TASK REPORT
DATA SCIENCE AND ITS IMPLEMENTATION IN STATISTICS FOR DATA SCIENCE

By:
Mohamad Irwan Afandi
Data Science Student

Data Fellowship
2020

Chapter 1
Introduction

Talking about the specific task of the data science certainly cannot be separated from data. Data science uses the data to find insights that can be used for support the business case and predict the future. To get the insight from the data, it is not enough just look at the values of the data. The first we need to find the data, create hypothesis from the data, analysis the data using functions or figures then we presenting that data. This is why data science need statistics knowledge that concern on the collection, organization, analytic interpreter and presenting the data. With this knowledge, the data science can determine what they have to do with their data and produce perfect analytic.

In this report the author wants to share basic concepts of statistics that often used by the data science. The concept will be delivered by solving some problems in practice case. The following are the problems that are trying to solve.

You've landed a great job with the Mallianzs insurance company as a data scientist. This insurance company wants to know its customer profile in a detailed way. Your team of engineers have to analyze the data that they have based on the predefined questions that your CEO gave.

Questions:

1. Perform basic exploratory data analysis which should include the following and print out your insights at every step:
   A. The shape of the data
   B. The data type of each attribute
   C. Checking the presence of missing values
   D. 5 points summary of numerical attributes
   E. Distribution of 'bmi', 'age' and 'charges' columns
   F. The measure of skewness of 'bmi', 'age', and 'charges' columns
   G. Checking the presence of outliers in 'bmi', 'age' and 'charges' columns

2. Answer the following questions with statistical evidence
   A. Do charges of people who smoke differ significantly from the people who don't? (Hypothesis Testing)

Chapter 2
Progress Report

In this chapter you will have to fill in the table below according to the progress of the project that you have made along the way. We need to know how long it takes for you and how big the effort that you have done in order to complete this task. We appreciate detailed information.

Day/Date	**Task**	**Level** (easy/medium/hard)	**Comments**
18/08/2020	Doing statistic's quizzes in iykra's website	easy	-
19/08/2020	Try to solve the problem using python and pandas library in google colabs	medium	Need to search and much explore. This is very new for me.
20/08/2020	Creating statistic practice case report	Easy	One of the best part, writing.

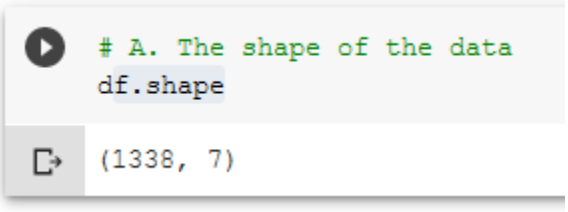
Chapter 3
Task Report

Perform basic exploratory data analysis which should include the following and print out your insights at every step:

1. The shape of the data

   Answer:

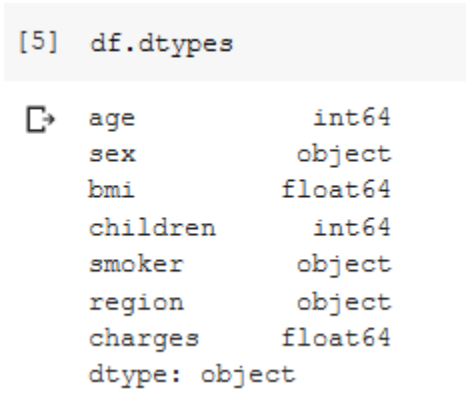
   Shape is the dimension of the data, we can check it using **df.shape** to show the dimension our data (row x columns). So the shape of our data is **1338 rows and 7 columns**



2. The data type of each attribute

   Answer:

   To get the data type for every attribute we can use **df.types** or **df.infor()**. The different is types just return the data type for every attributes but info will return count non null data, and its data type.



