STAD80 (Winter 2020) Analysis of Big Data **Assignment: 1**

Fundamentals of Statistics and Regression

Lecturer: Qiang Sun *Email: qsun@utstat.toronto.edu*

The due date for this homework is Feb 6th, Thursday 11:59pm. Please submit your homework via Quercus. You will have to submit '.rmd' R Markdown, '.tex' latex file (can be generated from R markdown) and '.pdf' file (can be generated from R Markdown). I should knit the '.rmd' file and compile '.tex' file without troubles.

Q1. Conceptual Challenges (30 points)

Select the answers as instructed (Note: each question **may have one or more** correct answers). Each question counts **5 points**. For each question, you **get zero point if any wrong answer is chosen**. If you miss one or more correct answers but all the chosen ones are correct, then you get **2 points**. If all the correct answers are chosen, you get full points.

(1) Let $\{X_i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} p_\theta(x)$ be n random samples (Let X be their population variable). We denote their realizations (or outcomes) to be $\{x_i\}_{i=1}^n$. Select all the WRONG statements.

A. The realizations $\{x_i\}_{i=1}^n$ are deterministic quantities.

B. Though a random sample X_i can fluctuate, its variance must be deterministic and finite.

C. Without the values of realizations, we cannot tell whether a statistic is consistent or not.

D. Without the values of realizations, we cannot give an estimate for the parameter of interest (e.g., θ).

E. The law of large numbers can be applied to both random samples $\{X_i\}_{i=1}^n$ and their realizations $\{x_i\}_{i=1}^n$.

F. The population variable X and the first random sample X_1 are identically distributed.

(2) Unbiasedness and Consistency. Select all the wrong statement:

A. Unbiasedness implies consistency.

B. Consistency implies unbiasedness.

C. Biased estimators can never be consistent.

D. Inconsistent estimators can be unbiased.

E. Let θ be the parameter of interest. If an estimator $\widehat{\theta}_n$ satisfies

$$\sqrt{n}\Big(\frac{\widehat{\theta}_n - \theta}{\sigma}\Big) \xrightarrow{D} N(0,1).$$

Then $\widehat{\theta}_n$ must be consistent.

(3) Law of Large Numbers (LLN) and Central Limit Theorem (CLT). Select all the WRONG statements:

A. Suppose $\{X_i\}_{i=1}^n$ are i.i.d. random samples and $\mathbb{E}X = \mu$. If $\sqrt{n}(\overline{X}_n - \mu) \xrightarrow{D} N(0, \sigma^2)$, then $\overline{X}_n \xrightarrow{P} \mu$.

B. If $\sqrt{n}(\widehat{\theta}_n - \theta) \xrightarrow{D} N(0,1)$, then $\widehat{\theta}_n - \theta \xrightarrow{D} N(0, n^{-1})$.

C. If $\sqrt{n}(\widehat{\theta}_n - \theta) \xrightarrow{D} N(0,1)$, then $\mathbb{P}\left(\theta \in [\widehat{\theta}_n - \frac{z_{\alpha/2}}{\sqrt{n}}, \widehat{\theta}_n + \frac{z_{\alpha/2}}{\sqrt{n}}]\right) \geq 1 - \alpha$ for sufficiently large n, where z_α is the α upper quantile of the standard normal distribution.

(4) Linear Regression and Ordinary Least Squares (OLS). Select all the WRONG statements:

A. In the regression model $Y = \boldsymbol{X}^T\boldsymbol{\beta} + \epsilon$, Y and $\boldsymbol{X}$ are deterministic quantities and ϵ_i is a random noise.

B. If the random samples $\{(Y_i, \boldsymbol{X}_i)\}$ independent follow the linear regression model $Y = \boldsymbol{X}^T\boldsymbol{\beta} + \epsilon$ where ϵ is independent of $\boldsymbol{X}$ and $\epsilon \sim N(0, \sigma^2)$, the OLS estimator of the coefficient vector $\widehat{\boldsymbol{\beta}}$ is unbiased.

C. The OLS estimator of the coefficient vector $\widehat{\boldsymbol{\beta}}$ is always unique.

(5) Assuming the true distribution of the data follows a linear model $Y = \beta_0 + \beta_1 X + \epsilon$. We fit an ordinary least squares regression using this true model on the data, as the number of data points goes to infinity, your estimator will have

A. variance approaching zero.

B. lower variance but not approaching zero.

C. same variance.

D. lower bias approaching zero.

E. lower bias but not approaching zero.

F. same bias.

(6) Select all the WRONG statements:

A. R^2 is only used when we are doing linear regression.

B. R^2 on training data can be negative if fitting is very bad.

C. R^2 on training data is always no smaller than that on testing data.

D. Recruiting more variables in the linear model never hurts R^2 on the training data for the OLS estimator.

E. The higher R^2 is, the better the fitted model is.

Q2. Short Question (15 points)

Prove or Give Counter Examples.

