.700, 97.580, 56.667, 111...
## $ Chla   <dbl> 50.000, 1.300, 15.600, 1.400, 10.500, 28.400, 3.200, 6....
## $ a1     <dbl> 0.0, 1.4, 3.3, 3.1, 9.2, 15.1, 2.4, 18.2, 25.4, 17.0, 1...
## $ a2     <dbl> 0.0, 7.6, 53.6, 41.0, 2.9, 14.6, 1.2, 1.6, 5.4, 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,...
## $ a4     <dbl> 0.0, 1.9, 0.0, 0.0, 0.0, 0.0, 3.9, 0.0, 0.0, 2.9, 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...
## $ a6     <dbl> 8.3, 0.0, 0.0, 0.0, 4.1, 12.6, 6.8, 8.7, 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, ...
```

PART 1A

```
# observations in each season
algae %>%
  group_by(season) %>%
  summarise(n())
```

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

PART 1B

```
# Count number of missing values
paste("Number of missing values", sum(is.na(algae)))
```

```
## [1] "Number of missing values 33"
```

```
# Mean for each chemical
chemMean <- algae %>%
  select(-c(season, size, speed, a1:a7, mxPH)) %>%
  summarise_all(mean, na.rm = TRUE)
```

```
chemMean
```

```
## # A tibble: 1 x 7
##    mnO2    Cl   NO3   NH4  oPO4   PO4  Chla
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  9.12  43.6  3.28  501.  73.6  138.  14.0
```

```
# Variance for each chemical
chemVar <- algae %>%
  select(-c(season, size, speed, a1:a7, mxPH)) %>%
  summarise_all(var, na.rm = TRUE)
chemVar
```

```
## # A tibble: 1 x 7
##    mnO2    Cl   NO3      NH4  oPO4    PO4  Chla
##   <dbl> <dbl> <dbl>    <dbl> <dbl>  <dbl> <dbl>
## 1  5.72 2193.  14.3 3851585. 8306. 16639.  420.
```

We can see that the quantities vary drastically from mean of 3.2824 for NO3 to mean of 137.5906 for PO4. Additionally, the variance is incredibly large for certain chemicals such as Cl NH4 and PO4.

PART 1C

```
# Median for each chemical
chemMedian <- algae %>%
  select(-c(season, size, speed, a1:a7, mxPH)) %>%
  summarise_all(median, na.rm = TRUE)
chemMedian
```

```
## # A tibble: 1 x 7
##    mnO2    Cl   NO3   NH4  oPO4   PO4  Chla
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1   9.8  32.7  2.68  103.  40.2  103.  5.48
```

```
# MAD for each chemical
chemMAD <- algae %>%
  select(-c(season, size, speed, a1:a7, mxPH)) %>%
  summarise_all(mad, na.rm = TRUE)
chemMAD
```

```
## # A tibble: 1 x 7
##    mnO2    Cl   NO3   NH4  oPO4   PO4  Chla
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  2.05  33.2  2.17  112.  44.0  122.  6.67
```

```
# Create dataframe for comparison
compareDF <- data.frame(rbind(chemMean, chemVar, chemMedian, chemMAD))
row.names(compareDF) <- c("Mean", "Var", "Median", "MAD")
compareDF
```

```
##            mnO2         Cl       NO3          NH4       oPO4        PO4
## Mean   9.117778   43.63628  3.282389     501.2958   73.59060   137.8821
## Var    5.718089 2193.17173 14.261756 3851584.6849 8305.84993 16639.3845
## Median 9.800000   32.73000  2.675000     103.1665   40.15000   103.2855
## MAD    2.053401   33.24953  2.172009     111.6175   44.04582   122.3212
##            Chla
## Mean    13.9712
```

