Homework 1

PSTAT 131/231, Winter 2019

Due on January 26, 2018 at 11:55 pm

Note: If you are working with a partner, please submit only one homework per group with both names and whether you are taking the course for graduate credit or not. Submit your Rmarkdown (.Rmd) and the compiled pdf on Gauchospace.

Predicting Algae Blooms

Background High concentrations of certain harmful algae in rivers constitute a serious ecological problem with a strong impact not only on river lifeforms, but also on water quality. Being able to monitor and perform an early forecast of algae blooms is essential to improving the quality of rivers.

With the goal of addressing this prediction problem, several water samples were collected in different European rivers at different times during a period of approximately 1 year. For each water sample, different chemical properties were measured as well as the frequency of occurrence of seven harmful algae. Some other characteristics of the water collection process were also stored, such as the season of the year, the river size, and the river speed.

Goal We want to understand how these frequencies are related to certain chemical attributes of water samples as well as other characteristics of the samples (like season of the year, type of river, etc.)

Data Description The data set consists of data for 200 water samples and each observation in the available datasets is in effect an aggregation of several water samples collected from the same river over a period of 3 months, during the same season of the year. Each observation contains information on 11 variables. Three of these variables are nominal and describe the season of the year when the water samples to be aggregated were collected, as well as the size and speed of the river in question. The eight remaining variables are values of different chemical parameters measured in the water samples forming the aggregation, namely: Maximum pH value, Minimum value of O_2 (oxygen), Mean value of Cl (chloride), Mean value of NO_3^- (nitrates), Mean value of NH_4^+ (ammonium), Mean of PO_4^3 (orthophosphate), Mean of total PO_4 (phosphate) and Mean of chlorophyll.

Associated with each of these parameters are seven frequency numbers of different harmful algae found in the respective water samples. No information is given regarding the names of the algae that were identified.

We can start the analysis by loading into R the data from the "algaeBloom.txt" file (the training data, i.e. the data that will be used to obtain the predictive models). To read the data from the file it is sufficient to issue the following command:

```
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")

glimpse(algae)
```

1. ***Descriptive summary statistics*** Given the lack of further information on the problem domain, it is wise to investigate some of the statistical properties of the data, so as to get a better grasp of the problem. It is always a good idea to start our analysis with some kind of exploratory data analysis. A first idea of the statistical properties of the data can be obtained through a summary of its descriptive statistics.

    (a) Count the number of observations in each season using `summarise()` in `dplyr`.

Table 1: Number of Observations in Each Season

season	freq
autumn	40
spring	53
summer	45
winter	62

Table 2: Missing Values for Each Chemical

mxPH	mnO2	Cl	NO3	NH4	oPO4	PO4	Chla
1	2	10	2	2	2	2	12

1. Are there missing values? Calculate the mean and variance of each chemical (Ignore a_1 through a_7). What do you notice about the magnitude of the two quantities for different chemicals?

There are missing values for several variables. One value is missing for mxPH. NO3, NH4, oPO4, and PO4 are all missing two values. Cl and Chla are each missing 10 and 12 variables.

The mean for NH4 is the largest, but it had a disproportionately large variance in observed measurements that resulted in this chemical having the largest variance to mean ratio at 7,683. PO4 had the second highest variance to mean ratio 120 followed by oPO4's variance to mean ratio of 112. These results highlight considerable variation among the measured values for NH4, PO4, and oPO4, which could be attributed to seasonal variation in uptake of these critical nutrients for algae that are more active in warmer temperatures but tend to die off in the winter. The mxPH had the smallest variance and the smallest variance to mean ratio of 0.04, which demonstrates consistency of maximum pH observed across the sample sites and seasons.

```
#. Mean and Variance is one measure of central tendency and spread of data.
Median and Median Absolute Deviation are alternative measures of central
tendency and spread.

    For a univariate data set $X_1, X_2, ..., X_n$, the Median Absolute Deviation (MAD) is defined as t

    Compute median and MAD of each chemical and compare the two sets of quantities (i.e., mean & varian
```

```
print('Median')
```

```
## [1] "Median"
```

```
chem_medians <- algae %>%
  select(c('mxPH','mnO2','Cl','NO3','NH4', 'oPO4','PO4','Chla')) %>%
  summarise_all(function (x) median(x, na.rm=TRUE))

print('MADs')
```

```
## [1] "MADs"
```

```
chem_mads <- algae %>%
  select(c('mxPH','mnO2','Cl','NO3','NH4', 'oPO4','PO4','Chla')) %>%
  summarise_all(function (x) mad(x, na.rm=TRUE))

med_mad_bind <- rbind(chem_medians, chem_mads)
rownames(med_mad_bind) <- c("Median", "MAD")
```

```
## Warning: Setting row names on a tibble is deprecated.
```

Table 3: Chemical Means and Variances

	mxPH	mnO2	Cl	NO3	NH4	oPO4	PO4	Chla
Mean	8.0117337	9.117778	43.63628	3.282389	501.2958	73.5906	137.8821	13.9712
Variance	0.3579693	5.718089	2193.17173	14.261757	3851584.6849	8305.8499	16639.3845	420.0827

Table 4: Chemical Medians and MADs

	mxPH	mnO2	Cl	NO3	NH4	oPO4	PO4	Chla
Median	8.060000	9.800000	32.73000	2.675000	103.1665	40.15000	103.2855	5.4750
MAD	0.504084	2.053401	33.24953	2.172009	111.6175	44.04582	122.3212	6.6717

```r
kable(med_mad_bind, "latex", booktabs = T,
      caption = "Chemical Medians and MADs") %>%
      kable_styling(bootstrap_options = "striped", full_width = F, position = "center")
```

