PSTAT 131 Homework 1

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.1     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

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

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

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

1a

```
#1 a
algae %>%
  dplyr::count(season)
```

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

1b

```
#1 b
algae[rowSums(is.na(algae))==1,]
```

```
## # A tibble: 7 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 autumn small  high   6.8   11.1  9     0.63     20   4     NA      2.7  30.3
## 2 spring small  high   8     NA    1.45  0.81     10   2.5    3      0.3  75.8
## 3 winter small  low   NA     12.6  9     0.23     10   5      6      1.1  35.5
## 4 autumn small  high   7.83  11.7  4.08  1.33     18   3.33   6.67  NA    14.4
## 5 winter medium high   9.7   10.8  0.222 0.406    10  22.4   10.1   NA    41
## 6 spring large  low    9      5.8 NA     0.9     142 102    186     68.0   1.7
## 7 winter large  high   8     10.9  9.06  0.825    40  21.1   56.1   NA    16.8
## # ... with 6 more variables: a2 <dbl>, a3 <dbl>, a4 <dbl>, a5 <dbl>, a6 <dbl>,
## #   a7 <dbl>
```

```
# Yes, there are some missing values.
data1 = algae%>%
  summarise(dplyr::across(c(mxPH, mnO2, Cl, NO3, NH4, oPO4, PO4, Chla), list(mean = mean, variance = va
data1
```

```
## # A tibble: 1 x 16
##   mxPH_mean mxPH_variance mnO2_mean mnO2_variance Cl_mean Cl_variance NO3_mean
##       <dbl>         <dbl>     <dbl>         <dbl>   <dbl>       <dbl>    <dbl>
## 1      8.01         0.358      9.12          5.72    43.6       2193.     3.28
## # ... with 9 more variables: NO3_variance <dbl>, NH4_mean <dbl>,
## #   NH4_variance <dbl>, oPO4_mean <dbl>, oPO4_variance <dbl>, PO4_mean <dbl>,
## #   PO4_variance <dbl>, Chla_mean <dbl>, Chla_variance <dbl>
```

```
# All chemcials except for the maximum pH value and minimum O2 value tend to have a high variance.
```

1c

```
# 1c
data2 = algae%>%
  summarise(across(c(mxPH, mnO2, Cl, NO3, NH4, oPO4, PO4, Chla), list(median = median, MAD = mad), na.r
data2
```