```
## Var    420.0827
## Median   5.4750
## MAD      6.6717
```

The MAD and the median are relatively close for each element except for mnO2 where the median is much larger than the MAD. Perhaps these measurements are closer than the mean and variance measurements because they are less sensitive to outliers.

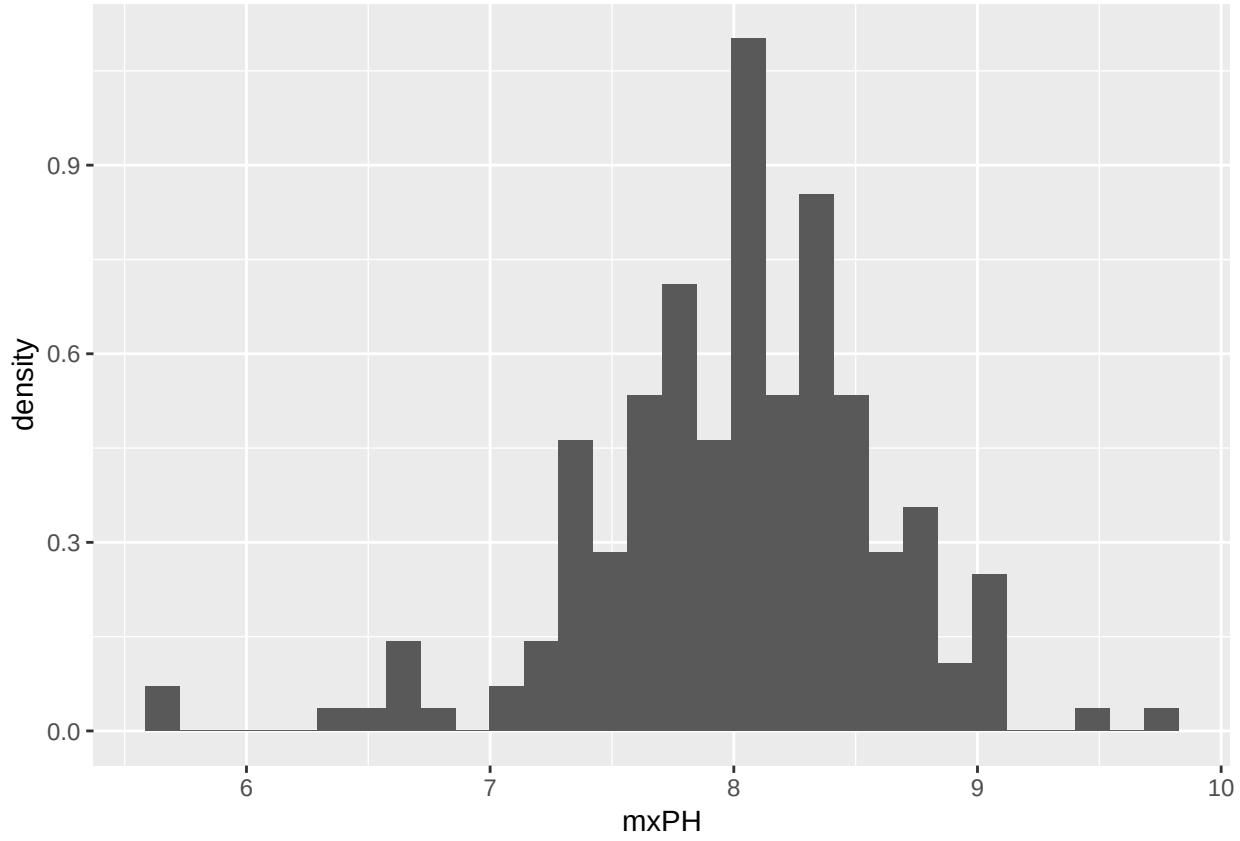
PROBLEM 2

PART 2A

```
# Histogram of mxPH
ggplot(algae, aes(mxPH), na.rm=TRUE) +
  geom_histogram(aes(y = stat(density)))
```

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

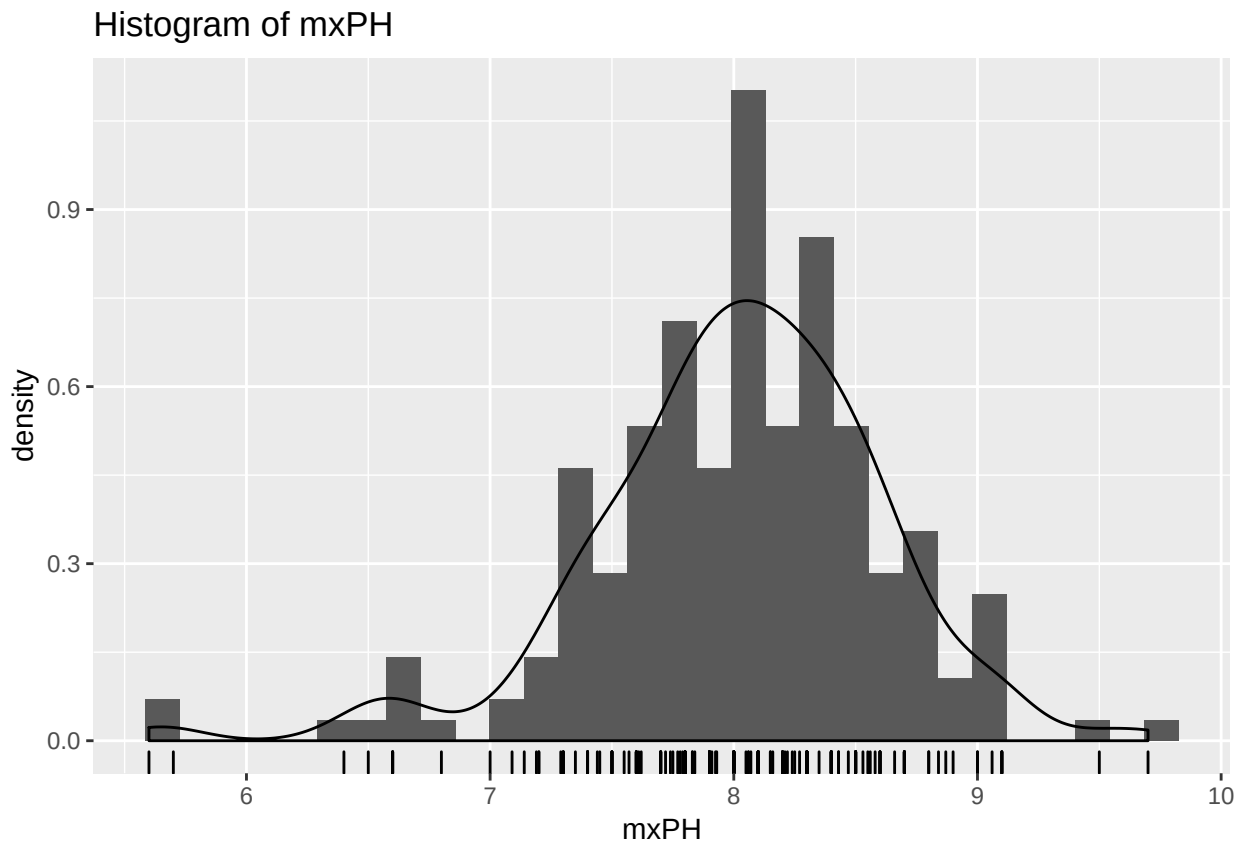
```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```