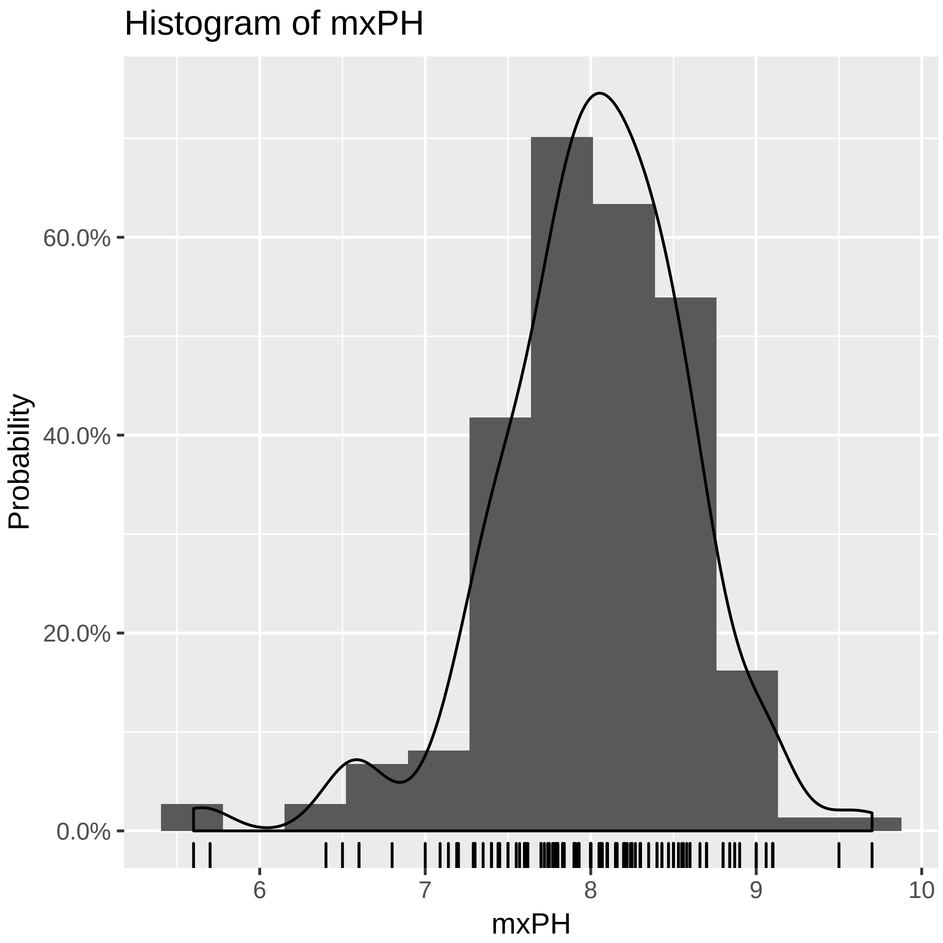
```r
mads_medians_ratio <- chem_mads/chem_medians
```

The ratio of MAD to median for all chemicals except mxPH are smaller in magnitude than the variance to mean ratio for each parameter, respectively. For most parameters, the MAD to median was many times smaller, and in the case of NH4 there was a reduction of three orders of magnitude as compared to its variance to mean ratio. PO4 and oPO4 MAD to median ratios were two orders of magnitude smaler than variance to mean ratios. These results highlight the significant difference between the results presented through two methods of central tendency. The considerably lower MAD to median ratios demonstrates that a few outlier measurements may be causing much higher variance in the calculation of means while the calculation of MAD will not be severely influenced by outliers because the MAD is the median of the absolute difference between each measurement and the mediant of the measurements. Bearing this in mind, it is important to consider what an appropriate representation of central tendency is and determine whether or not particular measures truly are outliers or not.

2. ***Data visualization*** Most of the time, the information in the data set is also well captured graphically. Histogram, scatter plot, boxplot, Q-Q plot are frequently used tools for data visualization. Use ggplot for all of these visualizations.

   (a) Produce a histogram of $mxPH$ with the title 'Histogram of mxPH' based on algae data set. Use an appropriate argument to show the probability instead of the frequency as the vertical axis. (Hint: look at the examples in the help file for function `geom_histogram()`). Is the distribution skewed?

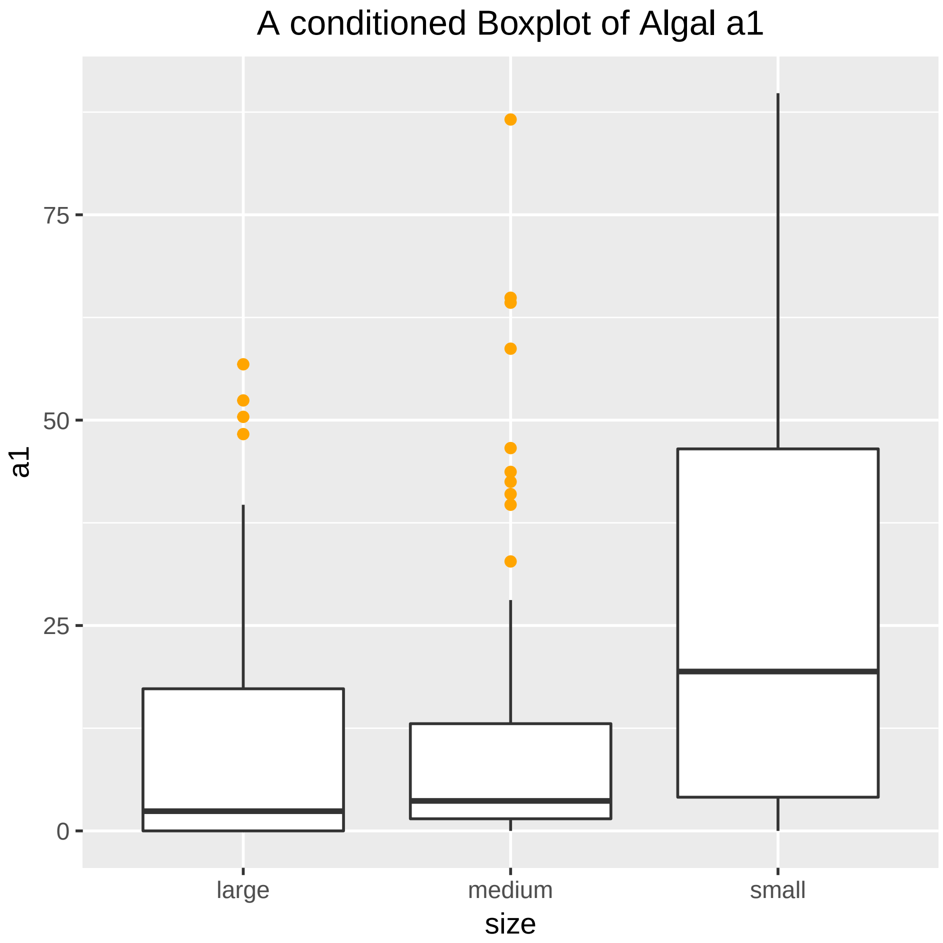
```r
hist_mxPH <- ggplot(algae, aes(x = mxPH)) +
geom_histogram(bins = 12, aes(y = ..density..)) + #2*(200)^1/3
labs(title = 'Histogram of mxPH') +
ylab('Probability') +
scale_y_continuous(labels =scales::percent) +
geom_density() +
geom_rug()

hist_mxPH
```



The histogram of mxPH is skewed with several more observations in the three bins immediately greater than the mean as compared to the three bins just below the mean.

