STAT 403 Final Project: Used Car Analysis

Chenxi Di, Nan Tang, Ying Wang, Yining Wang

June 14, 2019

Contents

1 Introduction

Cars are indispensable means of transportation that greatly facilitate people's life. When deciding which make and model to buy, it is common case that different retail stores sell at different prices under the similar manufacturer suggested retail prices. More variations on prices are seen on used cars, since many more factors can potentially affect the listed sales price. Our project focus on all available used car listed on TrueCar.com September 24th 2017. It contains 1.2M listings of used cars scraped from the website. We believe this data set could provide us insights about how used car prices are determined. The results from our project can help both sellers and buyers to optimize the prices set for used car transactions.

Our data set contains eight original variables, including three numerical variables. Price indicates the listed price of used car. Year represents a consistent model year, in which it records the starting year that the car is legally go on sale. All cars in the dataset have model year ranges from 1997 to 2017. Mileage is the numerical data deriving from car's

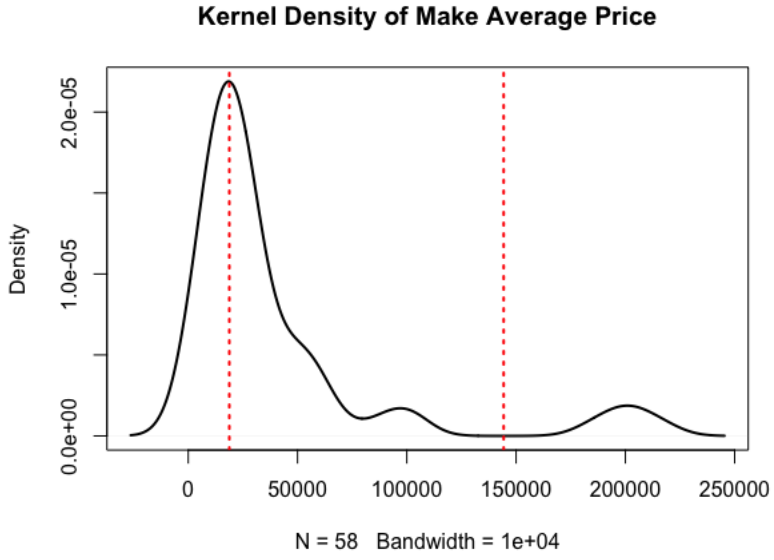
odometer reading. We also have City which is the current location of the cars, State the state where these cars placed in, Vin the unique identification of each car, serving as a vehicle's fingerprint, and Make and Model of the vehicle.

2 Methods

One goal is to build a model that helps to determine whether the used car price for a particular car provided on the used car website is reasonable based on the information provided in the data. We wish to explore the possible relationship between used car price and age/year, mileage, make and dealer location (city, state) first to estimate the plausible price. However, the large 1.2 million data points and so many categories render us unable to build one single model. In the regression modeling process, we turned to only focus on the age, mileage and make to predict the price. We compare the predicted price with the actual price. If they are similar, then the price is reasonable. We aim to build regression model in each 'brand' group. The influences of other remaining parameters on price of the used cars are investigated in other sections.

2.1 Factor Classify

The dataset consists of fifty car brands, of which covers all mainstream car make. Based on empirical experience and through common sources of information, we note that car brand has a heavy weight when dealer pricing the vehicle. Even note that strong correlation exists between brand and price, it's not practical to throw all fifty categorical levels into the mathematical model and come up with fifty different regressions. Therefore, we use **kernel density estimation on average price** of each car brand to determine clusters



By calculation, the lowest density occurs at price 144313, with density less than 10^{-6}, which is a perfect division point of this multimodal distribution: car-make that has an average price higher than 144313 is defined as high-end class. In the low end cluster, large variance still exists, therefore, we redivide evenly on this cluster, and classify car-make with average price lower than 18795 as low-end, price between 18795 and 144313 as mid-range class. Some typical car brands in high-end group are Ferrari, Porsche. Representative mid-range cars are Audi, Acura. Buick, Honda are representative of the low-end cars.

