# Problem 1: Predicting Honda Civic Prices

The database used to predict Honda Civic prices contains information from a car dealership. The dataset includes information on over 35 variables, which have a mix of numerical and categorical variables. The dataset also contains 1436 cases. Since our business problem is to predict Honda Civic prices, we will choose predictive variables that can use Price as the response variables.

1. **Selecting Variables**

Below we selected only 9 variables from the dataset. The variables selected were Price, Age, Mileage, HP, Automatic, CC, Doors, QuartTax, and Weight. The variables removed were ID, Model, MFG Month, Cylinders, and Gears. Variables such as ID and Model were removed since it was specific identifications of each case and bring no value to our model. The new name of the selected variables is selected.var.

**A screenshot of a computer

Description automatically generated**

1. **Setting Seeds and Selecting 1400 Cases**

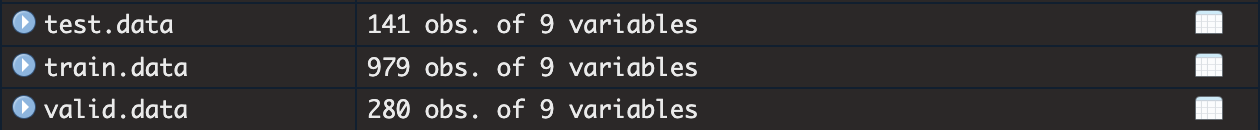
Below, the seed was set at 1, and only 1400 cases were selected from 1436 total cases. This new database with these new selection changes was named selected.var.df.

**A screenshot of a computer

Description automatically generated**

1. **Partitioning the Data**

Now, we partitioned the data into training, validation, and testing. From selected.var.df, 70% of the data was for training, 20% for validation, and the remaining 10% for testing. This partitioning resulted in 979 observations for training, 280 for validation, and 141 for testing.

****

1. **Why should the data be partitioned into training, validation, and testing sets? What will the training set be used for? What will the validation set be used for? How about testing set?**

When creating a model, we want to ensure how well our prediction model will perform when we apply it to new data. We do that by comparing the performance of different models to choose the best model. To make sure our chosen model is, in fact, the best, we use the concept of data partitioning to avoid overfitting. In this model, we partition our data into three, training, validation, and testing sets. Our training partition contains data used to build different models to be examined. It’s usually the largest partition. Our validation partition is used to assess the predictive performance of each model so that you can compare models and choose the best one. The test partition is used to assess the performance of the chosen model with new data. We rely on not only our training and validation partition but also our test partition. When we only use the validation data to create models and choose the best-performing model, we still encounter an overfitting problem. Applying the model to the test data provides an unbiased estimate of how well the model will perform with new data.

1. **Correlation Matrix**

Below we calculated the correlation matrix of 8 numerical predictors within our training data. Before creating the correlation matrix, we removed the variable Automatic.

**A black screen with white numbers

Description automatically generated**

Based on the correlation below, we highlighted the highly correlated pairs, which have the potential redundancy and can cause multicollinearity. The first highly correlated pair is **Price and Age**, with a **negative** correlation of **0.88**. There are also other correlations that are highly moderate such as **Weight and QuartTax** with a positive correlation of **0.68** and the pair **Weight and Price** with a positive correlation of **0.64**.

**A graph of numbers and a number of people

Description automatically generated with medium confidence**



1. **Simple Linear Regression Model based on Price with Highest Correlated Variable**

Based on the correlation matrix, Age has the highest correlation with Price. Here is the simple linear regression model for estimating the Honda Civic prices as a function of Age. The model provided the following **equation:**

A screenshot of a computer

Description automatically generated

To understand if this model is appropriate for predicting the prices of the Honda Civic, evaluate the model using the three criteria: Adjusted R-square, Statistical Tests, and Residual Analysis. For a simple regression model, it needs to pass all three criteria to classify the model as ‘good’.