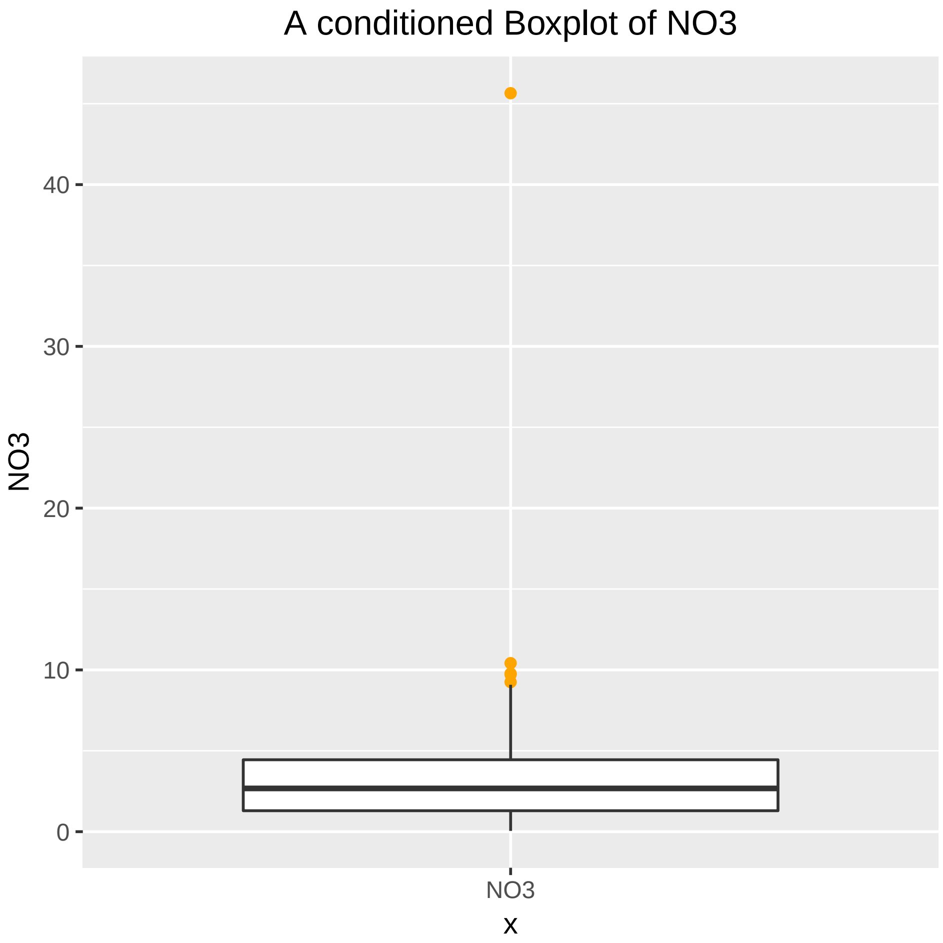
```
#. Add a density curve using `geom_density()` and rug plots using `geom_rug()` to above histogram.

#. Create a boxplot with the title 'A conditioned Boxplot of Algal $a_1$' for $a_1$ grouped by $size$.
```

```
algae_boxplot <- ggplot(algae, aes(size, a1)) +
  geom_boxplot(outlier.color = 'orange') +
  labs(title = 'A conditioned Boxplot of Algal a1') +
  theme(plot.title = element_text(hjust = 0.5))

algae_boxplot
```

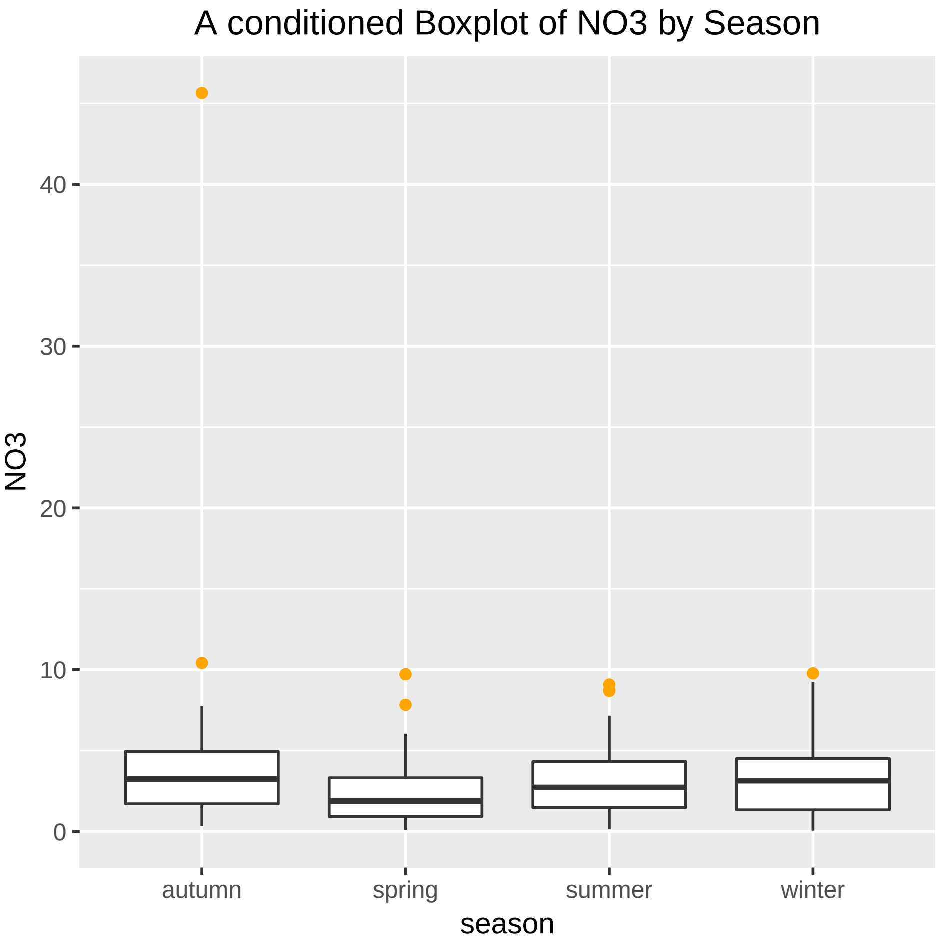


1. Are there any outliers for $NO3$ and $NH4$? How many observations would you consider as outliers? How did you arrive at this conclusion?

```
NO3_boxplot <- ggplot(algae, aes(x = 'NO3', y = NO3)) +
  geom_boxplot(outlier.color = 'orange') +
  labs(title = 'A conditioned Boxplot of NO3') +
  theme(plot.title = element_text(hjust = 0.5))

NO3_boxplot
```