2.2 Distribution of Factors

One of the purposes of this research is to figure out whether the distribution of model-year and mileage are identical for all car-makes. As we mentioned in previous part, it's unfeasible to compare all brands. We will compare the distributions for each class of the vehicle, for low-end, mid-range and high-end. If three classes of vehicle have similar mean and median on respectively year and mileage, then we can conclude that all car-makes are not different on year and mileage. **Permutation test** will be applied to test if differences exist on each of the distributions. For each iteration, we took sample size of 1000 from the data-pull, and assigned into two sample groups based on the ratio of their data-set size. And after that, calculate p-values of true sample difference on mean and median based on bootstrapped difference on mean and median.

2.3 Model Selection

To explore the regression model, we first apply Bayesian information criterion: The BIC values are listed as follows:

$$\text{BIC for lm(Price} \sim \text{Year + Mileage): } 26658558$$

$$\text{BIC for lm(Price} \sim \text{Year): } 26663474$$

$$\text{BIC for lm(Price} \sim \text{Mileage): } 26829432$$

Smaller BIC values refer to a better model to some degree. Therefore, we consider predicting the price using both of the car year and mileage first. To verify that works, we also employ the bootstrap method for confidence interval of correlation coefficients between year, mileage and price:

	5%	95%
Price and Year	0.4041356	0.4097733
Price and Mileage	-0.4241377	-0.1297690
Year and Mileage	-0.7671856	-0.2491677

The 95% bootstrapped confidence interval [0.4041356, 0.4097733] indicates that the association between price and year may be medium and positive. Furthermore, the interval [-0.4241377, -0.1297690] indicates the association between price and mileage may be

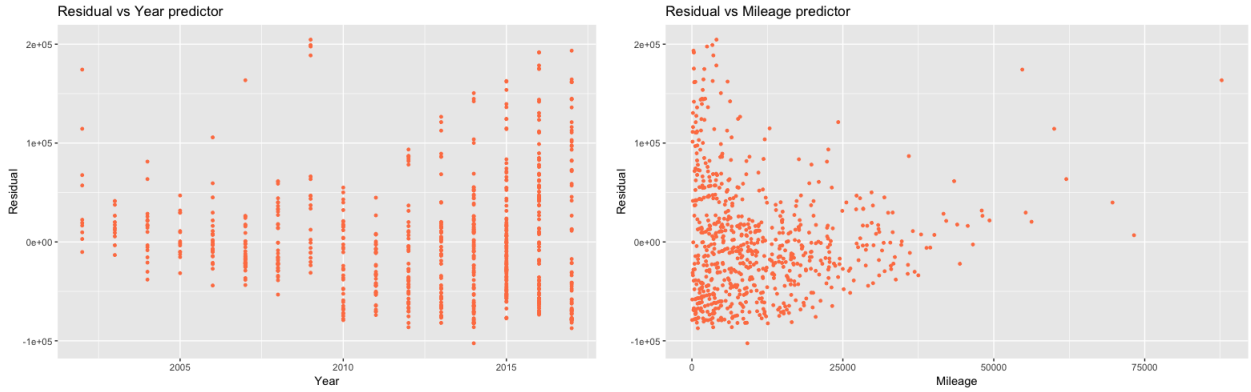
slightly medium and negative. Although the correlation between the two possible predictors: year, mileage are also medium, the absolute value is less than 0.8 which implies the multi-collinearity problem has not been present. Therefore, we insist on building the regression model for price based on the two predictors: year and mileage.
We used linear regression model fit in each car brand category first to have a glance at the general regression performance. To observe each regression coefficients deeply, we applied bootstrapped method to get the confidence interval for each regression coefficients. The detailed results are listed in the Result Section.
On the other hand, we noticed that the linear regression model has less reasonable performance, we chose to look at other non-parametric regression methods, eg. Multivariate Kernel Regression. To identify the optimal bandwidth, H, we used the R package named 'np.package'. The optimal bandwidth is chosen by least squares cross-validation method for the local constant estimator:

$$\hat{m}(x; 0, \mathbf{H}) = \sum_{i=1}^{n} \frac{K_{\mathbf{H}}(\mathbf{x} - \mathbf{X}_i)}{\sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{x} - \mathbf{X}_i)} Y_i = \sum_{i=1}^{n} \mathbf{W}_i^0(\mathbf{x}) Y_i$$

where $\mathbf{X}_i \in \mathbf{R}^2$ in this case, $\mathbf{H}$ is the bandwidth for the two predictors. (The least squares cross-validation is referred by minimizing the integrated square error between the estimated distribution and the actual one).

2.4 Re-sampling Approach

When bootstrapping the confidence interval for the linear regression coefficients, we considered two alternatives: wild bootstrap or residual bootstrap. To determine which approaches may yield a better model, we made the residuals vs. predictor plot:

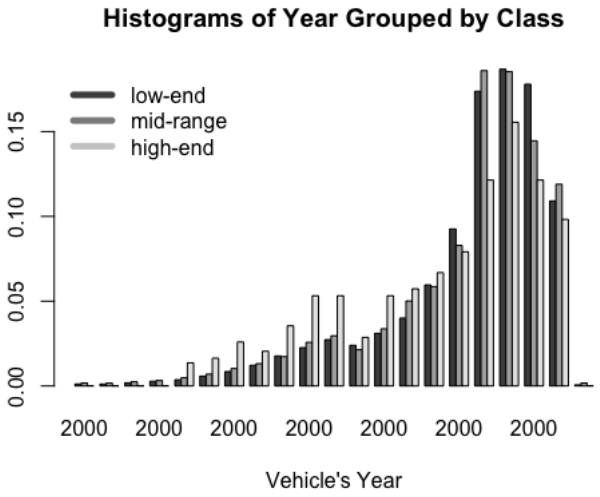


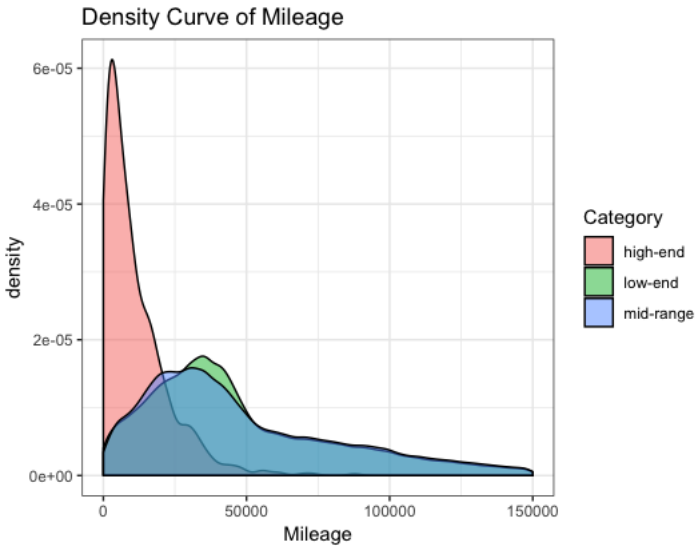
Noticed that the variation of residuals depends on the covariates, we decided to use the wild bootstrap method for the linear regression coefficients. The benefit of using bootstrap resampling method sits in giving a more reliable confidence interval for the predicted used car price. Instead of using the normal based standard error of regression coefficients carried by the R summary function() itself, the bootstrap method focus more on the data without normal assumption.

3 Result

3.1 Distribution Differences

We made a density plot for mileage and histogram for year, considered mileage is continuous and year is discrete data.





We may perceive from both graphs that high-end cars have quite distinguished distribution on both mileage and year compare to the other two classes. They are much denser on early model-year and low mileage. Therefore, we only test distributions difference of year and mileage on low-end and mid-range.

	Mileage	Year
Mean	0.508	0.447
Median	0.866	0.076

P-values that come out from permutation test are all greater than 0.05. Therefore, under the test size 0.05, we cannot reject the statement that distributions of car model year and mileage from low-end and mid-range classes are respectively identical.