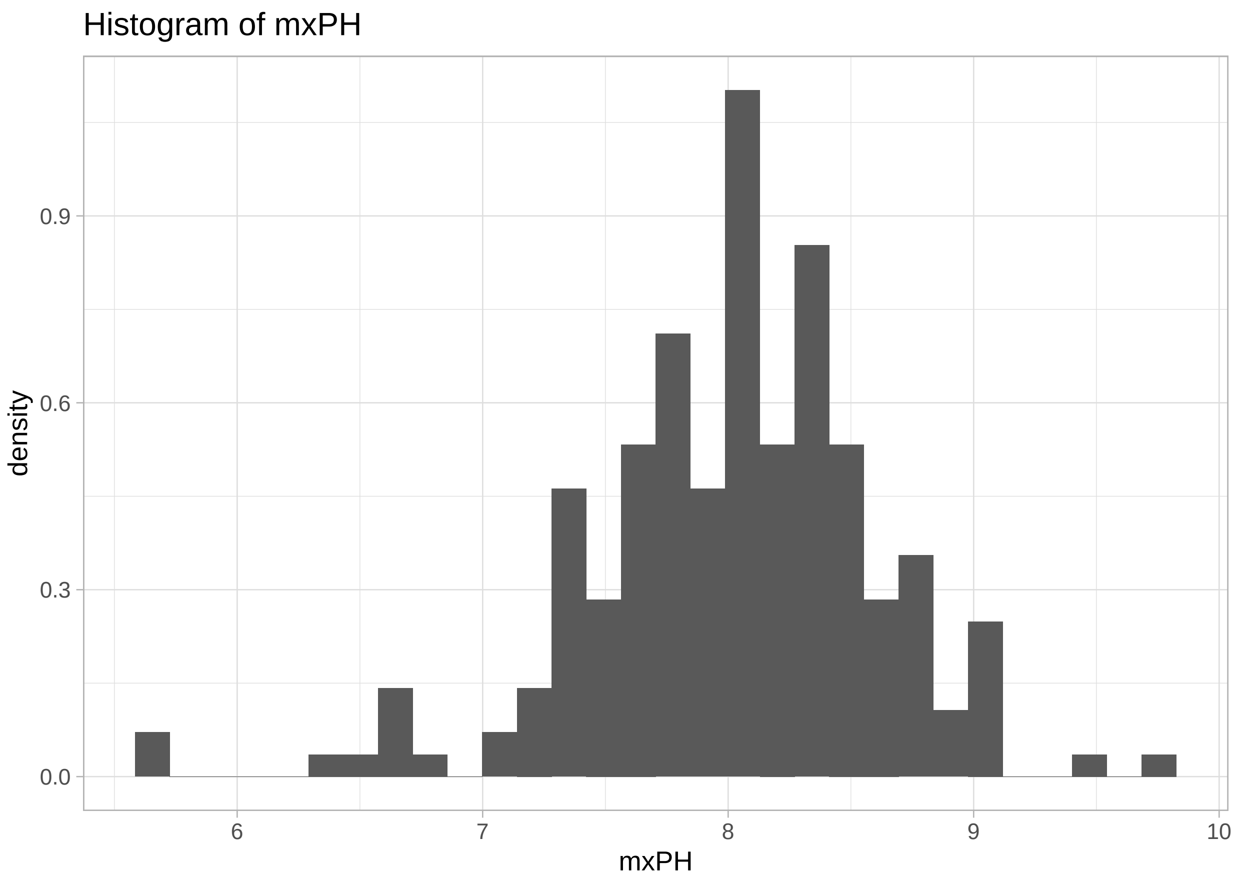
```
## # A tibble: 1 x 16
##   mxPH_median mxPH_MAD mnO2_median mnO2_MAD Cl_median Cl_MAD NO3_median NO3_MAD
##         <dbl>    <dbl>       <dbl>    <dbl>     <dbl>  <dbl>      <dbl>   <dbl>
## 1        8.06     0.34         9.8     1.38      32.7   22.4       2.68    1.46
## # ... with 8 more variables: NH4_median <dbl>, NH4_MAD <dbl>,
## #   oPO4_median <dbl>, oPO4_MAD <dbl>, PO4_median <dbl>, PO4_MAD <dbl>,
## #   Chla_median <dbl>, Chla_MAD <dbl>
```

```
# the numbers in median & MAD are smaller, look more reasonable. It seems outliers have smaller impact
```

2a

```
# 2a
library(ggplot2)
ggplot(data = algae)+
  geom_histogram(aes(x = mxPH, y = ..density..),na.rm = TRUE)+
  ggtitle("Histogram of mxPH")+
  theme_light()
```