PART 2B

```
# Add geom_density and geom_rug
ggplot(algae, aes(mxPH), na.rm=TRUE) +
  geom_histogram(aes(y = stat(density))) +
  geom_density() + geom_rug() + ggtitle('Histogram of mxPH')
```

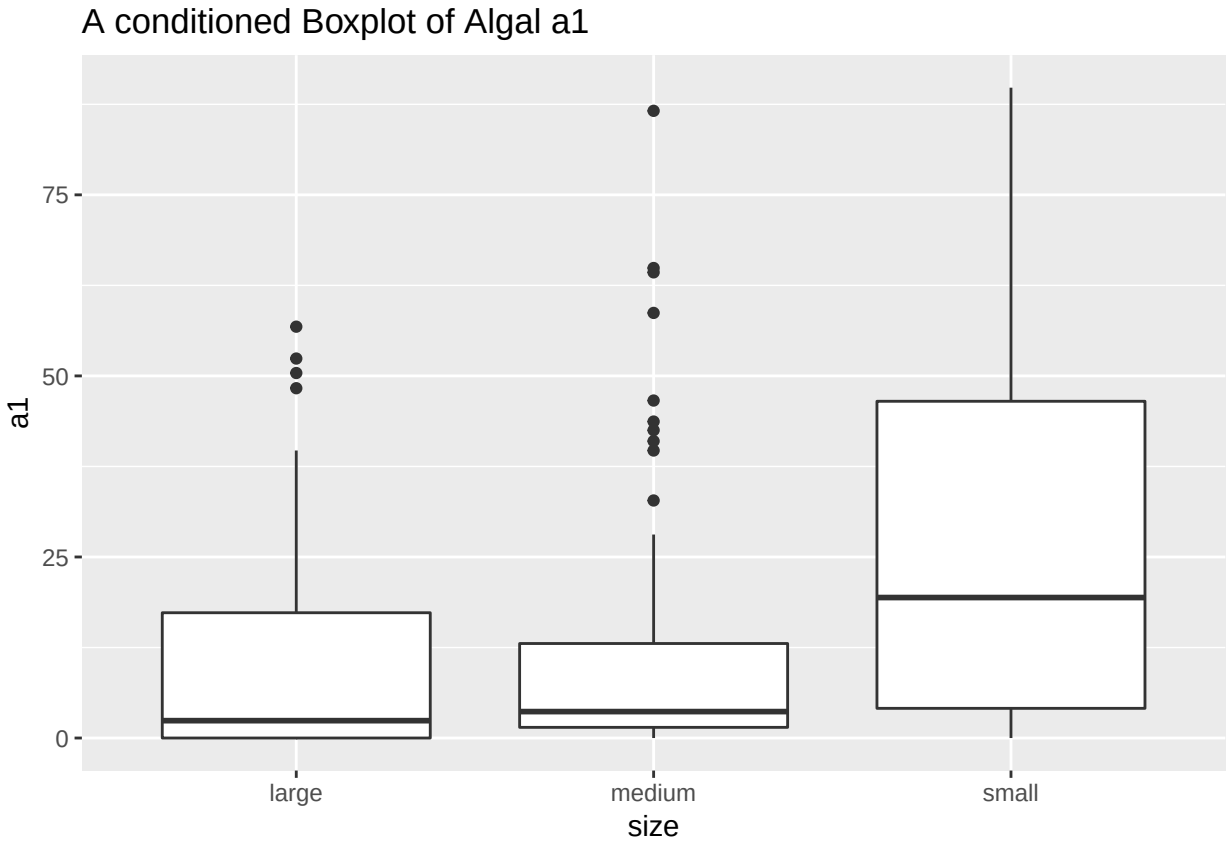
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 1 rows containing non-finite values (stat_bin).
## Warning: Removed 1 rows containing non-finite values (stat_density).
```



The distribution of the mxPH does not look significantly skewed. It looks like it comes from a normal distribution with a mean around 8 mxPH. If we wanted to confirm it comes from a normal distribution, we would use a normal QQ Plot. Furthermore, there seems to be outliers with low mxPH. Again, this is from visual inference because we have not definitively named them outliers.

PART 2C