3.2 Linear Regression and Kernel Regression in High-End

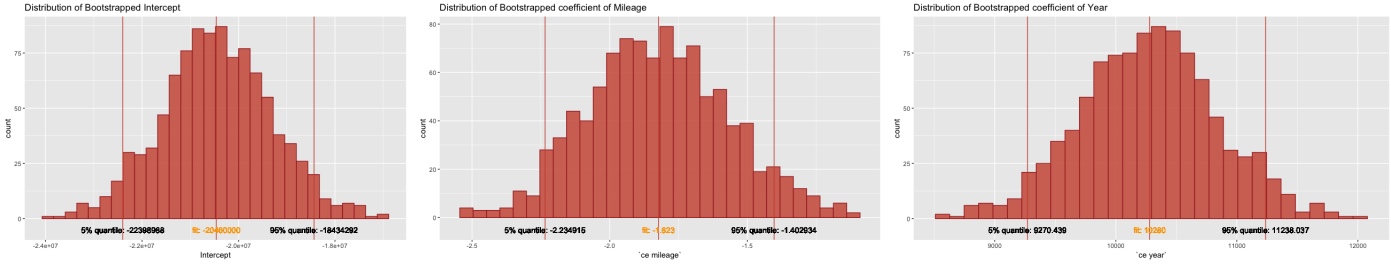
Using the linear regression fit for price based on the predictor: year and mileage gives us the following coefficients:

```
Coefficients:
(Intercept)          Year       Mileage
 -2.046e+07     1.028e+04    -1.823e+00
```

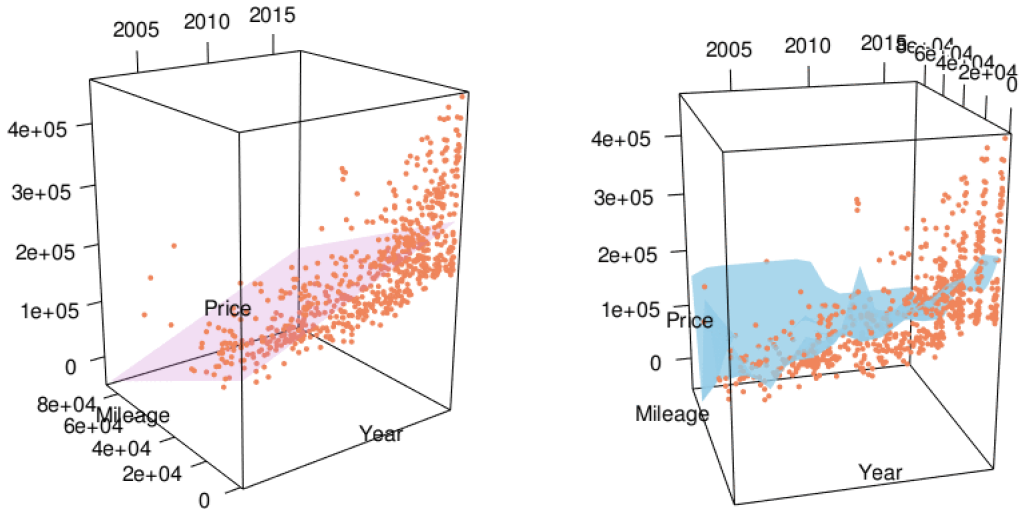
The bootstrapped confidence intervals for each coefficients are:

	5%	95%
Interval	-22449511	-18433568
Year	9269.249	11263.292
Mileage	-2.241013	-1.386211

The bootstrap sampling distribution for these parameters are:



We utilized the 3D plot to show how the linear regression model fits our data (the pink one). The linear regression model seems to have a good prediction when the car is old and mileage is high. However, at the corner of the 3D plot (the intersection of newer and lower mileage), the predicted price deviates from the actual one a lot. The Residual standard error we got in this fit is 58390(probably because the price of used high-end cars are already too high). The R-squared is 0.4722 which indicates the linear regression may not be a very reasonable model in this subgroup.



