PSTAT 131 Homework

Justin Lau

10/17/2020

Libraries

```
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(readr)
library(ggplot2)
library(plyr)
```

```
## ------------------------------------------------------------------------------
```

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

```
library(reshape2)
library(class)
library(boot)
```

Input 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")
```

```
##
## ── 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()
## )
```

```
attach(algae)
```

Question 1 a

```
group <- group_by(algae,season)
summarise(group, length(season))
```

```
##   length(season)
## 1            200
```

1b

```
chemicals <- list(mxPH, mnO2, Cl, NO3, Chla)
sapply(algae[4:11], mean, na.rm = TRUE)
```

```
##       mxPH       mnO2         Cl        NO3        NH4       oPO4        PO4
##   8.011734   9.117778  43.636279   3.282389 501.295828  73.590596 137.882101
##       Chla
##  13.971197
```

```
sapply(algae[4:11], var, na.rm = TRUE)
```

```
##         mxPH         mnO2           Cl          NO3          NH4         oPO4
## 3.579693e-01 5.718089e+00 2.193172e+03 1.426176e+01 3.851585e+06 8.305850e+03
##          PO4         Chla
## 1.663938e+04 4.200827e+02
```

1c

```
sapply(algae[4:11], median, na.rm = TRUE)
```

```
##     mxPH     mnO2       Cl      NO3      NH4     oPO4      PO4     Chla
##   8.0600   9.8000  32.7300   2.6750 103.1665  40.1500 103.2855   5.4750
```

```
sapply(algae[4:11], mad, na.rm = TRUE)
```

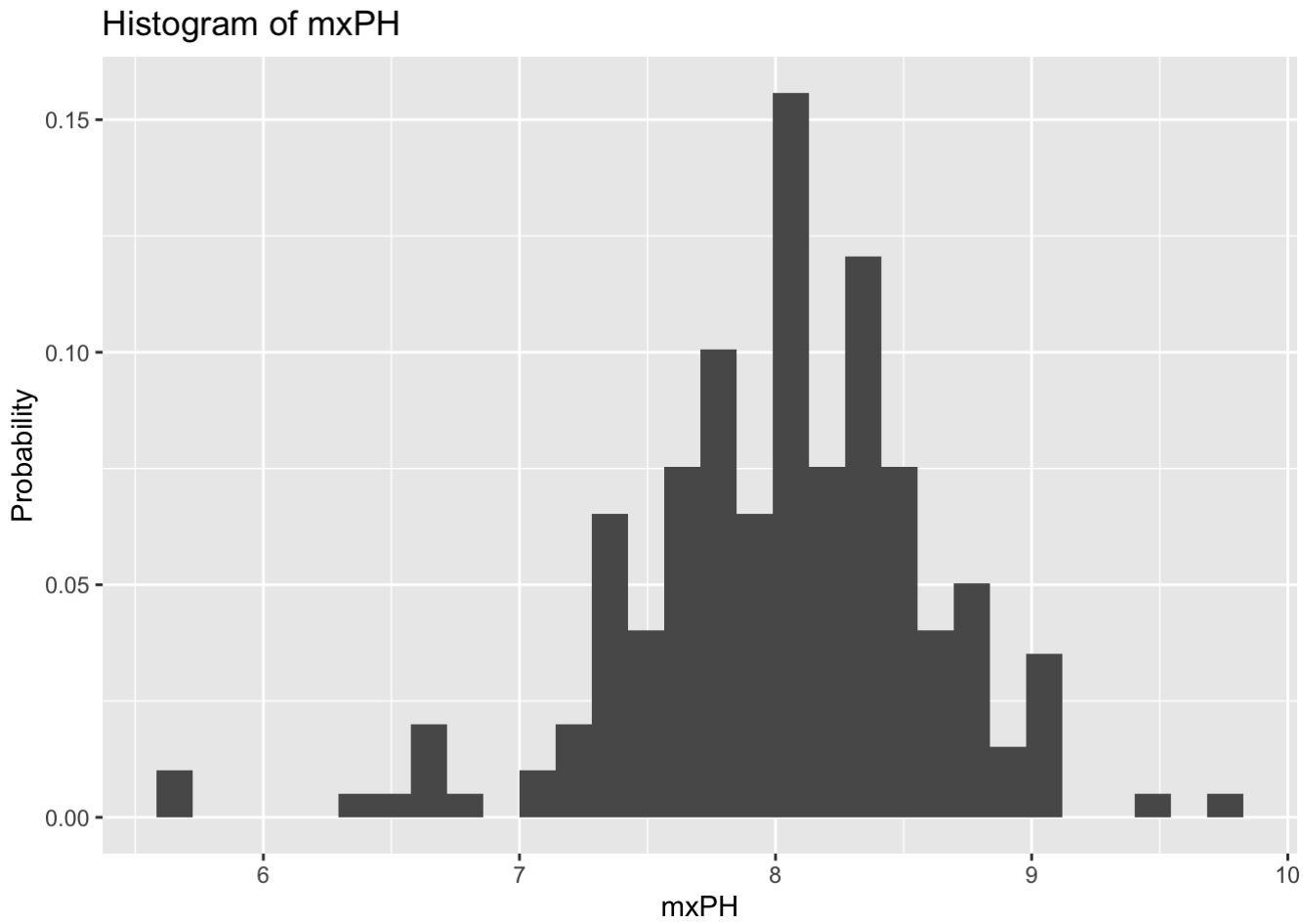
```
##       mxPH       mnO2         Cl        NO3        NH4       oPO4        PO4
##   0.504084   2.053401  33.249529   2.172009 111.617548  44.045822 122.321172
##       Chla
##   6.671700
```

Question 2a

