12/5/2021

Prepared by Norberto Limon & Zhenghao Gu

Turo Car Sharing Platform: Report & Analysis

(Region: Arizona)

**Abstract:** Given the use of a dataset provided from a popular car sharing website Turo.com, we derive a model based on factors that most contribute to the final price of a car trip. Using R, we attempt to explore this question by employing summary statistics, data cleaning and multiple linear regression to test a model of significant contributing factors.

**Sample Scope:** Data gathered is focused on patterns and insights from the state of Arizona.

**Analysis: (Walkthrough of Questions 1 – 6)**

*Question 1:*

**Load the data into RStudio.**

*Answer:*

The data is first loaded into a data frame using the built-in readRDS function and the provided turo.data.5140 dataset.

*Question 2:*

**Extract all the observations from the state your group is assigned to, and use this subset for the following tasks.**

*Answer:*

Next the Arizona data is isolated from the original dataset and read into a data frame for use throughout the remainder of the program using the readRDS function.

*Question 3:*

**Compute summary statistics and generate charts for ALL variables in the dataset, after excluding missing values. For each continuous variable, compute min, first quartile, medium, third quartile, max, mean, standard deviation, and skewness as summary statistics, and draw histogram. For each categorical variable, compute frequency and relative frequency distributions, and draw bar chart.**

*Answer:*

For this step, two list objects were created to store categorical and continuous variables separately. Using a for loop and a set of if statements, the first element of each column variable is tested (after having eliminated all NA values) to determine its classification status as either a continuous or a categorical variable.

# (3) Separate categorical and continuous data based on context.

continuous <- list()

categorical <- list()

for (name in names(az.df.clean)){

# Separated According to First Value of Each Variable

if (class(df[1, name]) == 'logical'){

# Boolean are categorical

categorical <- c(categorical, name)

}

else if(class(df[1, name]) == 'factor'){

# Factors AKA strings are also categorical

categorical <- c(categorical, name)

}

…

print(paste("Continuous Variable Count: ", length(continuous)))

print(paste("Categorical Variable Count: ", length(categorical)))

After this is determined, separate charts and statistics are computed for each item corresponding to the respective continuous or categorical lists. For both sets of variables, by employing the use of a for loop the corresponding statistics are extracted and charts generated (Standard Deviation, Skewness and Histograms for continuous variables; frequency, relative frequency and bar charts for the categorical variables). In order to calculate the summary statistics their respective names are matched to their column positions in the dataframe.

Continuous Variable Statistics Code:

> # Continuous Variables : min, first and third quartiles, max, mean

>

> N <- nrow(az.df.clean)

>

> print(summary(df[, continuous.pos])) # min, first and third quartiles, max, mean

### Number 3b: Summary Statistics

# extracting 'index positions'... (AKA: Column numbers)

continuous.pos <- match(continuous, names(df)) # index position of extracted columns classified as "continuous"

categorical.pos <- match(categorical, names(df)) # index position of extracted columns classified as "categorical"

Continuous Variable Statistics Output:

car.extra.mile.fee car.miles.included car.photo.num car.trip.price car.year

Min. :0.0100 Min. : 350 Min. : 1.000 Min. : 91.0 Min. :1929

1st Qu.:0.4800 1st Qu.: 750 1st Qu.: 4.000 1st Qu.: 294.0 1st Qu.:2013

Median :0.6800 Median :1000 Median : 7.000 Median : 448.0 Median :2016

Mean :0.7815 Mean : Inf Mean : 8.295 Mean : 641.1 Mean :2015

3rd Qu.:0.9500 3rd Qu.:1000 3rd Qu.:12.000 3rd Qu.: 714.0 3rd Qu.:2018

Max. :3.0000 Max. : Inf Max. :34.000 Max. :6993.0 Max. :2020

NA's :1275 NA's :449 NA's :42 NA's :47

host.tenure.in.weeks

Min. : 4.429

1st Qu.: 52.286

Median :112.857

Mean :126.728

3rd Qu.:186.857

Max. :461.000

NA's :119

Continuous Variables Standard Deviation, Skewness & Histograms:

> for (column in continuous){ # Standard Deviation and Skewness

+ print(paste(column,"': ","Standard Deviation: '", sqrt(N - 1 / N) \* sd(df[, column], na.rm = TRUE))) # <<<<<<<<< !!!!!!! See NOTES FOR LONGFORM !!!!!!