Applying the Multivariate Kernel Regression and use the least square cross validation gives us the following optimal bandwidth for each predictor: the Bandwidth for year is 0.2060929 while the bandwidth for mileage is 3388.956. We also utilized the 3D plot to show how the kernel regression model fits our data (the blue one). Now, the new regression model seems to give us a better fit since it follows the actual data more neatly. However, we still noticed some large residuals near the intersection of new age and low mileage. The R squared is 0.5506327 which suggests the Multivariate Kernel Regression might fit the actual data more closely compared to the simple linear regression.

The reason behind the phenomenon that both of the linear regression and kernel regression fails when the car is new and with low mileage might be in the high-end car categories such as Ferrari and Porsche, their prices will be very hard to predict due to the large gap between each brand.

4 Discussion

In the model selection procedure, we found the best model for predicting price is when both of the year and mileage are included. In the regression model procedure, we tried to apply linear regression method first and used wild bootstrap method to get the confidence interval for the linear regression coefficient without normality assumption. To improve our model, we also tried multivariate kernel regression technique where we ended up having R squared as 0.55 which implies that the model explains half of the variability of the response data around its mean. However, we already noticed that both of our models are likely to fail when it comes to relatively new and low mileage used cars in the 'high-end brand' groups due to the large price gap between this high-end cars. Hence, we would focus more on refining the current model, especially minimizing the large residuals in the certain scenario in the future. It is also noticeable that during the model selection procedure, the two predictors: year and mileage, already have some medium association. We will therefore look into more used car resource and incorporate new variables into our model to improve its performance. In addition, it will also be valuable to examine more yearly and monthly data to investigate seasonal/annual used car sale price pattern.

5 References/Bibliography

Regression Analysis: How Do I Interpret R-squared and Assess the Goodness-of-Fit?
https://blog.minitab.com/blog/adventures-in-statistics-2/
regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit

Package 'np' https://cran.r-project.org/web/packages/np/np.pdf

5.1 Kernel regression with mixed multivariate data
https://bookdown.org/egarpor/NP-UC3M/kre-ii-multmix.html

Likelihood Cross-Validation Versus Least Squares CrossValidation for Choosing the
Smoothing Parameter in Kernel Home-Range Analysis, JON S. HORNE, EDWARD
https://www.webpages.uidaho.edu/population_ecology/Horne\%20&\%20Garton\%202006\%20JWM.p

6 Appendix

(Include only r-codes)

Classify Factors

```
make_rank_dt <- read_csv('ordered_grouped_make.csv')
library(pastecs)
# use kde make average price, split by local minimum
# to reduce car make category's correlation to price
make_avg_den <- density(make_rank_dt$average, bw=8000)
ts_y<-ts(make_avg_den$y)
tp<-turnpoints(ts_y)
x_tp <- make_avg_den$x[tp$tppos]
y_tp <- make_avg_den$y[tp$tppos]
split_pt2 <- x_tp[which.min(y_tp)]
split_pt1 <- x_tp[which.max(y_tp)]

# now we have two split points
plot(density(make_rank_dt$average, bw=10000), lwd=2,
     main='Kernel Density of Make Average Price')
abline(v=c(split_pt1, split_pt2), lty=3, lwd=2, col='red')
make_category <- rep(NA, nrow(make_rank_dt))
for (ii in 1:nrow(make_rank_dt)) {
  avg <- make_rank_dt$average[ii]
  if (avg <= split_pt1) {
    make_category[ii] <- 'low-end'
  } else if (avg > split_pt1 & avg <= split_pt2) {
    make_category[ii] <- 'mid-range'
  } else {
    make_category[ii] <- 'high-end'
  }
}
car_make_category <- data.frame(Make=make_rank_dt$Make, Category=make_category)
write.csv(car_make_category, 'make_category.csv')

# updata car dataset
temp <- rep(NA, nrow(car_dt))
for (ii in 1:nrow(car_dt)) {
  temp_make <- as.character(car_dt$Make[ii])
  temp[ii] <- as.character(car_make_category[car_make_category$Make==temp_make,2])
}
new_tc <- data.frame(car_dt, Category=as.factor(temp))
write.csv(new_tc, 'new_tc.csv')
```