- If $X_n \xrightarrow{P} X$ and $Y_n \xrightarrow{P} Y$, then $X_n + Y_n \xrightarrow{P} X + Y$.
- If $X_n \xrightarrow{D} X$ and $Y_n \xrightarrow{D} Y$, then $X_n + Y_n \xrightarrow{D} X + Y$.
- If $X_n \xrightarrow{D} C$ for some constant C, then $X_n \xrightarrow{P} C$.

Q3. Maximum Likelihood Estimator (MLE) and Asymptotic Normality (20 points)

Maximum likelihood is one of the most fundamental principals in parameter estimation. Suppose we have n i.i.d. random samples $\{X_i\}_{i=1}^n$ that have probability density function $p_\theta(x)$. We are interested in estimating the parameter θ. Denote the correspondent MLE by $\widehat{\theta}_n$. In the lecture, we have known that under some regularity conditions, the MLE enjoys the asymptotic normality

$$\sqrt{n}(\widehat{\theta}_n - \theta) \xrightarrow{D} N(0, \frac{1}{I(\theta)}), \tag{0.1}$$

where

$$I(\theta) := \mathbb{E}_\theta\Big(-\frac{\partial^2}{\partial\theta^2}\log p_\theta(X)\Big) = -\int_{\mathcal{X}}\Big(\frac{\partial^2}{\partial\theta^2}\log p_\theta(x)\Big)p_\theta(x)dx$$

is the Fisher information and $\mathcal{X}$ is the range of X_i.

Part I. Let z_α denote the α upper quantile of standard normal distribution. Prove that

$$C_n = \left[\widehat{\theta}_n - \frac{z_{\alpha/2}}{\sqrt{nI(\widehat{\theta}_n)}}, \widehat{\theta}_n + \frac{z_{\alpha/2}}{\sqrt{nI(\widehat{\theta}_n)}}\right]$$

is a $(1-\alpha)$ asymptotic confidence interval for θ, i.e., $\lim_{n\to\infty} P(\theta \in C_n) = 1 - \alpha$. We assume $I(\theta)$ is a continuous function.
(Hint 1: According to Slutsky's Theorem, if $X_n \xrightarrow{D} X$ and $Y_n \xrightarrow{P} c$, where c is some constant, then $X_nY_n \xrightarrow{D} cX$. Use this result to prove that $\sqrt{nI(\widehat{\theta}_n)}(\widehat{\theta}_n - \theta) \xrightarrow{D} N(0,1)$.)

Part II. Suppose $\{X_i\}_{i=1}^n$ have the following probability density function:

$$p_\theta(x) = (\theta - 1)x^{-\theta} \cdot \mathbb{1}\{x \geq 1\},$$

where $\theta > 1$ is the parameter of interest. Denote this distribution by P_θ and MLE of θ by $\widehat{\theta}_n$.

(a) Derive the MLE $\widehat{\theta}_n$.

(b) From (0.1), we know that the MLE $\widehat{\theta}_n$ is asymptotically normal. Calculate the asymptotic variance in terms of θ.

(c) Derive a 95% asymptotic confidence interval (CI) for the parameter θ.

(d) Use simulations to verify the effectiveness of the CI in (c) when $n = 100$ and $\theta = 2$.
(Hint 1: For the simulation part, we need to figure out how to generate random variables that follow P_θ. Suppose we have obtained the cumulative distribution function $F(x)$ of P_θ, and we generate a random variable U that is uniformly distributed over $[0, 1]$. It is true that $F^{-1}(U) \sim P_\theta$.)
(Hint 2: By effectiveness of the CI, we mean whether the constructed CI will cover the true θ with probability around 95%. To verify the effectiveness of the CI, we can independently generate a large number of datasets (e.g., 10,000 datasets) with each having the sample size n. Calculate CI's for all generated datasets respectively and then summarize the frequency of CI's covering the true θ. If the frequency is around 95%, then we can claim that the constructed CI is effective.)

