Project 1 - Data Preparation, Exploration & Partitioning

Utkrist P. Thapa '21

2021-03-08

Load Packages

```
library(tidyverse)
library(rsample)
library(kableExtra)
library(lubridate)
library(reshape)
library(recipes)
```

Load Data

```
bike_data <- read_csv("data/bike_share_day.csv")
car_data <- read_csv("data/car_sales_summer_2014.csv")
```

Part 1 - Bike Rentals in Washington, DC

Question 1

I piped my dataframe bike_data into heasd(50) to get the first 50 rows. I used scroll_box() from kable() to create a scrollable window in order to accomodate the 50 rows without showing all 50 at the same time. Then, I printed out the dimensions and statistical summary of bike_data. I summarise the calculated the sum of is.na(bike_data) across all variables in bike_data and display the results. Since is.na() returns a boolean type (0 or 1 for FALSE or TRUE), we can sum these up to see how many missing values occur in the dataframe.

```
bike_data %>%
  head(50) %>%
  kable() %>%
  kable_styling() %>%
  scroll_box(width = "50%", height = "600px")
```

instant	dteday	season	yr	mnth	holiday	weekday
1	1/1/11	1	0	1	0	6
2	1/2/11	1	0	1	0	0
3	1/3/11	1	0	1	0	1
4	1/4/11	1	0	1	0	2
5	1/5/11	1	0	1	0	3
6	1/6/11	1	0	1	0	4
7	1/7/11	1	0	1	0	5
8	1/8/11	1	0	1	0	6
9	1/9/11	1	0	1	0	0
10	1/10/11	1	0	1	0	1
11	1/11/11	1	0	1	0	2
12	1/12/11	1	0	1	0	3
13	1/13/11	1	0	1	0	4
14	1/14/11	1	0	1	0	5
15	1/15/11	1	0	1	0	6

```
print(dim(bike_data))
```

```
## [1] 731  16
```

```
summary(bike_data)
```

```
##     instant         dteday              season           yr        
##  Min.   :  1.0   Length:731         Min.   :1.000   Min.   :0.0000  
##  1st Qu.:183.5   Class :character   1st Qu.:2.000   1st Qu.:0.0000  
##  Median :366.0   Mode  :character   Median :3.000   Median :1.0000  
##  Mean   :366.0                      Mean   :2.497   Mean   :0.5007  
##  3rd Qu.:548.5                      3rd Qu.:3.000   3rd Qu.:1.0000  
##  Max.   :731.0                      Max.   :4.000   Max.   :1.0000  
##       mnth          holiday           weekday        workingday    
##  Min.   : 1.00   Min.   :0.00000   Min.   :0.000   Min.   :0.000  
##  1st Qu.: 4.00   1st Qu.:0.00000   1st Qu.:1.000   1st Qu.:0.000  
##  Median : 7.00   Median :0.00000   Median :3.000   Median :1.000  
##  Mean   : 6.52   Mean   :0.02873   Mean   :2.997   Mean   :0.684  
##  3rd Qu.:10.00   3rd Qu.:0.00000   3rd Qu.:5.000   3rd Qu.:1.000  
##  Max.   :12.00   Max.   :1.00000   Max.   :6.000   Max.   :1.000  
##    weathersit         temp             atemp              hum        
##  Min.   :1.000   Min.   :0.05913   Min.   :0.07907   Min.   :0.0000  
##  1st Qu.:1.000   1st Qu.:0.33708   1st Qu.:0.33784   1st Qu.:0.5200  
##  Median :1.000   Median :0.49833   Median :0.48673   Median :0.6267  
##  Mean   :1.395   Mean   :0.49538   Mean   :0.47435   Mean   :0.6279  
##  3rd Qu.:2.000   3rd Qu.:0.65542   3rd Qu.:0.60860   3rd Qu.:0.7302  
##  Max.   :3.000   Max.   :0.86167   Max.   :0.84090   Max.   :0.9725  
##    windspeed          casual         registered        cnt      
##  Min.   :0.02239   Min.   :   2.0   Min.   :  20   Min.   :  22  
##  1st Qu.:0.13495   1st Qu.: 315.5   1st Qu.:2497   1st Qu.:3152  
##  Median :0.18097   Median : 713.0   Median :3662   Median :4548  
##  Mean   :0.19049   Mean   : 848.2   Mean   :3656   Mean   :4504  
##  3rd Qu.:0.23321   3rd Qu.:1096.0   3rd Qu.:4776   3rd Qu.:5956  
##  Max.   :0.50746   Max.   :3410.0   Max.   :6946   Max.   :8714
```

```
bike_data %>%
  summarise(across(everything(), ~ sum(is.na(.)))) %>%
  kable() %>%
  kable_styling() %>%
  scroll_box(height = "100px", width = "600px")
```

instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit
0	0	0	0	0	0	0	0	0

There are no missing values in this dataset.