3. Checking the presence of missing values

Answer:

To know the missing value of data we can compare the non-null values with all the range index in **df.info()**. If the non-null value is not equal with the range index, so we can say that the data has missing value. The second method is using **df.isnull().sum()** that will return the missing value directly. Based on insurance data, **we don't have missing value**

```
#C. Checking the presence of missing values
df.isnull().sum()
```

```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

4. 5 points summary of numerical attributes

   Answer:

   The summery of numerical in every attributes can be find one by one, but the simplest way is use **df.describe().** This function will return mean, standard deviation, min, max, quartile 1 (25%), quartile2 (50%) or mode and quartile3 (75%). But this only works on attribute with numerical data type. If you also want to get the summary of categorical data use **df.describe(include='all')**

```
# D. 5 points summary of numerical attributes
df.describe()
```

	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	13270.422265
std	14.049960	6.098187	1.205493	12110.011237
min	18.000000	15.960000	0.000000	1121.873900
25%	27.000000	26.296250	0.000000	4740.287150
50%	39.000000	30.400000	1.000000	9382.033000
75%	51.000000	34.693750	2.000000	16639.912515
max	64.000000	53.130000	5.000000	63770.428010

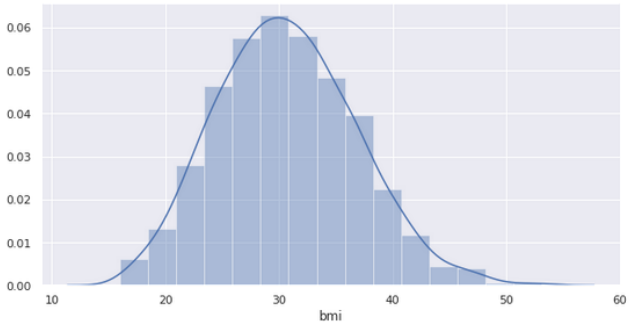
5. Distribution of 'bmi', 'age' and 'charges' columns.

   Answer:

   To get the distribution of the attribute we need to plot the data, we can use matplotlib but in this case I used seabor. The syntax is **sns.distplot(attribute, bins = value)**, but don't forget to import the seaborn library in your porject.

The distribution of bmi, looks like normal distribution, but we cannot say this is normal distribution before get the skew score.

```
# E. Distribution of 'bmi', 'age' and 'charges' columns
# F. The measure of skewness of 'bmi', 'age', and 'charges' columns
fig, ax = plt.subplots(figsize=(10,5))
sns.distplot(df['bmi'], bins=15)
plt.show()
```

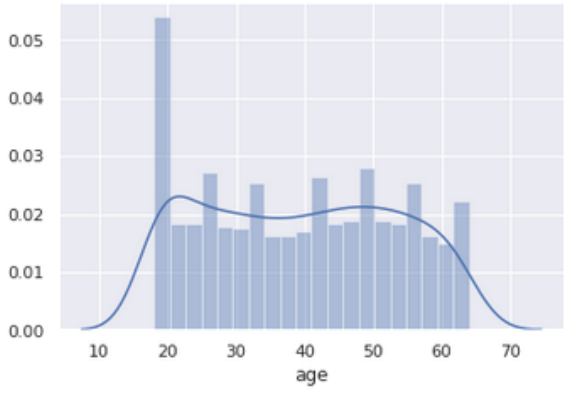



```
#skewness value
df['bmi'].skew()
```

```
0.2840471105987448
```

The distribution of age, bimodal distribution with 2 peak

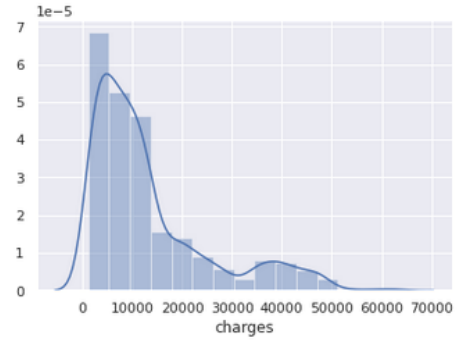
```
sns.distplot(df['age'], bins=20)
plt.show()
```




```
#skewness value
df['age'].skew()
```

The distribution of charges, the concentration of data frequency in the left side.

```
sns.distplot(df['charges'], bins=15)
# df['charges'].plot(kind = 'density')
plt.show()
```

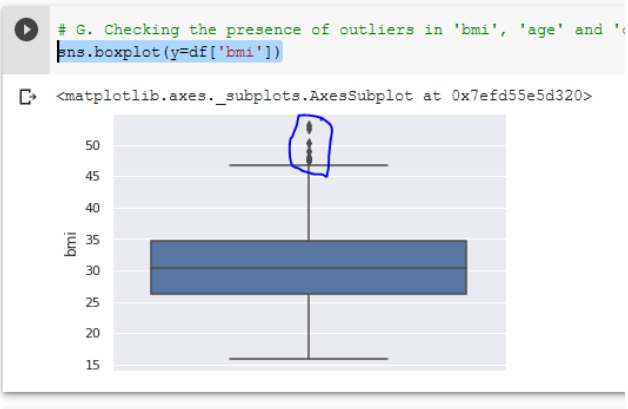