```
ggplot(algae, aes(x=mxPH, y = (..count..)/sum(..count..))) + labs(title = 'Histogram of
 mxPH', y = 'Probability') + geom_histogram()
```

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

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



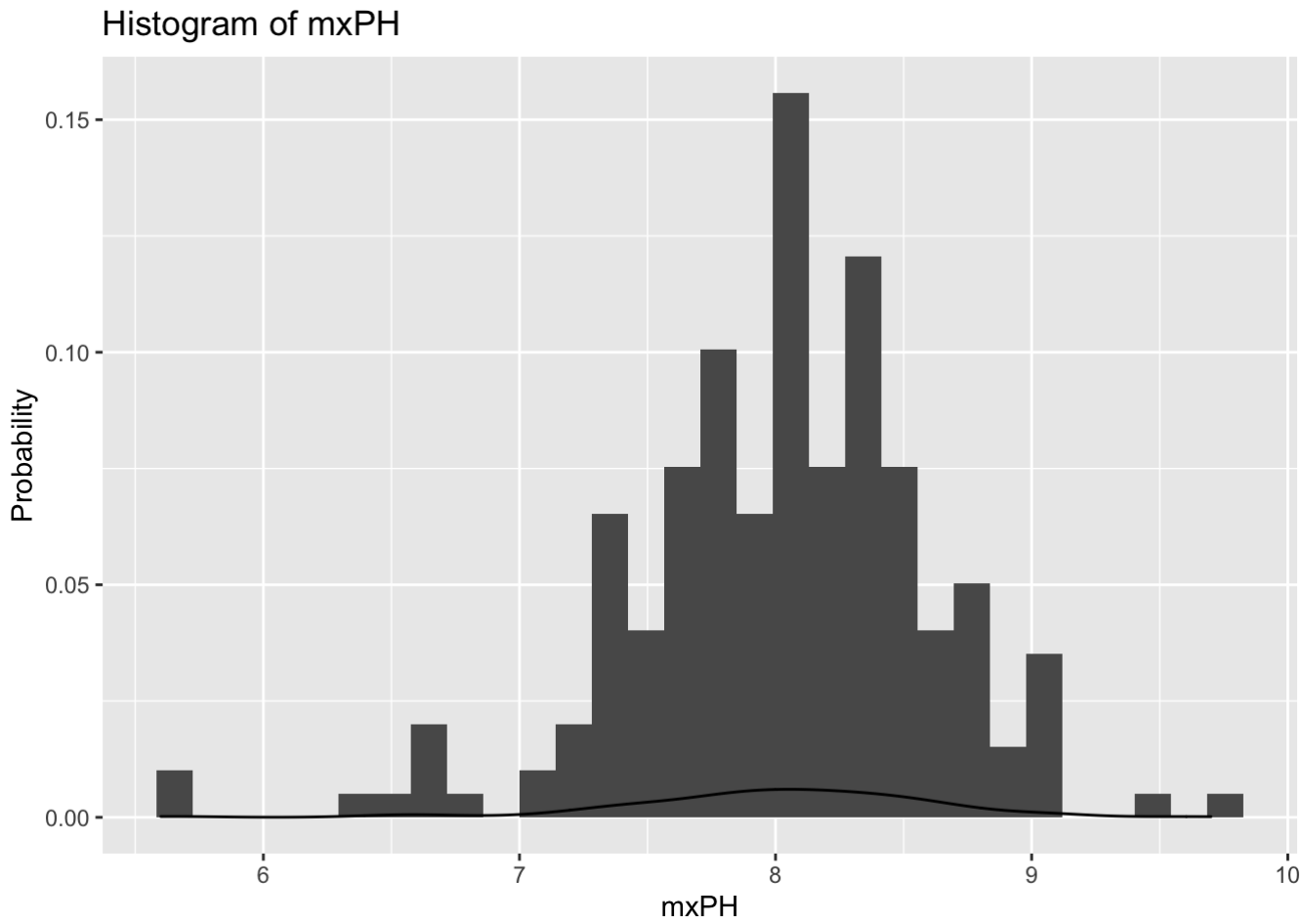
2b