```r
# Boxplot
ggplot(algae, aes(x=size, y=a1)) +
  geom_boxplot() +
  ggtitle('A conditioned Boxplot of Algal a1')
```



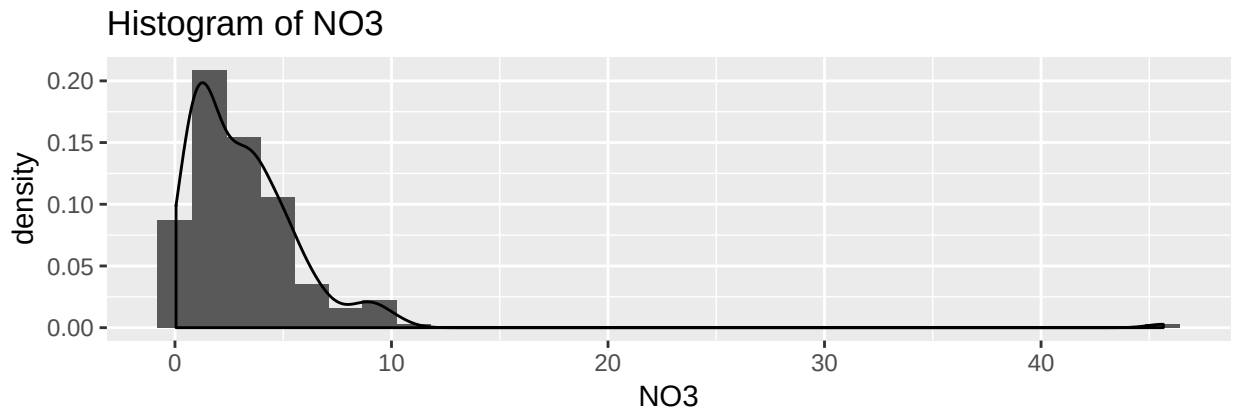
PART 2D

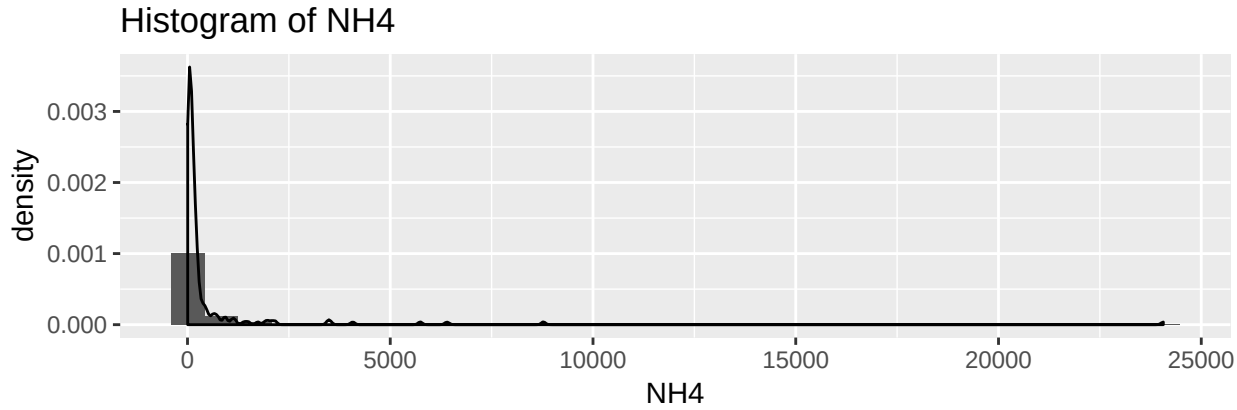
```
no3_hist <- ggplot(algae, aes(NO3, stat(density))) +
  geom_histogram() + ggtitle('Histogram of NO3') +
  geom_density()

nh4_hist <- ggplot(algae, aes(NH4, stat(density))) +
  geom_histogram() + ggtitle('Histogram of NH4') +
  geom_density()

# Plot histograms
grid.arrange(no3_hist, nh4_hist, nrow=2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 2 rows containing non-finite values (stat_bin).
## Warning: Removed 2 rows containing non-finite values (stat_density).
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 2 rows containing non-finite values (stat_bin).
## Warning: Removed 2 rows containing non-finite values (stat_density).
```





By visual observation, we see that there are high outliers for both NO_3 and NH_4 to the far right. We need an objective way to identify these outliers. Since we are not fitting a model, we can use the simple Interquartile range (IQR) method.

We will use the Interquartile range (IQR) method to find outliers for NO_3 and NH_4. This common objective method defines a boundary using the IQR and any observation outside the boundary is considered an outlier. The boundary is defined by

$$Upper.Q3 + 1.5 * IQR$$

$$Lower : Q1 - 1.5 * IQR$$

where Q2 and Q3 are the 25 and 75 quartiles and $IQR = Q3 - Q1$