Visualizations and Permutation Test

```
# remove outlier
remove_outliers <- function(x, na.rm = TRUE) {
  qnt <- quantile(x, probs=c(.25, .75), na.rm = na.rm)
  H <- 1.5 * IQR(x, na.rm = na.rm)
  y <- x
  y[x < (qnt[1] - H)] <- NA
  y[x > (qnt[2] + H)] <- NA
  y
}
low_end_dt <- new_tc %>%
  filter(Category == "low-end")
mid_range_dt <- new_tc %>%
  filter(Category == "mid-range")
high_end_dt <- new_tc %>%
  filter(Category == "high-end")

# tufteboxplot - mileage
# low-end
theme_set(theme_tufte())
g_le <- ggplot(data=low_end_dt)
g_le_ml_box <- g_le + geom_tufteboxplot(aes(x=Make, y=Mileage)) +
  ylim(0, 300000) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  theme(axis.title.x=element_blank(),
        axis.ticks.x=element_blank()) +
  labs(title="Low-end",
       y="Mileage")
# mid-range
g_md <- ggplot(data=mid_range_dt)
g_md_ml_box <- g_md + geom_tufteboxplot(aes(x=Make, y=Mileage)) +
  ylim(0, 300000) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  theme(axis.title.x=element_blank(),
        axis.ticks.x=element_blank()) +
  labs(title="Mid-range",
       caption="Source: TrueCar",
       y="Mileage")
# high-end
g_he <- ggplot(data=high_end_dt)
g_he_ml_box <- g_he + geom_tufteboxplot(aes(x=Make, y=Mileage)) +
  ylim(0, 300000) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  theme(axis.title.x=element_blank(),
```

```
        axis.ticks.x=element_blank()) +
  labs(title="High-end",
       y="Mileage")
grid.arrange(g_le_ml_box, g_he_ml_box, g_md_ml_box,
             widths = c(3,3,2),
             layout_matrix = rbind(c(1, 1, 2),c(3, 3, 3)))

# histogram - year
p <- multhist(list(low_end_dt$Year, mid_range_dt$Year, high_end_dt$Year), probability=T,
              main='Histograms of Year Grouped by Class', xlab='Vehicle\'s Year')
legend('topleft', c('low-end', 'mid-range', 'high-end'),
        col=c('gray30', 'gray55', 'gray80'),lwd=5, cex = 1, bty='n')

# mileage density curve
theme_set(theme_bw())
g_ml <- ggplot(data=new_tc, aes(Mileage))
g_ml_den <- g_ml + geom_density(aes(fill=Category), alpha=0.5) +
  xlim(0, 150000) +
  labs(title="Density Curve of Mileage")

# permutation test - mileage
diff_median <- abs(median(low_end_dt$Mileage) - median(mid_range_dt$Mileage))
diff_mean <- abs(mean(low_end_dt$Mileage) - mean(mid_range_dt$Mileage))
data_pull <- c(low_end_dt$Mileage, mid_range_dt$Mileage)
n_total <- length(low_end_dt$Mileage) + length(mid_range_dt$Mileage)
low_rt <- length(low_end_dt$Mileage) / n_total
mid_rt <- length(mid_range_dt$Mileage) / n_total

B = 1000
per_diff_means <- rep(NA, B)
per_diff_medians <- rep(NA, B)
for (ii in 1:B) {
  sp_index <- sample(n_total, 1000, replace=F)
  sp_dt <- data_pull[sp_index]
  sp_dt1 <- sp_dt[1:round(1000*low_rt)]
  sp_dt2 <- sp_dt[(round(1000*low_rt)+1):1000]
  per_diff_means[ii] <- mean(sp_dt1) - mean(sp_dt2)
  per_diff_medians[ii] <- median(sp_dt1) - median(sp_dt2)
}
med_p <- length(which(per_diff_medians>diff_median | per_diff_medians < -diff_median))/B
mean_p <- length(which(per_diff_means>diff_mean | per_diff_means < -diff_mean))/B

# permutation test - mean
diff_median <- abs(median(low_end_dt$Year) - median(mid_range_dt$Year))
diff_mean <- abs(mean(low_end_dt$Year) - mean(mid_range_dt$Year))
```

```
data_pull <- c(low_end_dt$Year, mid_range_dt$Year)
n_total <- length(low_end_dt$Year) + length(mid_range_dt$Year)
low_rt <- length(low_end_dt$Year) / n_total
mid_rt <- length(mid_range_dt$Year) / n_total

B = 1000
per_diff_means <- rep(NA, B)
per_diff_medians <- rep(NA, B)
for (ii in 1:B) {
  sp_index <- sample(n_total, 1000, replace=F)
  sp_dt <- data_pull[sp_index]
  sp_dt1 <- sp_dt[1:round(1000*low_rt)]
  sp_dt2 <- sp_dt[(round(1000*low_rt)+1):1000]
  per_diff_means[ii] <- mean(sp_dt1) - mean(sp_dt2)
  per_diff_medians[ii] <- median(sp_dt1) - median(sp_dt2)
}
med_p <- length(which(per_diff_medians>diff_median | per_diff_medians < -diff_median))/B
mean_p <- length(which(per_diff_means>diff_mean | per_diff_means < -diff_mean))/B
```