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

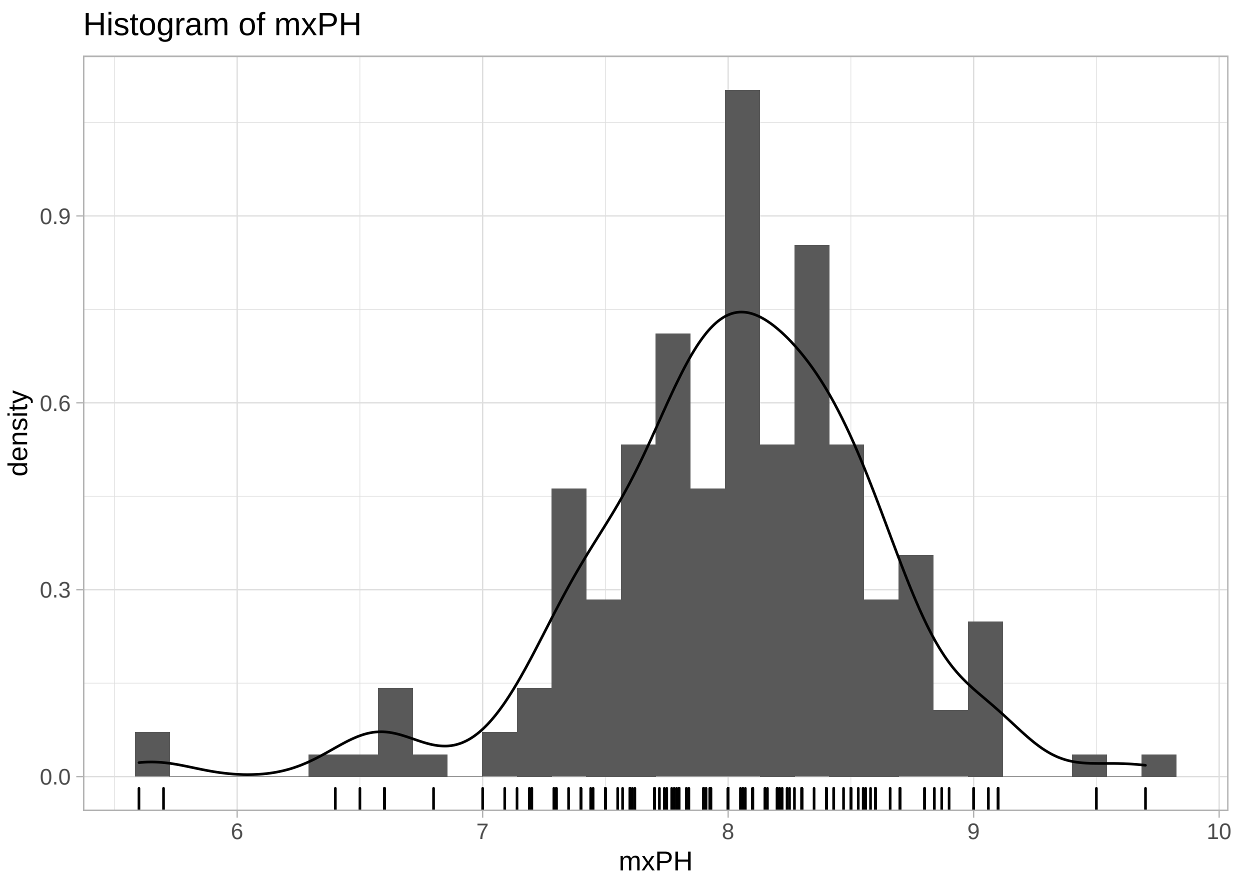


```
# the distribution is not skewed
```

2b

```
# 2b
ggplot(data = algae)+
  geom_histogram(aes(x = mxPH, y = ..density..),na.rm = TRUE)+
  geom_density(aes(x = mxPH),na.rm = TRUE) +
  geom_rug(aes(x = mxPH))+
  ggtitle("Histogram of mxPH")+
  theme_light()
```