+ print(paste(column,"': ", "Skewness: '", skewness(df[, column], na.rm = TRUE)))

+ hist(df[, column], main = paste("Histogram of",column), xlab = column)

+ }

[1] "car.extra.mile.fee ': Standard Deviation: ' 16.522521766424"

[1] "car.extra.mile.fee ': Skewness: ' 1.99167355884491"

[1] "car.miles.included ': Standard Deviation: ' NaN"

[1] "car.miles.included ': Skewness: ' NaN"

[1] "car.photo.num ': Standard Deviation: ' 178.320381278836"

[1] "car.photo.num ': Skewness: ' 0.608224551790603"

[1] "car.trip.price ': Standard Deviation: ' 22583.0659105013"

[1] "car.trip.price ': Skewness: ' 4.11023554579958"

[1] "car.year ': Standard Deviation: ' 151.484724737546"

[1] "car.year ': Skewness: ' -4.8662168534807"

[1] "host.tenure.in.weeks ': Standard Deviation: ' 2966.91525528998"

[1] "host.tenure.in.weeks ': Skewness: ' 0.684016843441089"

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generated

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generated

Chart, histogram

Description automatically generated Chart, histogram

Description automatically generated

Categorical Variables: Freq & Relative Freq

for (column in categorical){ # freq and relative freq distribution

freq <- table(az.df.clean[,column])

rel.freq <- table(column) / length(N)

barplot(freq, main = paste("Bar Chart of",column), xlab = column)

}

Categorical Variables: Some Sample Barplots

Chart, box and whisker chart

Description automatically generatedChart, bar chart

Description automatically generated

Chart, histogram

Description automatically generated Chart, bar chart

Description automatically generated

*Question 4*:

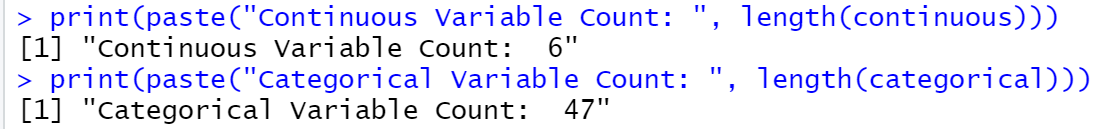
**Use Inter Quartile Range (IQR) method to identify outliers of all continuous variables, then remove all observations containing outliers.**

*Answer:*

Initially, the total number of records in the dataset was 31,261 rows and 53 columns. The first step in detecting the outlier is removing all the observations with missing values, i.e., all rows with NA's are deleted. After removing NA's, the cleaned data set includes 1285 rows and 53 columns. Based on the definition of each variable, the categorical and continuous variables are identified. There are 6 continuous and 47 categorical variables in the cleaned dataset. The code to identify the continuous and categorical dataset is as shown:

|  |
| --- |
| # List all categorical and continuous data separately based on context.  continuous <- list()  categorical <- list()  for (name in names(az.df.clean)){  # Separated According to First Value of Each Variable  if (class(df[1, name]) == 'logical'){  # Boolean are categorical  categorical <- c(categorical, name)  }  else if(class(df[1, name]) == 'factor'){  # Factors AKA strings are categorical  categorical <- c(categorical, name)  }  else if(class(df[1, name]) == 'integer'){  categorical <- c(categorical, name)  }  else if(df[1,name] == 0){  categorical <- c(categorical, name)  }  else if(df[1,name] == 1){  categorical <- c(categorical, name)  }  else if(name == "car.doors"){  categorical <- c(categorical, name)  }  else{  # All others belong in the list of continuous variables  continuous <- c(continuous, name)  }  }  print(paste("Continuous Variable Count: ", length(continuous)))  print(paste("Categorical Variable Count: ", length(categorical))) |

Output:

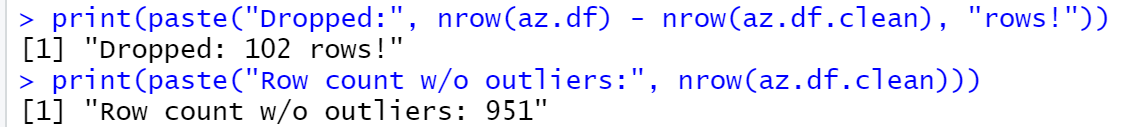


Using loop control statements, the interquartile range, upper quantiles, and lower quantiles are calculated for each continuous variable. Thus, all those records above the (upper quantile + 1.5 x Interquartile range) or lower than the (lower Quantile -1.5 x Interquartile range) are considered outliers. The rows detected as outliers are cleaned and removed. The total number of observations detected as outliers is 102. Thus, after removing the outlier, the dataset has 1053 rows and 53 columns. The code to detect outliers is:

Code:

|  |
| --- |
| az.df<- az.df.clean  # The reduction of outlier is performed on az.df.clean.  # The dataset is retained in the dataframe az.df  for (column in continuous){  iqr <- quantile(az.df[, column], 0.75) - quantile(az.df[, column], 0.25)  lowerlimit <- quantile(az.df[, column], 0.25) - 1.5 \* iqr  upperlimit <- quantile(az.df[, column], 0.75) + 1.5 \* iqr  ### Remove all observations w/ missing values  for (row in az.df.clean[, column]){  if(row < lowerlimit | row > upperlimit){  az.df.clean <- az.df.clean[-row,]  }  }}  print(paste("Dropped:", nrow(az.df) - nrow(az.df.clean), "rows!"))  print(paste("Row count w/o outliers:", nrow(az.df.clean))) |

Output:



*Question 6:*

**Build a multiple linear regression model using car.trip.price as dependent variable. Select at least five independent variables. Treat each categorical variable as a single variable although it may be broken into multiple dummy variables. Try different models and choose the best one you can find.**

*Answer:*

The steps we took to identify the best multiple regression model to estimate the car.trip.price using various combinations of independent variables are as follows:

1. First, all categorical variables with more than 40 categories are dropped. Thus, car.city, car.make, car.insurance, car.model, and car.state are excluded from modeling.
2. A full model is created with all of the independent variables is created using function lm() as shown below. This full model has many insignificant predictors, and the model can predict 60.14% variation in car.trip.price by varying all of the predictors. To identify all of the significant predictors, the step function is used.

|  |
| --- |
| fit<- lm(formula = car.trip.price ~ ., data = az.lm) |

1. The step function in R is used for stepwise detection of significant variables. It focuses on minimizing the AIC of the model. Code: step(fit) detects the final model:

Table

Description automatically generated

The model with the above predictor is:

Table

Description automatically generated

This model can predict a 59.4% variation in the car.trip.price. Thus, the model's predictability is insignificantly reduced, with many predictors not included. However, the model still contains insignificant predictors as the p-value of many predictors are greater than 0.05.

1. The next step is to remove all of those insignificant predictors one by one, based on the p-values. The predictor with the highest p-value (highest p>0.05) is removed first, and the model is recreated.
2. If there remains any insignificant predictor after the recreation of the model, then step 4 is repeated until all of the remaining predictors remain significant. The final model with backward linear regression step is:

Text

Description automatically generated

**Conclusion:** After conducting our analysis we determined that the resulting model can predict a 58.2% variation in the car.trip.price with an adjusted R squared of 0.5788, approximately the same as r-squared. Thus, the model's predictability is insignificantly reduced, with all of the predictors included in the model becoming significant. In other words, only those variables that are statistically significant are left in the model with all others being removed. With this model the predictability remains comparable to that of the initial model with all variables present. (Please see attached: Appendix 1 - R Code).

