Homework 1

Leonel Carlen, Stefan Ciric

October 18, 2019

```
## 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()
## )
```

```
## 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, ...
```

Problem 1

##1.a

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

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

1.b

```
notNA <- algae %>%
  summarise_at(.funs=funs(sum(!is.na(.))), .vars = vars(mxPH:Chla))
```

```
## Warning: funs() is soft deprecated as of dplyr 0.8.0
## please use list() instead
##
##   # Before:
##   funs(name = f(.))
##
##   # After:
##   list(name = ~ f(.))
## This warning is displayed once per session.
```

```
#Compute mean of each chemical
chemMean <- algae %>%
  summarise_at(vars(mxPH:Chla), mean, na.rm=TRUE)
#Computer var of each chemical
chemVar <- algae %>%
  summarise_at(vars(mxPH:Chla), var, na.rm=TRUE)
Names <- c("Count", "Mean", "Variance")
cbind(Names, rbind(notNA, chemMean, chemVar))
```

```
##      Names        mxPH       mnO2         Cl        NO3          NH4
## 1    Count 199.0000000 198.000000  190.00000 198.000000     198.0000
## 2     Mean   8.0117337   9.117778   43.63628   3.282389     501.2958
## 3 Variance   0.3579693   5.718089 2193.17173  14.261756 3851584.6849
##        oPO4        PO4     Chla
## 1  198.0000   198.0000 188.0000
## 2   73.5906   137.8821  13.9712
## 3 8305.8499 16639.3845 420.0827
```

```
#The large variance of NH4, oPO4 and PO4 indicate, that the mean isn't very useful.
```

1.c

```
# Compute median of each chemical
chemMed <- algae %>%
  dplyr::select(mxPH:Chla) %>%
  summarise_all(function(z) median(z, na.rm=TRUE))
# Compute mad of each chemical
chemMad <- algae %>%
```

```
  summarise_at(vars(mxPH:Chla), funs(mad), na.rm = TRUE)
rnames <- c("Med", "Mean", "Mad", "Var")
# cbind(rnames,rbind(chemMed, chemMean, chemMad, chemVar))
tab<-rbind(chemMed, chemMean, chemMad, chemVar)
cbind(rnames,tab)
```

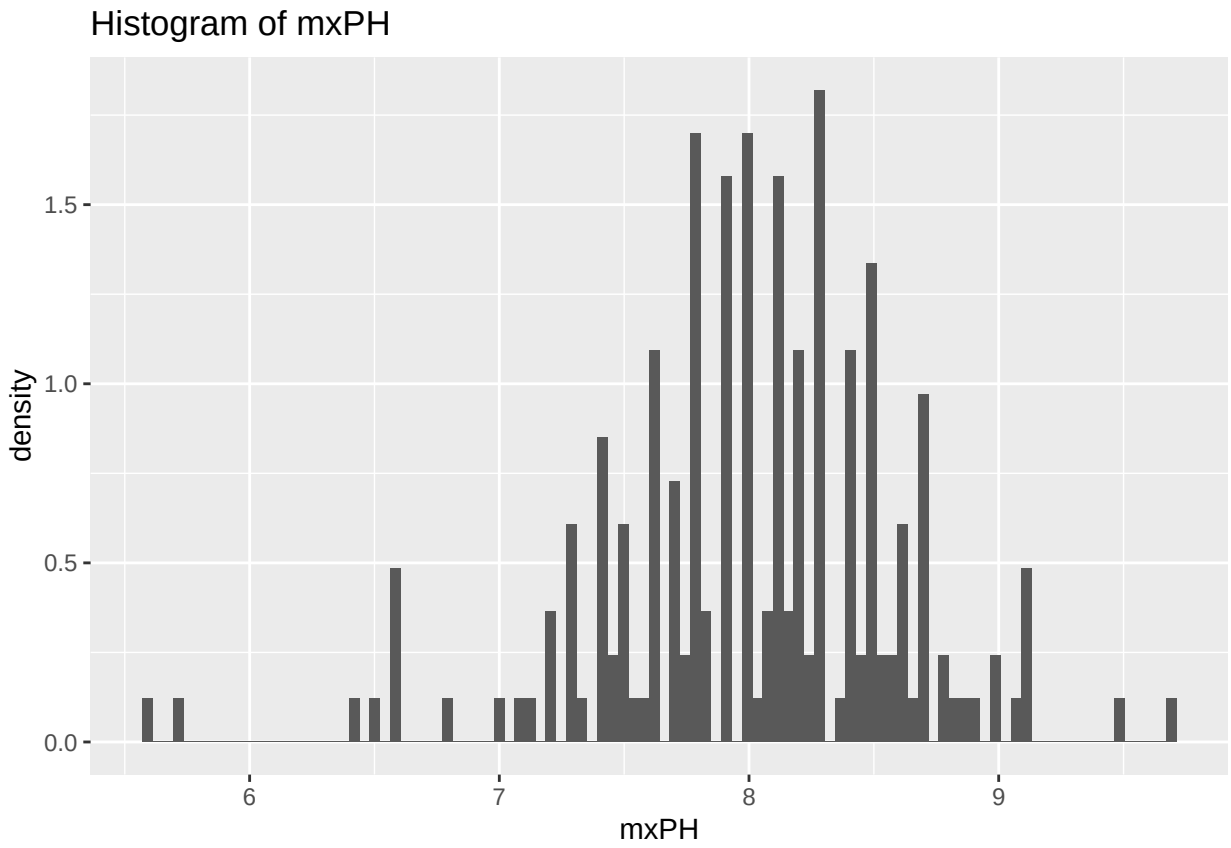
```
##   rnames      mxPH     mnO2         Cl       NO3          NH4       oPO4
## 1    Med 8.0600000 9.800000   32.73000  2.675000     103.1665   40.15000
## 2   Mean 8.0117337 9.117778   43.63628  3.282389     501.2958   73.59060
## 3    Mad 0.5040840 2.053401   33.24953  2.172009     111.6175   44.04582
## 4    Var 0.3579693 5.718089 2193.17173 14.261756 3851584.6849 8305.84993
##          PO4     Chla
## 1   103.2855   5.4750
## 2   137.8821  13.9712
## 3   122.3212   6.6717
## 4 16639.3845 420.0827
```