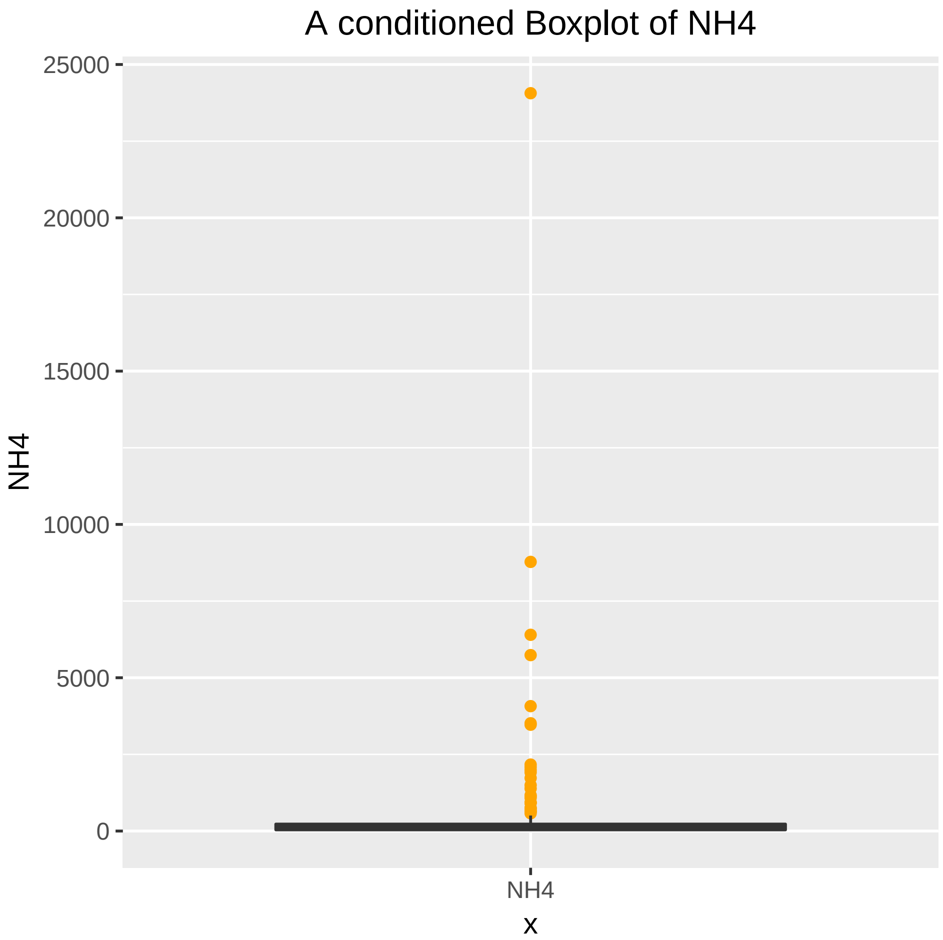


```
NO3season_boxplot <- ggplot(algae, aes(season, y = NO3)) +
  geom_boxplot(outlier.color = 'orange') +
  labs(title = 'A conditioned Boxplot of NO3 by Season') +
  theme(plot.title = element_text(hjust = 0.5))

NO3season_boxplot
```

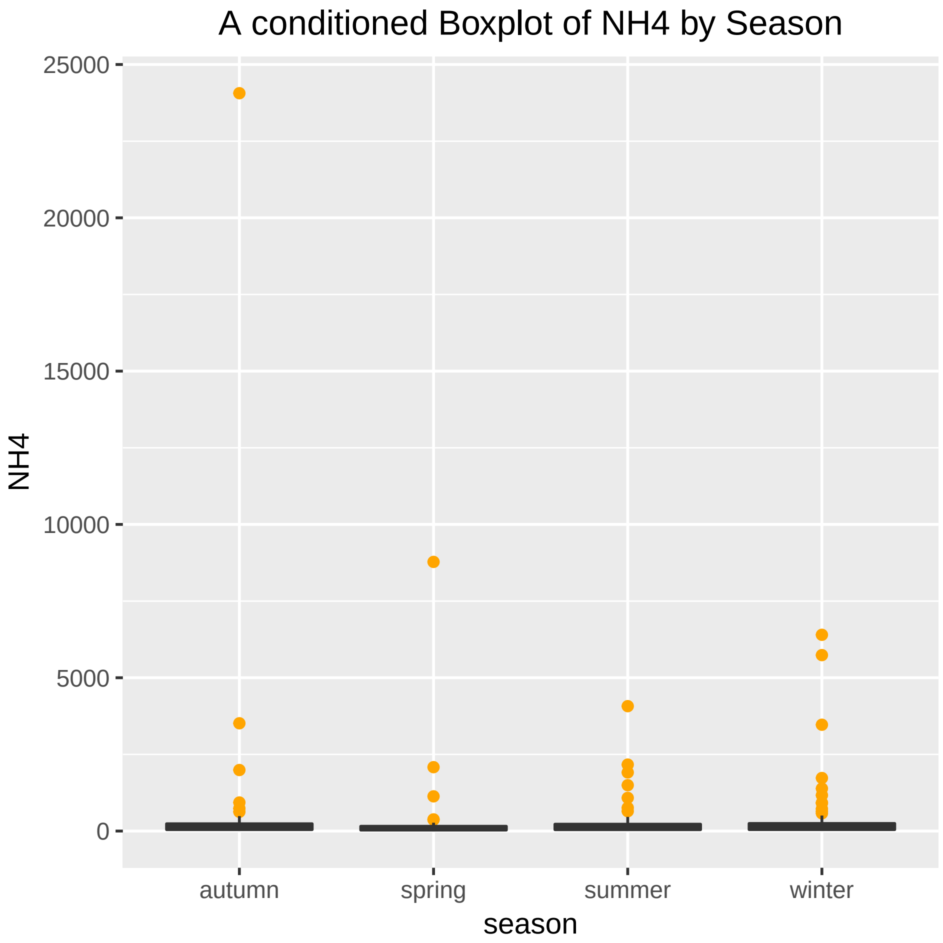


By separating the NO3 data into seasons, we can see that there is a clear outlier in autumn that is significantly further from the Interquartile Range than outliers at more than four times the value measured for NO3 in other months. Note: outliers are indicated in orange. Thus, we will only consider this one autumn outlier as a true outlier.

```
NH4_boxplot <- ggplot(algae, aes(x = 'NH4', y = NH4)) +
  geom_boxplot(outlier.color = 'orange') +
  labs(title = 'A conditioned Boxplot of NH4') +
  theme(plot.title = element_text(hjust = 0.5))

NH4_boxplot
```



```
NH4season_boxplot <- ggplot(algae, aes(season, y = NH4)) +
  geom_boxplot(outlier.color = 'orange') +
  labs(title = 'A conditioned Boxplot of NH4 by Season') +
  theme(plot.title = element_text(hjust = 0.5))

NH4season_boxplot
```