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

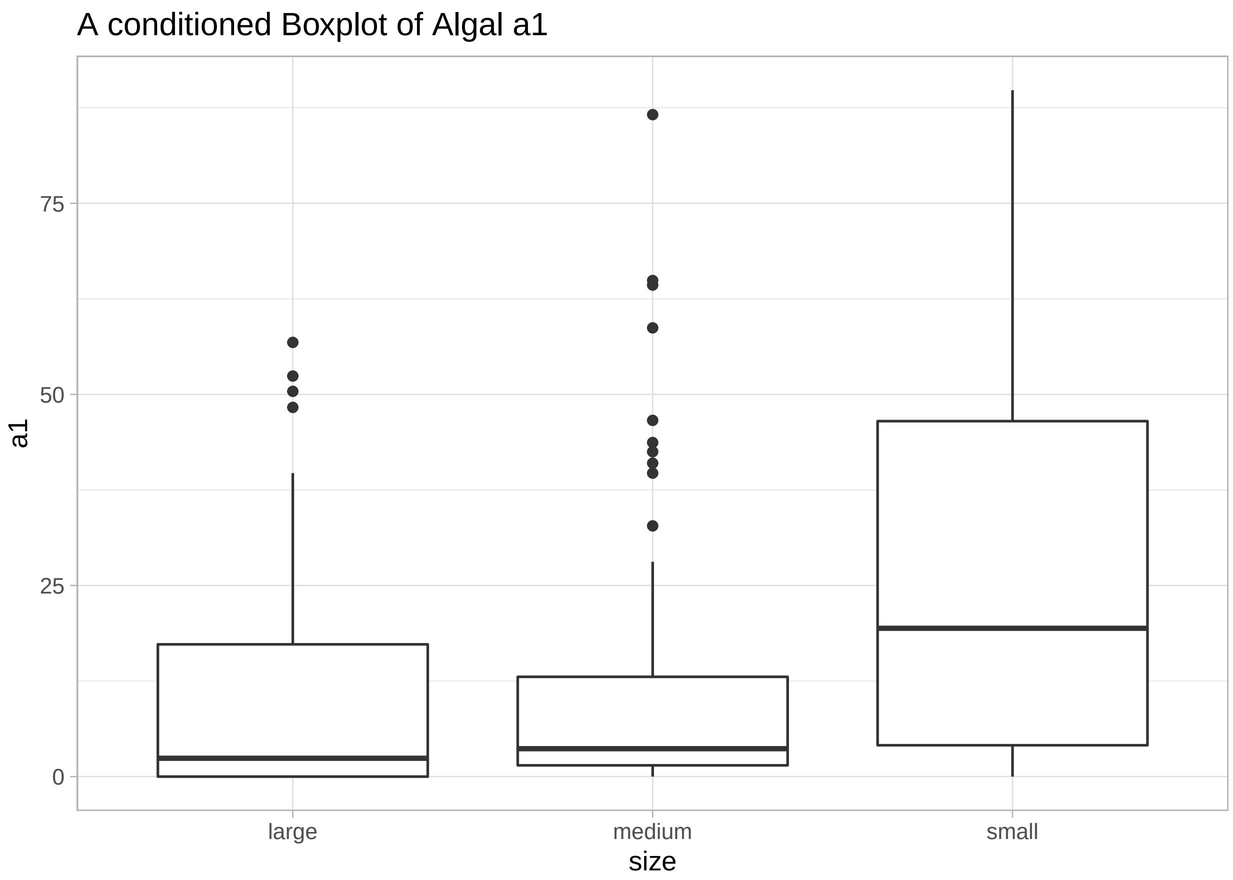


2c

```
a_one <- algae %>% group_by(size) %>% summarise(a1)
```

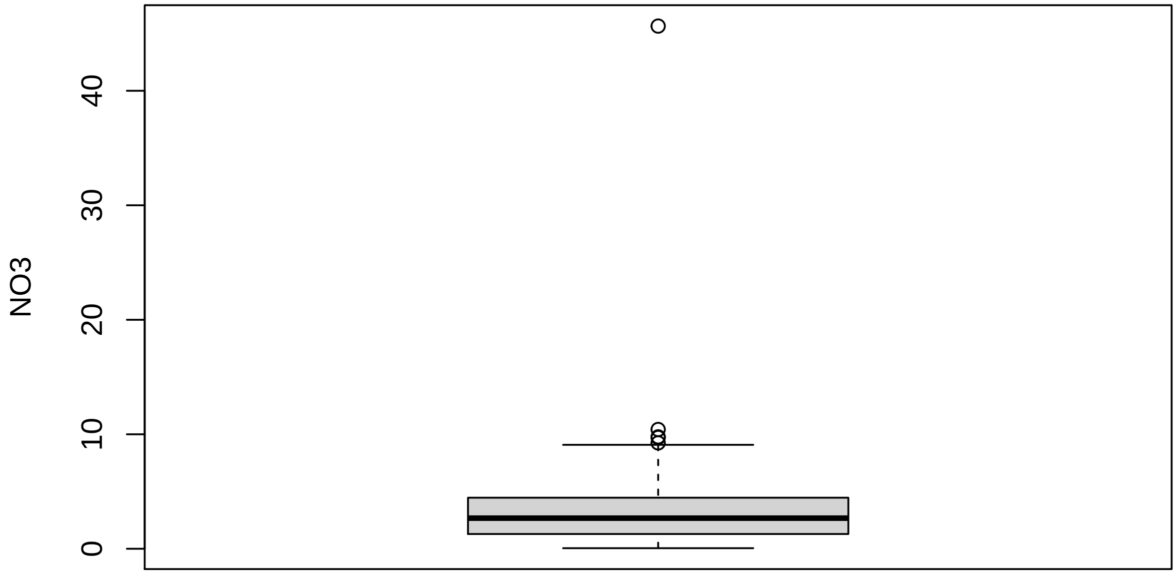
```
## `summarise()` has grouped output by 'size'. You can override using the `.groups` argument.
```

```
a_one %>% ggplot(aes(size,a1))+
  geom_boxplot()+
  ggtitle("A conditioned Boxplot of Algal a1")+
  theme_light()
```