The mean for each attribute is in general higher than the median The variance is larger than the mad. The differences indicate that the average was skewed by extremely high values. Therefore, the data is more broadly spread around the mean than around the median. The much smaller MAD vs. the Var indicates the presence of influential points, potentially outliers.

2

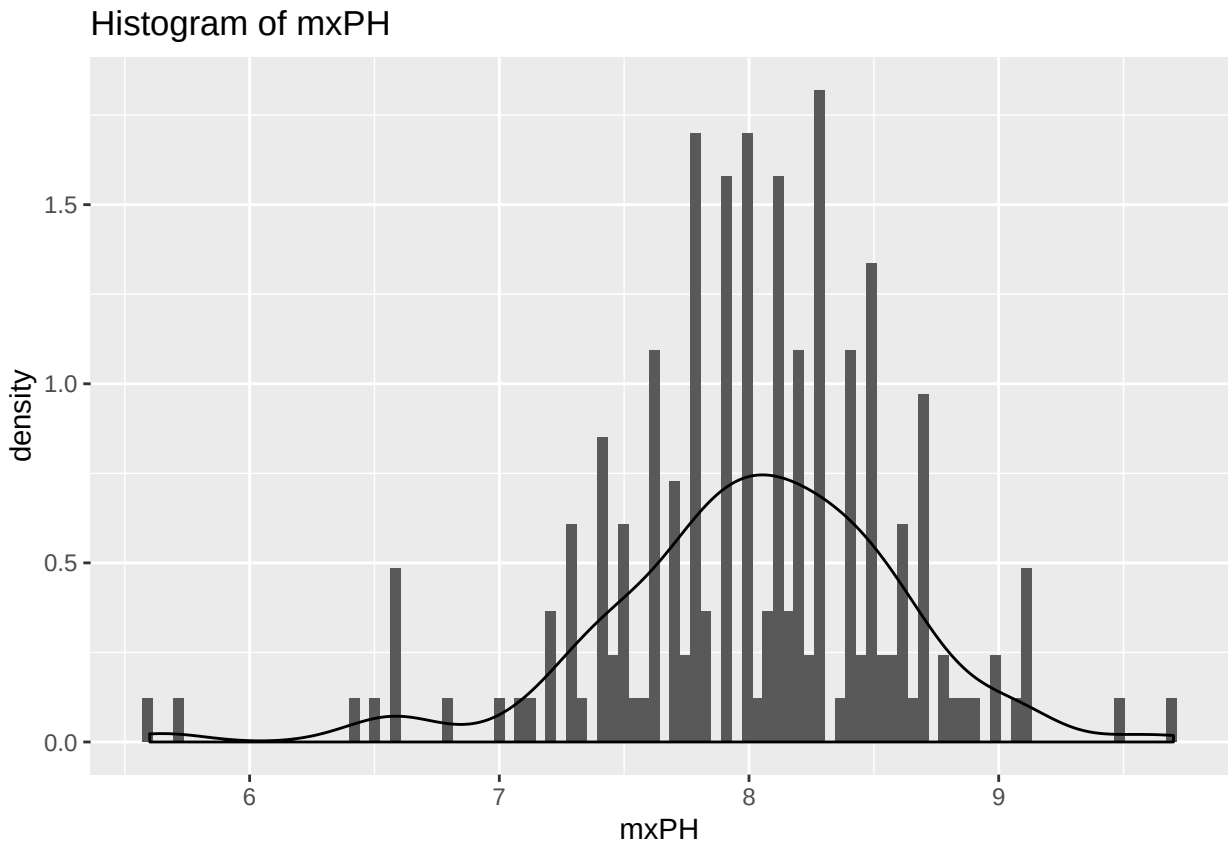
2.a

```
mxphPlot <- algae%>%
  drop_na(mxPH)%>%
  ggplot(aes(mxPH,stat(density))) +
  geom_histogram(bins = 100) + ggtitle("Histogram of mxPH")
mxphPlot
```