Question 2

I have factored season, holiday, workingday and weathersit variables with appropriate levels. I have rearranged the level order of season variable in order to make spring the baseline level. I have used glimpse() function to take a quick look at what bike_data looks like.

```
bike_data$season <- factor(bike_data$season)
levels(bike_data$season)[1:4] <- c("winter", "spring", "summer", "fall")

bike_data$season <- factor(bike_data$season, levels = c("spring", "summer", "fall", "winter"))

bike_data$holiday <- factor(bike_data$holiday)
levels(bike_data$holiday)[1:2] <- c("no", "yes")

bike_data$workingday <- factor(bike_data$workingday)
levels(bike_data$workingday)[1:2] <- c("no", "yes")

bike_data$weathersit <- factor(bike_data$weathersit)
levels(bike_data$weathersit)[1:4] <- c("clear", "mist", "light precipitation", "heavy precipitation")

glimpse(bike_data)
```

```
## Rows: 731
## Columns: 16
## $ instant    <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,…
## $ dteday     <chr> "1/1/11", "1/2/11", "1/3/11", "1/4/11", "1/5/11", "1/6/11"…
## $ season     <fct> winter, winter, winter, winter, winter, winter, winter, wi…
## $ yr         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …
## $ mnth       <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …
## $ holiday    <fct> no, no, no, no, no, no, no, no, no, no, no, no, no, no, no…
## $ weekday    <dbl> 6, 0, 1, 2, 3, 4, 5, 6, 0, 1, 2, 3, 4, 5, 6, 0, 1, 2, 3, 4…
## $ workingday <fct> no, no, yes, yes, yes, yes, yes, no, no, yes, yes, yes, ye…
## $ weathersit <fct> mist, mist, clear, clear, clear, clear, mist, mist, clear,…
## $ temp       <dbl> 0.3441670, 0.3634780, 0.1963640, 0.2000000, 0.2269570, 0.2…
## $ atemp      <dbl> 0.3636250, 0.3537390, 0.1894050, 0.2121220, 0.2292700, 0.2…
## $ hum        <dbl> 0.805833, 0.696087, 0.437273, 0.590435, 0.436957, 0.518261…
## $ windspeed  <dbl> 0.1604460, 0.2485390, 0.2483090, 0.1602960, 0.1869000, 0.0…
## $ casual     <dbl> 331, 131, 120, 108, 82, 88, 148, 68, 54, 41, 43, 25, 38, 5…
## $ registered <dbl> 654, 670, 1229, 1454, 1518, 1518, 1362, 891, 768, 1280, 12…
## $ cnt        <dbl> 985, 801, 1349, 1562, 1600, 1606, 1510, 959, 822, 1321, 12…
```

Question 3

I have used the information available in the codebook to calculate the values for these new variables. I have added these new columns (variables) to bike_data.

```
bike_data <- bike_data %>%
  mutate(raw_temp = 41 * temp) %>%
  mutate(feel_temp = 50 * atemp) %>%
  mutate(humidity = 100 * hum) %>%
  mutate(cal_windspeed = 67 * windspeed)
```

Question 4

Here, I have calculated the difference between the sum of all values in the variable casual and registered, against the variable cnt. If the difference is zero, that must mean that the values in variables casual and registered must add up to the values in cnt.

```
sum(bike_data$casual + bike_data$registered
    - bike_data$cnt) == 0
```

```
## [1] TRUE
```