Q4. Law of Large Numbers and Central Limit Theorem (20 points)

This question helps you to better understand the concepts of convergence in probability and convergence in distribution. In particular, you will visualize the Law of Large Numbers and Central Limit Theorem for a Discrete Distribution:

$$\mathbb{P}(X = 1) = \mathbb{P}(X = -1) = 1/2.$$

Generate $N = 10,000$ datasets, each of which has n data points.
(Hint: Write a function in R that samples from the uniform distribution between 0 and 1 using the built-in `runif` function. If the result is less than 0.5, set it to -1. Otherwise, set it to 1.)
Let $\overline{X}_n^{(i)}$ be the average of i^{th} dataset, $\mu = EX$ and $\sigma^2 = \mathrm{Var}(X)$. We consider $n = \{10, 100, 1000, 10000\}$ for this simulation.
(Hint: You will not need the individual data points from each dataset. Therefore, to save memory, you need only store the $\overline{X}_n^{(i)}$ rather than all the data points. It is highly recommended that you do this to avoid freezing or crashing your computer.)

Plot and intepret the following:

(a) $\log_{10}(n)$ v.s. $\overline{X}_n^{(1)} - \mu$;
(Hint: This plot illustrates how the deviation $\overline{X}_n^{(1)} - \mu$ converges to 0 as n goes to infinity).

(b) Draw $\log_{10}(n)$ v.s. $\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\{|\overline{X}_n^{(i)} - \mu| > \epsilon\}$ for $\epsilon = 0.5, \epsilon = 0.1, \epsilon = 0.05$;
(Hint 1: This plot illustrates the law of large numbers. Please explain why.)
(Hint 2: For some statement S, the indicator function $\mathbb{1}\{S\}$ is defined as $\mathbb{1}\{S\} = 1$ if S is true and $\mathbb{1}\{S\} = 0$ otherwise.)

(c) Draw histograms and Q-Q plots of $\sqrt{n}(\overline{X}_n^{(i)} - \mu)/\sigma$ for N datsets for $n = 10, n = 1,000, n = 10,000$. You may choose your histogram bins or you may let R choose automatically–any meaningful plot will do.
(Hint: This plot illustrates the Central Limit Theorem. Please explain why.)

(d) Generate i.i.d. standard normal $Y_1, \ldots, Y_N$ that are independent to previous random variables. Plot

$$\log_{10}(n) \text{ v.s. } \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\{|\sqrt{n}(\overline{X}_n^{(i)} - \mu)/\sigma - Y_i| > \epsilon\} \text{ for } \epsilon = 0.001.$$

(Hint: This plots illustrates the difference between convergence in probability and convergence in distribution. Explain why this plot shows that $\sqrt{n}(\overline{X}_n^{(i)} - \mu)/\sigma$ does not converge to Y in probability.)

Instructions: This problem tries to familiarize you with `R` programming. You should produce 11 graphics in total – one plot for (a), three plots for (b), six plots for (c), and one plot for (d). You may format and combine the plots together for each part in whichever way you like, but we must be able to clearly read and interpret your results.

Q5. Basic R Programming for Big Data (20 points)

This problem will help you practice coding in `R` under a big data setting. The dataset comes from a dating website "LibimSeti." It consists of two files, `ratings.dat` and `users.Rdata`. The `ratings.dat` contains 3,000,000 ratings of profiles made by LibimSeTi users. It is organized as a comma-separated matrix with 3,000,000 rows and 3 columns. The `users.Rdata` contains the basic information of the users. It has $135,358$ rows and 7 columns. `Readme.txt` contains a detailed description of the data.

(a) Load `ratings.dat` into `R` using the package `bigmemory`, and name the columns by UserID, ProfileID, Rating. To evaluate every profile's score, we use the weighted rank (also used by IMDB):

$$\text{Weighted Rank (WR)} = (v \div (v+m)) \times R + (m \div (v+m)) \times C, \text{ where}$$

$$\begin{array}{rcl} R & = & \text{average rate for the profile} \\ v & = & \text{number of votes for the profile} \\ m & = & \text{4182 (the 250th largest number of ratings for a single profile)} \\ C & = & \text{the mean rate across the whole data.} \end{array}$$

Write an `R` function `weighted.rank(ProfileID)` returning the weighted rank of the profile with its ID as input. Use the functions `which` and `apply` to compute the weighted ranks of all the profiles who were rated by UserID 100. Report your result by plotting a histogram of those scores you obtained. You can also use the `mwhich` and `lapply` functions. [This teaches you how to use bigmemory to load big data and how to write a R function.]

(b) Load `users.Rdata` into `R`. Then you will have a matrix `User` in your working environment. Based on `users.Rdata` and `ratings.dat`, plot the boxplots of ratings from 1) male users coming from New York State and 2) female users coming from California. (Hint: you can use the `unique` or `grep` functions to find out the different ways each state is recorded in the dataset.)

