Report 2

Lab2 Group A: Lee, Joshua; Liu, Kaiyi; Pulsone, Nathaniel; Wang, Mengyao; Xu, Zexian; Yang, Xiaojing

Loading the Library and DataSet

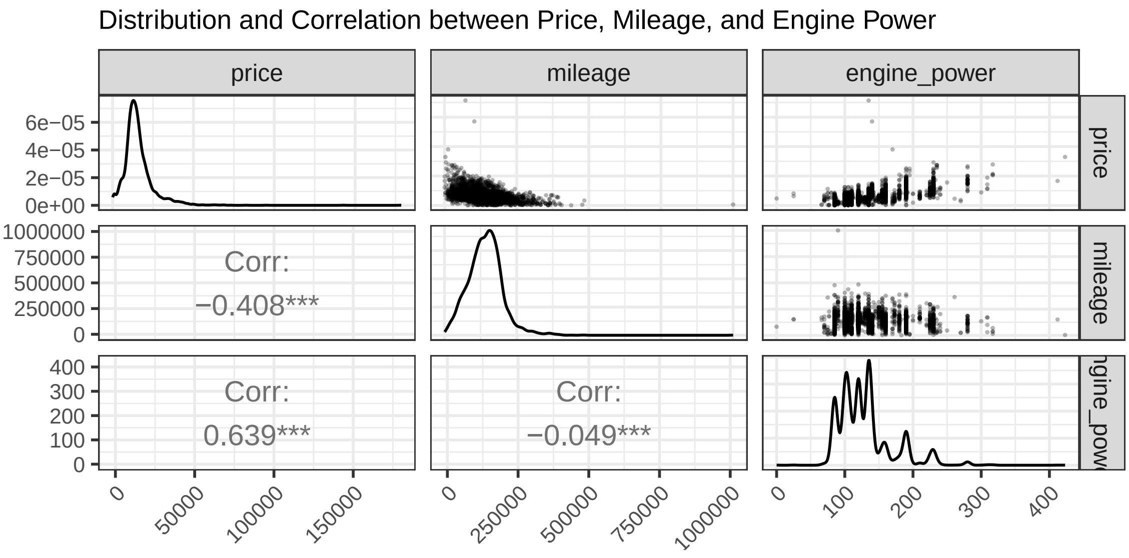
```r
mylibrary <- c("dplyr", "ggplot2", "cowplot", "GGally", "MASS")
invisible(lapply(mylibrary, library, character.only = TRUE))
### load the data
dat_bmw <- read.csv("BMWpricing_updated.csv")
### Assigning NA to data points with negative mileage
dat_bmw[dat_bmw$mileage < 0, "mileage"] <- NA
### Converting mileage, engine power, and price into numeric
dat_bmw[,c(3,4,17)] <- apply(dat_bmw[,c(3,4,17)], 2, as.numeric)
### Create a data frame only has the variable of interests
dat_bmw2 <- dat_bmw[,c("price", "mileage", "engine_power")]
### Drop the data point with NA
dat_bmw2 <- dat_bmw2[complete.cases(dat_bmw2),]
```

Variable Selection

Preliminary data visualization and analysis will utilize an ordinary least-square simple linear regression model to study variable relationships. The response variable for the model will be the auction price of a BMW car, and the first predictor will be the car's mileage. Most of the models constructed in this study will use price as the response variable, as the price of a car is what we want to be able to predict. Given that the mileage of a car tends to be the most important factor people consider when buying a used car, it would be useful to study the relationship between price and mileage.

As we shall see, car mileage appears to be the most appropriate predictor for a simple linear regression of price for a number of reasons. First, the mileage has the largest variance of all possible predictors, which will give an estimate for the model's slope with relatively low standard error. In addition, mileage seems to have the most 'continuous' distribution between the two numeric variables (the other of which being the car's engine power), meaning it has minimal "variance clustering" around particular values, and also minimal gaps across the range of collected data. Notably, there are a few weaknesses to the SLR model that should be considered. Most importantly, the price seems to exponentially decay with mileage, rather than sharing a linear relationship. Related to this, there is clearly heteroscedasticity in price across mileage, as the data is clustered around its lower boundary (price = 0).