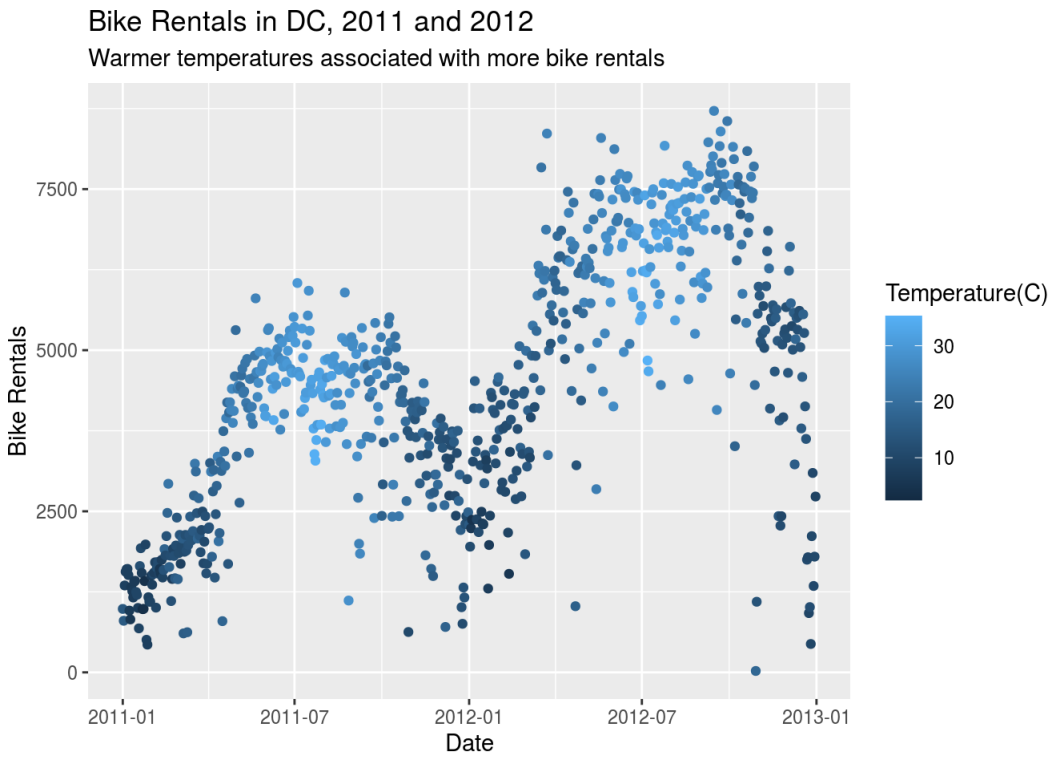
Question 5

First, I converted the date column into a Date object type. I have used the newly created variable raw_temp in order to color the scatterplot as shown in the figure. I have limited the date (xlim) from beginning of 2011 to the end of 2012 according to the figure given to us.

```
bike_data <- bike_data %>%
  mutate(dteday = mdy(dteday))

bike_data %>%
  ggplot(mapping = aes(x = dteday, y = cnt, color = bike_data$raw_temp)) +
  geom_point(show.legend = TRUE) +
  labs(title = "Bike Rentals in DC, 2011 and 2012",
       subtitle = "Warmer temperatures associated with more bike rentals",
       x = "Date",
       y = "Bike Rentals",
       color = "Temperature(C)") +
  xlim(c(ydm("2011-01-01"), ymd("2012-31-12")))
```