(c) In order to predict a user's rating for a profile, fit a linear regression using the average rating given by the user and the average rating of the profile from all users as predictors. Report your results. Specifically, report the R-squared value and the three coefficients (intercept, coefficient for average rating given by a user and coefficient for average ratings given to a profile). Informally, the model is

(User i's rating on Profile j) $= \theta_1 + \theta_2$(Average Rating given by User i)$+\theta_3$(Average Rating given to Profile j)$+\epsilon_{ij}$

where $\{\theta_1, \theta_2, \theta_3\} \in \mathbb{R}^3$ and ϵ_{ij} is a Gaussian noise term. You are required to 1) run the regression over the entire dataset using the the `R` function `biglm` in the package of `biganalytics`; 2) apply the subsampling technique, which means you do regression over multiple small subsamples respectively and take the mean of coefficients fitted by all subsamples as the aggregated estimand of coefficients. You can refer to Hint 3 below for further instructions on the subsampling technique. You are encouraged to compare the difference between the two methods in terms of the coefficient values, the R-squared value and the CPU time. You might need to recall how to calculate the R-squared value from previous courses.

Hint 1: Simple things become painful under the big data setting. The most time-consuming part is to calculate average ratings involved in the linear model. The most natural way of doing this is to use `mwhich` for each user (profile) to

locate the ratings associated with the given user (profile) and then do the average, but once you run the code you will feel the pain: the progress is very slow. Here we suggest another way of doing this, which we hope can make your life easier. Try to understand the following code for calculation of average ratings, and if you like it, you can copy the following code from the given `R` script file `AveRating.r`. :)

```
N=3000000   # number of rating records
Nu=135359   # maximum of UserID
Np=220970   # maximum of ProfileID
user.rat=rep(0,Nu)  # user.rat[i] denotes the sum of ratings given by user i
user.num=rep(0,Nu)  # user.num[i] denotes the number of ratings given by user i
profile.rat=rep(0,Np)  # profile.rat[i] denotes the sum of ratings given to profile i
profile.num=rep(0,Np)  # user.rat[i] denotes the number of ratings given to profile i
for (i in 1:N){ # In each iteration, we update the four arrays, i.e. user.rat,
    user.num, profile.rat and profile.num, using one rating record.
  user.rat[X[i,'UserID']]=user.rat[X[i,'UserID']]+X[i,'Rating']   # The matrix X here
      comes from the file 'ratings.dat'
  user.num[X[i,'UserID']]=user.num[X[i,'UserID']]+1
  profile.rat[X[i,'ProfileID']]=profile.rat[X[i,'ProfileID']]+X[i,'Rating']
  profile.num[X[i,'ProfileID']]=profile.num[X[i,'ProfileID']]+1
  if (i %% 10000==0) print(i/10000)
}
user.ave=user.rat/user.num
profile.ave=profile.rat/profile.num
X1=X
colnames(X1)=c('UsrAveRat','PrfAveRat','Rat')
X1[,'UsrAveRat']=user.ave[X[,'UserID']]
X1[,'PrfAveRat']=profile.ave[X[,'ProfileID']]  # X1 is the new data matrix we will
    work with in regression.
```

With the code given above, we can finish calculating all the average ratings by only one for-loop over the entire dataset.

Hint 2: Using `biglm` on the entire dataset can take around one hour on a descent laptop. Be patient. Here are several tips for running codes on big data: a) Test you code first on a small dataset; b) In order to know that your codes are still working, you should have your code display the current progress of your program (the percentage of work finished, the number of loops it is working on, etc.) For example, if there is a huge for-loop in your code, print an asterisk('*') after every 10,000 iterations; c) Have your code automatically save your results as an rda file every so often. This allows you to check your results in a different R console while the program is running and retain your results if your R program prematurely terminates; d) You can run R from the command line to avoid using the R Studio interface. This will typically make your code faster.

Hint 3: We first need to figure out what the data matrix $\mathbf{X}_1$ is for our regression task. It is worth noting that $\mathbf{X}_1$ is now different from the matrix $\mathbf{X}$ that we load from `ratings.dat`. $\mathbf{X}_1$ is of the same dimensions as $\mathbf{X}$, but to get $\mathbf{X}_1$, we need to replace the ProfileID's and UserID's in $\mathbf{X}$ with the associated average ratings as explained in the model. (See the code given in Hint 1 for reference.) The subsampling technique essentially means that we randomly choose a small number rows of $\mathbf{X}_1$, whose row indices form a set I, and do regression only over X_{1_I}, which is a sub-matrix of $\mathbf{X}_1$ that consists of only rows I. Since we use only a small number of samples, the regression coefficients we get will not be accurate. A natural way to solve this problem is to apply this subsampling trick for multiple times and average the coefficients we get. In this way, we expect the result to be more stable and accurate.