**Adjusted R-square**

The model provided an adjusted r-square of 0.7709. It can be interpreted as **77.09%** of the variation in our dependent variable of Honda Civic prices can be explained by our independent variable, Age. Since our r-squared is greater than 0.64 (**0.7182 > 0.64**), our **model passes** the first criteria.

**Statistical Tests: F-test and Individual t-tests***F-test:*

To conduct the f-test, we compare the p-value to our assumed alpha of .05. Our p-value is 2.2e-16:

our p-value is smaller than our alpha, meaning we **reject our null hypothesis in favor of the alternative**. Age is found to have a significant influence on Price at a 95% confidence level.

*t-tests:*

We set up our Null and Alternative Hypothesis above. Our null hypothesis states that the slope for age is equal to zero, and our alternative hypothesis is not equal to zero. For Age, our t-value was -57.38, and our p-value was 2e-16. Since our p-value is less than 0.05 that means we **reject the null hypothesis in favor of the alternative**. In other words, Age has a statistically significant relationship with the price of the Honda Civic.

Based on the 2 statistical tests, the model **passes the second criterion** and is **statistically significant**.

**Residual Analysis**

A graph of residuals and statistical data

Description automatically generated

In the graph above, it shows residual values versus predicted values. The plot doesn't show any distinct patterns or curves, which supports the assumption of linearity between the predictor (Age) and the response (Price). Since there is **no clear pattern**, the model **passes** the criterion

A simple regression model needs **to pass all three criteria**. Fortunately, this model

passes all criteria, meaning that this model does a great job of predicting the prices of the Honda Civic.

# Problem 2: Multiple Regression Analysis

1. **Scatterplots**

To explore the relationship between spending vs. Freq and Spending vs. the Last Update, 2 scatterplots were created, as seen below.

A graph with black and white dots

Description automatically generated

In the first scatter plot, the predictor represents Freq, and the response represents Spending in dollars. The plot suggests a weak and somewhat scattered relationship between frequency and spending. Most data points are clustered at low spending levels, between 0 and 5 dollars, regardless of frequency, with a dense cluster appearing around frequencies below 500. As frequency, or the number of transactions in the preceding year, increases, there are instances of higher spending values, which are sparse and widely spread out. This indicates that, while increased frequency might lead to higher spending in some cases, there is no consistent, strong correlation between the two variables and **no linear relationship**.

A graph of a number of dots

Description automatically generated with medium confidence

In this second scatter plot, the x-axis represents the number of days since the Last Update, while the y-axis shows Spending in dollars. The plot reveals that most data points are concentrated along the left side, where Last Update values are relatively low, under 500. Here, spending values range widely from 0 to over 3000 dollars, suggesting high variability in spending for these lower Last Update values. As the number of days since the Last Update increases beyond 500, spending occurrences decrease significantly, and the few points in this range exhibit lower spending values overall. This indicates that spending tends to decrease as the number of days since the Last Update values increase, though the relationship is not entirely consistent and has considerable variation at lower Last Update values and **no linear relationship**.

1. **Partitioning the Data**

The following steps were taken to partition the 2000 records.

A black screen with white numbers

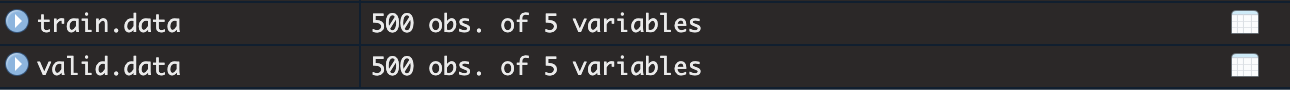
Description automatically generated

From the original data frame imported, only 1000 records were selected and named tacko1000.df. The data set contains 1000 observations and 6 variables.