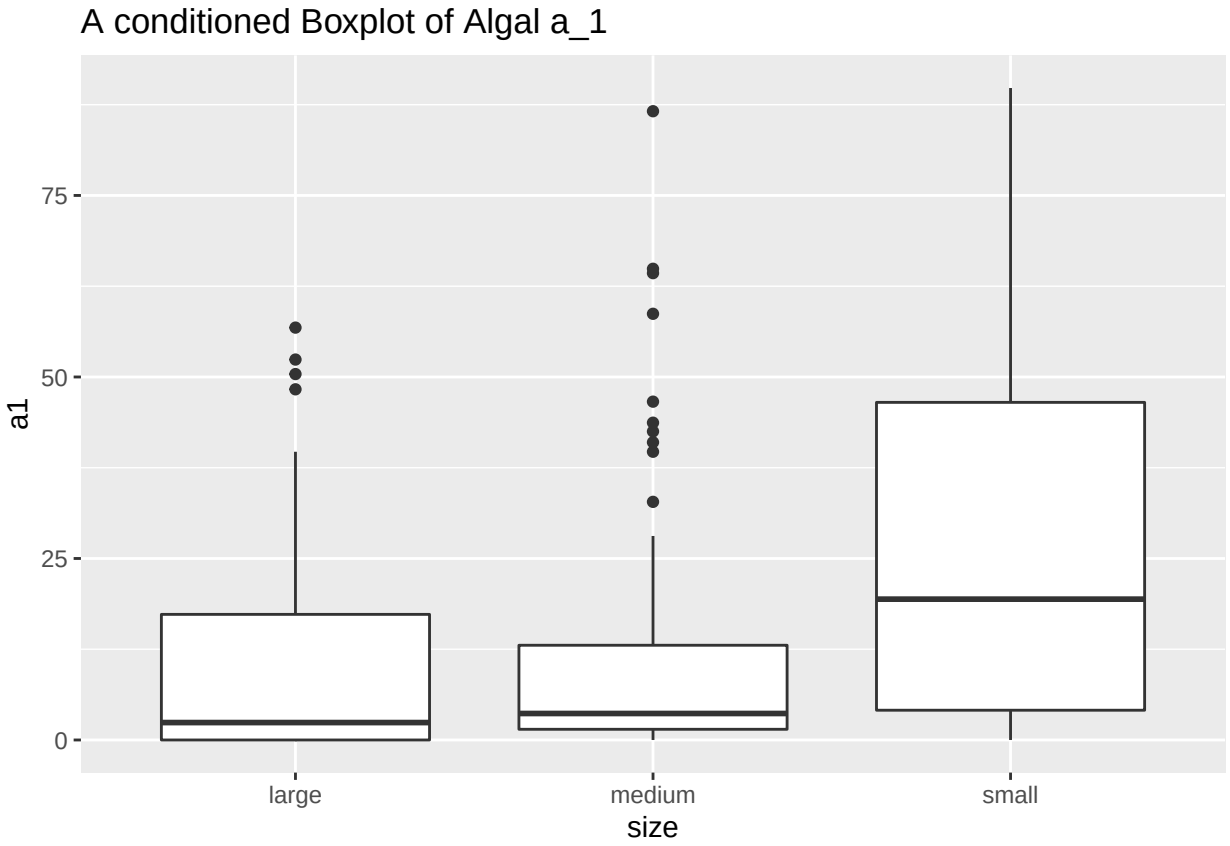
2.b

```
algae%>%
  drop_na(mxPH)%>%
  ggplot(aes(mxPH,stat(density))) +
  geom_histogram(bins = 100) + ggtitle("Histogram of mxPH") +
  geom_density(inherit.aes = TRUE)
```