Determining the number of outliers for NH4 measurements is slightly more challenging, but splitting up the box plot into seasonal observations helps once again. There are four clear true outliers across all seasons (one in both autumn and summer, two in winter) with values above the 5000 threshold that are well beyond the range of other measurements. It also seems reasonable to count the three outlier points (one in each of autumn, summer and winter) above the ~3000 value threshold as additional true outliers. While this is a somewhat arbitrary threshold, these three points seemed similarly distinct from the other outlier points closer to the extent of the Interquartile Range. However, it is possible that the presence of these three measures of NH4 at ~3000 across three seasons may suggest that this is not an outlier but a characteristic of the study site. The remaining outlier points below NH4 = 2500 are not considered true outliers because they are not too far beyond the Interquartile Range for each season and it would be challenging to tease out any further groupings to distinguish some of these points from others as ture outliers. Thus a total of seven true outliers were chosen for NH4 observations.

```
#. Compare mean & variance vs. median & MAD for $NO3$ and $NH4$. What do you notice? Can you conclude wh
```

The ratios of the variances to means for both nitrate and ammonium were 4.3 and 7683, respectively. The particularly high variance to mean ratio for ammonium results from the considerable variation in the data set that can be visually observed in the above graph. For ratio of variance to mean for nitrate is much smaller in magnitude because it only has a single extreme outlier that causes the variance to four times larger than its mean. On the other hand, the ratios of MADs to medians for both of these chemical are 0.81 and 1.08, respectively. These ratios demonstrate that the MADs for these two chemicals are much closer to their respective medians. These results demonstrates that using MADs and medians are much more robust to outliers as measures of central tendency.

Table 5: Counts of Missing Values

season	size	speed	mxPH	mnO2	Cl	NO3	NH4	oPO4	PO4	Chla	a1	a2	a3	a4	a5	a6	a7
0	0	0	1	2	10	2	2	2	2	12	0	0	0	0	0	0	0

Predicting Algae Blooms

Some water samples contained unknown values in several chemicals. Missing data are very common in real-world problems, and may prevent the use of certain data mining techniques that are not able to handle missing values.

In this homework, we are going to introduce various ways to deal with missing values. After all the missing values have been taken care of, we will build a model to investigate the relationship between the variable `a1` and other 11 predictors (`season`, `size`, `speed`, `mxPH`, `mnO2`, `Cl`, `NO3`, `NH4`, `oPO4`, `PO4`, `Chla`) utilizing cross-validation in the next problem.

Dealing with missing values

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

There are 16 observations that contain at least one missing value for a parameter with a total of 33 missing variable values across all of the observations. There is one missing value for mxPH, and two missing for each of mnO2, NO3, NH4, oPO4, and PO4. Ten values are missing for Cl and 12 are missing for Chla.

```
#. **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`.

    Hint: `complete.cases()` may be useful.
```

```
algae.del <- algae %>%
  filter_all(all_vars(!is.na(.)))
```

There are 184 observations in algae.del

```
#. \label{imputation} **Imputing unknowns with measures of central
tendency**: the simplest and fastest way of filling in (imputing) missing
values is to use some measures of central tendency such as mean, median and
mode.

    Use `mutate_at()` and `ifelse()` in `dplyr` to fill in missing values
    for each chemical with its median, and save the imputed dataset as
    `algae.med`. Report the number of observations in `algae.med`.  Display
    the values of each chemical for the $48^{th}$, $62^{th}$ and $199^{th}$
    obsevation in `algae.med`.
```

```
algae.med <- algae %>%
  mutate_at(.vars = c('mxPH','mnO2','Cl','NO3','NH4',
  'oPO4','PO4','Chla'), .funs = funs(ifelse(is.na(.), median(., na.rm = TRUE), .)))

algae.med.48 <- algae.med[48, 4:11]
algae.med.62 <- algae.med[62, 4:11]
algae.med.199 <- algae.med[199, 4:11]

algae.med.rows <- rbind(algae.med.48, algae.med.62, algae.med.199)
```

Table 6: Chemical Observations for Specific Rows After Imputing

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

```r
rownames(algae.med.rows) <- c("48", "62", "199")
```

```
## Warning: Setting row names on a tibble is deprecated.
```

There are 200 observations in algae.med.

```
    This simple strategy, although extremely fast and thus appealing for
    large datasets, imputed values may have large bias that can influence
    our model fitting. An alternative for decreasing bias of imputed values
    is to use relationships between variables.

#. **Imputing unknowns using correlations**: another way to impute missing
values is to use correlation with another variable. For a highly
correlated pair of variables, we can fill in the unknown values by
predicting one based on the other with a simple linear regression model,
provided the two variables are not both unknown.

    Compute pairwise correlation between the continuous (chemical) variables.
```

```r
algae.cor = algae %>%
  select(c('mxPH','mnO2','Cl','NO3','NH4',
  'oPO4','PO4','Chla','a1','a2','a3','a4','a5', 'a6', 'a7')) %>%
  cor(use = "complete.obs")
```

```
    Then, fill in the missing value for `PO4` based on `oPO4` in the
    $28^{th}$ observation. What is the value you obtain?

    Hint: use `lm()` and `predict()` function.
```

```r
PO.lm <- lm(PO4 ~ oPO4, data = algae %>%
             select(c('PO4','oPO4')))
oPO4 <- algae %>%
  select(c('oPO4'))
prediction28 = predict(PO.lm, oPO4[28,])
```

The model predicts a value of 48.07 units of P04 in the $28^{th}$ observation.

```
#. **Questioning missing data assumptions**:  When might imputation using only the observed data lead yo
```

Imputing missing values using only observed data could lead to issues with conclusion generation if there is systemic bias to the way in which data points are missing. Furthermore, imputing missing values based on the presence points will reduce the variation in the entire data set. There could be some sort of survivorship bias in the missing values for this algae data set similar to the example given in class about using bullet hole locations in airplanes as a means of deciding where to reinforce the body of the plane. More careful examination is necessary to deduce if there is some systemic bias in the missing values in this algae data set. For example, if the data was missing for certain chemicals at one site on multiple sampling days because inclement weather and flooding during a particular season made measurements infeasible, then there could be a bias in the missing data. Imputing the missing values based on other observations would likely fail to capture the chemical composition of that river during flooding events that was directly measured.

Estimating the Test Error with Cross Validation (CV)

Using `algae.med` dataset obtained in (**??**), we will build a linear regression model to predict the levels of algae type `a1` based on 11 variables (`season`, `size`, `speed`, `mxPH`, `mnO2`, `Cl`, `NO3`, `NH4`, `oPO4`, `PO4`, `Chla`), and test generalization of model to data that have not been used for training.

4. **Cross-validation**: in class we talked about how to use cross-validation (CV) to estimate the "test error". In k-fold CV, each of k equally sized random~ partitions of data (chunks) are used in a heldout set (called validation set or test set). After k runs, we average the held-out error as our final estimate of the validation error. For this part, we will run cross-validation on only a single model, as a way to estimate our test error for future predictions (we are not using it here for model selection since we are considering only one model). Perform 5-fold cross-validation on this model to estimate the (average) test error.

   (a) First randomly partition data into 5 equal sized chunks.