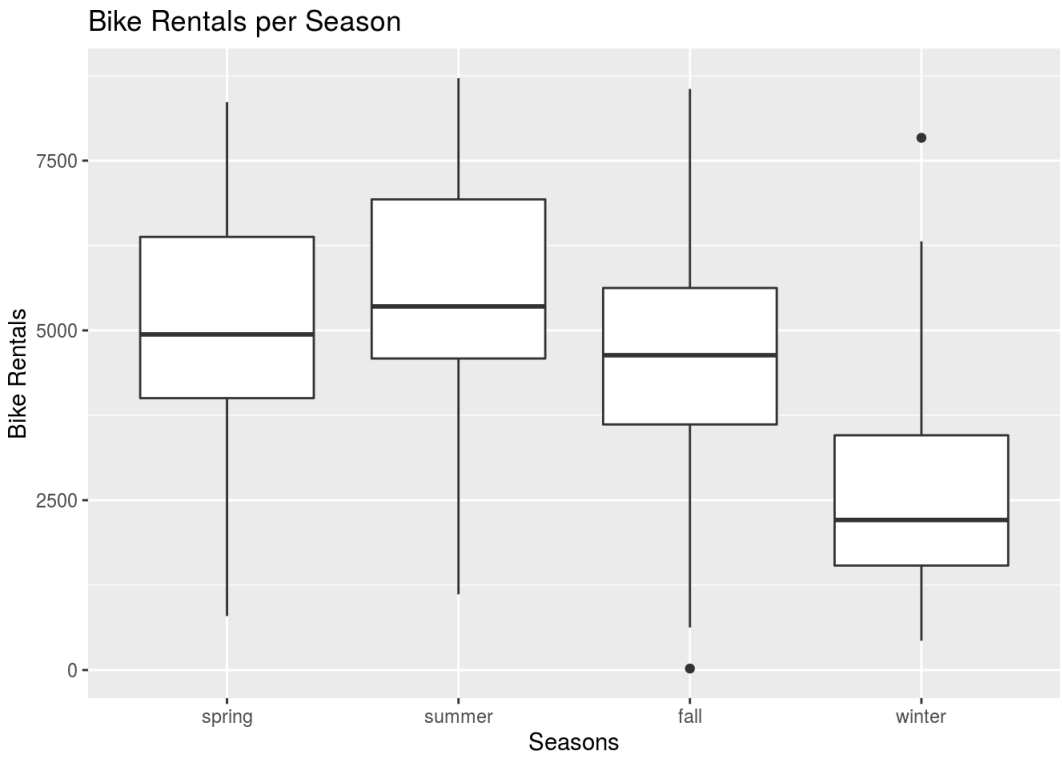
```
## Warning: All formats failed to parse. No formats found.
```



Question 6

I decided to visualize this data via boxplots. The visualization clearly expresses what the scatterplot previously stated: higher amount of bike rentals are associated with warmer temperatures since the median for the first three boxplots are higher than winter.

```
bike_data %>%
  ggplot(aes(x = season, y = cnt)) +
  geom_boxplot() +
  labs(title = "Bike Rentals per Season",
       x = "Seasons",
       y = "Bike Rentals")
```



Question 7

I have used the rsample package in order to partition the data into training and test sets. Then I have used kable() in order to display the partitions in a nice way. In order to make the partition, I set the seed and then use initial_split() function from rsample package to make splits according to a specified proportion.

```
set.seed(2021)
bike_data_split <- bike_data %>%
  initial_split((prop = 0.75))
bike_train <- training(bike_data_split)
bike_test <- testing(bike_data_split)

# displaying the partitions
print(paste("Rows: ", nrow(bike_train)))
```

```
## [1] "Rows:  549"
```

```
bike_train %>%
  head(10) %>%
  kable() %>%
  kable_styling() %>%
  scroll_box(height = "50%", width = "600px")
```

instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit
1	2011-01-01	winter	0	1	no	6	no	mist
2	2011-01-02	winter	0	1	no	0	no	mist
3	2011-01-03	winter	0	1	no	1	yes	clear
4	2011-01-04	winter	0	1	no	2	yes	clear
5	2011-01-05	winter	0	1	no	3	yes	clear
6	2011-01-06	winter	0	1	no	4	yes	clear
8	2011-01-08	winter	0	1	no	6	no	mist
9	2011-01-09	winter	0	1	no	0	no	clear
10	2011-01-10	winter	0	1	no	1	yes	clear
11	2011-01-11	winter	0	1	no	2	yes	mist

```
print(paste("Rows: ", nrow(bike_test)))
```

```
## [1] "Rows:  182"
```

```
bike_test %>%
  head(10) %>%
  kable() %>%
  kable_styling() %>%
  scroll_box(height = "50%", width = "600px")
```

instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit
7	2011-01-07	winter	0	1	no	5	yes	mist
13	2011-01-13	winter	0	1	no	4	yes	clear
20	2011-01-20	winter	0	1	no	4	yes	mist
27	2011-01-27	winter	0	1	no	4	yes	clear
28	2011-01-28	winter	0	1	no	5	yes	mist
41	2011-02-10	winter	0	2	no	4	yes	clear
43	2011-02-12	winter	0	2	no	6	no	clear
48	2011-02-17	winter	0	2	no	4	yes	clear
49	2011-02-18	winter	0	2	no	5	yes	clear
57	2011-02-26	winter	0	2	no	6	no	clear