A black screen with white numbers

Description automatically generated

Then, there were only five variables selected. As seen above, the variables selected were Spending, Freq, Last\_Update, Web.order, and Geder. The select variables were name select.var, still containing 1000 observations. The variable removed was Address Residential. Before going to the next step, the seed was set to 1.



Finally, we partitioned our data into two equal parts. As seen above, 50% for training data, which now includes 500 observations and 5 variables, and 50% for validation data, which now includes 500 observations and 5 variables.

1. **Multiple Linear Regression Model**

Below is the multiple linear regression model for estimating the amount spent as

a function of the number of Freq transactions, the number of days since the Last update, whether the customer purchased by Web, and Gender. The model provided this **equation**:

The **Coefficient Interpretations** are as follows:

**Freq:** For every one transaction in the preceding year, customer spending **increases by 81.025 dollars**.

**Last Update:** For every one day since the latest update, customer spending has **decreased by 0.008 dollars**.

**Web Order:** When customers purchase by web at least once, customer spending **increases by 21.579 dollars**.

**Gender:** When customers are male, customer spending **decreases by 10.042** dollars

A screenshot of a computer

Description automatically generated

To understand if this model is appropriate for predicting customer spending, we evaluate the Multiple Regression model using the four criteria: Adjusted R-square, Statistical Tests, Residual Analysis, and Multicollinearity. For a multiple regression model, it needs to pass at least 2 out of 4 criterias to classify the model as ‘good’.

**Adjusted R-square**

The model provided an adjusted r-square of 0.3907. It can be interpreted as **39.07%** of the variation in our dependent variables (freq, last update, web order, and gender) can be explained by our independent variable (spending). Since our r-squared is less than 0.64 (**0.3907 > 0.64**), our **model fails** the first criterion.

**Statistical Tests: F-test and Individual t-tests**

*F-test:*

*Individual t-tests:*

|  |  |
| --- | --- |
| Freq | <2e-16 **<** 0.05 |
| Last Update | 0.228 **>** 0.05 |
| Web Order | 0.128 **>** 0.05 |
| Gender | 0.470 **>** 0.05 |

Based on the statistical testing, the predictor Freq is the only variable that is significantly related to the dependent variable, Spending. The p-value of Freq is less than our significant level of 0.05, resulting in rejecting our null hypothesis in favor of our alternative hypothesis. Our other predictors, Last Update, Web Order and Gender are not significantly related to Spending. These predictors have a p-value greater than our significant level of 0.05, resulting in failing to reject our null hypothesis. Based on our statistical testing our model **does not pass the second criteria**.

**Residual Analysis**

A graph with black dots and white text

Description automatically generated

This residual vs. predicted plot above illustrates the relationship between the residuals and the predicted values from the multiple regression model. Usually, residuals should be randomly scattered around zero across all predicted values, indicating that the model fits well across the range. However, in this plot, the residuals show a **pattern**. Around the lower predicted values (below 200), residuals are tightly clustered near zero, but as predicted values increase, the residuals spread more widely. This pattern suggests heteroscedasticity, meaning the variance of errors increases with the predicted values, which violates the assumption of constant variance in linear regression. A linear model may not be the best fit for this data, an alternative modeling approach could improve the fit by addressing these issues. The model **does not pass** the residual analysis criteria.

**Multicollinearity**

The VIF test is used to test for multicollinearity. VIF values range between 5 to 10. If, in the model, the VIF value of variables is greater than 5, then there is multicollinearity, and if the VIF is greater than 10, then there is severe multicollinearity. Below is our VIF test result:

A screenshot of a computer

Description automatically generated

Based on the results, all variables **passed** the test. The output was FALSE for all predictors, meaning none of the variables have a square root of their VIF greater than 5. This indicates that **multicollinearity is not present**.

Overall, the multiple regression model to predict customer spending **is not statistically significant** since it **failed three out of four criteria**. To fix this, we can try different models and compare them, and we can remove redundant predictors.