```
ggplot(algae, aes(x=mxPH, y = (..count..)/sum(..count..))) + labs(title = 'Histogram of
 mxPH', y = 'Probability') + geom_histogram() + geom_density()
```

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

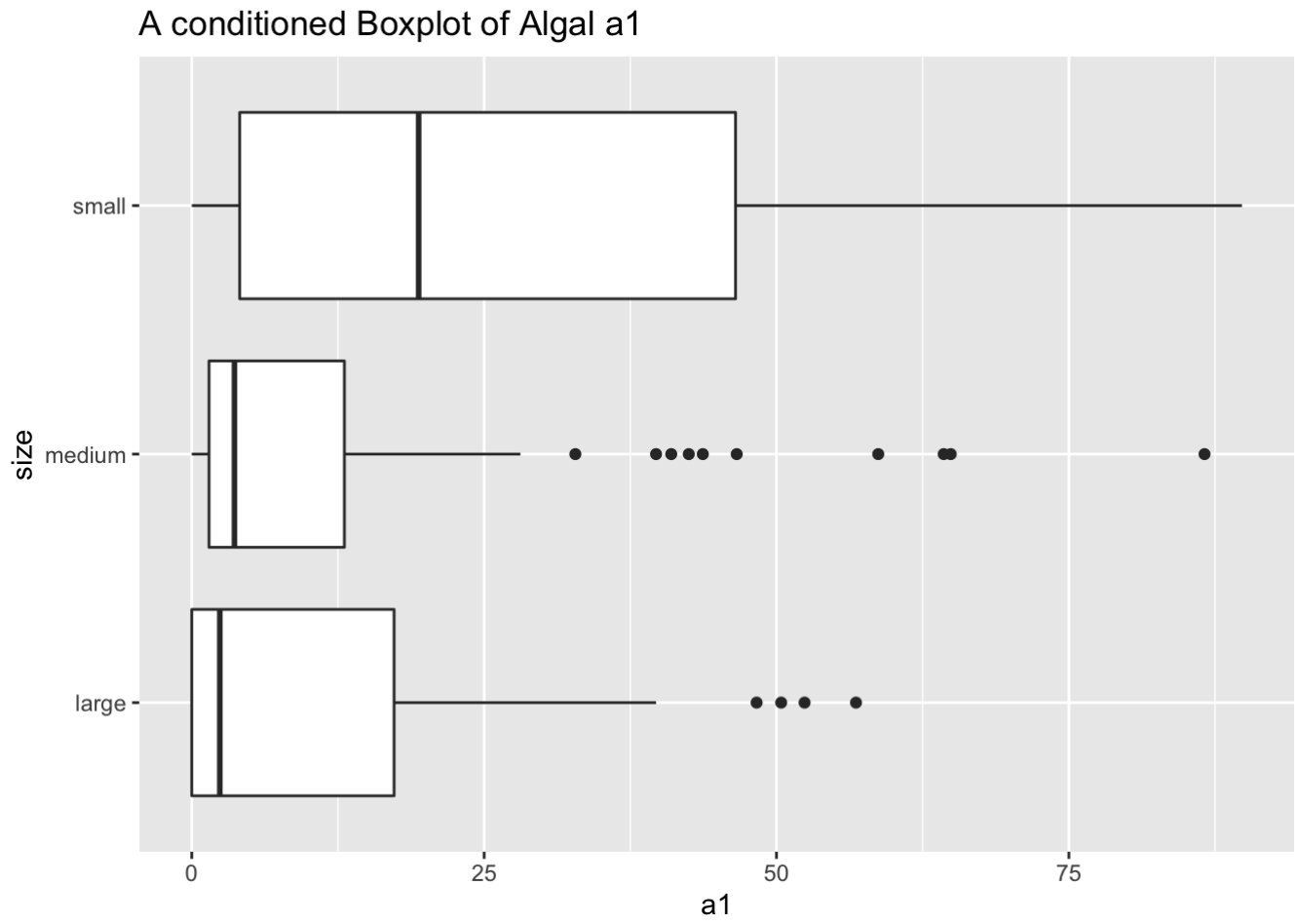
```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```