Part 2: Toyota Corolla Dataset

Question 8

I decided to create a reduced subset by using the subset variable names.

```
car_subset <- car_data[c("Id", "Model", "Price", "Age_08_04", "KM",
                         "Fuel_Type", "HP", "Met_Color", "Automatic",
                         "cc", "Doors", "Quarterly_Tax", "Weight")]
glimpse(car_subset)
```

```
## Rows: 1,436
## Columns: 13
## $ Id            <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, …
## $ Model         <chr> "TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors", "TOYOT…
## $ Price         <dbl> 13500, 13750, 13950, 14950, 13750, 12950, 16900, 18600,…
## $ Age_08_04     <dbl> 23, 23, 24, 26, 30, 32, 27, 30, 27, 23, 25, 22, 25, 31,…
## $ KM            <dbl> 46986, 72937, 41711, 48000, 38500, 61000, 94612, 75889,…
## $ Fuel_Type     <chr> "Diesel", "Diesel", "Diesel", "Diesel", "Diesel", "Dies…
## $ HP            <dbl> 90, 90, 90, 90, 90, 90, 90, 90, 192, 69, 192, 192, 192,…
## $ Met_Color     <dbl> 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0…
## $ Automatic     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ cc            <dbl> 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 1800, 1…
## $ Doors         <dbl> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3…
## $ Quarterly_Tax <dbl> 210, 210, 210, 210, 210, 210, 210, 210, 100, 185, 100, …
## $ Weight        <dbl> 1165, 1165, 1165, 1165, 1170, 1170, 1245, 1245, 1185, 1…
```

Question 9

I have displayed the dimensions and the statistical summary of the car_subset dataframe. Then, I summarise the calculated the sum of is.na(car_subset) across all variables in car_subset and display the results. Since is.na() returns a boolean type (0 or 1 for FALSE or TRUE), we can sum these up to see how many missing values occur in the dataframe.

```
print(dim(car_subset))
```

```
## [1] 1436   13
```

```
summary(car_subset)
```

```
##        Id            Model                Price          Age_08_04
##  Min.   :   1.0   Length:1436        Min.   : 4350   Min.   : 1.00
##  1st Qu.: 361.8   Class :character   1st Qu.: 8450   1st Qu.:44.00
##  Median : 721.5   Mode  :character   Median : 9900   Median :61.00
##  Mean   : 721.6                      Mean   :10731   Mean   :55.95
##  3rd Qu.:1081.2                      3rd Qu.:11950   3rd Qu.:70.00
##  Max.   :1442.0                      Max.   :32500   Max.   :80.00
##        KM          Fuel_Type              HP          Met_Color
##  Min.   :     1   Length:1436        Min.   : 69.0   Min.   :0.0000
##  1st Qu.: 43000   Class :character   1st Qu.: 90.0   1st Qu.:0.0000
##  Median : 63390   Mode  :character   Median :110.0   Median :1.0000
##  Mean   : 68533                      Mean   :101.5   Mean   :0.6748
##  3rd Qu.: 87021                      3rd Qu.:110.0   3rd Qu.:1.0000
##  Max.   :243000                      Max.   :192.0   Max.   :1.0000
##    Automatic             cc            Doors        Quarterly_Tax
##  Min.   :0.00000   Min.   : 1300   Min.   :2.000   Min.   : 19.00
##  1st Qu.:0.00000   1st Qu.: 1400   1st Qu.:3.000   1st Qu.: 69.00
##  Median :0.00000   Median : 1600   Median :4.000   Median : 85.00
##  Mean   :0.05571   Mean   : 1577   Mean   :4.033   Mean   : 87.12
##  3rd Qu.:0.00000   3rd Qu.: 1600   3rd Qu.:5.000   3rd Qu.: 85.00
##  Max.   :1.00000   Max.   :16000   Max.   :5.000   Max.   :283.00
##      Weight
##  Min.   :1000
##  1st Qu.:1040
##  Median :1070
##  Mean   :1072
##  3rd Qu.:1085
##  Max.   :1615
```

```
car_subset %>%
  summarise(across(everything(), ~ sum(is.na(.)))) %>%
  kable() %>%
  kable_styling() %>%
  scroll_box(height = "100px", width = "600px")
```

Id	Model	Price	Age_08_04	KM	Fuel_Type	HP	Met_Color	Automatic	cc
0	0	0	0	0	0	0	0	0	0

There are no missing values in this dataset.