Q6. A Simple Linear Regression (25 points)

Read in the `housingprice.csv` file using `read.csv()` function.

(a) Rank the zipcodes by their average housing prices. What are the top 3 zipcodes whose average housing prices are most expensive? Create three boxplots of housing prices for these 3 zipcodes respectively.

(Hint: First convert the `zipcode` column into factors. Then use `tapply()` and `sort()` functionals to compute the result.)

(b) Visualize the relationship between `sqft_living` and `housing price` by creating a scatter plot.

The following questions continue from above questions. Load the training data `train.data.csv` and testing data `test.data.csv`. We'll build our regression model on the training data and evaluate the model on the testing data.

(c) Build a linear model on the training data using lm() by regessing the housing price on these variables: `bedrooms`, `bathrooms`, `sqft_living`, and `sqft_lot`. What's the R^2 of the model on training data? What's the R^2 on testing data?

(d) Add `zipcode` in your linear model. What's the R^2 of the new model on the training data and testing data respectively?

(e) The image below is Bill Gates' house. Load the file `fancyhouse.csv` to obtain the features of the house. Guess the price of his house using your linear model. Do you think the predicted price is reasonable?

Figure 1: Image fom Wikipedia Commons

(f) Suppose we have a linear regression problem with n training samples and d covariates. If $n > d + 1$, show that adding another covariate in the model never hurts R^2 over the training data.

(Hint: By definition, the ordinary least squares (OLS) estimator $\widehat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta} \in R^d} \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2$, where $\boldsymbol{Y}$ is the response and $\boldsymbol{X}$ is the design matrix. Denote the new design matrix with the additional covariate by $\boldsymbol{X}_1$. Then the OLS estimator for the new regression problem $\widehat{\boldsymbol{\beta}}_1 = \text{argmin}_{\boldsymbol{\beta} \in R^{d+1}} \|\boldsymbol{Y} - \boldsymbol{X}_1\boldsymbol{\beta}\|_2^2$. Compare $\|\boldsymbol{Y} - \boldsymbol{X}_1\widehat{\boldsymbol{\beta}}_1\|_2^2$ and $\|\boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|_2^2$).

Q7. Feature Engineering (20 points)

Let's continue to improve the linear model we have. Instead of throwing only the raw data into the statistical model, we might want to use our intuition and domain expertise to extract more meaningful features from the raw data. This step is called feature engineering. Using meaningful features in the model is often crucial for successful data analysis.

(a) Add another variable by multiplying the number of bedrooms by the number of bathrooms, which describes the combined benefit of having more bedrooms and bathrooms. Add this variable to the linear model we have in Question 6 (d). What's the R^2 of the new model on the training data and testing data respectively?

(Hint: You don't have to create a new column in the data frame. Try this trick in `lm()`: `lm( y ~ x1 + x2 + x1 * x2, data = your.data))`

(b) Using R^2 on the testing data as the metric for evaluating your model, propose another feature engineering that further improves the model you have in Question 7 (a).

(c) Polynomial regression is a general technique that allows you to add nonlinear features in your statistical model. Based on the model we have in Question 6 (d), add polynomial terms of the bedrooms and bathrooms variables of degrees 2 and 3 (no cross terms) in your model. Find out the R^2 of the new model on training data and testing data.

(Hint: You don't have to create a new data frame. Try this trick in `lm()`. `lm( y ~ poly(x1, degree) + x2 + x3, data = your.data))`

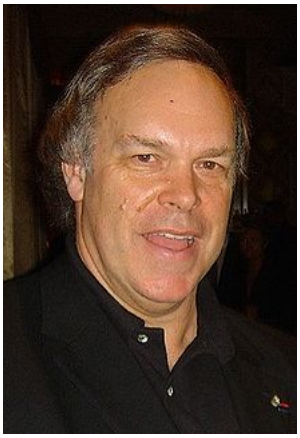
Q8. Wine Pricing (20 points)

Wine pricing is a challenging job. Though wine is produced every year in a similar way, its price and quality varies between years. Since all the wines are meant to be aged, it is very hard to tell if wine will be good or not on the market in the future. Traditionally, wine companies solely rely on expert tasters to assess wine quality. However, in March 1990, Orley Ashenfelter, an economics professor in Princeton, claimed he could predict wine quality without actually tasting the wine; his method was surprisingly simple: just use ordinary linear regression! In fact, his method is so simple that Robert Parker, one of the most famous wine experts, once commented:

"Arshenfelter is an absolute total sham".