Choosing Re-sampling Method

```
> library(readr)
> library(dplyr)
> library(ggplot2)
> tc_origin <- read_csv("tc.csv")
> unique(tc_origin$Make)
> View(tc_origin)
> library(np)
> library(rgl)
> library(magick)
> tc <- read_csv("new_tc.csv")
> data1 <- tc %>%
  select(Price, Year,Mileage)
> B = 1000
> data1_cor_BT = matrix(NA, nrow=B, ncol=3) # to compute two correlation
> for(i_BT in 1:B){
  w = sample(n,n,replace=T)
  data1_BT = data1[w,]
  data1_cor_BT[i_BT, 1] = cor(data1_BT)[1,2] ## price vs year
  data1_cor_BT[i_BT, 2] = cor(data1_BT)[1,3]  ## price vs mileage
  data1_cor_BT[i_BT, 3] = cor(data1_BT)[2,3]  ## year vs mileage
}
> colnames(data1_cor_BT) = c("price vs year", "price vs mileage", "year vs mileage")
> head(data1_cor_BT)
> quantile(data1_cor_BT[, "price vs year"], c(0.05,0.950))
```

```
> quantile(data1_cor_BT[, "price vs mileage"], c(0.05,0.950))
> quantile(data1_cor_BT[, "year vs mileage"], c(0.05,0.950))
> fit = lm(Price ~ Year + Mileage, data = temp3)
> predicted = predict(fit)
> Year = temp3$Year
> Mileage = temp3$Mileage
> res = price - predicted
> df = data.frame("Year" = Year, "Mileage" = Mileage, "Residual" = res)
> ggplot(df, aes(Mileage, Residual)) +
  geom_point(colour = "coral", size = 1) +
   labs(title = "Residual vs Mileage predictor")
> ggplot(df, aes(Year, Residual)) +
  geom_point(colour = "coral", size = 1) +
   labs(title = "Residual vs Year predictor")
```