2d

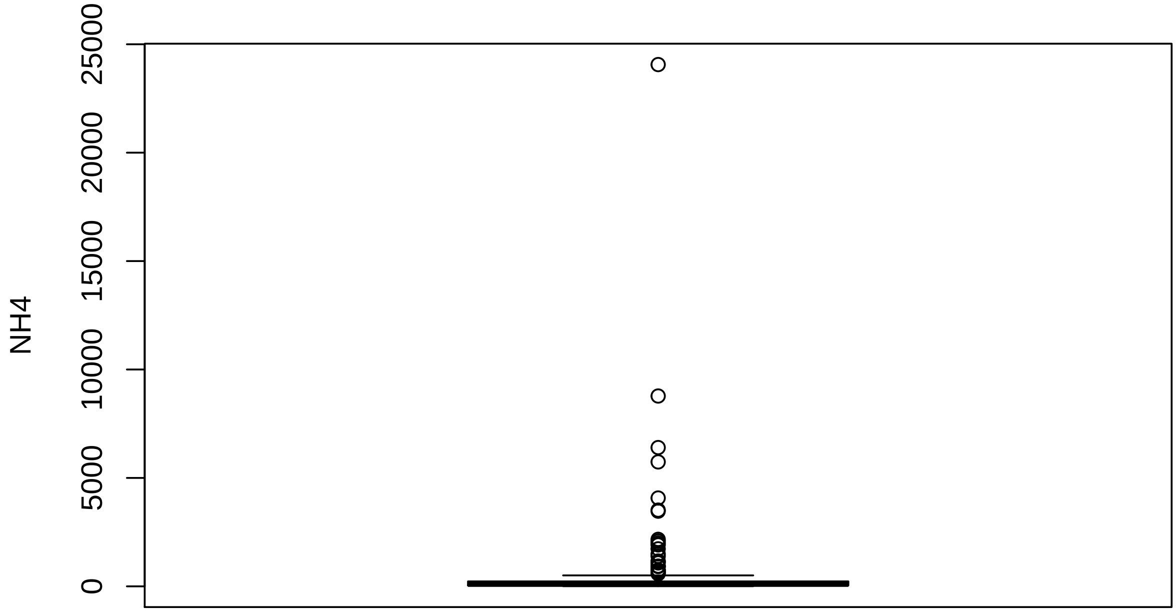
```
boxplot(algae$NO3,
  ylab = "NO3"
)
```



```
boxplot.stats(algae$NO3)$out
```

```
## [1] 10.416  9.248  9.773  9.715 45.650
```

```
boxplot(algae$NH4,
        ylab = "NH4")
```



```
boxplot.stats(algae$NH4)$out
```

```
##  [1]   578.000  8777.600  1729.000  3515.000  6400.000  1911.000   647.570
##  [8]  1386.250  2082.850  2167.370   737.500   914.000  5738.330  4073.330
## [15]   758.750   931.833   723.667  3466.660   920.000  1990.160 24064.000
## [22]  1131.660  1495.000   643.000   627.273  1168.000  1081.660
```

```
# yes there are some outliers for NO3 and NH4.
# I arrive at this conclusion by ploting the boxplot
```

2e

```
# 2e
algae%>%
  summarise(across(c(NO3, NH4), list(mean = mean, variance = var), na.rm = TRUE))
```

```
## # A tibble: 1 x 4
##   NO3_mean NO3_variance NH4_mean NH4_variance
##      <dbl>        <dbl>    <dbl>        <dbl>
## 1     3.28         14.3     501.     3851585.
```

```
algae%>%
  summarise(across(c(NO3, NH4), list(median = median, MAD = mad), na.rm = TRUE, constant = 1))
```

```
## # A tibble: 1 x 4
##   NO3_median NO3_MAD NH4_median NH4_MAD
##        <dbl>   <dbl>      <dbl>   <dbl>
## 1       2.68    1.46       103.    75.3
```

```
# the variance for NO3 is fine. but the variance for NH4 is crazy because of some outliers
# median&MAD data look more reasonable.
# median&MAD data are more robust when outliers are present.
```

3a

```
# 3a
nrow(algae[rowSums(is.na(algae))== FALSE,])
```

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

```
#  16 observations contain missing values.
algae %>%
  summarize(across(c(season, size, speed, mxPH, mnO2, Cl, NO3, NH4, oPO4, PO4, Chla, a1,a2,a3,a4,a5,a6,
```

```
## # A tibble: 1 x 18
##   season_missing size_missing speed_missing mxPH_missing mnO2_missing Cl_missing
##            <int>        <int>         <int>        <int>        <int>      <int>
## 1              0            0             0            1            2         10
## # ... with 12 more variables: NO3_missing <int>, NH4_missing <int>,
## #   oPO4_missing <int>, PO4_missing <int>, Chla_missing <int>,
## #   a1_missing <int>, a2_missing <int>, a3_missing <int>, a4_missing <int>,
## #   a5_missing <int>, a6_missing <int>, a7_missing <int>
```