1. **Can you determine if the gender is a contributing factor for predicting the spending amount?**

Based on the multiple regression model, gender is not a significant factor for predicting the spending amount. When conducting statistical tests, the p-value for the gender coefficient is 0.470, which is greater than the significance level of 0.05. This indicates that we fail to reject the null hypothesis. In other words, there is not enough evidence to suggest that gender has a significant impact on spending.

1. **Predictive Accuracy**

A screenshot of a computer

Description automatically generated

The provided predictive accuracy metrics indicate a less-than-ideal model performance. The model overestimates the actual values, as evidenced by the positive Mean Error of 2.91. The high Root Mean Squared Error of 143.42 and Mean Absolute Error of 83.54 suggest significant deviations between predicted and actual values. The NaN and Inf values for Mean Percentage Error (MPE) and Mean Absolute Percentage Error (MAPE) are likely from issues with predictions close to zero or negative, leading to division by zero errors.

1. **Suppose we used the stepwise regression method to reduce the number of predictors. Which model would be your suggested model? Explain the meaning of this model in the context of this problem.**

We used the stepwise regression method to reduce the number of predators. Specially we used backward elimination which starts with all predictors and eliminates the least useful predictors one by one. It stops when all remaining predictors have statistically significant contributions. The following model was suggested:

A screen shot of a computer

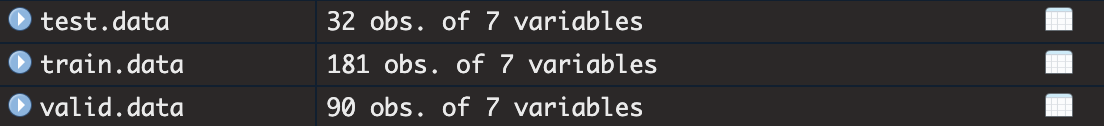
Description automatically generated

Our suggested model suggests using two predictors Freq and Web Order. Based on the adjusted R-squared, the Stepwise regression is a better model than the multiple regression model we created in part c. The R-squared for the Stepwise regression model was 0.3908 which is higher than the multiple regression model r-square of 0.3907.

# Problem 3: Principal Components

1. **Partitioning Data Set**

As seen below, the dataset that contains information about 303 patients was partitioned into 3: training, validation, and testing. For the training set, it included 60% of our data, which was 181 observations. The validation had 30%, which was 90 observations, and our testing had 10%, which included 32 observations.

****

1. **Correlation Matrix**

To create a correlation matrix, only Age, Blood Pressure, Cholesterol, and Heart Rate were used as quantitative measurements.

**A screenshot of a graph

Description automatically generated**

According to the correlation matrix above, the pair with the highest correlation is **Age and Heart Rate**. This pair has a **negative** correlation of **0.37**. The pair with the lowest correlation is **Cholesterol and Heart Rate**. The pair has a **negative** correlation of **0.04**. Ultimately, all the pairs exhibit low correlations, below .5.

1. **Principle Component Analysis**

Below is a principal component analysis on the training data:

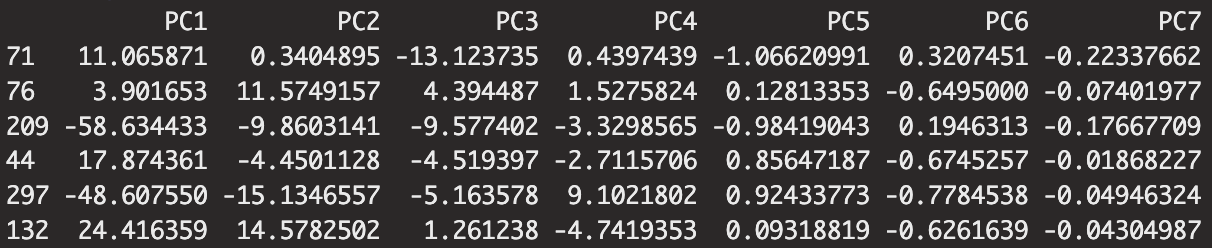
**A computer screen with numbers and symbols