Linear Regression and Regression Coefficients Bootstrap

```
> n3= nrow(temp3)
> y_predict3 = predict(fitlinearl3)
> B = 1000
> coeff_BT_wild3 = matrix(NA, nrow=B, ncol=3)
> for(i_BT in 1:B){
  y_bt = y_predict3+rnorm(n3, sd=abs(fitlinearl3$residuals))
  data1_BT = data.frame(years=temp3$Year, mileage = temp3$Mileage, price=y_bt)
  fit_BT = lm(price~years + mileage, data=data1_BT)
  coeff_BT_wild3[i_BT,] = fit_BT$coefficients
}
> colnames(coeff_BT_wild3) = c("Intercept","ce year", "ce mileage")
> quantile(coeff_BT_wild3[,'Intercept'],c(0.05,0.95))
> quantile(coeff_BT_wild3[,'ce year'],c(0.05,0.95))
> quantile(coeff_BT_wild3[,'ce mileage'],c(0.05,0.95))

> df3 = as.data.frame(coeff_BT_wild3)
> ggplot(df3, aes(`ce year`)) +
  geom_histogram(fill = "tomato3", col = "brown", alpha = 0.8) +
  labs(title = "Distribution of Bootstrapped coefficient of Year") +
  geom_vline(xintercept = c(9270.439, 11238.037, 10280), col ="tomato3") +
  geom_text(aes(x=9270.439, label="5% quantile: 9270.439", y=-5), angle=0) +
  geom_text(aes(x=11238.037, label="95% quantile: 11238.037", y=-5), angle=0) +
  geom_text(aes(x=10280, label="fit: 10280", y=-5), col = "orange", angle=0)

> summary(fitlinearl3)
> str(summary(fitlinearl3))
 $ r.squared    : num 0.474
```

```
 $ adj.r.squared: num 0.472
> out <- capture.output(
+   bw_temp3 <- npregbw(formula = Price ~ ordered(Year) + Mileage, data = temp3,
+                           regtype = "lc", nmulti = 2)
+ )
> summary(bw_temp3)

> fit_temp3 <- np::npreg(bw_temp3)
> summary(fit_temp3)

> x_grid <- sort(unique(temp3$Year))
> y_grid <- sort(unique(temp3$Mileage))
> xy_grid <- expand.grid(x_grid, y_grid)
> colnames(xy_grid) <- c("Year", "Mileage")
> f = as.data.frame(xy_grid)
> f_1 <- f %>%
  group_by(Year) %>%
  arrange(Year)
> fit_graph <- predict(fit_temp3, newdata = f_1)
> f_year <- matrix(data=fit_graph, nrow=16, byrow=T)
> open3d()
> plot3d(x = temp3$Year, y = temp3$Mileage, z = temp3$Price, xlab = "Year",
ylab = "Mileage", zlab = "Price", col = "coral")
> rgl::surface3d(x = x_grid, y = y_grid, z = f_year,
               col = "skyblue", alpha = 0.8, lit = FALSE)
> rgl::rglwidget()
> par3d(windowRect = c(20, 30, 500, 500))
> movie3d(spin3d(), duration = 20, clean=FALSE,dir='/Users/claire/Desktop/test',
type = "gif", convert=NULL)
```

Kernel Regression

```
> x_gridl3 <- sort(unique(temp3$Year))
> y_gridl3 <- sort(unique(temp3$Mileage))
> xy_gridl3 <- expand.grid(x_gridl3, y_gridl3)
> colnames(xy_gridl3) <- c("Year", "Mileage")
> fl3 = as.data.frame(xy_gridl3)
> f_new_l3 <- fl3 %>%
  group_by(Year) %>%
  arrange(Year)
> fit_graphl3 <- predict(fitlinearl3, newdata = f_new_l3)
> f_yearl3 <- matrix(data=fit_graphl3, nrow=16, byrow=T)
> open3d()
> plot3d(x = temp3$Year, y = temp3$Mileage, z = temp3$Price, xlab = "Year",
ylab = "Mileage", zlab = "Price", col = "coral")
```

```
> rgl::surface3d(x = x_gridl3, y = y_gridl3, z = f_yearl3,
               col = "orchid", alpha = 0.25, lit = FALSE)
> rgl::rglwidget()
> par3d(windowRect = c(20, 30, 500, 500))
> movie3d(spin3d(), duration = 10, clean=TRUE,dir='/Users/claire/Desktop/test',
type = "gif", convert=NULL)
```