3b

```
# 3b
algae.del <- algae[complete.cases(algae),]
nrow(algae.del)
```

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

```
# 184 observations are in algae.del.
```

3c

```
# 3c
algae.med <-algae %>%
  mutate_at(vars(season, size, speed, mxPH, mnO2, Cl, NO3, NH4, oPO4, PO4, Chla, a1,a2,a3,a4,a5,a6,a7),
```

```
## Warning: 'funs()' was deprecated in 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))
```

```
algae.med[c(48,62,199), ]
```

```
## # A tibble: 3 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
## 2 summer small medi~  6.4    9.8  32.7  2.68  103.  40.2    14   5.48  19.4   0
## 3 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

```
# 3d
algae%>%
  dplyr::select(4:11) %>%
  cor(., use = "na.or.complete")
```

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

```
fit <- lm(data = algae, PO4 ~ oPO4)
1.293 * algae[28,9] + 42.897
```

```
##     oPO4
## 1 48.069
```

3e

```
# 3e
# Because of the survivorship bias, we cannot simply use other obeserved data to fill in missing values
# Wen examing how bullet patterns affect the probability of survial of planes, we don't have data from
# for example, engine down might be the biggest impact, but planes won't come back if the engine is dow
# This means most of our observations have a functioning engine which may underestimate its impact.
# Imputing data using either the median or correlation method could also increase the bias in the predi
```

4a

```
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
```

```
## The following object is masked from 'package:purrr':
##
##     compact
```

```
# 4 a
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(as.matrix((predYtr-Ytr)^2)), # compute and store training error
             val.error = mean(as.matrix((predYvl-Yvl)^2)))# compute and store test error
}
set.seed(1)
nfold = 5
folds = cut(1:nrow(algae.med), breaks = nfold, labels = FALSE) %>%
  sample()
```

4b

```
# 4 b
tmp = ldply(1:nfold,do.chunk,chunkdef=folds,dat=algae.med)
tmp
```

```
##   fold train.error val.error
## 1    1    278.5925  342.0506
## 2    2    262.1875  404.3828
## 3    3    293.1881  272.5459
## 4    4    259.5677  477.9522
## 5    5    306.8601  246.4158
```

5

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

```
fit_all <- lm(a1~., data = algae.med[1:12])
X <- algae.Test[,1:11]
pred <- predict(fit_all,X)
real_error <- mean((pred - algae.Test$a1)^2)
real_error
```

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