Question 10

I summarise the calculated sum of boolean type output from str_detect() across all variables. This helps us see how many extraneous question marks occur in each of the variables. Then, I mutate the Model variable using str_remove() in order to remove the question marks.

```
car_subset %>%
  summarise(across(everything(), ~ sum(str_detect(., pattern = "\\?")))) %>%
  kable() %>%
  kable_styling() %>%
  scroll_box(height = "100px", width = "600px")
```

Id	Model	Price	Age_08_04	KM	Fuel_Type	HP	Met_Color	Automatic	cc
0	147	0	0	0	0	0	0	0	0

```
# removing the extraneous question marks
car_subset <- car_subset %>%
  mutate(Model = str_remove(Model, pattern = "\\?"))

car_subset %>%
  head(50) %>%
  kable() %>%
  kable_styling() %>%
  scroll_box(width = "70%", height = "600px")
```

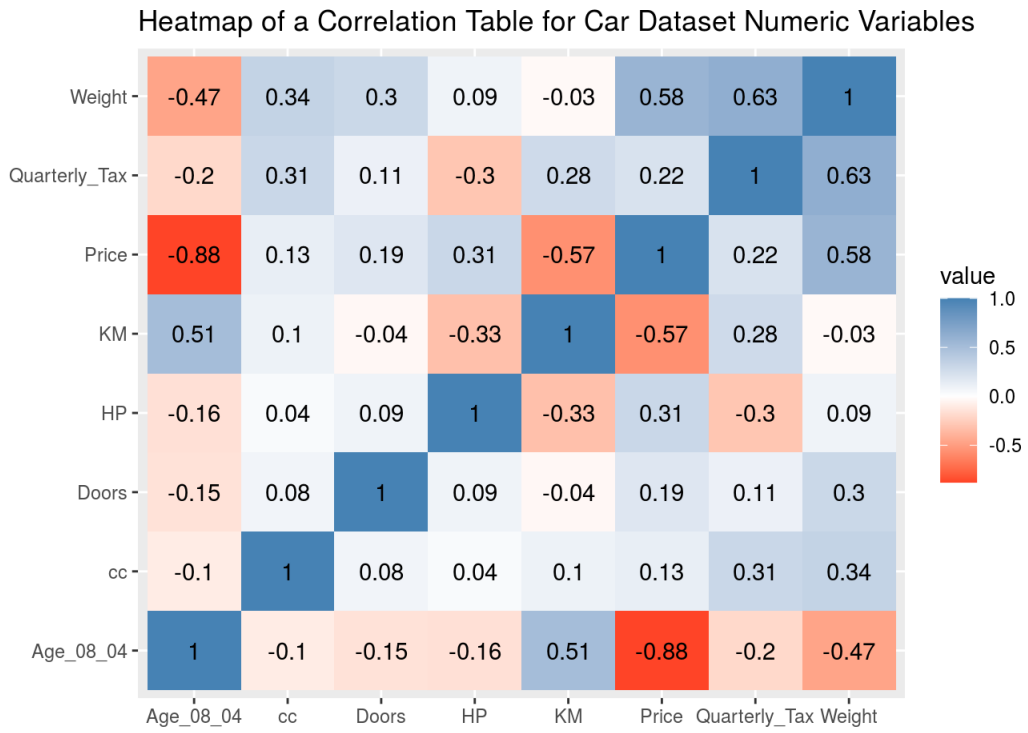
Id	Model	Price	Age_08_04	KM	Fuel_Type	HP	Met_Color	Autom
1	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13500	23	46986	Diesel	90	1	
2	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13750	23	72937	Diesel	90	1	
3	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13950	24	41711	Diesel	90	1	
4	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	14950	26	48000	Diesel	90	0	

There are no other variables with extraneous question marks aside from model. There are 147 cases of extraneous question marks in the variable model.

Question 11

The price and age are heavily negatively correlated. Similarly, age and weight seem to be negatively correlated as well. Weight is highly correlated with quarterly_tax. Similarly, price and weight are highly correlated as well.

```
cor_matrix <- round(cor(car_subset[c(3, 4, 5, 7, 10, 11, 12, 13)]), 2)
melt(cor_matrix) %>%
  ggplot(aes(x = X1, y = X2, fill = value)) +
  geom_tile() +
  geom_text(aes(x = X1, y = X2, label = value)) +
  scale_fill_gradient2(low = "red", high = "steelblue", guide = "colorbar") +
  labs(title = "Heatmap of a Correlation Table for Car Dataset Numeric Variables",
       x = "", y = "")
```