```
# IQR Method
# Compute boundaries
outlier_cutoff_upper <- quantile(algae$NO3, 0.75, na.rm = TRUE) + 1.5 * IQR(algae$NO3, na.rm = TRUE)
outlier_cutoff_lower <- quantile(algae$NO3, 0.25, na.rm = TRUE) - 1.5 * IQR(algae$NO3, na.rm = TRUE)

# Extract observations outside boundaries
index_outlier <- which(algae$NO3 > outlier_cutoff_upper | algae$NO3 < outlier_cutoff_lower)
length(index_outlier)
```

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

Using the Interquartile Range Method, we found 5 outliers for NO_3. We will repeat the method for NH_4.

```
# IQR Method
# Compute boundaries
outlier_cutoff_upper_nh4 <- quantile(algae$NH4, 0.75, na.rm = TRUE) + 1.5 * IQR(algae$NH4, na.rm = TRUE)
outlier_cutoff_lower_nh4 <- quantile(algae$NH4, 0.25, na.rm = TRUE) - 1.5 * IQR(algae$NH4, na.rm = TRUE)
```

```
# Extract observations outside boundaries
index_outlier_nh4 <- which(algae$NH4 > outlier_cutoff_upper_nh4 | algae$NH4 < outlier_cutoff_lower_nh4)
length(index_outlier_nh4)
```

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

Using the IQR method, we discovered 27 outliers fro NH4.

PART 2E

```
mean_no3_nh4 <- algae %>%
  select(c(NO3, NH4)) %>%
  summarise_all(mean, na.rm = TRUE)

# Variance for each chemical
var_no3_nh4 <- algae %>%
  select(c(NO3, NH4)) %>%
  summarise_all(var, na.rm = TRUE)

median_no3_nh4 <- algae %>%
  select(c(NO3, NH4)) %>%
  summarise_all(median, na.rm = TRUE)


MAD_no3_nh4 <- algae %>%
  select(c(NO3, NH4)) %>%
  summarise_all(mad, na.rm = TRUE)

# Create dataframe for comparison
df_no3_nh4 <- data.frame(rbind(mean_no3_nh4, var_no3_nh4, median_no3_nh4, MAD_no3_nh4))
row.names(df_no3_nh4) <- c("Mean", "Var", "Median", "MAD")
df_no3_nh4
```

```
##              NO3          NH4
## Mean    3.282389     501.2958
## Var    14.261756 3851584.6849
## Median  2.675000     103.1665
## MAD     2.172009     111.6175
```

After looking at the data.frame above, we noticed that the median of NH4 is much lower than the mean. Additionally, the variance of NH4 is very high because of the abudance of outliers in the data.

Aftering comparing NO3 and NH4, it appears that the median and median absolute deviation are more robust when outliers are present because of the large differences in mean and variance between the two chemicals.

PROBLEM 3

PART 3A

```
# Count observations with missing columns
rowMiss <- sum(!complete.cases(algae))
paste("Num of observations with NA: ", rowMiss)
```

```
## [1] "Num of observations with NA:  16"
```

```
# Print missing values per column
sapply(algae, function(x) sum(is.na(x)))
```

```
## 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
```

Above is a table showing missing values for each chemical.

PART 3B

```
# Filter data by complete cases
algae.del <- filter(algae, complete.cases(algae))

# Count up complete cases
tally(algae.del)
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1   184
```

The data set `algae.del` has 184 total observations

PART 3C

```
# Imputate: NA --> Median of column
algae.med <- algae %>%
  mutate_at(.vars = vars(mxPH:Chla), funs(ifelse(is.na(.),median(., na.rm = TRUE),.)))

# Print 1st 3 rows
head(algae.med, 3)
```

```
## # A tibble: 3 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 medium  8      9.8  60.8  6.24  578   105   170   50     0
## 2 spring small medium  8.35   8    57.8  1.29  370   429.  559.   1.3   1.4
## 3 autumn small medium  8.1   11.4  40.0  5.33  347.  126.  187.  15.6   3.3
## # ... with 6 more variables: a2 <dbl>, a3 <dbl>, a4 <dbl>, a5 <dbl>,
## #   a6 <dbl>, a7 <dbl>
```

```
# Display 48, 62, 199
cbind(observation = c(48,62,199), rbind(algae.med[48,1:11], algae.med[62,1:11], algae.med[199,1:11]))
```