**Appendix 1**

R Source Code:

* Cleaning, Detecting Outliers and Performing Multiple Linear Regression Analysis

|  |
| --- |
| * # Group Project Code: Multiple Regression Analysis * # (1) Import Data set * df = readRDS('turo.data.5140') * # (2) Select the state * az.df.clean <- df[df$car.state == 'az',] * head(az.df.clean,1) * # (5) Remove All NA's * az.df.clean<- na.omit(az.df.clean) * # (3) Separate categorical and continuous data based on context. * continuous <- list() * categorical <- list() * for (name in names(az.df.clean)){ * # Separated According to First Value of Each Variable * if (class(df[1, name]) == 'logical'){ * # Boolean are categorical * categorical <- c(categorical, name) * } * else if(class(df[1, name]) == 'factor'){ * # Factors AKA strings are also categorical * categorical <- c(categorical, name) * } * else if(class(df[1, name]) == 'integer'){ * # In this case the int values are used to represent Binary Boolean values (1's and 0's) * categorical <- c(categorical, name) * } * else if(df[1,name] == 0){ * # Binary Boolean values (1's and 0's) are categorical * categorical <- c(categorical, name) * } * # Binary Boolean values (1's and 0's) are categorical * else if(df[1,name] == 1){ * categorical <- c(categorical, name) * } * else if(name == "car.doors"){ * categorical <- c(categorical, name) * } * else{ * # All others belong in the list of continuous variables * continuous <- c(continuous, name) * } * } * print(paste("Continuous Variable Count: ", length(continuous))) * print(paste("Categorical Variable Count: ", length(categorical))) * ### Number 3b: Summary Statistics * # extracting 'index positions'... (AKA: Column numbers) * continuous.pos <- match(continuous, names(df)) # index positions of extracted columns classified as "continuous" * categorical.pos <- match(categorical, names(df)) # index positions of extracted columns classified as "categorical" * # Continuous Variables : Compute -> min, first and third quartiles, max, mean, sd & skewness; histograms * N <- nrow(az.df.clean) * print(summary(df[, continuous.pos])) # min, first and third quartiles, max, mean * for (column in continuous){ # Standard Deviation and Skewness * print(paste(column,"': ","Standard Deviation: '", sqrt(N - 1 / N) \* sd(df[, column], na.rm = TRUE))) # <<<<<<<<< !!!!!!! See NOTES FOR LONGFORM !!!!!! * print(paste(column,"': ", "Skewness: '", skewness(df[, column], na.rm = TRUE))) * hist(df[, column], main = paste("Histogram of",column), xlab = column) * } * # Categorical Variables: Compute -> frequency and relative frequency distributions; bar charts * print(categorical) # extracted column names of the df classified as "continuous" * for (column in categorical){ # freq and relative freq distribution * freq <- table(az.df.clean[,column]) * rel.freq <- table(column) / length(N) * barplot(freq, main = paste("Bar Chart of",column), xlab = column) * } * # (4) Using the IQR method, detect and remove all the rows having outliers. * az.df<- az.df.clean * for (column in continuous){ # Standard Deviation and Skewness * iqrange <- quantile(az.df[, column], 0.75) - quantile(az.df[, column], 0.25) * lowerlimit <- quantile(az.df[, column], 0.25) - 1.5 \* iqrange * upperlimit <- quantile(az.df[, column], 0.75) + 1.5 \* iqrange * # (5) Remove all observations w/ missing values * for (row in az.df.clean[, column]){ * if(row < lowerlimit | row > upperlimit){ * az.df.clean <- az.df.clean[-row,] * } * } * } * print(paste("Dropped:", nrow(az.df) - nrow(az.df.clean), "rows!")) * print(paste("Row count w/o outliers:", nrow(az.df.clean))) * # Multiple Linear Regression * az.lm<-az.df.clean * az.lm$car.city<-NULL * az.lm$car.make<-NULL * az.lm$car.insurance<-NULL * az.lm$car.model<-NULL * az.lm$car.state<-NULL * fit<-lm(car.trip.price~.,az.lm) * step(fit) * # Backward Step method to remove insignificant variables. * Finalfit<-lm(formula = car.trip.price ~ car.displayed.user.review.num.past.18m + car.displayed.user.review.num.past.6m + * car.doors + car.extra.mile.fee + car.extra.post.trip.cleaning +car.transmission + * car.year + host.verified.email, data = az.lm) * summary(Finalfit) |