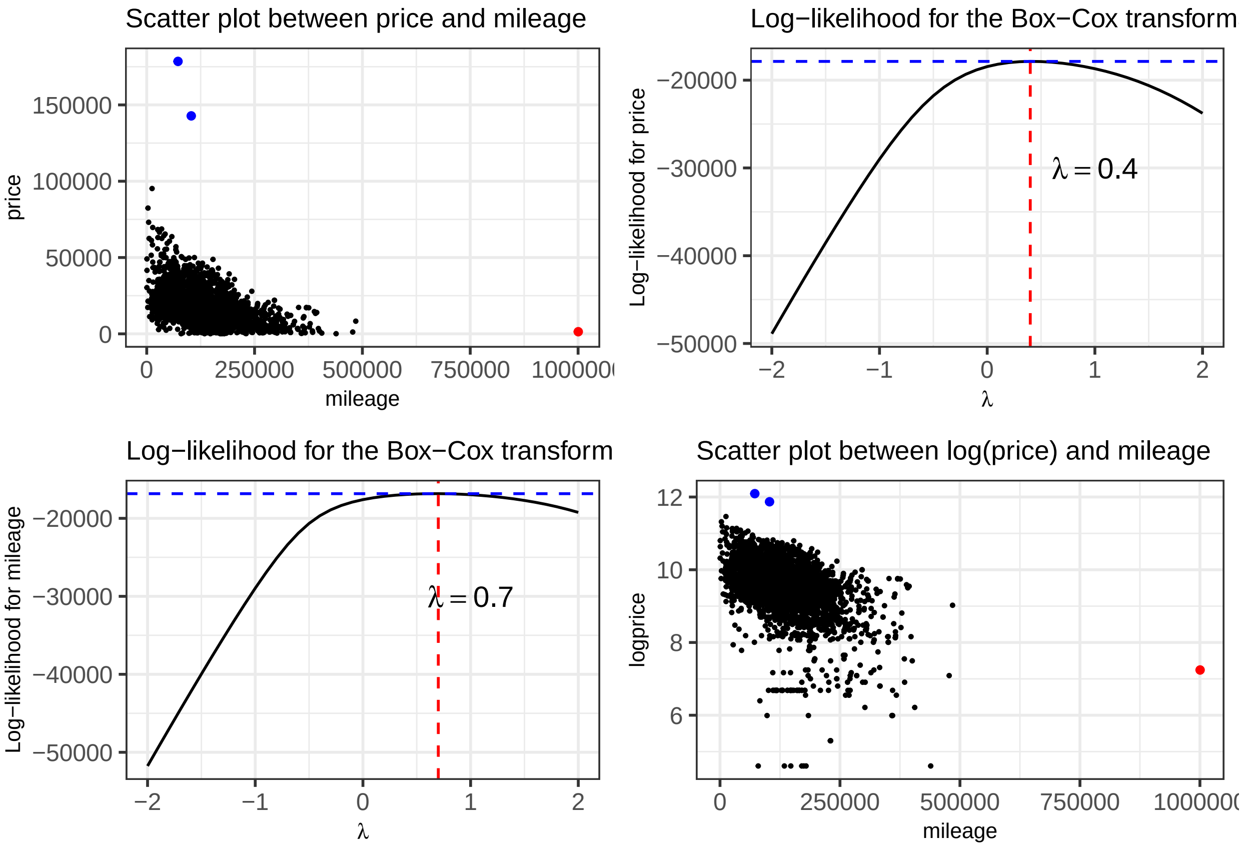
Scatterplot Matrix



Scatter Plot of Price against Mileage

We first set the negative mileage values (in one sample) as missing data, and excluded it in our following analysis.
The scatter plot shows a negative correlation between price and mileage, indicating that price tends to decrease as mileage increases. The relationship is not strongly linear, so a straight-line regression model may not be the best option. We find some problems in our data: 1) Both price and mileage follow highly right-skewed distributions; 2) High-leverage points appear on the right side of the scatter plot, particularly the red point with approximately 1,000,000 miles (likely an error in data entry, which will be removed); 3) Outliers with extreme high sold prices are observed on the top-left of the plot, especially these blue points with price over 100000.
To improve normality, we applied the Box-Cox transformation to the price variable. Though the λ with the greatest log-likelihood ($\lambda_{price} = 0.4, \lambda_{mileage} = 0.7$) and its 95% CI do not include 0, 0.5, and 1, there are slight differences in log-likelihood for price when $\lambda \in [0, 0.5]$ and for mileage when $\lambda \in [0.5, 1]$. For interpretability, we finally apply log-transformation on the price and keep the mileage in the data.
In the scatter plot of log(price) and mileage, we observe some potential outliers with low prices and short mileage.

```r
# Scatter plot of price vs. mileage with highlighted outliers
scatterPlotPriceMileage <- ggplot(data = dat_bmw2, mapping = aes(x = mileage, y = price)) +
  geom_point(pch=19, cex=0.3) +  # Default points
  geom_point(data = subset(dat_bmw2, mileage > 500000), mapping = aes(x = mileage, y = price), pch=19, c
  geom_point(data = subset(dat_bmw2, price > 100000), mapping = aes(x = mileage, y = price), pch=19, ce
  labs(title = "Scatter plot between price and mileage") +
  theme_bw() + theme(title = element_text(size=8))

# Box-Cox transformation for the price variable
boxcox_price <- boxcox(lm(dat_bmw2$price ~ 1), plotit = F)
boxcox_priceDF <- data.frame("lambda" = boxcox_price$x, "ll" = boxcox_price$y)
maxlabmda_price <- boxcox_price$x[which.max(boxcox_price$y)]