```
#The true error here is 250.1794, which is roughly what I expected based on the CV estimated test error
```

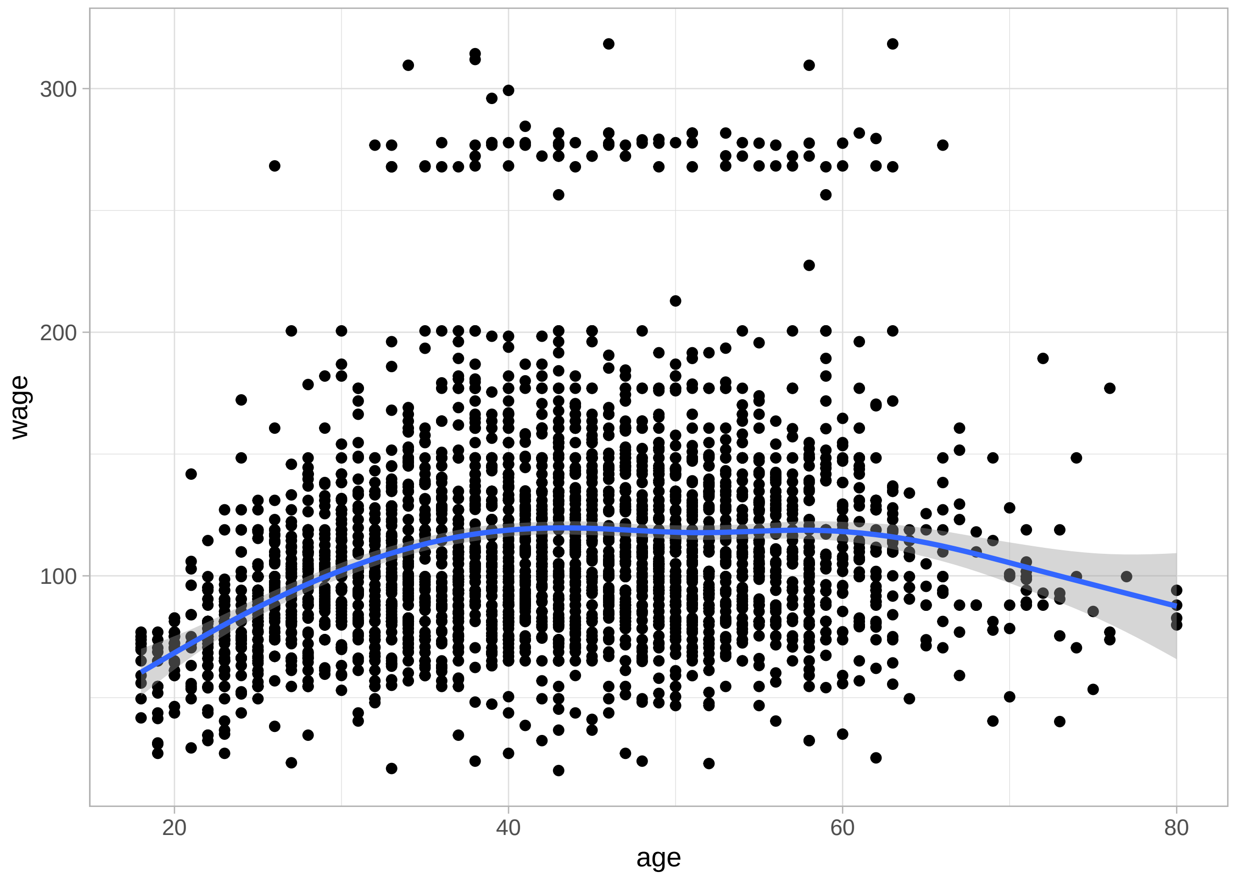
6

```
# 6 a
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
```

```
ggplot(data = Wage, aes(x = age, y = wage))+
  geom_point()+
  geom_smooth()+
  theme_light()
```

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



```
# wage grows as age grows, when age >40 & age < 60, the pattern is flat,
# when age > 60, the wage goes down as age increases.
# It matches what I expect. It is reasonable in daily life. For example, people who get first jobs usua
```

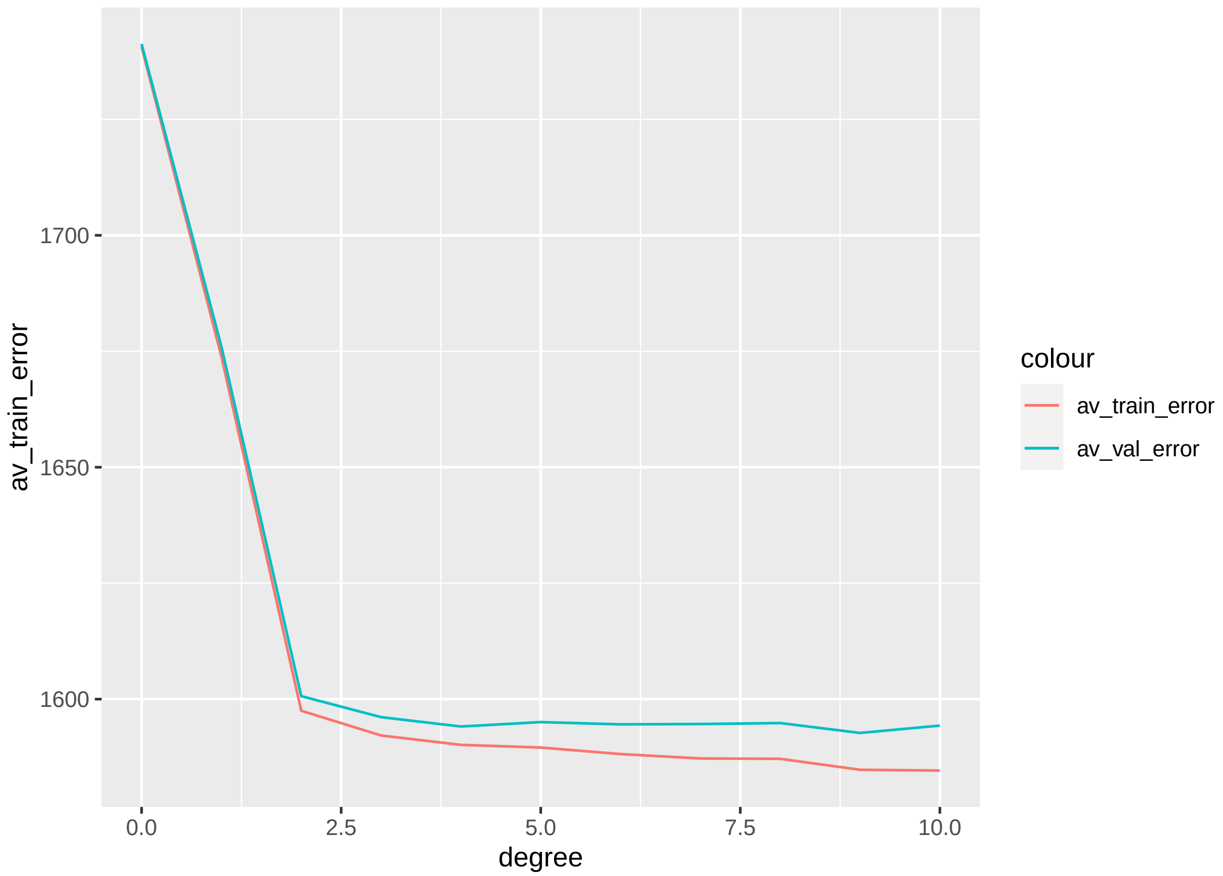
6b