We would see in this problem whether Ashenfelter is a sham or not through real data analytics.

Orley Arshenfelter



Robert Parker

In this study, we will use a dataset `wine.csv`. We start with a description of the covariates of this dataset . For wine in each year from 1952 to 1978, we collect the logarithm of its price in the $1990-1991$ wine auction, average growing season temperature (`AGST`), winter rain amount, harvest rain amount, age of wine and etc.

Part I. Preliminary Analysis

Load `wine.csv`. Give four scatter plots of `Price` v.s. `AGST`, `Price` v.s. `WinterRain`, `Price` v.s. `HarvestRain` and `Price` v.s. `Age`. `Price` should be on the y-axis. Which variable do you think is most correlated with `Price`? Justify your observation by calculating the Pearson's correlation.

Part II. Marginal Regression Analysis.

Fit a marginal regression model: `Price` $\sim$ `AGST`. Report the fitted coefficient values and R^2.

Part III. Multiple Regression Analysis.

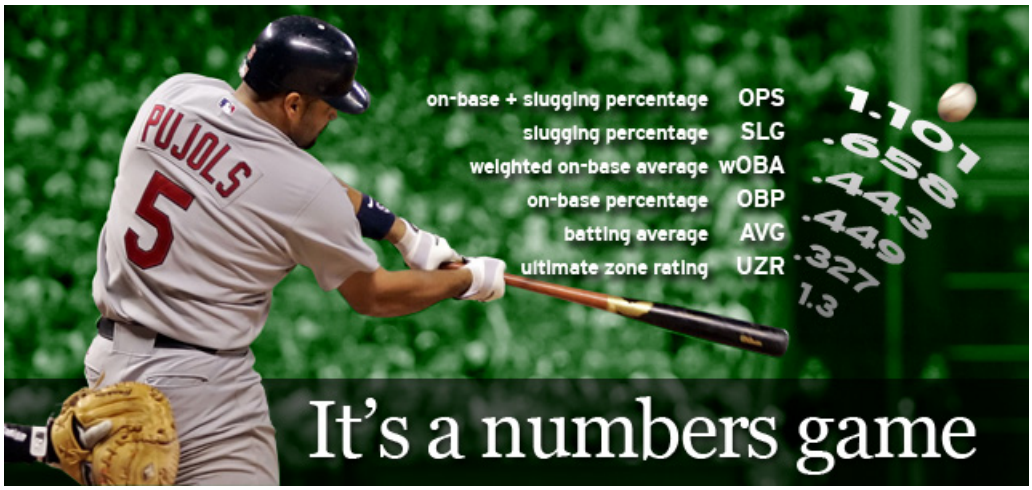
Add `HarvestRain`, `Age`, `WinterRain` and `FrancePop` to your model in Part II one by one. Report how R^2 on the training data changes as we add more and more covariates. For each model, also report your R^2 over the testing data provided in `winetest.csv`. Which model should we choose based on R^2? According to Prof. Ashenfelter's system in 1990, heavy rains in the winter followed by a hot summer improve wine quality, while rainfall before the harvest damages it. Is your model consistent with Prof. Ashenfelter's finding?

The ending of the story is that regression beats Parker! Parker thought the wine in 1986 was "very good to sometimes exceptional", while Ashenfelter's model predicted that the wine in 1986 was mediocre and the wine produced in 1989 instead would be "the wine of the century". It turned out that the 1989 wine was sold for more than twice the price of 1986. Here we see the power of simple regression in wine price prediction; it can sometimes beat human experts.

Note: More details about this story can be found in a fascinating article from 1990 New York Times:

`http://www.nytimes.com/1990/03/04/us/wine-equation-puts-some-noses-out-of-joint.html`

Q9. Moneyball: The Analytics Edge in Sports (30 points)



Baseball is one of the most symbolic sports in the US. Back into the 20th century, baseball teams relied on respected and experienced scouts to recruit players. However, as statistical analytics were introduced to this game, the rule changed. Michael Lewis's bestselling book *Moneyball: The Art of Winning an Unfair Game* tells the story of how Billy Beane, the general manager of the Oakland Athletics in 1998, effectively used statistical analytics to turn this losing team into a winning team with very stringent payroll budget. His great success was due to the discovery of significant features that create runs through statistical analysis. In particular, Billy found that on-base percentage (`OBP`) and slugging percentage (`SLG`) have much better performance in measuring offensive capability than on batting

average (BA), which has always been baseball's most famous and well-published statistic. In this problem, we would use the real data to verify Billy's claim, and see how regression analysis can reshape decision making procedures in baseball team management and lead to dramatic enhancement in team performance.