```
##   observation season  size  speed mxPH mnO2    Cl   NO3      NH4  oPO4
## 1          48 winter small    low 8.06 12.6  9.00 0.230  10.0000  5.00
## 2          62 summer small medium 6.40  9.8 32.73 2.675 103.1665 40.15
## 3         199 winter large medium 8.00  7.6 32.73 2.675 103.1665 40.15
##        PO4  Chla
## 1   6.0000 1.100
## 2  14.0000 5.475
## 3 103.2855 5.475
```

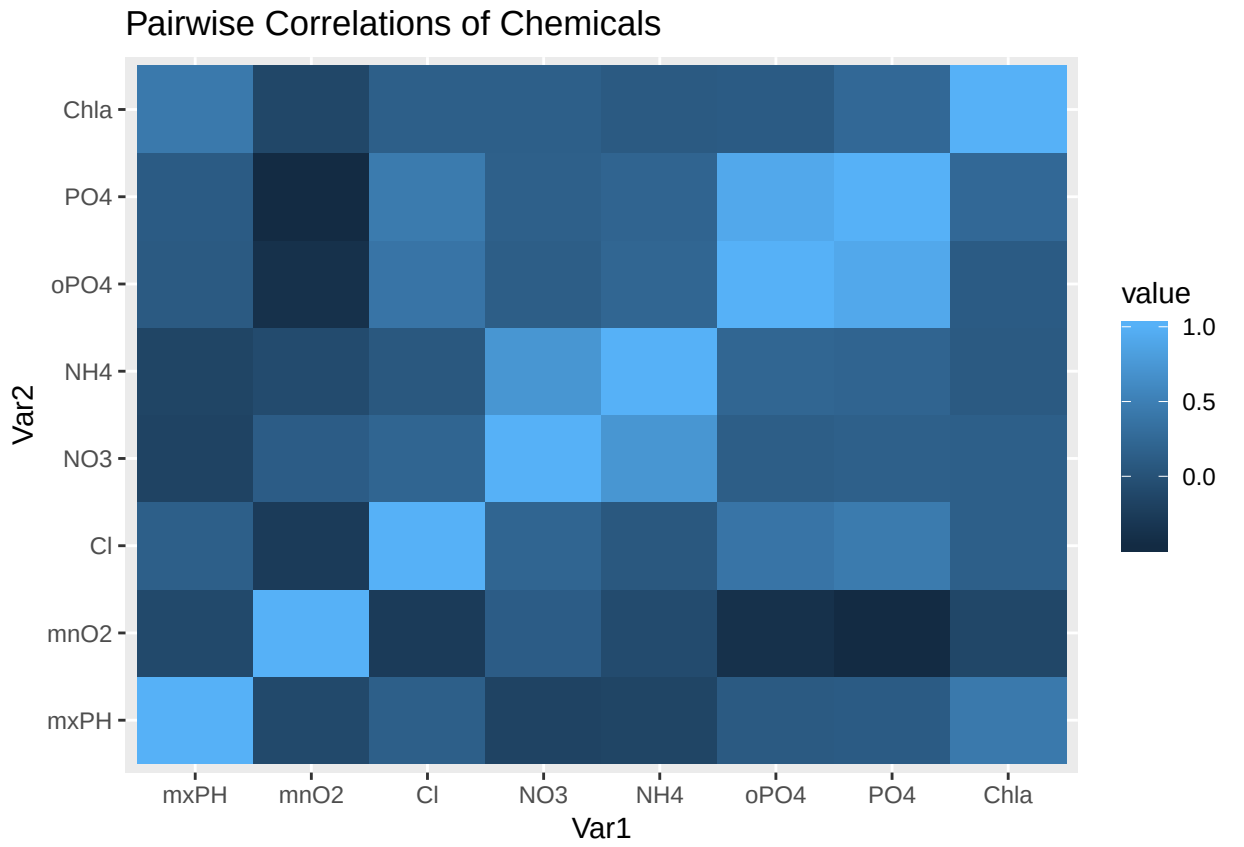
Above, we imputed missing values with the median of the column and then printed a table containing the 48th, 62nd, and 199th observations.

PART 3D

```
library(reshape2)
xmat <- algae %>% select(c(mxPH:Chla))
algae_corr = cor(xmat, use = "complete.obs")
algae_corr
```

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

```
ggplot(data = melt(algae_corr), aes(x=Var1, y=Var2, fill=value)) +
  geom_tile() + ggtitle("Pairwise Correlations of Chemicals")
```




```
model <- lm(data = algae, PO4 ~ oPO4)

algae$oPO4[28] <- predict(model, algae[28,])
paste("Imputed Value from Regression:", algae$oPO4[28])
```

```
## [1] "Imputed Value from Regression: 48.0692899726549"
```

The value we obtained for the 28th oberservation has the PO4 value is 48.06929. The pairwise correlation can be seen in `algae.cor` above. A heat map of the correlations can also be shown above.

PART 3E

Similar to the surviorship bias with the airplanes in lecture 2, we can apply a similar principle to this dataset. Because there may be bias in previously measured data, imputution may not be a proper substitute for missing data values.

The data for the algae was collected from European rivers at different times during a period of approximately 1 year. Some of the algae might have more or less concentration of algae based on the season of the year. In addition, different parts of the river may have different concentrations of algae and thus would not be ideal for imputation.

PROBLEM 4

PART 4A