Description automatically generated**

The principal component analysis reveals that the first few principal components capture a significant portion of the data's variability. PC1 appears to be associated with cholesterol levels and age, with higher values indicating older individuals with higher cholesterol. PC2 is primarily driven by heart rate, with higher values linked to increased heart rate. PC3 is associated with blood pressure, with higher values indicating higher blood pressure. PC4 seems related to chest pain type, while PC5 and PC7 are associated with fasting blood sugar levels. Finally, PC6 appears gender-related, with higher values associated with female patients. Overall, this PCA analysis suggests that these principal components represent key Heart Rate risk factors and can provide valuable insights into the underlying structure of the data.

1. **Principle Component Scores**

Below are the principal component scores. The scores provided 181 rows, which was too large to display, so instead, the first 6 rows were provided.

****

The summary statistics of each principal component are included to provide a clear representation of each principal component.

**A screenshot of a computer screen

Description automatically generated**

The summary statistics show the distribution and range of values for each principal component scores. For example, PC1 ranges from -113.550 to 161.928, with a median of -6.686 and a mean of 0. This suggests a wide range of values with a slight negative skew. Similarly, PC2 ranges from -77.011 to 47.383, with a median of 3.398 and a mean of 0, indicating a wider distribution with a slight positive skew. The other PCs also exhibit varying ranges and central tendencies, with some being more symmetrically distributed than others.

1. **How many principal components should be used? List and discuss your suggested principal components. What proportion of the original information can be represented by your suggested principal components?**

**A black and white background with numbers

Description automatically generated**

Based on my outputs, **PC1** and **PC2**, should be used. To determine the number of components, we need to be aware of the cumulative performance. The goal is to retain enough components to capture a high proportion of the total variance, typically around 80-90%. Looking at the 1st cumulative proportion, we see that PC1 already accounts for 70.15% of the variance, which is a relatively large percentage. The first 2 principal components (PC1 and PC2) explain **88.53%** of the variance (of the original information), which meets the common threshold of 80%. Based on this analysis, using the first 2 components is enough to capture most of the variation in the data, and the rest can be eliminated.

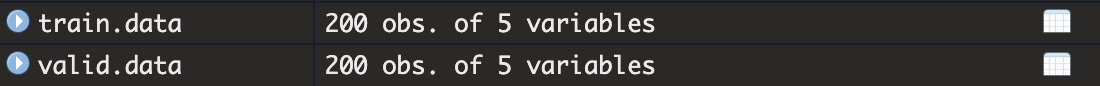
1. **Should the data be normalized?**

Yes, the data should be normalized. When we normalize data, it is usually if the variables are measured in different units so that it is unclear how to compare the variability of different variables, or if variables are measured in the same units, the scale does not reflect importance, data should be normalized. In this data, we had variables that used different units, such as Age and Blood Pressure. Age is usually measured in years, and Blood pressure is measured in mm Hg. Not only are they different units but the ranges are different. It’s also important to normalize before conducting a principal component analysis since PCA aims to maximize variance captured across components. Without normalization, PCA will capture the most variance from variables with larger numerical ranges, even if those variables are not necessarily the most important. Normalizing ensures that each variable contributes equally to the variance, regardless of its original scale. Normalizing our data is generally important but is necessary when conducting a principal component analysis to obtain meaningful principal components.

# Problem 4: Logistic Regression Model

1. **Partitioning Data Set**

The data stored in Admission.csv contained 400 cases. The data was partitioned into two, the training set and the validation set. 200 of those cases went toward the training set, and the rest went toward the validation set. We then set our seed to 1.

****

1. **Logistic Regression Model**

Below is our Logistic Regression Model that shows how the Graduate Record Exam score, Grade Point Average, and the ranking of a student’s undergraduate institution affect their admission into graduate school.