# Box-cox Plot for the price variable
boxcoxPlotPrice <- ggplot(data = boxcox_priceDF, mapping = aes(x = lambda, y = ll)) +
  geom_line(color = "black", linewidth = 0.5) +
  geom_vline(xintercept = maxlabmda_price, linetype = "dashed", color = "red") +
  geom_hline(yintercept = max(boxcox_price$y) - qchisq(0.95, df = 1)/2, linetype = "dashed", color = "bl
  labs(title = "Log-likelihood for the Box-Cox transformation",
       x = expression(lambda),
       y = "Log-likelihood for price") +
  annotate("text", label = paste0("lambda ==", maxlabmda_price), x = 1, y = -30000, parse = T) +
  theme_bw() + theme(title = element_text(size=8))

# Box-cox transformation for the mile variable
boxcox_mile <- boxcox(lm(dat_bmw2$mileage ~ 1), plotit = F)
boxcox_mileDF <- data.frame("lambda" = boxcox_mile$x, "ll" = boxcox_mile$y)
maxlabmda_mile <- boxcox_mile$x[which.max(boxcox_mile$y)]

# Box-cox Plot for the mile variable
boxcoxPlotMile <- ggplot(data = boxcox_mileDF, mapping = aes(x = lambda, y = ll)) +
  geom_line(color = "black", linewidth = 0.5) +
  geom_vline(xintercept = maxlabmda_mile, linetype = "dashed", color = "red") +
  geom_hline(yintercept = max(boxcox_mile$y) - qchisq(0.95, df = 1)/2, linetype = "dashed", color = "blu
  labs(title = "Log-likelihood for the Box-Cox transformation",
       x = expression(lambda),
       y = "Log-likelihood for mileage") +
  annotate("text", label = paste0("lambda ==", maxlabmda_mile), x = 1, y = -30000, parse = T) +
  theme_bw() + theme(title = element_text(size=8))

# Create a column of logprice in dat_bmw2
dat_bmw2 <- dat_bmw2 %>% mutate(logprice = log(price))

# Scatter plot of logprice vs. mileage with highlighted outliers
scatterPlotLogPriceMileage <- ggplot(data = dat_bmw2, mapping = aes(x = mileage, y = logprice)) +
  geom_point(pch=19, cex=0.3) +
  geom_point(data = subset(dat_bmw2, mileage > 500000), mapping = aes(x = mileage, y = logprice), pch=1
  geom_point(data = subset(dat_bmw2, logprice > log(100000)), mapping = aes(x = mileage, y = logprice),
  labs(title = "Scatter plot between log(price) and mileage") +
  theme_bw() + theme(title = element_text(size=8))

plot_grid(scatterPlotPriceMileage, boxcoxPlotPrice, boxcoxPlotMile, scatterPlotLogPriceMileage, align =
```