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



2c

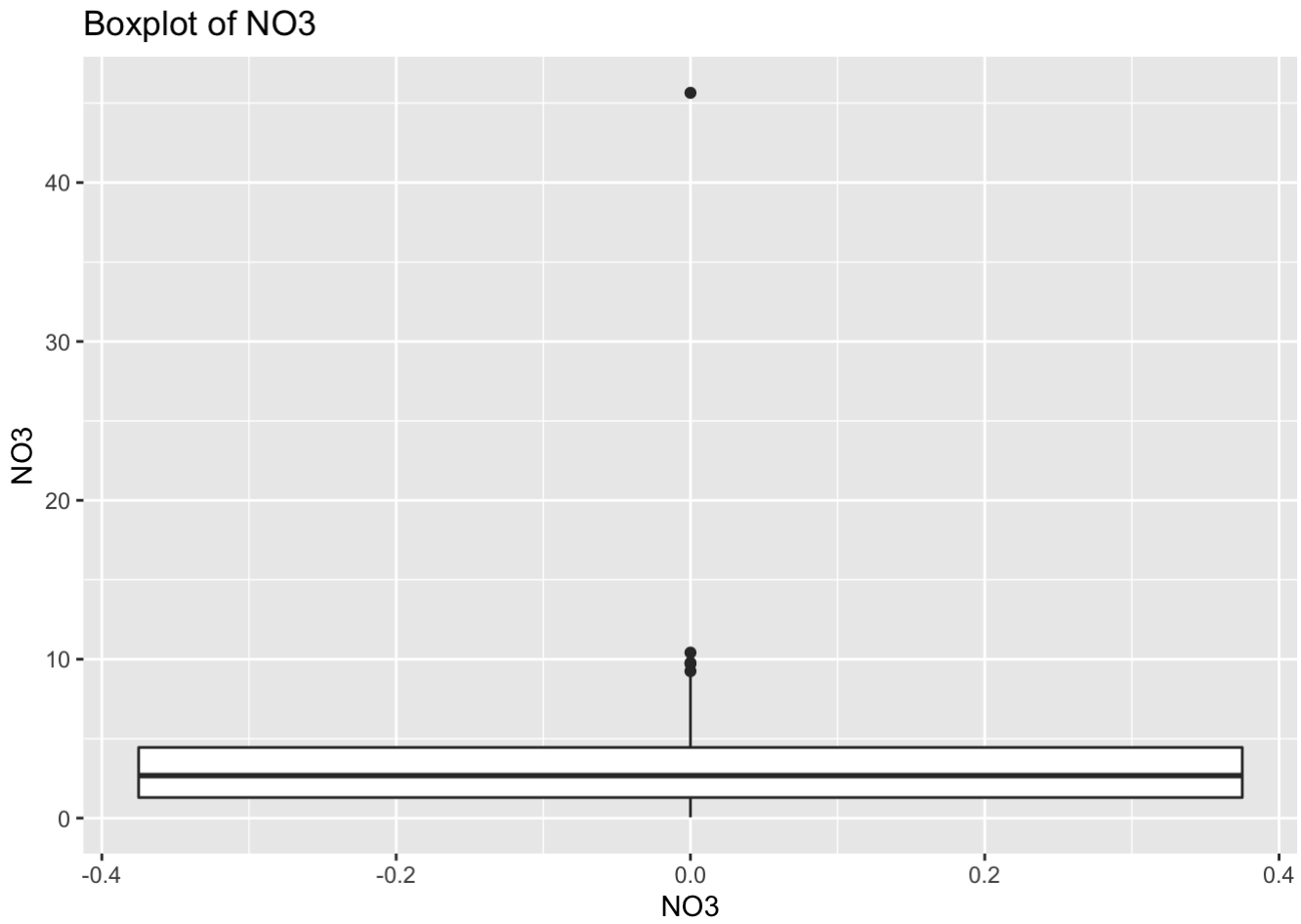
```
ggplot(algae, aes(x=a1, y=size)) + geom_boxplot() + labs(title = 'A conditioned Boxplot
 of Algal a1')
```