**A computer screen shot of a program

Description automatically generated**

1. **Logistic Regression Equation**

Below is the Logistic Regression **Equation**:

**Coefficient Interpretations:  
GRE**: For every one point better in the Graduate Record Exam, the chance of being admitted **increases** by **0.002**

**GPA:** For every one point added to the student’s Grade Point Average, the likeliness of being admitted into graduate school increases by **0.777**

**Rank:** The lower the ranking of the student’s undergraduate institution, the likeliness of being admitted into the graduate school **decreases** by **0.560**

**GRE is significant** to the logistic regression model. This conclusion was made by testing its p-value. The GRE p-value is 0.036, which is less than the significant level of 0.05. When the p-value of a variable is less than the significant coefficient, we reject the null hypothesis in favor of the alternative. This means the Graduate Record Exam scores have a significant value to the admission into graduate school.

()

1. **Confusion Matrix**

Using the Confusion Matrix provided below, the performance of the Logistic Regression model will be evaluated.

**A screenshot of a computer screen

Description automatically generated**

The Confusion Matrix provided the following:

**True Negatives**: 124 cases where the model predicted no admission (0), and the actual value was also no admission (0).

**False Positives**: 53 cases where the model predicted admitted (1), but the actual value was no admission (0).

**False Negatives**: 9 cases where the model predicted no admissions (0), but the actual values were admitted (1).

**True Positives**: 14 cases where the model predicted admitted(1), and the actual value was also admitted(1)

The Logistic Regression Model had an overall **accuracy** of **69%**. Meaning that it correctly predicted admission status for 69% of the validation data

1. **Sensitivity and Specificity of the Model**

The confusion matrix provided a **Sensitivity of 70.06%** and a **Specificity of 60.87%**. Having a sensitivity of 70.06% indicates that the model correctly identifies about 70% of actual admitted students (1). On the other hand, Specificity measures the model’s ability to correctly identify negative cases (0). Based on our sensitivity, our model tends to miss many true admission cases, which could become problematic. Our model is decent at predicting admission, but it could benefit from altercating the model to help improve both sensitivity and specificity.

1. **CSV**

Csv is provided as an attachment

# R-Code

Part 1. Predicting Honda Civic Prices

**#Part a**

hondacivic.df <- read.csv("HondaCivic.csv")

View(hondacivic.df)

selected.var <- hondacivic.df[-c(1,2,5,11,12)]

View(selected.var)

**#Part b**

set.seed(1)

selected.var.df <- selected.var[1:1400,]

**#Part c**

#training 70%

train.rows <- sample(rownames(selected.var.df),dim(selected.var.df)[1]\*0.7)

train.data <- selected.var.df[train.rows,]

**#validation 20%**

valid.rows <- sample(setdiff(rownames(selected.var.df),train.rows),dim(selected.var.df)[1]\*0.2)

valid.data <- selected.var.df[valid.rows,]

**#testing**

test.rows <- setdiff(rownames(selected.var.df),union(train.rows,valid.rows))

test.data <- selected.var.df[test.rows,]

**#Part e**

**#Correlation Matrix**

selected.cor <- train.data[-c(5)]

CM1 <- cor(selected.cor)

View(CM1)

round(cor(CM1),3)

library(corrplot)

corrplot(CM1, type = "upper", method = "number", tl.srt=45, tl.col = "black")

**#Part f**

**#Simple regression**

simplereg <- lm(Price ~ Age, data = train.data)

View(simplereg)

summary(simplereg)

**#Predicted values**

pred\_y\_sr <- fitted(simplereg)

**#Standardized residual**

residual\_sr <- rstandard(simplereg)

**#residual vs. predict**

plot(pred\_y\_sr, residual\_sr, main = "Residual vs. Predicted Plot")

Part 2. Predicting Honda Civic Prices

tacko.df <- read.csv("Tacko.csv")