```
# Create 5 groups
# set.seed(343)
```

```r
partitions <- cut(1:200, label = FALSE, breaks = 5) %>%
  sample()
```

```r
# Cross Validation function
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
}
```

```r
# 5 folds
lapply(1:5, do.chunk, chunkdef = partitions, dat=algae.med)
```

```
## [[1]]
##   fold train.error val.error
## 1    1    292.8355  271.2005
##
## [[2]]
##   fold train.error val.error
## 1    2    299.4782  265.8164
##
## [[3]]
##   fold train.error val.error
## 1    3    247.0347  493.8911
##
## [[4]]
##   fold train.error val.error
## 1    4     272.378  374.1106
##
## [[5]]
##   fold train.error val.error
## 1    5     285.502  563.3714
```

PROBLEM 5

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

```
## Parsed with 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()
## )
```

```
# Define model using algae.med
model <- lm(a1~., data = algae.med[, 1:12])

# Use model on algae.Test
# Predict + Calculate true error
mean((algae.Test$a1 - predict.lm(model, algae.Test)) ^ 2)
```

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

This true error of 250.1794, is what we expect because the validation error from part 4 varies from 250 to 450 (with the training errors closer to 250) depending on the randoming sorting of the 5 folds. Considering that we are testing on more data in algae.Test (coming from the same distribution as alga.med), it makes sense that the test error is closer to 250.

PROBLEM 6

```
# Load in packag for data
library(ISLR)

# First few rows of Wage data
head(Wage)
```

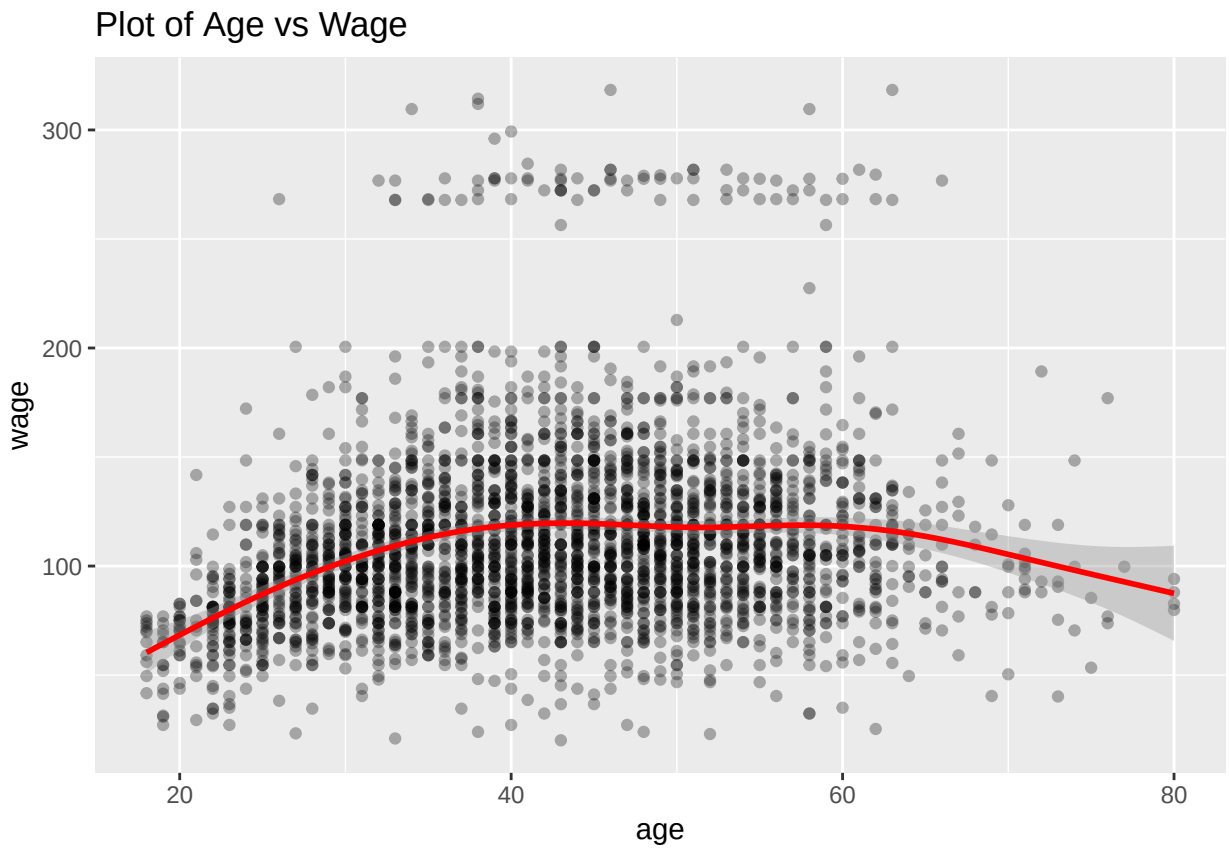
```
##        year age           maritl     race       education
## 231655 2006  18 1. Never Married 1. White    1. < HS Grad
## 86582  2004  24 1. Never Married 1. White 4. College Grad
## 161300 2003  45       2. Married 1. White 3. Some College
## 155159 2003  43       2. Married 3. Asian 4. College 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
```

PART 6A