First of all, let's familiarize ourselves with some key terminologies.

1. Plate appearances: It is counted every time a player comes to bat regardless of the outcome of that time at the plate

2. At bat: It is counted only the times a player gets a hit or make an out.

3. Batting average (BA): It represents the percentage of at bats that result in hits for a particular baseball player.

4. On-base percentage (OBP). It is a measure of the number of times a player gets on base by hit, walk, or hit by pitch, expressed as a percentage of his total number of plate appearances.

5. Slugging percentage (SLG): It is calculated as total bases divided by at bats.

In this problem, we would use the dataset *baseball.csv* that has statistics of performance of all Major League Baseball (MLB) teams from 1962 to 2012. We list below meanings of important features in the dataset. You are encouraged to google the detailed information if you are not pretty sure about the terminologies here.

1. RS: runs scored

2. RA: runs allowed

3. W: the number of winning games

4. OBP: on-base percentage

5. SLG: slugging percentage

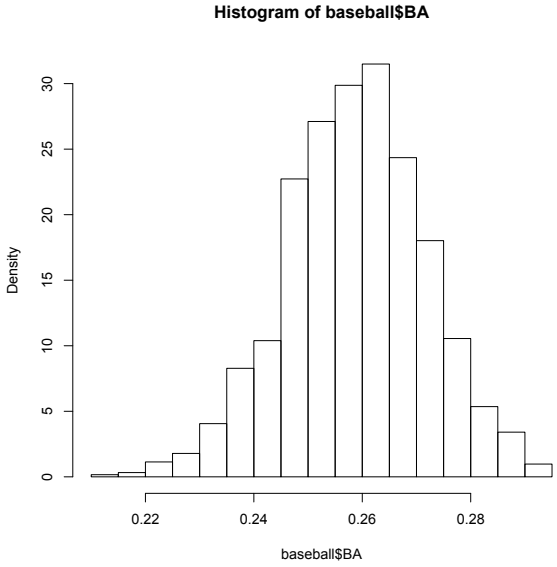
6. BA: batting average

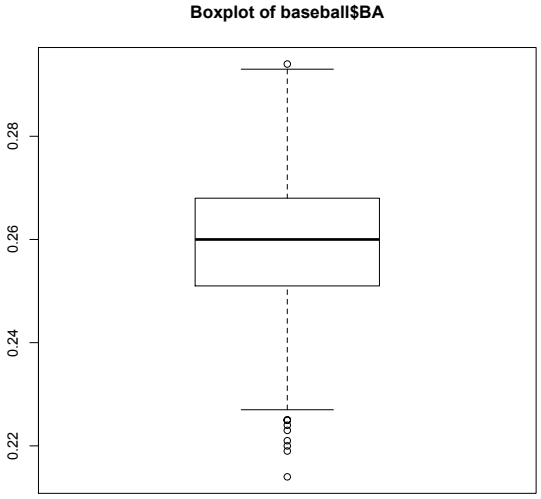
7. Playoffs: whether making a playoff, 1 means yes and 0 means no

8. OOBP: opponent on-base percentage

9. OSLG: opponent slugging percentage

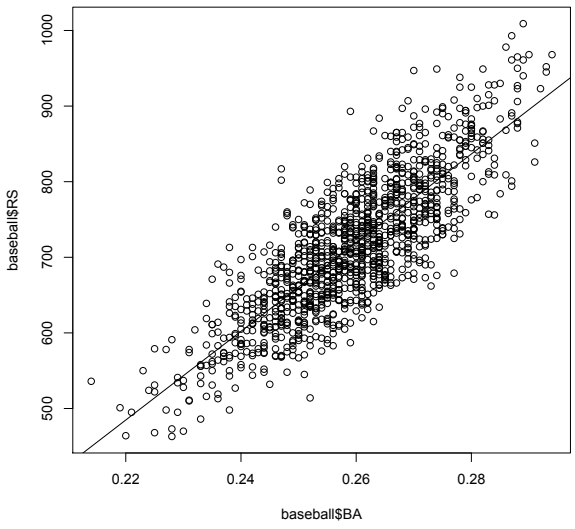
Part I. Preliminary Analysis. Load the dataset *baseball.csv*. Plot histograms and boxplots for OBP, SLG, BA. Also report the mean and median for these three statistics. The purpose of doing this is to get an initial idea of how the data are distributed and detect potential outliers. Below we present the histogram and boxplot for BA for your reference.