Scatter plot between price and mileage
Log−likelihood for the Box−Cox transform
Log−likelihood for the Box−Cox transform
Scatter plot between log(price) and mileage

Outliers

Before, we fit the model, we removed the outliers we observed from the scatter plot (price > 100000 and mileage > 500000). We reasonably conclude that these points are errors in data entry that skew the model too heavily.

```
modelDF <- dat_bmw2
### Assigning NA to car with mileage > 50,0000 miles
modelDF[modelDF$mileage > 500000, "mileage"] <- NA
### Assigning NA to car with price > 100000 usd
modelDF[modelDF$price > 100000, "price"] <- NA
```

```r
modelDF <- modelDF[complete.cases(modelDF),]
modelDF$logprice <- log(modelDF$price)
```

Model Fitting

We applied an Ordinary Least Square regression model to predict the natural logarithm of BMW car prices from mileage. The model may be represented as:

$$log(price) = \beta_0 + \beta_1 \times mileage + \epsilon$$

The residuals represent the difference between the model-predicted values and the actual log(price) values. Ideally, they should be symmetrically distributed around zero for a well-behaved model. However, the minimum of residual is –5.1907. That means that, for at least one observation, the model-predicted log(price) is much larger than the actual value.

The Coefficients section provides us with detailed statistics for the intercept and the mileage coefficient. The intercept is estimated to be 10.17 with very small standard error (0.02175) and thus having an extremely large t-value of 467.7, corresponding with a p-value of less than 2×10^{-16}. The test provides strong evidence against the null hypothesis that the intercept is zero. Similarly, mileage coefficient is also estimated as ˘4.746×10^{-06}, indicating that for one unit increase in mileage, logprice decreases by about 4.746×10^{-06} units. Its standard error is 1.425×10^{-07}, which yields a t-value of –33.3 and a p-value similarly less than 2×10^{-16}. These allow us to reject the null hypothesis for both coefficients, meaning mileage is a statistically significant predictor of log(price).

Besides the individual coefficients, the fit of the entire model is also tested by means of the F-test. The F-statistic of 1109 with degrees of freedom 1 and 4837, and p-value smaller than 2.2×10^{-16} tests whether the model that includes mileage as a predictor explains significantly more variation in log(price) than does a model with no predictors. This result confirms that our model as a whole is statistically significant although mileage by itself explains only about 18.65% of the variation in log(price) (as indicated by the Multiple R-squared of 0.1865). The relatively low R-squared value suggests that while mileage is a significant predictor, other variables likely contribute to the variation in car prices.