   Hint: a simple way to randomly assign each observation to a chunk is to do the following. First, use `cut(..., label=FALSE)` to divide observation ids (1, 2, . . .) into equal numbers of chunk ids. Then, randomize output of `cut()`by using `sample()`.

```
IDs <- c(1:200)
IDs.cut <- cut(IDs, 5, label = FALSE) %>%
  sample()
```

~~~
#. Perform 5-fold cross-validation with training error and validation
errors of each chunk determined from \eqref{chunkids}.

    Since same computation is repeated 5 times, we can define the following
    function for simplicity.


    ```r
    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)^2), # compute and store training error
                   val.error = mean((predYvl - Yvl)^2)) # compute and store test error

    }
    ```

    First argument `chunkid` indicates which chunk to use as validation set
    (one of 1:5). Second argument `chunkdef` is chunk assignments from
    \eqref{chunkids}. Third argument `dat` will be `algae.med` dataset.
~~~

Table 7: Training and Validation Errors for 5-fold Cross Validation

fold	train.error	val.error
1	267.7011	386.9186
2	276.5036	346.3671
3	316.3755	193.6436
4	273.2023	347.5212
5	276.0718	509.0541
Mean	281.9709	356.7009

```
In order to repeatedly call `do.chunk()` for each value of `chunkid`,
use functions `lapply()` or `ldply()`. Note that `chunkdef` and `dat`
should be passed in as optional arguments (refer to help pages).

Write the code and print out the `train.error` and `val.error` five times (e.g. for each chunk).
```

```r
nfold = 5
error.folds = NULL

algae.med.df = as.data.frame(algae.med)

set.seed(654)

tmp = ldply(1:nfold, do.chunk,
          chunkdef=IDs.cut, dat=algae.med.df)

error.folds=rbind(error.folds, tmp)
mean.val = mean(error.folds$val.error)
mean.train = mean(error.folds$train.error)
mean.errors = data.frame(fold = "Mean", train.error = mean.train, val.error = mean.val)
error.folds = rbind(error.folds, mean.errors)


kable(error.folds, "latex", booktabs = T,
      caption = "Training and Validation Errors for 5-fold Cross Validation") %>%
      kable_styling(bootstrap_options = "striped", full_width = F, position = "center")
```

5. **Test error on additional data**: now imagine that you actually get *new* data that wasn't available when you first fit the model.

   (a) Additional data can be found in the file `algaeTest.txt`.

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

   This data was not used to train the model and was not (e.g. wasn't used in the CV procedure to estimate the test error). We can get a more accurate measure of true test error by evaluating the model fit on this held out set of data. Using the same linear regression model from part 4 (fit to all of the training data), calculate the "true" test error of your predictions based on the newly collected measurements in `algaeTest.txt`. Is this roughly what you expected based on the CV estimated test error from part 4?

```
# Evaluate fit of previous model against new algae.Test data

# The error results from Question 4 part B help to identify that there is the smallest validation error

# re-train a model on all of the original dataset
lm.a1.full <- lm(a1~., data = algae.med.df[1:12]) # should this include the a1 column?

# predict the model output for algae a1 using the new algae.Test data that wasn't used to create the mo
predYvl = predict(lm.a1.full,algae.Test)  # predict validation values
val.a1 = algae.Test$a1 # actual values of a1 in validation (algae.Test) data frame

val.error.data = mean((predYvl - val.a1)^2) # compute and store test error

val.error.data
```

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

When predictions for a1 are made using a linear model trained on the original data set and the new algae.Test data, a validation error is found to be 250.18. In Q4b (the 5 fold CV with test data coming from the same data set as the training data), the mean validation error was 354.89 while the mean test error was 282.03. This presents a counterintuitive case in that the validation error from question 5 (using algae.Test) would be expected to be closer to the mean validation error from Q4b, when in fact we've found the validation error from Q5 to be closer to the TRAINING ERROR from Q4b. This result was likely caused by random chance given the two data sets used but demonstrates an important lesson in checking possible anomolies in model generation and validation.

Cross Validation (CV) for Model Selection

In this problem, we will be exploring a dataset of wages from a group of 3000 workers. The goal in this part is to identify a relationship between wages and age.

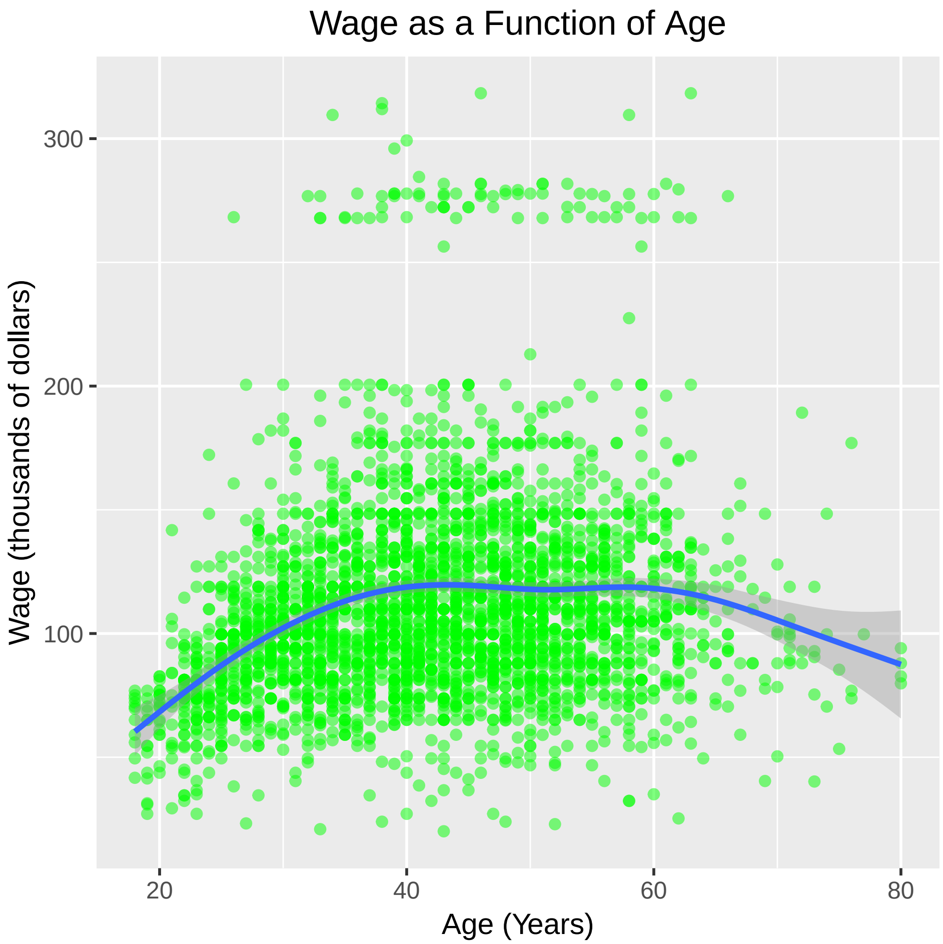
6. First, install the `ISLR` package, which includes many of the datasets used in the ISLR textbook. Look at the variables defined in the `Wage` dataset. We will be using the `wage` and `age` variables for this problem.

```
library(ISLR)
head(Wage)

Wage = ISLR::Wage
```

