Lab 2- Data Exploration and Analysis

Learning Outcomes

At the end of the session, you will be able to:

- Explore and visualize a dataset using basic R code.
- Create insight and analysis for the dataset

Activity 1 – Data Exploration

1. Go to Kaggle.com and download some historical data in CSV file. You may start with a sample data provided, Churn_Train.csv.
2. Install `tidyverse` , `dplyr` and `dlookr` libraries.
3. Perform univariate analysis.
   - Calculating descriptive statistics using `describe()`
   - Test of normality on numeric variables using `normality()`
   - Visualization of normality of numerical variables using `plot_normality()`

4. Perform bivariate/multivariate analysis.
   - Calculation of correlation coefficient using `correlate()`
   - Visualization of the correlation matrix using `plot.correlate()`

5. Perform EDA based on target variable

   - To perform EDA based on target variable, you need to create a `target_by` class object. `target_by()` creates a `target_by` class with an object inheriting data.frame or data.frame. `target_by()` is similar to `group_by()` in `dplyr` which creates `grouped_df`. The difference is that you specify only one variable.

     ```
     categ <- target_by(data,Category)
     ```

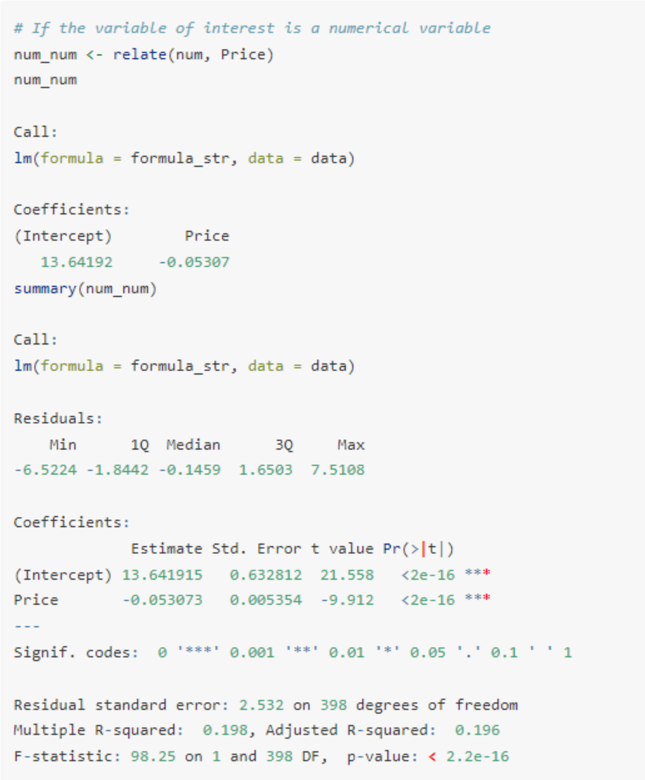
   - If the variable of interest is a numerical variable, you can use `relate()` to show the relationship between the target variable and the predictor.

     ```
     cat_num <- relate(categ, Sales)
     cat_num
     summary(cat_num)
     ```

   - Visualize the relate class object created by `relate()`. The relationship between target and selected variable (predictor) is visualized by *density plot*.
     `plot(cat_num)`

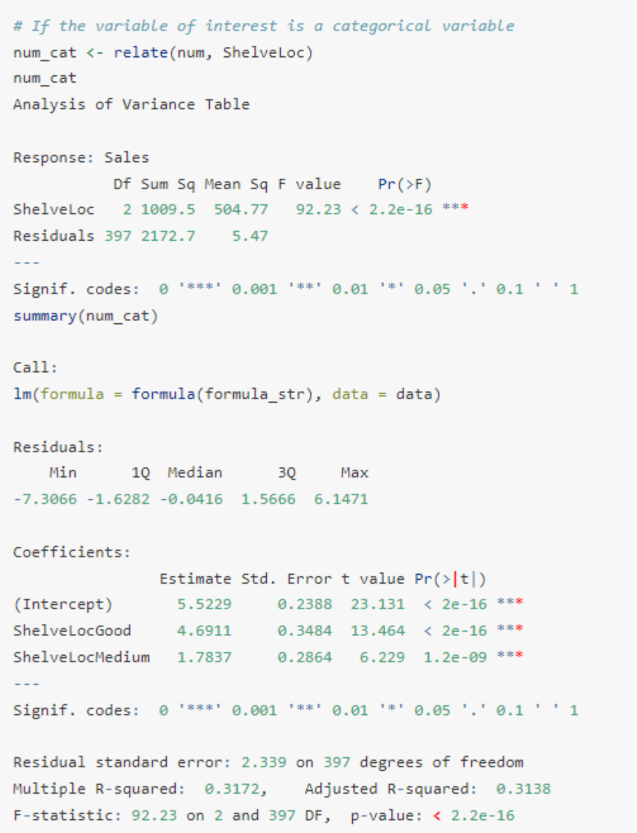
   - Cases where predictors are categorical variable, The relationship between target and predictor is represented by a *mosaics plot.*