2d

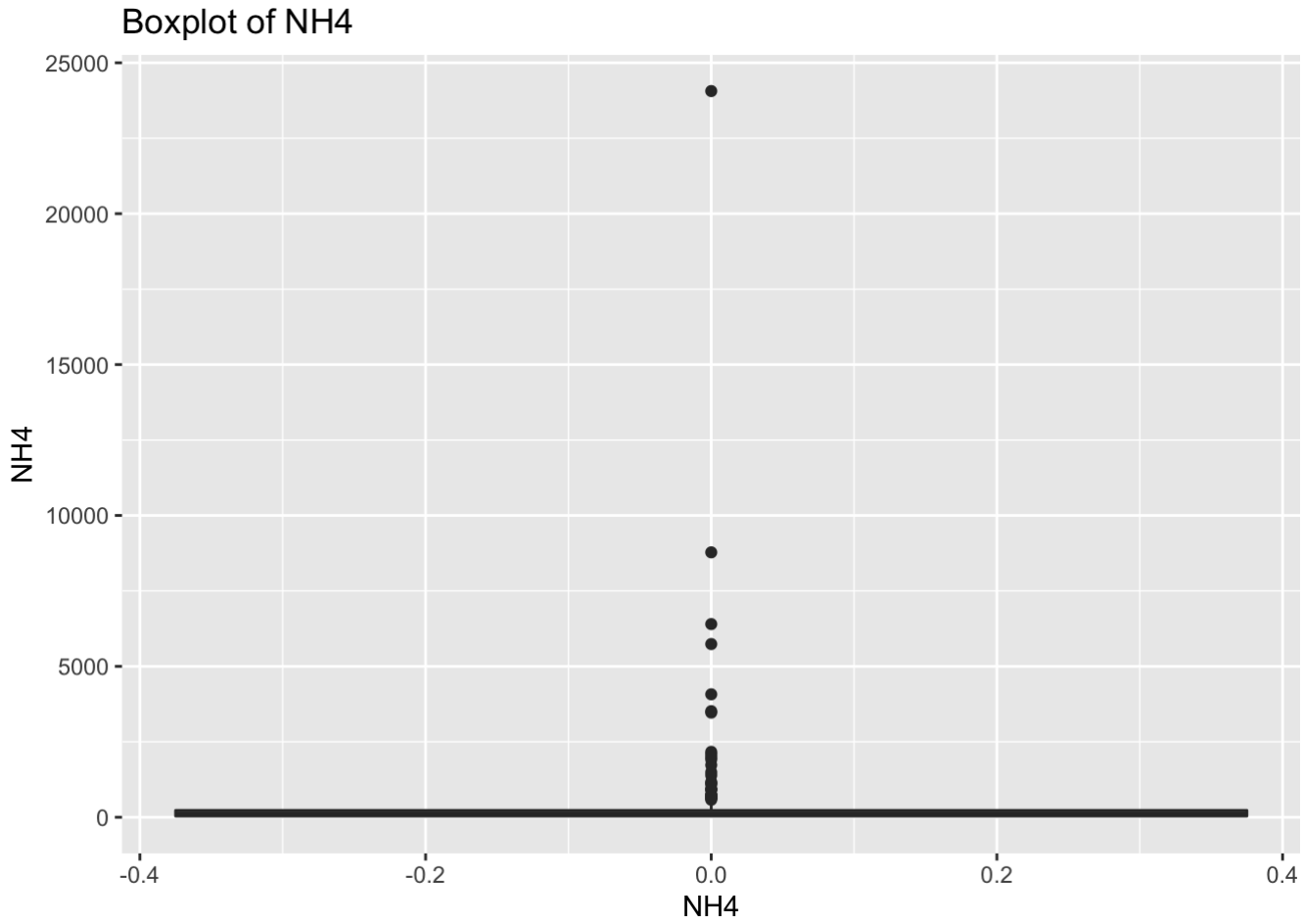
```
ggplot(algae, aes(y=NO3)) + labs(x='NO3',title='Boxplot of NO3') + geom_boxplot()
```

```
## Warning: Removed 2 rows containing non-finite values (stat_boxplot).
```




```
ggplot(algae, aes(y=NH4)) + labs(x='NH4',title='Boxplot of NH4') + geom_boxplot()
```

```
## Warning: Removed 2 rows containing non-finite values (stat_boxplot).
```




```
lowerq = quantile(algae$NO3, na.rm =TRUE)[2]
upperq = quantile(algae$NO3, na.rm =TRUE)[4]
iqr = upperq - lowerq
upper.threshold.NO3 = (iqr * 1.5) + upperq
lower.threshold.NO3= lowerq - (iqr * 1.5)
count(algae$NO3 > upper.threshold.NO3)
```

```
##       x freq
## 1 FALSE  193
## 2  TRUE    5
## 3    NA    2
```

```
count(algae$NO3 < lower.threshold.NO3)
```

```
##       x freq
## 1 FALSE  198
## 2    NA    2
```

```
lowerq = quantile(algae$NH4, na.rm =TRUE)[2]
upperq = quantile(algae$NH4, na.rm =TRUE)[4]
iqr = upperq - lowerq
upper.threshold.NH4 = (iqr * 1.5) + upperq
lower.threshold.NH4= lowerq - (iqr * 1.5)
count(algae$NH4 > upper.threshold.NH4)
```

```
##       x freq
## 1 FALSE  171
## 2  TRUE   27
## 3    NA    2
```

```
count(algae$NH4 < lower.threshold.NH4)
```

```
##       x freq
## 1 FALSE  198
## 2    NA    2
```

There are 5 outliers for N03 and 27 outliers for NH4. This is calculated using the IQR of the data and setting upper and lower thresholds of 1.5 to test for data points outside of the range.