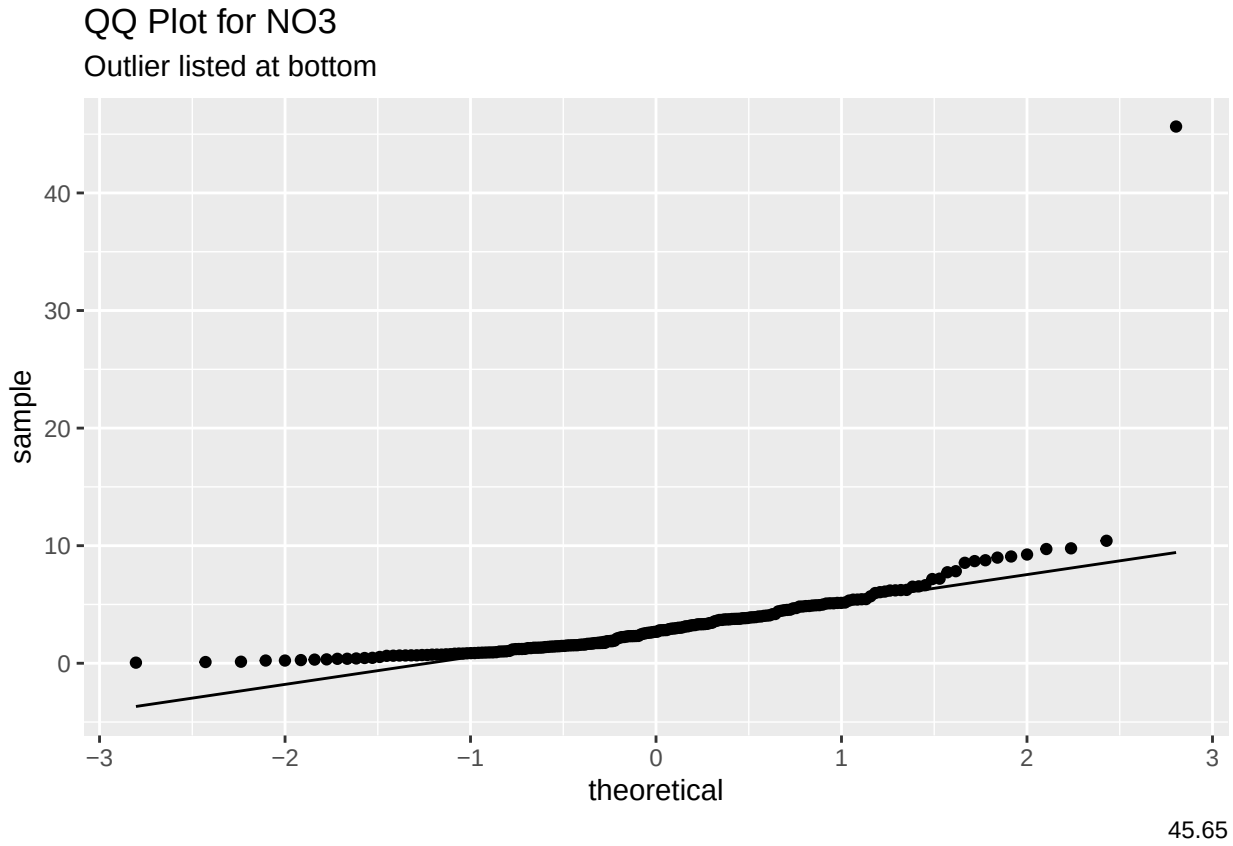
2.c

```
a1Box <- ggplot() + geom_boxplot(data=algae, aes(y=a1, x=size)) +
  ggtitle('A conditioned Boxplot of Algal a_1')
a1Box
```