   - If EDA when target variable is numerical variable and predictors are numeric variables, it shows the result of a simple linear model of the `target ~ predictor` formula. The `summary()` function expresses the details of the model. `plot()` visualizes the relationship between the target and predictor variables with a *scatter plot*. Example of `summary()` for numerical target variable is shown below.



```
# If the variable of interest is a numerical variable
num_num <- relate(num, Price)
num_num

Call:
lm(formula = formula_str, data = data)

Coefficients:
(Intercept)        Price
   13.64192     -0.05307
summary(num_num)

Call:
lm(formula = formula_str, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-6.5224 -1.8442 -0.1459  1.6503  7.5108

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 13.641915   0.632812  21.558   <2e-16 ***
Price       -0.053073   0.005354  -9.912   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.532 on 398 degrees of freedom
Multiple R-squared:  0.198, Adjusted R-squared:  0.196
F-statistic: 98.25 on 1 and 398 DF,  p-value: < 2.2e-16
```

- Cases where predictors are categorical variable but the target is numeric value. The results are expressed in terms of ANOVA. The `summary()` function shows the *regression coefficients* for each level of the predictor. In other words, it shows detailed information about *simple regression analysis* of target ~ predictor relationship as shown in the example below.



```
# If the variable of interest is a categorical variable
num_cat <- relate(num, ShelveLoc)
num_cat
Analysis of Variance Table

Response: Sales
           Df Sum Sq Mean Sq F value    Pr(>F)
ShelveLoc   2 1009.5  504.77   92.23 < 2.2e-16 ***
Residuals 397 2172.7    5.47
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
summary(num_cat)

Call:
lm(formula = formula(formula_str), data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-7.3066 -1.6282 -0.0416  1.5666  6.1471

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)       5.5229     0.2388  23.131  < 2e-16 ***
ShelveLocGood     4.6911     0.3484  13.464  < 2e-16 ***
ShelveLocMedium   1.7837     0.2864   6.229  1.2e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.339 on 397 degrees of freedom
Multiple R-squared:  0.3172,    Adjusted R-squared:  0.3138
F-statistic: 92.23 on 2 and 397 DF,  p-value: < 2.2e-16
```

The relationship between target and predictor is represented by a box plot.

Activity 2 – Automated Report

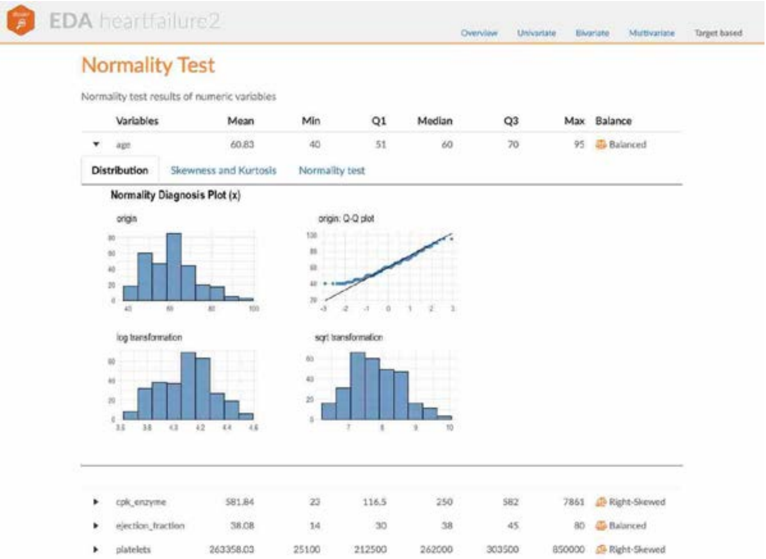
`dlookr` provides two automated EDA reports:

- Web page-based dynamic reports can perform in-depth analysis through visualization and statistical tables.
- Static reports generated as pdf files or html files can be archived as output of data analysis.

1. Create a dynamic report using `eda_web_report()`for Churn dataset. Example of script is shown below:

```
heartfailure %>%
  eda_web_report(target = "death_event", subtitle = "heartfailure",
                 output_dir = "./", output_file = "EDA.html", theme = "blue")
```

The dynamic contents of the report is shown in the following figure:



Normality Test

Normality test results of numeric variables

	Variables	Mean	Min	Q1	Median	Q3	Max	Balance
▾	age	60.83	40	51	60	70	95	Balanced
▸	cpk_enzyme	581.84	23	116.5	250	582	7861	Right-Skewed
▸	ejection_fraction	38.08	14	30	38	45	80	Balanced
▸	platelets	263358.03	25100	212500	262000	303500	850000	Right-Skewed

2. Create a EDA report using `eda_paged_report()`for static report for object inherited from `data.frame(tbl_df, tbl, etc)` or `data.frame.` Sample of script to create a static report is shown below.

```
heartfailure %>%
  eda_paged_report(target = "death_event", subtitle = "heartfailure",
                   output_dir = "./", output_file = "EDA.pdf", theme = "blue")
```

Reference: Choonghyun Ryu, Exploratory Data Analysis, 2023.

Week 4 Lab Submission

1. For activity 1, publish your work to GitHub and share it with GA.
2. Submit EDA report in PDF format through ULearn.

Deadline: 2 Oct 2023