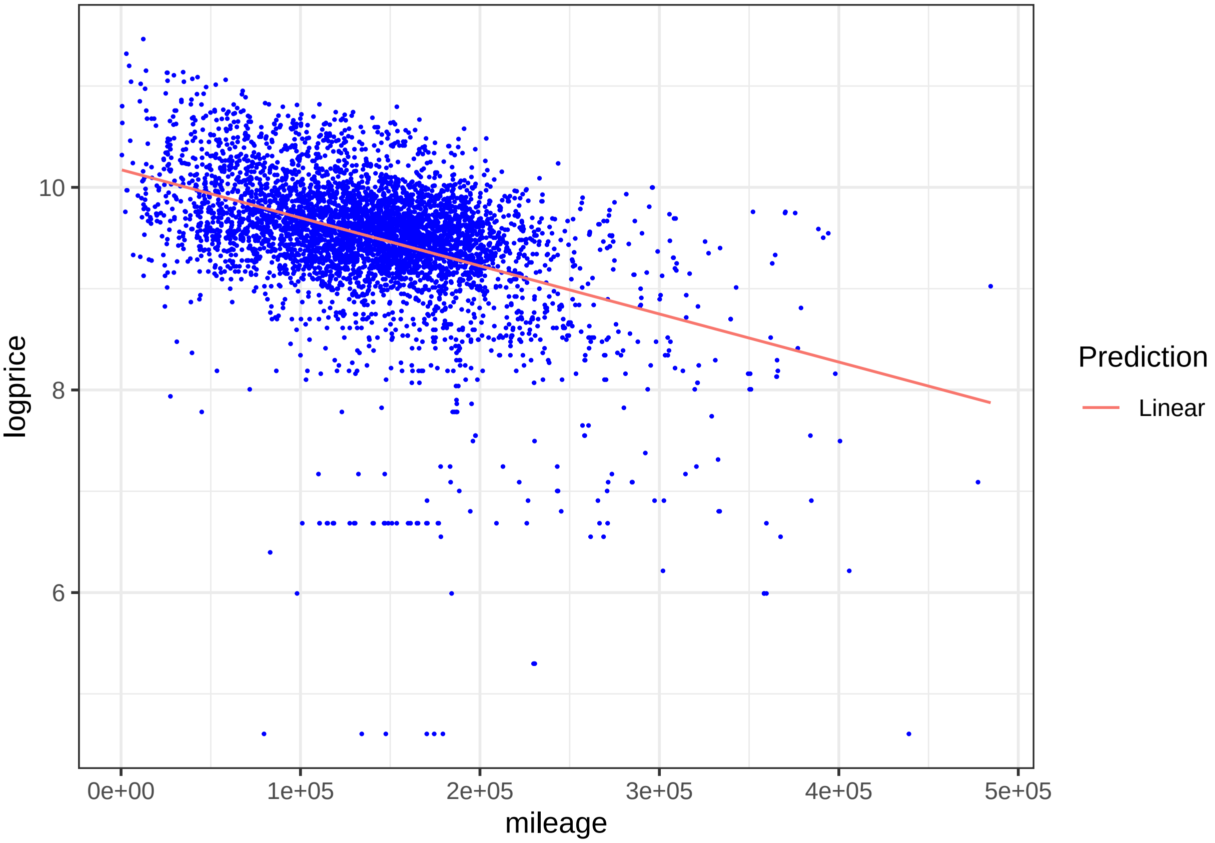
```r
mod1 <- lm(logprice ~ mileage, data = modelDF)
summary(mod1)
```

```
##
## Call:
## lm(formula = logprice ~ mileage, data = modelDF)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1907 -0.2404  0.0496  0.3229  1.3558
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.017e+01  2.175e-02   467.7   <2e-16 ***
## mileage     -4.746e-06  1.425e-07   -33.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5838 on 4837 degrees of freedom
## Multiple R-squared:  0.1865, Adjusted R-squared:  0.1864
## F-statistic:  1109 on 1 and 4837 DF,  p-value: < 2.2e-16
```

Scatter Plot with Best fit Line

In the plot below, we overlay the scatter plot of mileage vs price with the linear regression line. We can see that as mileage increases, log-price will decrease overall. But there is significant noise around the best-fit line. Particularly, there are a few points at the low end of log-price (close to 5–6 on the y-axis) that are rather far away from the rest of the data—these points are potential high-leverage points for the regression because they are far out in mileage and have exceptionally low log-prices. They can pull the regression line down more than if they were not present. Despite those outliers, the linear fit still shows the general inverse relation (higher mileage → lower log-price), but it does not explain all of the price variability, given the broad scatter of points. So we can conclude that simple linear regression is not a good fit here. A multiple regression model would be a better choice as it would account for the variation in price that is not accounted for by mileage alone.

```r
# Create a scatter plot with the best-fit line
# Add predicted values to the dataset
modelDF$predLinear <- predict(mod1)

# Create the scatter plot with the best-fit line
ggplot(modelDF, aes(x = mileage, y = logprice)) +
  geom_point(size = 0.15, color = 'blue') +                          # Scatter plot points
  geom_line(aes(x = mileage, y = predLinear, color = "blue")) +      # Best-fit line from predictions
  scale_color_discrete(name = "Prediction", labels = c("Linear")) + # Legend for the line
  theme_bw()
```



Contribution

Nathan Pulsone: wrote the intro variable selection section

Zexian Xu: revised Report 1 according to comments from TA

Mengyao Wang: draw and interpreted scatterplot matrix and scatter plot of price and mileage

Xiaojing Yang: revised and edit the analysis for report 2

Kaiyi Liu: Fitted the linear model, and wrote the interpretation for it.