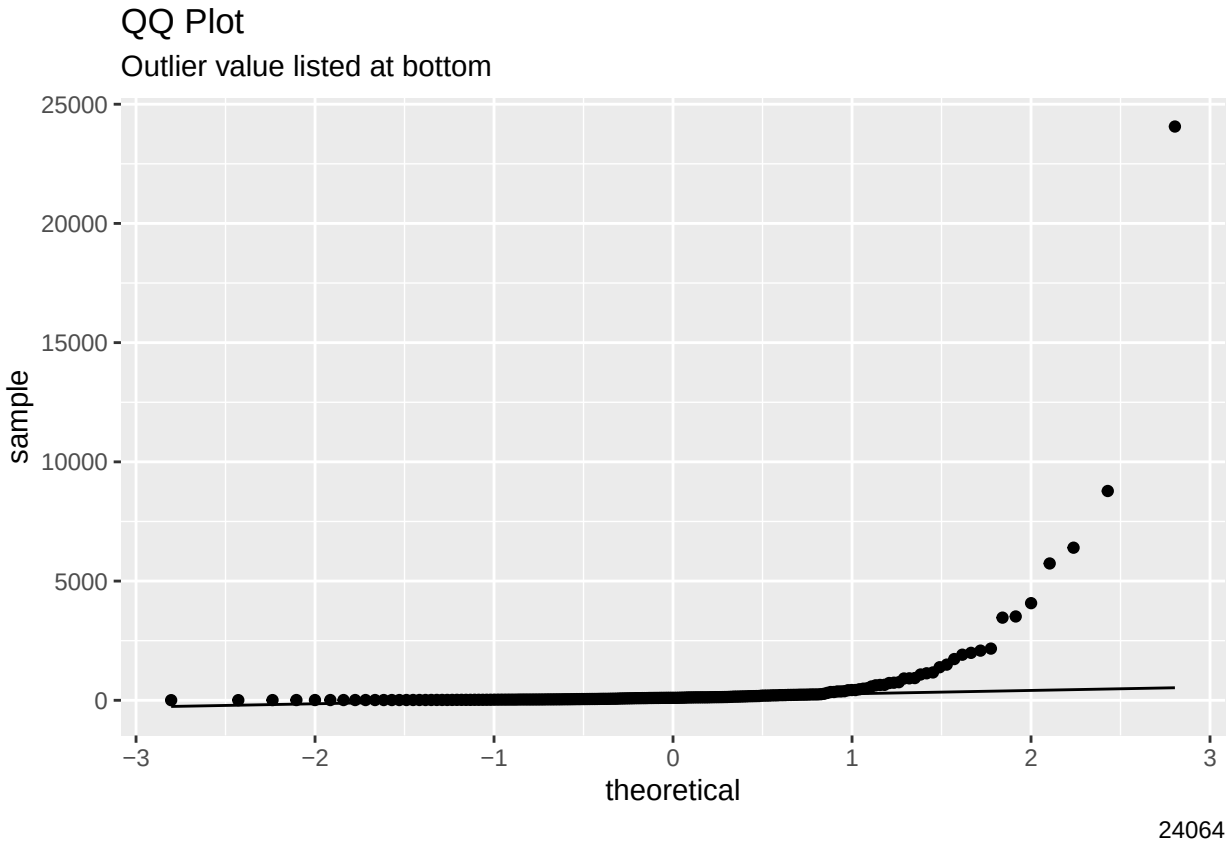
2.d

```
# Use ggplot function stat_qq() and stat_qq_line to find outliers in NO3
outNO3 <- algae%>%
  drop_na(NO3)%>%
  ggplot(aes(sample = NO3)) + stat_qq()+stat_qq_line()+
  labs(title = "QQ Plot for NO3", subtitle = "Outlier listed at bottom",
       caption = max(algae$NO3, na.rm=TRUE))
outNO3
```



The data contains about 30 outliers. One of these outliers is a critical lever point.

```
# Find outlier in NH4
outNO4 <- algae%>%
  drop_na(NH4)%>%
  ggplot(aes(sample = NH4)) + stat_qq()+stat_qq_line()+
  labs(title = "QQ Plot", subtitle = "Outlier value listed at bottom", caption = max(algae$NH4, na.rm=T
outNO4
```



The data points show a systematic deviation. Seven outliers are critical leverage points

2.e

```
# Compute median of each chemical
medNO3NH4 <- algae%>%
  dplyr::select(NO3, NH4)%>%
  summarise_all(function(z) median(z, na.rm=TRUE))
# Compute mad of each chemical
madNO3NH4 <- algae%>%
  summarise_at(vars(NO3, NH4), funs(mad), na.rm = TRUE)
#Compute mean of each chemical
meanNO3NH4 <- algae%>%
  summarise_at(vars(NO3, NH4), mean, na.rm=TRUE)
#Computer var of each chemical
varNO3NH4 <- algae%>%
  summarise_at(vars(NO3, NH4), var, na.rm=TRUE)
myTable <- rbind(medNO3NH4,meanNO3NH4, madNO3NH4,varNO3NH4)
Stat <- c("Med", "Mean", "Mad", "Var")
nTable <- cbind(Stat, myTable)
nTable
```

```
##   Stat       NO3          NH4
## 1  Med  2.675000     103.1665
## 2 Mean  3.282389     501.2958
## 3  Mad  2.172009     111.6175
## 4  Var 14.261756 3851584.6849
```

As expected, the mean is much higher than the median and the variance is much higher than the MAD, which indicates that not only is the measure of central tendency more temperamental in the presence of outliers, but the spread appears much broader when the variance is used, than the MAD would indicate. Hence the median and MAD are again the more robust measures.