```
#skewness value
df['charges'].skew()
```

```
1.5158796580240388
```

6. The measure of skewness of 'bmi', 'age', and 'charges' columns
   Answer:
   To get the value of measurement from the skewness we can use **df[attribute].skew()** where it will return a float value. (see the picture in question 5 for the skew score)
   Skewness of bmi: 0.284 (positive skew)
   Skewness of age: 0.05567 (positive skew)
   Skewness of charges: 1.5159 (positive skew)

7. Checking the presence of outliers in 'bmi', 'age' and 'charges' columns
   Answer:
   To know the the data has outliers or not, you can use box plot with the syntax **sns.boxplot(y=attribute).** But you can also using Inter Quartile (IQR) method to know the lower bound and upper bound of the data. This is the result
   bmi : 9 outliers

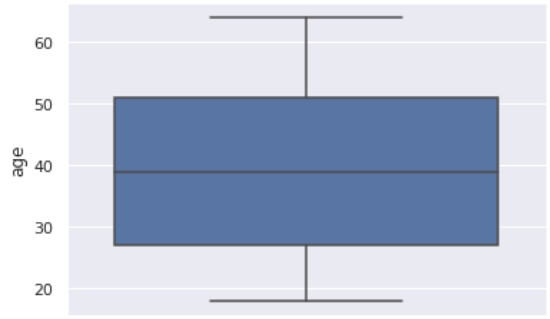


If there is a dot above or under the line boundaries, it indicates that there is outlier in that data

age : 0 outliers

```
sns.boxplot(y=df['age'])
```

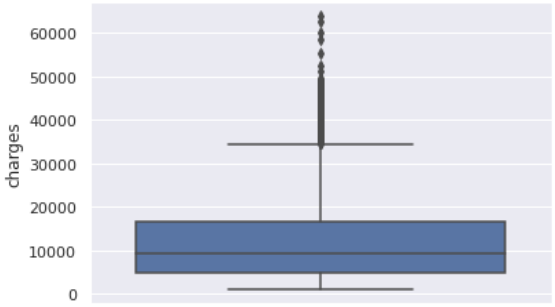
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efd55e76a58>
```



charges : 139 outliers

```
sns.boxplot(y=df['charges'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efd55cc3080>
```



Do charges of people who smoke differ significantly from the people who don't? (Hypothesis Testing)

Answer:
Determine H0 and H1
H0 : mean insurance charges of smokers and non-smokers are equals
H1 : mean insurance charges of smokers and non-smokers are not equals

And then, separate data between smoker and non-smoker data. From the data we get 274 smoker and 1064 non-smoker data. Apply the Confident Interval of 95% means that the conclusion of this test will be valid. Then calculate the t-statistic using this formula.

$$t = \frac{M_x - M_y}{\sqrt{\frac{S_x^2}{n_x} + \frac{S_y^2}{n_y}}}$$

M = mean
n = number of scores per group

$$S^2 = \frac{\sum (x - M)^2}{n - 1}$$

x = individual scores
M = mean
n= number of scores in group

The next step is calculating the critical t-value from the t-distribution, the last is compare the t-statistic with t-value. In this step I used scipay library (state) and this is the result of the calculation.

```
t2, p2 = stats.ttest_ind(rokok['charges'],norokok['charges'])
print("t = " + str(t2))
print("p = " + str(p2))
```

```
t = 46.664921172723716
p = 8.271435842177219e-283
```

From this result we can see that the value of t-statistic is greater than the critical t-value. Based on hypothesis rule if t-statistic > critical t value, it means there is a statistically significant difference between the two populations. So we need to **reject HO** and **accept H1** as the conclusion of our hypothesis. So, **the insurance charges of smokers and non-smokers are not equals (differ significantly)**

Visit this link for the code :
https://colab.research.google.com/drive/1MffTptNmW3Y2xf1fkAw1EPqiXBsktE77?usp=sharing

Conclusion

Statistic is very helpful to explore the data and see the pattern of the data. From that pattern we can know what we have to do for the next. Like if there is a missing value and the distribution is not normal so we use median to handle that missing value, etc. If you really understand about statistics then you will get the better results of data from cleansing process.