2e

```
cat('The mean of N03 =', mean(algae$NO3, na.rm = TRUE), '\n', 'The variance of NO3 =', v
ar(algae$NO3, na.rm = TRUE), '\n')
```

```
## The mean of N03 = 3.282389
##  The variance of NO3 = 14.26176
```

```
cat('The median of N03 =', median(algae$NO3, na.rm = TRUE), '\n', 'The MAD of NO3 =', ma
d(algae$NO3, na.rm = TRUE), '\n')
```

```
## The median of N03 = 2.675
##  The MAD of NO3 = 2.172009
```

```
cat('The mean of NH4 =', mean(algae$NH4, na.rm = TRUE), '\n', 'The variance of NH4 =', v
ar(algae$NH4, na.rm = TRUE), '\n')
```

```
## The mean of NH4 = 501.2958
##  The variance of NH4 = 3851585
```

```
cat('The median of NH4 =', median(algae$NH4, na.rm = TRUE), '\n', 'The MAD of NH4 =', ma
d(algae$NH4, na.rm = TRUE), '\n')
```

```
## The median of NH4 = 103.1665
##  The MAD of NH4 = 111.6175
```

It appears that median and Mad tend to hold up more to outliers, this is caused because using the mean it is suceptible to skewing the data when there are extremem outliers in the data.

Question 3a

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

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

```
summary(algae)
```

```
##     season              size              speed                mxPH
##  Length:200         Length:200         Length:200         Min.   :5.600
##  Class :character   Class :character   Class :character   1st Qu.:7.700
##  Mode  :character   Mode  :character   Mode  :character   Median :8.060
##                                                           Mean   :8.012
##                                                           3rd Qu.:8.400
##                                                           Max.   :9.700
##                                                           NA's   :1
##       mnO2               Cl               NO3              NH4
##  Min.   : 1.500   Min.   :  0.222   Min.   : 0.050   Min.   :    5.00
##  1st Qu.: 7.725   1st Qu.: 10.981   1st Qu.: 1.296   1st Qu.:   38.33
##  Median : 9.800   Median : 32.730   Median : 2.675   Median :  103.17
##  Mean   : 9.118   Mean   : 43.636   Mean   : 3.282   Mean   :  501.30
##  3rd Qu.:10.800   3rd Qu.: 57.824   3rd Qu.: 4.446   3rd Qu.:  226.95
##  Max.   :13.400   Max.   :391.500   Max.   :45.650   Max.   :24064.00
##  NA's   :2        NA's   :10        NA's   :2        NA's   :2
##       oPO4             PO4              Chla               a1
##  Min.   :  1.00   Min.   :  1.00   Min.   :  0.200   Min.   : 0.00
##  1st Qu.: 15.70   1st Qu.: 41.38   1st Qu.:  2.000   1st Qu.: 1.50
##  Median : 40.15   Median :103.29   Median :  5.475   Median : 6.95
##  Mean   : 73.59   Mean   :137.88   Mean   : 13.971   Mean   :16.92
##  3rd Qu.: 99.33   3rd Qu.:213.75   3rd Qu.: 18.308   3rd Qu.:24.80
##  Max.   :564.60   Max.   :771.60   Max.   :110.456   Max.   :89.80
##  NA's   :2        NA's   :2        NA's   :12
##        a2               a3               a4               a5
##  Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.000
##  Median : 3.000   Median : 1.550   Median : 0.000   Median : 1.900
##  Mean   : 7.458   Mean   : 4.309   Mean   : 1.992   Mean   : 5.064
##  3rd Qu.:11.375   3rd Qu.: 4.925   3rd Qu.: 2.400   3rd Qu.: 7.500
##  Max.   :72.600   Max.   :42.800   Max.   :44.600   Max.   :44.400
##
##        a6               a7
##  Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 0.000   1st Qu.: 0.000
##  Median : 0.000   Median : 1.000
##  Mean   : 5.964   Mean   : 2.495
##  3rd Qu.: 6.925   3rd Qu.: 2.400
##  Max.   :77.600   Max.   :31.600
##
```

3b

```
algae.del <- algae %>% filter(complete.cases(algae))
print('There are 184 complete observations')
```

```
## [1] "There are 184 complete observations"
```

3c

```
algae.med = algae %>% mutate_at(vars(4:11), funs(ifelse(is.na(.), median(., na.rm=TRUE), .)))
```

```
## 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.
```

```
algae.med[48,]
```

```
## # A tibble: 1 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
## # … with 5 more variables: a3 <dbl>, a4 <dbl>, a5 <dbl>, a6 <dbl>, a7 <dbl>
```

```
algae.med[62,]
```

```
## # A tibble: 1 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 summer small medi…   6.4   9.8  32.7  2.68  103.  40.2    14  5.48  19.4     0
## # … with 5 more variables: a3 <dbl>, a4 <dbl>, a5 <dbl>, a6 <dbl>, a7 <dbl>
```

```
algae.med[199,]
```

```
## # A tibble: 1 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 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