3

3.a

```
fAlgae <- filter(algae, is.na(mxPH)|is.na(mnO2)|is.na(Cl)|is.na(NO3)|is.na(NH4)|is.na(oPO4)|
                   is.na(PO4)|is.na(Chla))

cat("The number of observations that contain one or more missing values is",nrow(fAlgae), "\n")
```

```
## The number of observations that contain one or more missing values is 16
```

```
isNA = notNA
for(i in 1:length(notNA))
  isNA[[i]] = 200 - notNA[[i]]
print("The number of missing values in each column is listed in the table below:")
```

```
## [1] "The number of missing values in each column is listed in the table below:"
```

```
isNA
```

```
## # A tibble: 1 x 8
##    mxPH  mnO2    Cl   NO3   NH4  oPO4   PO4  Chla
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     1     2    10     2     2     2     2    12
```

3.b

```
algae.del <- filter(algae, !is.na(mxPH)&!is.na(mnO2)&!is.na(Cl)&!is.na(NO3)&!is.na(NH4)
                    &!is.na(oPO4)&!is.na(PO4)&!is.na(Chla))

cat("There are",nrow(algae.del),"observations without missing values in the dataset.")
```

```
## There are 184 observations without missing values in the dataset.
```

3.c

```
algae.med <- algae%>%
  mutate_at(.vars = vars(4:11), .funs = funs(ifelse(is.na(.), median(., na.rm = TRUE), .)))
print("The number of observations in algae.med is")
```

```
## [1] "The number of observations in algae.med is"
```

```
nrow(algae.med)
```

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

```
print("The chemicals for the 48th, 62nd, and 199th rows are displayed in the table below")
```

```
## [1] "The chemicals for the 48th, 62nd, and 199th rows are displayed in the table below"
```

```
Row <- c(48, 62, 199)
cbind(Row, rbind(algae.med[48,4:11], algae.med[62,4:11], algae.med[199,4:11]))
```

```
##   Row mxPH mnO2    Cl   NO3      NH4  oPO4      PO4  Chla
## 1  48 8.06 12.6  9.00 0.230  10.0000  5.00   6.0000 1.100
## 2  62 6.40  9.8 32.73 2.675 103.1665 40.15  14.0000 5.475
## 3 199 8.00  7.6 32.73 2.675 103.1665 40.15 103.2855 5.475
```

3.d

```
require(utils)
#pairs(algae[4:11])
x <- algae.del[4:11]
x.cor <- cor(x)
x.cor
```

```
##             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
```

```
reg <- lm(algae$PO4~algae$oPO4)
algae$PO4[28] <- predict(reg)[28]
algae$PO4[28]
```

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

3.e

Imputation might cause us to have incorrect conclusions becuase of relying too heavily on the observed data only. Lets say that in a scenario of the algae data we have new data that is far from the prediction based on oPO4 or far from the medians of each chemical, then we will have very high test error, and a model that is too overfitted on the training data.

In the context of Correlation Method: In Lecture 2 we learned about Survivorship Bias. Many datesets have Survivorship Bias where the data that we have is insufficient in telling us about a certain variable. In the context of the algae data, using oPO4 to impute values of PO4 might be inducing Survivorship bias because oPO4 might actually not be sufficient in predicting PO4. Leading us to have high test error again

In the context of the Median Method: this might lead us to the wrong conclusions by introducing alot of Bias because we are using only the observed data to impute. New data might be very far from the Median causing us to have values that have high test error.

4

4.a

```
set.seed(500)
id<-cut(1:nrow(algae.med), 5, label=FALSE) %>% sample()
almed1 <- cbind(id, algae.med)
```

4.b

```
  error <- data.frame("fold"=NULL, "train.error"=NULL, "val.error"=NULL)
  dat <- almed1
  for(i in 1:5){
    train=(dat$id != i)
    Xtr = dat[train,1:11] # get training set
    Ytr = dat[train,12] # get true response values in trainig set
    Xvl = dat[!train,1:12] # get validation set
    Yvl = dat[!train,12] # get true response values in validation set
    lm.a1 <- lm(a1~., data = dat[train,1:13])
    predYtr = predict(lm.a1) # predict training values
    predYvl = predict(lm.a1,Xvl) # predict validation values
    error_d<-list(i,mean((predYtr - Ytr)^2), # compute and store trainin
                  mean((predYvl - Yvl)^2)) # compute and store test er
    error <- rbind(error, error_d)
  }

#}
colnames(error) <- c("Fold", "Training Error", "Test Error")
error
```

```
##   Fold Training Error Test Error
## 1    1       856.6085   583.4071
## 2    2       810.8734  1247.1013
## 3    3       763.1246  1007.4681
## 4    4       889.1585   691.7369
## 5    5       781.9331   957.9058
```

We didn't use the given function instead we used the code for a forloop We had to adjust the parameters for lm.a1 and Xvl so we could use the given function.