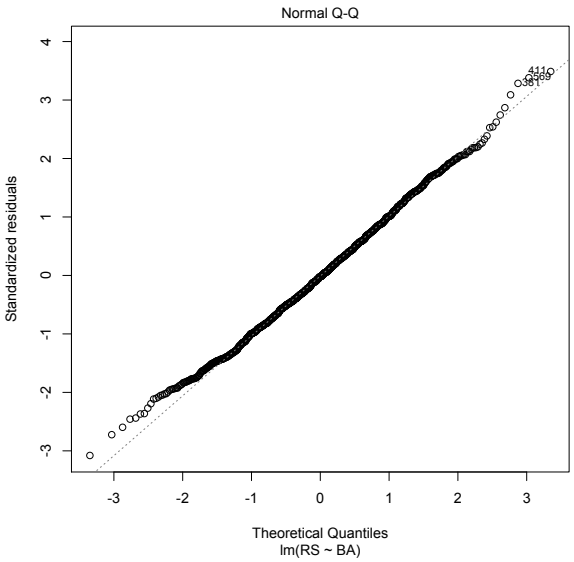




The mean and median of BA are 0.259 and 0.26, meaning that the distribution is not skewed at all. You can also verify this from the boxplot and histogram. Analyze the other two quantities similarly.

Part II. Marginal Regression Analysis. As characterized before, BA, OBP and SLG are important quantities in determining the team performance, which is usually quantified by RS. Marginally regress RS on BA, OBP, SLG respectively and see how they are correlated. Give the scatter plot of the data and the fitted line. Report the coefficient values and R^2. Also give the QQ-plot of the fitted residuals, which is to verify that the residuals are not skewly distributed and the model is reasonable. Below we show the regression result using the model $\texttt{RS} \sim \texttt{BA}$. The intercept and slope are -805.51 and 5864.84 respectively, and $R^2 = .6839$.





Traditionally, BA is thought to be most responsible for RS since it removes contribution of lucky scoring like walk or hit by pitch and honestly reflects how well the batter hits the ball. Compare R^2 you obtained for BA, SLG and OBP. Is that consistent with the intuition?

Part III. Multiple Regression Analysis.

The plots above only tell us part of the story. We now need to examine a regression output to see how BA, SLG and OBP relate together in predicting our target variable RS. Fit the model $\texttt{RS} \sim \texttt{BA} + \texttt{SLG} + \texttt{OBP}$. Report the estimated coefficients for these covariates (along with their significance). Check the model by giving QQ plots of the residuals. Is the fitting result consistent with that in Part II, especially the fitted coefficient of BA? Summarize and justify your findings. Also fit the model $\texttt{RS} \sim \texttt{BA} + \texttt{SLG}$. Compare R^2 of the two models. Which model do you prefer?

Part IV. Back to 2001 and Reshape the Baseball World.

Through the marginal regression analysis in Part II, we witness how OBP, SLG and BA have a strong positive correlation with RS. However, of those three statistics, the traditional and most often used BA had the lowest correlation. Furthermore, when we examined how these variables work together in predicting RS in Part III, we noticed that OBP and SLG alone could do just as good a job predicting RS than a model that included BA. This suggests that team managers would best work on focusing on his team's OBP and SLG over BA to improve his team's run output for the season. Also in determining the payroll distribution, batters with higher OBP should be given higher salaries. This is the reason why Billy Beans could hire excellent players with low salaries since at that time OBP is not valued!

Suppose we were data analytics for Oakland Athletics back in 2001, and our job was to predict how many games we would win in 2002. Based on historical players statistics, we estimated that in 2002 $\texttt{OBP} = .349$, $\texttt{SLG} = .430$, $\texttt{OOBP} = .307$ and $\texttt{OSLG} = .373$ for Oakland Athletics. Add a column $\texttt{RD} = \texttt{RS} - \texttt{RA}$. Fit the models $\texttt{W} \sim \texttt{RD}$, $\texttt{RS} \sim \texttt{OBP} + \texttt{SLG}$ and $\texttt{RA} \sim \texttt{OOBP} + \texttt{OSLG}$ using the data before (not including) 2002, and try to predict how many games Oakland would win in 2002 by combining these three models. Check whether your prediction is accurate by looking at our dataset.

Acknowledgement

The origin of the dataset housingprice.csv is from the Coursera open course Machine Learning Foundations: A Case Study Approach by Prof. Carlos Guestrin and Prof. Emily Fox. The open course also inspired the linear

regression part of this assignment.

We greatly appreciate your insightful questions after class, which give lots of inspiration in creating Q1 and Q2. We also appreciate the MIT online course *Analytics in Edge*, which provides interesting stories on power of statistical analytics in real applications and the relevant datasets.