Question 12

I factor Met_Color and Automatic in order to avoid problems with step_dummy(). I have used recipes package in order to create the dummy variables for the three categorical variables. I preserve the original variables and simply add the new dummy variables to the dataframe. Following the syntax of the recipes package, I pipe car_subset to recipe(), then create the dummy variables using step_dummy(), prep this step using prep() and finally apply it to the dataframe using bake().

```
car_subset <- car_subset %>%
  mutate(Met_Color = factor(Met_Color, levels = c(0, 1))) %>%
  mutate(Automatic = factor(Automatic, levels = c(0, 1)))

car_dummy <- car_subset %>%
  recipe(~ .) %>%
  step_dummy(Fuel_Type, Met_Color, Automatic, one_hot = TRUE, preserve = TRUE) %>%
  prep(training = car_subset) %>%
  bake(new_data = car_subset)
glimpse(car_dummy)
```

```
## Rows: 1,436
## Columns: 20
## $ Id                <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1…
## $ Model             <fct> TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors, TOYOT…
## $ Price             <dbl> 13500, 13750, 13950, 14950, 13750, 12950, 16900, 186…
## $ Age_08_04         <dbl> 23, 23, 24, 26, 30, 32, 27, 30, 27, 23, 25, 22, 25, …
## $ KM                <dbl> 46986, 72937, 41711, 48000, 38500, 61000, 94612, 758…
## $ Fuel_Type         <fct> Diesel, Diesel, Diesel, Diesel, Diesel, Diesel, Dies…
## $ HP                <dbl> 90, 90, 90, 90, 90, 90, 90, 90, 192, 69, 192, 192, 1…
## $ Met_Color         <fct> 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1…
## $ Automatic         <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ cc                <dbl> 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 1800…
## $ Doors             <dbl> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3…
## $ Quarterly_Tax     <dbl> 210, 210, 210, 210, 210, 210, 210, 210, 100, 185, 10…
## $ Weight            <dbl> 1165, 1165, 1165, 1165, 1170, 1170, 1245, 1245, 1185…
## $ Fuel_Type_CNG     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ Fuel_Type_Diesel  <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0…
## $ Fuel_Type_Petrol  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1…
## $ Met_Color_X0      <dbl> 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0…
## $ Met_Color_X1      <dbl> 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1…
## $ Automatic_X0      <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
## $ Automatic_X1      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
```