```
# Plot age vs wage
ggplot(Wage, aes(x = age, y = wage)) +
  geom_point(alpha = 0.3) +
  geom_smooth(color = 'red') +
  ggtitle('Plot of Age vs Wage')
```

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



This plot matches what we expect. As age increases, wage goes from increasing to plateauing to decreasing. This follows expected career trajectories (from entry level position to promotions to retirement).

PART B.i

```
# Loop to fit model for each p = 0:10
p = 0
while (p < 10) {
  if (p==0) {
    fit <-lm(Wage$wage~1)    # fit model on intercept
    }
```

```
  else {
    fit <-lm(wage~poly(age, p), data=Wage)   # fit model with polynomial p
    }
  print(fit)
  p = p+1
}
```

```
##
## Call:
## lm(formula = Wage$wage ~ 1)
##
## Coefficients:
## (Intercept)
##       111.7
##
##
## Call:
## lm(formula = wage ~ poly(age, p), data = Wage)
##
## Coefficients:
##  (Intercept)  poly(age, p)
##        111.7         447.1
##
##
## Call:
## lm(formula = wage ~ poly(age, p), data = Wage)
##
## Coefficients:
##   (Intercept)  poly(age, p)1  poly(age, p)2
##         111.7          447.1         -478.3
##
##
## Call:
## lm(formula = wage ~ poly(age, p), data = Wage)
##
## Coefficients:
##   (Intercept)  poly(age, p)1  poly(age, p)2  poly(age, p)3
##         111.7          447.1         -478.3          125.5
##
##
## 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
##
##
## 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
##       -35.81
##
##
## 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
##       -35.81          62.71
##
##
## 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
##       -35.81          62.71          50.55
##
##
## 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
##       -35.81          62.71          50.55         -11.25
##
##
## 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
```

PART 6B.ii

```r
set.seed(96)

# Create 5 partitions
partitions2 <- cut(1:nrow(Wage), label=FALSE, breaks=5) %>% sample()
head(partitions2)
```

```
## [1] 3 5 5 1 5 2
```

```r
# CV function
do.chunk2 <- function(chunkid, chunkdef, dat, l){ # function argument

  train = (chunkdef != chunkid)

  training = dat[train,]
  testing = dat[!train,]

  # fit training data to model
  if (l==0) {
    fitwage <-lm(wage ~ 1, data = dat[train,])
    }
  else {
    fitwage <-lm(wage ~ poly(age, degree=l), data = dat[train,])
    }

  predYtr = predict(fitwage) # predict training values
  predYvl = predict(fitwage, testing) # predict validation values

  data.frame(fold = chunkid,
             train.error = mean((predYtr - training$wage)^2), # compute and store training error
             val.error = mean((predYvl - testing$wage)^2)) # compute and store test error
}
```

```r
test.errors=NULL
train.errors=NULL

set.seed(131)
# Train and test for each polynomial l
for (i in 0:10) {

  # Get 5 fold CV for polynomial l
  tmp = lapply(1:5, do.chunk2, chunkdef=partitions2, dat=Wage[,c("age","wage")], l=i)

  # Get average training error over 5 folds
  mean.err.train = mean(c(tmp[[1]][["train.error"]], tmp[[2]][["train.error"]],
                          tmp[[3]][["train.error"]], tmp[[4]][["train.error"]],
                          tmp[[5]][["train.error"]]))

  # Get average testing error over 5 folds
  mean.err.test = mean(c(tmp[[1]][["val.error"]], tmp[[2]][["val.error"]],
                         tmp[[3]][["val.error"]], tmp[[4]][["val.error"]],
                         tmp[[5]][["val.error"]]))

  # Append to vector
  train.errors = c(train.errors, mean.err.train)
  test.errors = c(test.errors, mean.err.test)
}

# Create dataframe with polynomials and errors
polynomial <- c(0:10)
error.df <- data.frame(polynomial, train.errors, test.errors)

# Print errors
```

```
grid.table(error.df)
```

	polynomial	train.errors	test.errors
1	0	1740.41410954854	1743.2255797719
2	1	1673.76174248224	1676.86460529663
3	2	1597.24950516628	1602.88019964527
4	3	1591.95115381406	1598.03359071443
5	4	1589.84198382523	1596.80044958599
6	5	1589.37175681131	1596.76342747909
7	6	1587.98323280856	1596.16325780143
8	7	1587.06626112094	1595.90517826737
9	8	1587.00933695994	1595.99341650875
10	9	1584.59251358254	1594.41860290788
11	10	1584.42738389805	1595.94586821052