   (a) Plot wages as a function of age using `ggplot`. Your plot should include the datapoints (`geom_point()`) as well as a smooth fit to the data (`geom_smooth()`). Based on your visualization, what is the general pattern of wages as a function of age? Does this match what you expect?

```
wage_age_plot <- ggplot(ISLR::Wage, aes(x = age, y = wage)) +
  geom_point(color = 'green', alpha = 0.5) +
  geom_smooth() +
  xlab("Age (Years)") + ylab("Wage (thousands of dollars)") +
  ggtitle("Wage as a Function of Age") +
  theme(plot.title = element_text(hjust = 0.5))


wage_age_plot
```

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



There is a general pattern of increasing wage with age until it peaks in the early 40s. After that there is a slight dip in wage until retirement but it is generally a plateau until 62 after which point there is a more clear decline after people have retired. Based on current knowledge of trends in wage as a function of age and the justifications outlined above, the shape of this graph is generally what would be expected.

```
#.  In this part of the problem, we will find a polynomial function of age that best fits the wage data

    #.  Fit a linear regression to predict wages as a function of $age$, $age^2$, ... $age^p$ (you shoul

    #.  Use 5-fold cross validation to estimate the test error for this model. Save both the test error
```

```
do.wage <- function(chunkid, chunkdef, p, dat){
    Wage = dat
    train = (chunkdef != chunkid)

    if(p<1){
      predictors.df <- data.frame(age0 = 1)
      wage = data.frame(Wage$wage)
      colnames(wage) = "wage"

      predictors_new = cbind(predictors.df, wage)

      Xtr = predictors_new[train,1]  # get training set
      Ytr = predictors_new[train,2]  # get true response values in trainig set
```

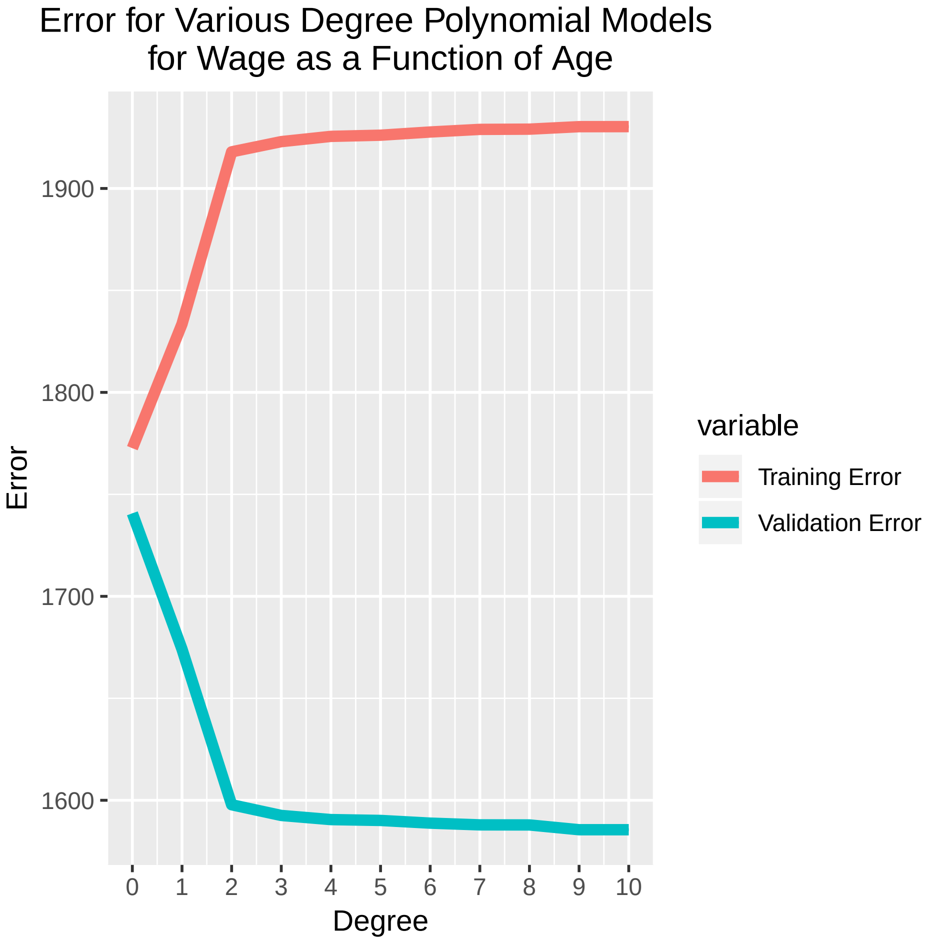
```r
      Xvl = as.data.frame(predictors_new[!train,1]) # get validation set
      colnames(Xvl) <- paste("age", 0, sep = "")
      Yvl = predictors_new[!train,2] # get true response values in validation set
    }else{
      predictors <- poly(Wage$age, p, raw = TRUE)

      predictors.df <- as.data.frame(predictors)

      colnames(predictors.df) <- paste("age", 1:p, sep = "")

      wage = data.frame(Wage$wage)
      colnames(wage) = "wage"

      Xtr = predictors.df[train,1:p]  # get training set
      Ytr = wage[train,1]  # get true response values in trainig set

      Xvl = as.data.frame(predictors.df[!train,1:p]) # get validation set
      colnames(Xvl) <- paste("age", 1:p, sep = "")

      Yvl = wage[!train,1] # get true response values in validation set

      predictors_new = cbind(wage, predictors.df)

    }
      lm.wage <- lm(wage~., data = predictors_new)

      predYtr = predict(lm.wage)  # predict training values
      predYvl = predict(lm.wage,Xvl)  # predict validation values

      data.frame(degree = p,
               fold = chunkid,
               train.error = mean((predYtr - Ytr)^2), # compute and store train error
               val.error = mean((predYvl - Yvl)^2)) # compute and store test error


}
```

```r
nfold = 5

folds <- cut(1:3000, nfold, label = FALSE) %>%
  sample()

error.folds.wages = data.frame()

allP = 0:10

set.seed(5555)

for (j in allP){
  tmp.wage = ldply(1:nfold, do.wage,
    chunkdef=folds, p = j, dat = ISLR::Wage)

  error.folds.wages = rbind(error.folds.wages, tmp.wage)
}
```

```
error.means = aggregate(error.folds.wages[, 3:4], list(error.folds.wages$degree), mean)
colnames(error.means) = c("degree", "Training Error","Validation Error")

error.melt = melt(error.means, id.vars=c('degree'), value.name='error')
```

```
#. Plot both the test error and training error (on the same plot) for each of the models estimated abov
```

```
wage.error.plot <- ggplot(error.melt, aes(x=degree, y=error, col = variable))+
    geom_line() +
    stat_summary(aes(group=variable), fun.y="mean", geom='line', size=2) +
    scale_x_continuous(breaks = seq(0,10, by = 1)) +
    xlab("Degree") + ylab("Error") +
    ggtitle("Error for Various Degree Polynomial Models \nfor Wage as a Function of Age") +
    theme(plot.title = element_text(hjust = 0.5))


wage.error.plot
```