Question 13

I have partitioned the car_subset dataframe using the rsample package. I set the random seed, and initially split the car_subset into 50-50. Then I take the second half, and split it into 60-40 splits in order to get the final 50-30-20 train, validation and test splits.

```
set.seed(2021)
car_split <- car_dummy %>%
  initial_split(prop = 0.5)
car_train <- training(car_split)
car_rest <- testing(car_split)

# splitting the rest of the car data into testing and validation sets
set.seed(2021)
car_val_split <- car_rest %>%
  initial_split(prop = 0.6)

car_val <- training(car_val_split)
car_test <- testing(car_val_split)

glimpse(car_train)
```

```
## Rows: 718
## Columns: 20
## $ Id                <dbl> 1, 2, 4, 5, 6, 9, 11, 13, 14, 15, 19, 21, 22, 25, 26…
## $ Model             <fct> TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors, TOYOT…
## $ Price             <dbl> 13500, 13750, 14950, 13750, 12950, 21500, 20950, 196…
## $ Age_08_04         <dbl> 23, 23, 26, 30, 32, 27, 25, 25, 31, 32, 24, 30, 29, …
## $ KM                <dbl> 46986, 72937, 48000, 38500, 61000, 19700, 31461, 321…
## $ Fuel_Type         <fct> Diesel, Diesel, Diesel, Diesel, Diesel, Petrol, Petr…
## $ HP                <dbl> 90, 90, 90, 90, 90, 192, 192, 192, 192, 192, 110, 11…
## $ Met_Color         <fct> 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1…
## $ Automatic         <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0…
## $ cc                <dbl> 2000, 2000, 2000, 2000, 2000, 1800, 1800, 1800, 1800…
## $ Doors             <dbl> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3…
## $ Quarterly_Tax     <dbl> 210, 210, 210, 210, 210, 100, 100, 100, 100, 100, 19…
## $ Weight            <dbl> 1165, 1165, 1165, 1170, 1170, 1185, 1185, 1185, 1185…
## $ Fuel_Type_CNG     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ Fuel_Type_Diesel <dbl> 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ Fuel_Type_Petrol <dbl> 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
## $ Met_Color_X0      <dbl> 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0…
## $ Met_Color_X1      <dbl> 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1…
## $ Automatic_X0      <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1…
## $ Automatic_X1      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0…
```

```
glimpse(car_val)
```

```
## Rows: 431
## Columns: 20
## $ Id                <dbl> 3, 7, 8, 10, 12, 16, 20, 23, 27, 29, 33, 34, 36, 41,…
## $ Model             <fct> TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors, TOYOT…
## $ Price             <dbl> 13950, 16900, 18600, 12950, 19950, 22000, 16950, 159…
## $ Age_08_04         <dbl> 24, 27, 30, 23, 22, 28, 30, 28, 27, 28, 27, 26, 26, …
## $ KM                <dbl> 41711, 94612, 75889, 71138, 43610, 18739, 64359, 563…
## $ Fuel_Type         <fct> Diesel, Diesel, Diesel, Diesel, Petrol, Petrol, Petr…
## $ HP                <dbl> 90, 90, 90, 69, 192, 192, 110, 110, 110, 110, 97, 97…
## $ Met_Color         <fct> 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1…
## $ Automatic         <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ cc                <dbl> 2000, 2000, 2000, 1900, 1800, 1800, 1600, 1600, 1600…
## $ Doors             <dbl> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5…
## $ Quarterly_Tax     <dbl> 210, 210, 210, 185, 100, 100, 85, 85, 85, 85, 85, 85…
## $ Weight            <dbl> 1165, 1245, 1245, 1105, 1185, 1185, 1105, 1120, 1120…
## $ Fuel_Type_CNG     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ Fuel_Type_Diesel <dbl> 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0…
## $ Fuel_Type_Petrol <dbl> 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1…
## $ Met_Color_X0      <dbl> 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0…
## $ Met_Color_X1      <dbl> 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1…
## $ Automatic_X0      <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
## $ Automatic_X1      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
```

```
glimpse(car_test)
```

```
## Rows: 287
## Columns: 20
## $ Id               <dbl> 17, 18, 24, 28, 44, 54, 55, 66, 77, 80, 81, 90, 92, …
## $ Model            <fct> TOYOTA Corolla 1.8 16V VVTLI 3DR T SPORT 2/3-Doors, …
## $ Price            <dbl> 22750, 17950, 16950, 15750, 16950, 21950, 15500, 169…
## $ Age_08_04        <dbl> 30, 24, 28, 29, 27, 27, 25, 26, 31, 30, 25, 19, 20, …
## $ KM               <dbl> 34000, 21716, 32220, 41415, 110404, 49866, 49163, 32…
## $ Fuel_Type        <fct> Petrol, Petrol, Petrol, Petrol, Diesel, Petrol, Petr…
## $ HP               <dbl> 192, 110, 110, 110, 90, 192, 110, 110, 110, 97, 110,…
## $ Met_Color        <fct> 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
## $ Automatic        <fct> 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0…
## $ cc               <dbl> 1800, 1600, 1600, 1600, 2000, 1800, 1600, 1600, 1600…
## $ Doors            <dbl> 3, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 3, 3, 3, 3, 3, 3, 3…
## $ Quarterly_Tax    <dbl> 100, 85, 85, 85, 234, 100, 100, 19, 85, 85, 100, 234…
## $ Weight           <dbl> 1185, 1105, 1120, 1120, 1255, 1195, 1165, 1075, 1130…
## $ Fuel_Type_CNG    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ Fuel_Type_Diesel <dbl> 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0…
## $ Fuel_Type_Petrol <dbl> 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1…
## $ Met_Color_X0     <dbl> 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ Met_Color_X1     <dbl> 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
## $ Automatic_X0     <dbl> 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1…
## $ Automatic_X1     <dbl> 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0…
```

The training partition is used for training the model. The validation partition is then used to test the model with unseen data in order to gauge underfitting/overfitting. Validation partitions can also be used to tweak the parameters of a model. The test partition is used to evaluate the performance of the model with new data.

Project Log

I have only used materials posted to the class website as well as my personal notes and past assignments.

The Pledge

On my honor, I have neither given nor received any unacknowledged aid on this project.

Utkrist P. Thapa

March 8, 2021, Monday