5

```
alTest <- 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()
## )
```

5.a

```
tSet = algae.med[,1:11]
vSet = alTest[,1:11]
vSet2 = alTest[,12]

lmAll <- lm(a1~., data = algae.med[,1:12])
predvSet = predict(lmAll, vSet)
sum((predvSet-vSet2)^2)/length(predvSet)
```

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

```
#MSE

# The mean square error is much lower than the test errors in part 4.But this is probably due to our mo
```

6

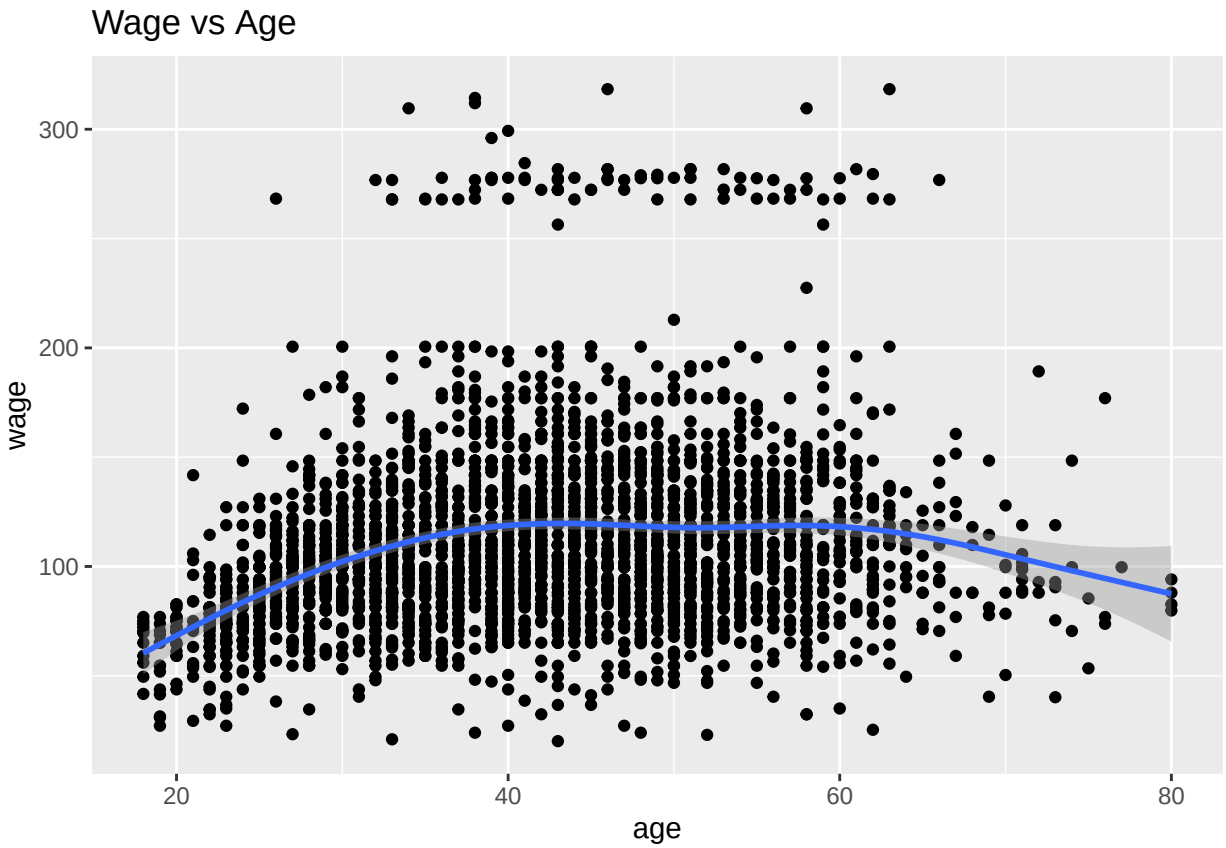
```
library(ISLR)
head(Wage)
```

```
##        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
```

6.a

```
ExpSalary = ggplot(Wage, aes(x=age, y=wage))
   ExpSalary + geom_point() + geom_smooth()+ggtitle("Wage vs Age")
```

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



Apart from a few outliers, it seems that wages rise with age up till a peak point around 40-50 years of age and then slowly decrease again with increase in age. It matches our expectations.

