Assignment 2 Report

Student: Lina Mi @01377283

I. R commands used:

1. Load "usedcars_noisy.csv" dataset and store it into a local R variable "usedcars_noisy":

```
>usedcars_noisy<-read.csv('M:/milina/Registered_Courses_2018_Spring/Data Mining/week2/usedcars_noisy.csv')
```

2. Display the content of variable "usedcars_noisy"

```
> usedcars_noisy
```

3. Find out the mean of price and median of mileage for all of the used cars:

```
>mean_price<-mean(usedcars_noisy$price)
> median_mileage<-median(usedcars_noisy$mileage)
```

4. Get the index vector of elements in 'price' column where the values of elements equals zero

```
>index_errprice<-which(usedcars_noisy$price %in% 0)
```

5. Get the index vector of the elements with noise value of '-1' in 'mileage' column of data frame 'usedcars_noisy'.

```
>index_errmileage<-which(usedcars_noisy$mileage %in% -1)
```

6. Replace the "0" price values with the mean price of all cars in the dataset

```
>price_clean<-replace(usedcars_noisy$price, index_errprice, mean_price)
```

7. Replace the '-1' mileage values with the median mileage of all cars in the dataset

```
>mileage_clean<-replace(usedcars_noisy$mileage, index_errmileage, median_mileage)
```

8. Create new variable of data frame to store the cleaned data

```
>usedcars_clean<-usedcars_noisy
```

9. Store the cleaned price column and mileage column to the created variable

```
>usedcars_clean$price<-price_clean
> usedcars_clean$mileage<-mileage_clean
```

10. Save the cleaned dataset to 'usedcars_clean.csv' file

```
>write.csv(usedcars_clean, file='M:/milina/Registered_Courses_2018_Spring/Data Mining/week2/usedcars_clean.csv')
```

II. Mean of price and median of mileage

```
>mean_price<-mean(usedcars_noisy$price)
> mean_price
[1] 12717.55

> median_mileage<-median(usedcars_noisy$mileage)
> median_mileage
[1] 36120
```

III. Summary of the preprocessing on noisy 'usedcars' dataset.

The cleaning on this noisy dataset is relatively easy, as the noise value in price column and mileage column are easy to detect. The challenging part is to find the position of noisy data, namely the index of noise data in the corresponding vector, for example, the indexes of elements with value '0' in 'price' column of 'usedcars_noisy' data frame. In R, command 'which' is used to find the vector of indexes of the elements which meet some condition. Once the index vector of noise data is obtained, it is easy to use command 'replace' to correct these noise data.

In the case of massive dataset, the cleaning process would be much more challenging, how to detect the noise data, the noise data for each field of dataset may have several different values, not like in this case of used cars dataset, where the noise data are apparent and has only one value ('0' for price field and '-1' for mileage field respectively). How to clean these noise data in the massive dataset is also not as simple as it is in this case of used car dataset.