```
# 6 b
set.seed(3)
folds2 = cut(1:nrow(Wage), breaks = nfold, labels = FALSE) %>%
  sample()

do.chunk3 <- function(chunkid, chunkdef, dat, degree){
  train = (chunkdef != chunkid)
  Xtr = dat[train, 1:10]
  Ytr = dat[train, 11]
  Xvl = dat[!train, 1:10]
  Yvl = dat[!train, 11]
  lm <- lm(wage~poly(age, degree, raw = F), data = dat[train, 1:11])
  predYtr = predict(lm)
  predYvl = predict(lm, Xvl)
  data.frame(fold = chunkid,
             p = degree,
             train.error = mean((predYtr - Ytr)^2), # compute and store training error
             val.error = mean((predYvl - Yvl)^2))
}
do.chunk4 <- function(chunkid, chunkdef, dat){
  train = (chunkdef != chunkid)
  Xtr = dat[train, 1:10]
  Ytr = dat[train, 11]
  Xvl = dat[!train, 1:10]
  Yvl = dat[!train, 11]
  lm <- lm(wage~1, data = dat[train, 1:11])
  predYtr = predict(lm)
  predYvl = predict(lm, Xvl)
  data.frame(fold = chunkid,
             train.error = mean((predYtr - Ytr)^2), # compute and store training error
             val.error = mean((predYvl - Yvl)^2))
}
final = NULL
temp4 <- ldply(1:5, do.chunk4, chunkdef = folds2, dat = Wage)
final = temp4%>%
  summarize(av_train_error = mean(train.error), av_val_error = mean(val.error), degree = 0)
for(i in 1:10){
  temp = (ldply(1:5, do.chunk3, chunkdef = folds2, dat = Wage, degree = i))
  temp2 = temp%>%
  summarize(av_train_error = mean(train.error), av_val_error = mean(val.error), degree = i)
  final <- rbind(final, temp2)
}
final
```

```
##    av_train_error av_val_error degree
## 1        1740.631     1741.269      0
## 2        1673.857     1676.008      1
## 3        1597.495     1600.641      2
## 4        1592.163     1596.118      3
## 5        1590.139     1594.100      4
## 6        1589.561     1595.048      5
## 7        1588.160     1594.560      6
## 8        1587.204     1594.635      7
## 9        1587.134     1594.839      8
## 10       1584.773     1592.717      9
## 11       1584.603     1594.295     10
```

```
ggplot(data = final)+
  geom_line(aes(x = degree, y = av_train_error, color = "av_train_error"))+
  geom_line(aes(x = degree, y = av_val_error, color = "av_val_error"))
```



```
# As p increases, both errors go down.
# training error is even lower.
which(final$av_val_error == min(final$av_val_error)) - 1
```

```
## [1] 9
```

```
# we should choose 9 as the degree.
```