```
for(i in 4:11){
  print(paste0(colnames(algae)[i]))
  print(cor(algae[,i],algae[,(i+1):11],use = 'pairwise.complete.obs'))
}
```

```
## [1] "mxPH"
##            mnO2         Cl         NO3          NH4      oPO4        PO4      Chla
## mxPH -0.1686123 0.1361078 -0.1309805 -0.09353577 0.1589994 0.1899081 0.4459618
## [1] "mnO2"
##              Cl         NO3          NH4       oPO4         PO4        Chla
## mnO2 -0.2783325 0.09944373 -0.08747825 -0.4161629 -0.4874862 -0.1532648
## [1] "Cl"
##          NO3        NH4      oPO4      PO4      Chla
## Cl 0.2250409 0.07191298 0.3910535 0.457449 0.1498565
## [1] "NO3"
##           NH4      oPO4      PO4      Chla
## NO3 0.7214435 0.1445878 0.168601 0.1396792
## [1] "NH4"
##          oPO4       PO4       Chla
## NH4 0.2272372 0.2081804 0.08894652
## [1] "oPO4"
##            PO4      Chla
## oPO4 0.9143652 0.1156213
## [1] "PO4"
##          Chla
## PO4 0.2536213
## [1] "Chla"
##              a1 Chla
## Chla -0.2779866    1
```

```
fit <- lm(algae$PO4 ~ algae$oPO4)
PO4_pred <- predict(fit)

PO4_pred[28]
```

```
##       29
## 76.51663
```

3e There can be surviorship bias that is using data from what was there to impute onto data that was missing. Another issue with this is that it reduces the actual variance of the data alongside the standard error. Imputation of the median while sometimes necessary messes with the relationship of variables.

Question 4a

```
nfold = 5
set.seed(66)
folds = cut(1:nrow(algae.med), breaks=nfold, labels=FALSE) %>% sample()
folds
```

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

4b

```
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
}
```

```
error.folds = NULL
allK = 1:50
set.seed(67)
for (j in allK){
    tmp = ldply(1:nfold, do.chunk, chunkdef=folds, dat=algae.med)
    error.folds = rbind(error.folds, tmp)
}
tmp
```

```
##   fold train.error val.error
## 1    1    290.3775  285.3887
## 2    2    240.6154  506.5678
## 3    3    296.3188  256.5233
## 4    4    280.9803  400.1096
## 5    5    299.8153  257.5973
```

Question 5

```
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()
## )
```

```
model <- glm(a1~ season + size + speed + mxPH + mnO2 + Cl + NO3 + NH4 + oPO4 + PO4 + Chl
a, data = algae.Test)
predictY <- predict(model)
Ytr <- algae.Test$a1

test.error = mean((predictY - Ytr)^2)
test.error
```

```
## [1] 218.2218
```

This test error is close to the training error produced in question 4b

Question 6a

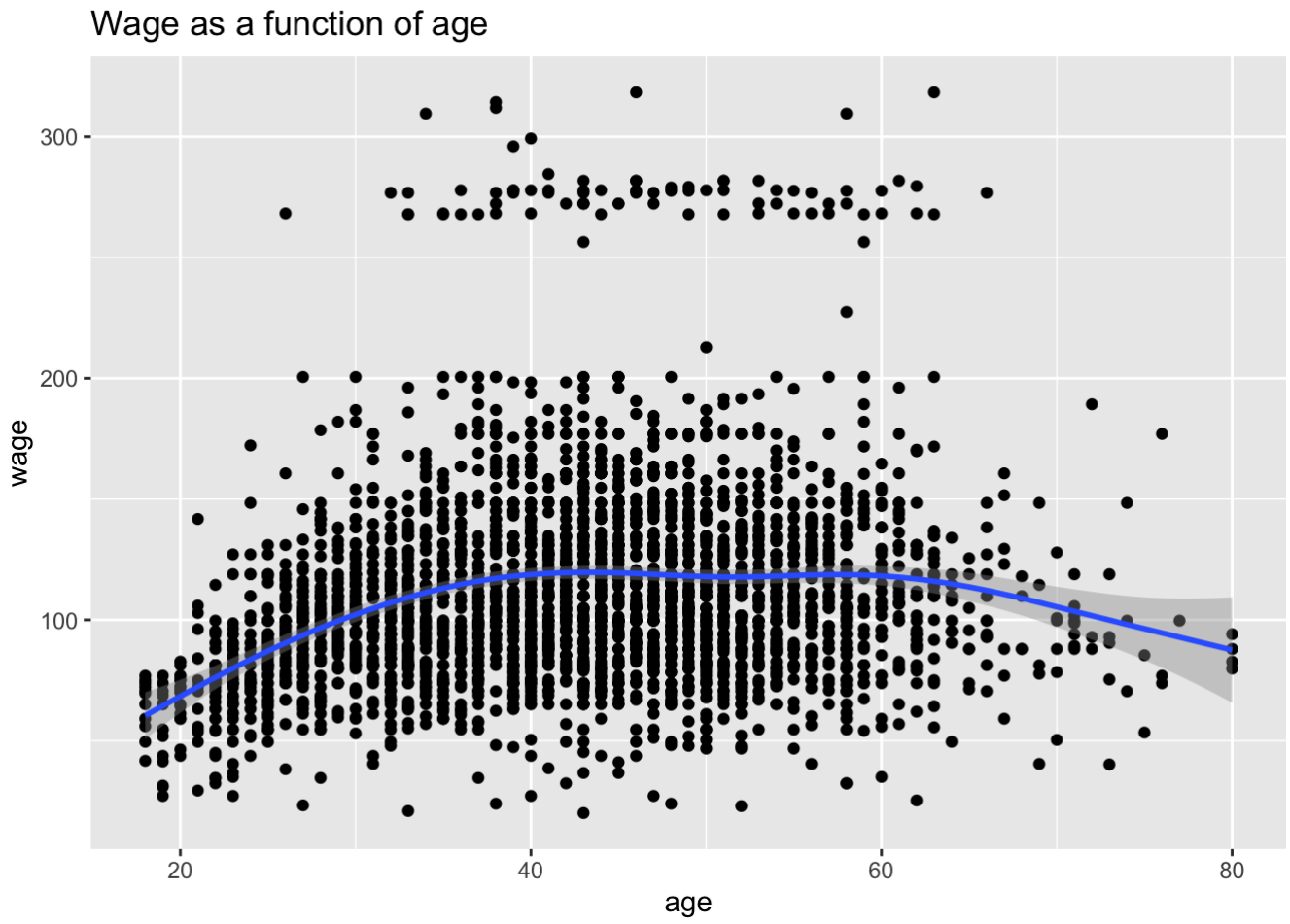
```
library(ISLR)
head(Wage)
```