**#part a**

**#Spending vs. freq**

plot(tacko.df$Freq ~ tacko.df$Spending, main = "Spending vs. Freq", xlab = "Freq", ylab = "Spending (in dollars)")

**#spending vs. last update**

plot(tacko.df$Last\_Update ~ tacko.df$Spending, main = "Spending vs. Last Update", xlab = "Last Update", ylab = "Spending (in dollars)")

**#part b**

**# Select the first 1000 records**

tacko1000.df <- tacko.df[1:1000,]

**#Select the first five variables**

select.var <- tacko1000.df[-c(6)]

**#sample seed 1**

set.seed(1)

**#partition**

train.rows <- sample(rownames(select.var), dim(select.var)[1]\*0.5)

train.data <- select.var[train.rows,]

valid.rows <- setdiff(rownames(select.var),train.rows)

valid.data <- select.var[valid.rows,]

**#Part c**

reg <- lm(Spending~., data = train.data)

summary(reg)

**#residuals**

**#predicted values for Y**

pred\_y\_mr <- fitted(reg)

**#standardized residual**

residual\_mr <- rstandard(reg)

**#plot Residual vs. predicted**

plot(pred\_y\_mr, residual\_mr, main = "Residual vs. Predicted Plot")

**#multicollinearity**

library(car)

vif(reg)

sqrt(vif(reg))>5

**#part e**

library(forecast)

**#use predict() to make predictions on a new set**

valid.lm.pred <- predict(reg,valid.data)

acc <- accuracy(valid.lm.pred, valid.data$Spending)

View(acc)

**#part f**

**#stepwise backwards**

reg.step.back <- step(reg, direction = "backward", data = train.data)

summary(reg.step.back)

Part 3. Predicting Honda Civic Prices

heart.df <- read.csv("Heart.csv")

**#part a**

#paritioning training validation and testing

train.rows <- sample(rownames(heart.df), dim(heart.df)[1]\*0.6)

train.data <- heart.df[train.rows,]

valid.rows <- sample(setdiff(rownames(heart.df),train.rows), dim(heart.df)[1]\*0.3)

valid.data <- heart.df[valid.rows,]

test.rows <- setdiff(rownames(heart.df), union(train.rows, valid.rows))

test.data <- heart.df[test.rows,]

**#part b**

CM1 <- cor(train.data[c("Age", "BP", "Chol", "HR")])

View(CM1)

library(corrplot)

corrplot(CM1, type = "upper", method = "number", tl.srt = 45, tl.col = "black")

**#part c**

pc2 <- prcomp(train.data,)

print(pc2)

**#part d**

pc\_scor <- pc2$x

print(pc\_scor)

head(pc\_scor)

summary(pc\_scor)

**#part e**

summary(pc2)

Part 4. Predicting Honda Civic Prices

admission.df <- read.csv("Admission.csv")

admission.df <- admission.df[,-c(1)]

**#part a**

set.seed(1)

train.rows <- sample(rownames(admission.df),dim(admission.df)[1]\*0.5)

train.data<- admission.df[train.rows,]

valid.rows <- setdiff(rownames(admission.df),train.rows)

valid.data <- admission.df[valid.rows,]

**#part b**

library(caret)

library(ggplot2)

library(lattice)

multi.log <- glm(Admission ~ GRE+GPA+Rank, data = admission.df, family="binomial")

summary(multi.log)

**#part d**

#confusion matrix

multi.log.pred <- predict(multi.log, newdata = train.data, type = "response")

train.data$pred.glm = ifelse(multi.log.pred > 0.5, "1", "0")

train.data$pred.glm = as.factor(train.data$pred.glm)

str(train.data$pred.glm)

train.data$Admission = as.factor(train.data$Admission)

str(train.data$Admission)

confusionMatrix(train.data$Admission,train.data$pred.glm)

View(train.data)

**#part f**

write.csv(train.data, "MidtermPart4TrainData.csv")