6.b

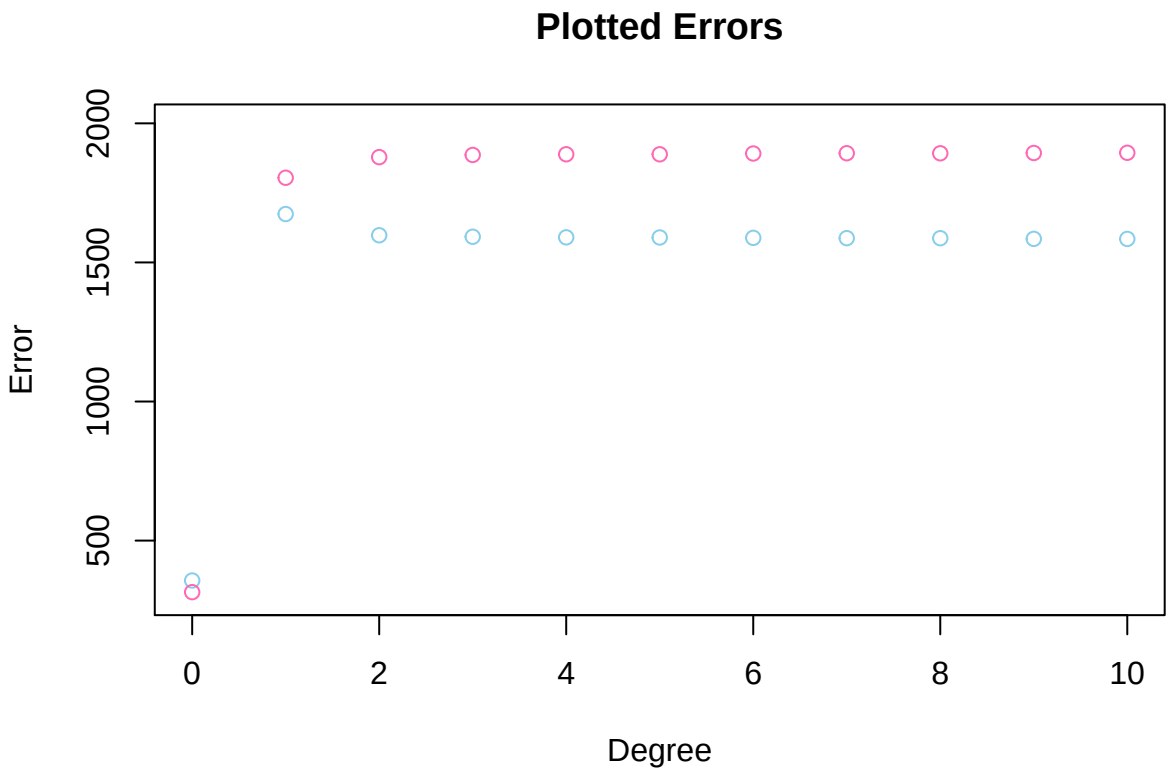
```
modelErrors <- data.frame("Model"=NULL, "Train Error"=NULL, "Test Error"=NULL)
nums <- rep(1:5, each = length(Wage)/5)
id <- sample(nums)
age <- Wage$age
wage <- Wage$wage
data <- data.frame("ID" = id, "AGE" = age, "WAGE" = wage)
for (i in 1:5){
  sumtrain = 0
  sumtest = 0
    inTrain = data[data[,1]!=i,2]
    outTrain = data[data[,1]!=i,3]
    inTest = data[!(data[,1]!=i),2:3]
    outTest = data[!(data[1]!=i),3]
    fit <- lm(WAGE~1, data[data[,1]!=i,2:3])
    length(data)
    pTrain = predict(fit)
```

```
    pTest = predict(fit, inTest)
    sumtrain <- sumtrain + mean((pTrain - outTrain)^2)
    sumtest <- sumtest + mean((pTest - outTest)^2)
}
modelErrors <- rbind(modelErrors, list(0, sumtrain/5, sumtest/5))
for(j in 1:10){
  sumtrain = 0
  sumtest = 0
  for (i in 1:5){
    inTrain = data[data[,1]!=i,2]
    outTrain = data[data[,1]!=i,3]
    inTest = data.frame(data[data[,1]!=i, 2])
    outTest = data[!(data[1]!=i),3]
    fit <- lm(data[data[,1]!=i, 3]~poly(data[data[,1]!=i, 2], j, raw = FALSE), data = data)
    pTrain = predict(fit)
    pTest = predict(fit, inTest)
    sumtrain <- sumtrain + mean((pTrain - outTrain)^2)
    sumtest <- sumtest + mean((pTest - outTest)^2)
  }
  modelErrors <- rbind(modelErrors, list(j, sumtrain/5, sumtest/5))
}
colnames(modelErrors) <- list("Degree", "Train Error", "Test Error")
modelErrors
```

```
##    Degree Train Error Test Error
## 1       0    356.5212   314.7029
## 2       1   1673.9497  1804.4275
## 3       2   1597.6477  1878.6136
## 4       3   1592.3507  1886.4644
## 5       4   1590.2267  1888.8229
## 6       5   1589.7791  1888.7296
## 7       6   1588.3729  1891.6231
## 8       7   1587.3635  1892.5502
## 9       8   1587.1468  1892.4138
## 10      9   1584.7455  1893.4671
## 11     10   1584.5093  1894.2452
```

##6.c

```
plot(modelErrors$Degree, modelErrors$`Train Error`, col = 'skyblue', ylab = "Error",
     xlab = "Degree", ylim = c(300, 2000), main = "Plotted Errors")
points(modelErrors$Degree, modelErrors$`Test Error`, col = 'hotpink')
```



It looks like they are all equally bad except for the intercept-only mode.