```
##        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
```

```
data(Wage)
```

```
ggplot(Wage, aes(x=age, y=wage)) + labs(title = 'Wage as a function of age') + geom_poin
t()  + geom_smooth()
```

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



6b

```
Wage <- Wage %>% select(c(age,wage))
model <- glm(wage~poly(age,10), data=Wage)

nfold = 5
folds = cut(1:nrow(Wage), breaks=nfold, labels=FALSE) %>% sample()



do.chunks <- function(chunkid, chunkdef, dat){ # function argument
  train = (chunkdef != chunkid)

  Ytr = dat[train,2] # get true response values in trainig set
  for (p in c(1:11)){
    lm.wage <- lm(wage~poly(age,p), data = dat[train,1:2])
    predYtr = predict(lm.wage) # predict training values
    train.error = mean((predYtr - Ytr)^2)
    nam <- paste("A", p, sep = "")
    assign(nam, data.frame(polynomial.degree = p, train.error = train.error))
  }
  error <- rbind(A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11)
  print(error)
}
```

```
for (p in c(1:11)) {
  model <- lm(wage~poly(age,p), data = Wage)
  predictY <- predict(model)
  Ytr <- Wage$wage
  test.error = mean((Ytr - predictY)^2)
  nam <- paste("A", p, sep = "")
  assign(nam, data.frame(polynomial.degree = p, test.error = test.error))
}
test.error <- rbind(A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11)
train.error <- do.chunks(4, folds, Wage)
```

```
##    polynomial.degree train.error
## 1                  1    1732.165
## 2                  2    1652.475
## 3                  3    1644.965
## 4                  4    1643.936
## 5                  5    1642.915
## 6                  6    1642.208
## 7                  7    1641.433
## 8                  8    1641.289
## 9                  9    1639.645
## 10                10    1639.643
## 11                11    1639.529
```

```
both.error <- merge(train.error, test.error, by.y = 'polynomial.degree' )
print(both.error)
```

```
##    polynomial.degree train.error test.error
## 1                  1    1732.165   1674.072
## 2                  2    1652.475   1597.810
## 3                  3    1644.965   1592.558
## 4                  4    1643.936   1590.535
## 5                  5    1642.915   1590.107
## 6                  6    1642.208   1588.796
## 7                  7    1641.433   1587.945
## 8                  8    1641.289   1587.902
## 9                  9    1639.645   1585.568
## 10                10    1639.643   1585.567
## 11                11    1639.529   1585.532
```

6c

```
ggplot(both.error, aes(x=polynomial.degree))  + geom_line(aes(y=train.error, color='Trai
ning Error')) + geom_line(aes(y=test.error, color='Testing Error')) + labs(title = 'Trai
ning and Testing Error of wages as a polynomial function of age', y = 'MSE', x='Degree o
f Polynomial', color = 'Type of Error')
```