As the degree (p) of the polynomial model used increases, the training error increases. For the first three degrees (0 through 2), there's a rapid increase in training error and then there is a slightly greater increase to p = 4 before it plateaus with only a slightly further increase as the degree of the polynomial model approaches 10. A similar pattern except in the decreasing direction is observed for the validation error with a rapid decrease in validation error from p = 0 to 2 and then a gradual decrease in validation error to p = 4 and minimal change as the degree of the model goes to p = 10. Based on these results, the fourth degree polynomial should be selected because it has a low validation error similar to higher degree polynomials and the training error has yet to increase much more. At p = 0 through 3, the model is not flexible enough and will have a higher bias. Choosing a higher degree polynomial model than p = 4 won't make much difference in improving the model's validation error and the training error will also increase. While selecting a higher degree polynomial would increase flexibility of the model, it may result in overfitting with higher variance in modeled results. Additionally, a polynomial degree 4 would have a generally parabolic shape with slightly greater flexibility, which is generally the shape that might be expected to describe wage as a function of age (increase in pay until a peak age where there is a plateau and then a gradual drop off). Given these concerns, a fourth degree polynomial model seems most appropriate for this scenario.

```
Note: `poly(age, degree=p, raw=TRUE)` will return a matrix with $p$ columns, where the $p$-th column i
Hint: A function similar to `do.chunk` from problem 4 will be helpful here as well.
```

---

7. **(231 Only) The bias-variance tradeoff**. Prove that the mean squared error can be decomposed into the variance plus bias squared. That is, who $E[(\hat{\theta} - \theta)^2] = \text{Var}(\hat{\theta}) + \text{Bias}(\hat{\theta})^2$ where $\text{Bias}(\hat{\theta}) = E[\hat{\theta} - \theta]$. Here $\hat{\theta}$ is an estimator (a random variable) of the fixed unknown constant θ. Hint: reogranize terms in the MSE by adding and subtracting $E[\hat{\theta}]$.

Proof that $MSE(\hat{p}) = Var_p(\hat{p}) + Bias_p(\hat{p}, p)^2$ where, $\hat{p}$ is the MSE estimator of p:
We have,

$$MSE(\hat{p}) = E_p\left[(\hat{p} - p)^2\right]$$

$$= E_p\left[\left(\hat{p} - E_p(\hat{p}) + E_0(\hat{p}) - p\right)^2\right]$$

$$= E_P\left[\hat{p} - E_p(\hat{p})^2\right] + E_p\left[2\left(\hat{p} - E_p(\hat{p})\left(E_p(\hat{p}) - p\right)\right] + E_p\left[\left(E_p(\hat{p}) - p\right)^2\right]$$

$$= E_p\left[\hat{p} - E_p(\hat{p})^2\right] + 2\left(E_p(\hat{p}) - p\right)\left(E_p(\hat{p}) - E_p(\hat{p})\right) + \left(E_p(\hat{p}) - p\right)^2 \text{ constants: } E_p(\hat{p}) - p \text{ and } E_p(\hat{p})$$

$$= E_p\left[\hat{p} - E_p(\hat{p})^2\right] + \left(E_p(\hat{p}) - p\right)^2$$

$$= \text{Var}(\hat{p}) + \text{Bias}(\hat{p}, p)^2 \text{ by definition.}$$

8. **(231 Only)** As we discussed in class, distance metrics satisfy the following properties:

- *Positivity*:
  - $d(x, y) \geq 0$
  - $d(x, y) = 0$ only if $x = y$
- *Symmetry*:
  - $d(x, y) = d(y, x)$ for all x and y
- *Triangle Inequality*:
  - $d(x, z) \leq d(x, y) + d(y, z)$ for $x,\ y,$ and z

Show that the following measures are distance metrics by showing the above properties hold:

(a) $d(x, y) = \|x - y\|_2$ Proof that d_2 is a metric in $\mathbb{R}^2$

Let $x, y \in \mathbb{R}^2$. We have $d_2(x, y) = \|x - y\|_2 = \left[(x_1 - y_1)^2 + (x_2 - y_2)^2\right]^{1/2} \geq 0 \quad \forall x, y \neq 0$

Note, , $d_2(x, y) = 0 \Leftrightarrow x_i = y_i \quad \forall_{i \in \mathbb{Z}}$ (positivity)

We have $d_2(x, y) = \|x - y\|_2 = \left[(x_1 - y_1)^2 + (x_2 - y_2)^2\right]^{1/2}$

$$= \left[(y_1 - x_1)^2 + (y_2 - x_2)^2\right]^{1/2} \qquad \text{(symmetry)}$$

$$= \|x - y\|_2 = d_2(y, x)$$

Take $z \in \mathbb{R}^2$ then,

$$\begin{aligned} d_2(x,z) = \|x-z\|_2 &= \|(x-y)+(y-z)\|_2 \\ &\leq \|x-y\|_2 + \|y-z\|_2 = d(x,y) + d_2(y,z) \end{aligned} \text{ (triangle inequality).}$$

(b) $d(x,y) = \|x-y\|_\infty$

Proof that $d_\infty(x,y)$ is a metric in $\mathbb{R}^2$:
Take $x, y \in \mathbb{R}^2$. We have $d_\infty(x,y) = \|x-y\|_\infty = \max\{|x_1-y_1|, |x_2-y_2|\} \geq 0 \quad \forall x, y \neq 0$
Note, $d_\infty(x,y) = 0 \Leftrightarrow x_i = y_1 = 0 \quad \forall_i \in Z$ (positivity)
We have $d_\alpha(x,y) = \|x-y\|_\infty = \max\{|x_1-y_1|, |x_2-y_2|\}$
$= \max\{|y_1-x_1|, |y_2-x_2|\}$
$= \|y-x\|_\infty = d_\infty(y,x)$ (symmetry)
Take $z \in \mathbb{R}^2$ then,
$d_\infty(x,z) = \max\{|x_1-z_1|, |x_2-z_2|\} \leq \max\{|x_1-y_1|, |x_2-y_2|, |y_1-z_1|, |y_2-z_2|\}$
$= \max\{|x_1-y_1| \, x_2-y_2|\} + \max\{|y_1-z_1|, |x_2-z_2|\}$
$= d_\infty(x,y) + d_\infty(y,z)$ by definition.