TASK REPORT
DATA SCIENCE AND ITS IMPLEMENTATION IN BASIC DATA ANALYSIS

By:
Mohamad Irwan Afandi
Data Science Student

Data Fellowship
2020

Chapter 1
Introduction

Data is an information where we can get a lot of insights if we can able to cultivate and try to analysis it. Before we analyst the data we need to understand the business process behind the data and then try to make questions that maybe can be obtained from that data. We can answer the question by selecting, grouping, calculation, shorting or even cooler by using the data visualization. This why as data scientist we need to understand the method how to analyst the data to get quick insight from it.

In this report the author wants to share about basic data analysis that 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 Ritz-Jager Hotel operator as a data scientist. This hotel operator wants to improve their business efficiency by utilizing their historical data and they want to find out what happened in their previous bookings, knowing their customer better, and optimizing the promo timing.

Your team of engineer have to **analyze the data** that they have based on the pre-defined questions that your CEO gave.

Questions:

1. Where do the guests come from?
2. How much do guests pay for a room per night?
3. How does the price per night vary over the year?
4. Which are the busiest months?
5. How long do people stay at the hotels?
6. Bookings by market segment
7. How many bookings were cancelled?
8. Which month has the highest number of cancellations?

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 basic data analysis quizzes in iykra's website	easy	-
19/08/2020	Try to solve the problem using python and library (pandas, matplotlib, numpy, and seaborn) in google colabs	medium	Need searching skill ☺
20/08/2020	Creating practice case report for basic data analysis.	Easy	Try to imagine, start translating and learn grammer from this step

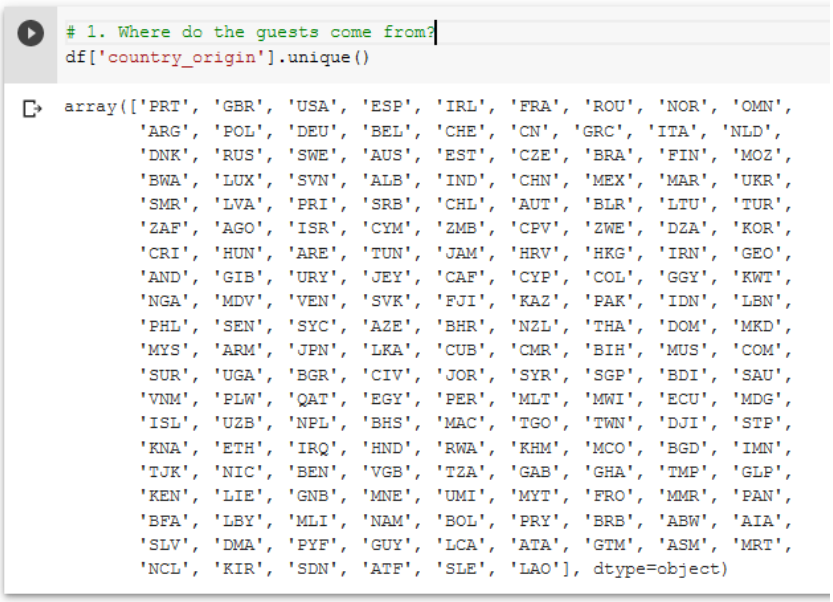
Chapter 3
Task Report

Before we try to analyst the data, we need to check the data whether contain missing value or not. If it's contain outlier, we need to handle it by filling the missing value or drop the data. This rules usually use to handle missing value:

- If the data is categorical, we can use mode to fill the missing values.
- If the data is nominal you have to check the skewness. If the skewness is symmetry you can apply mean to fill the missing value or median if the skewness is positive or negative.
- If the 80% data of an attribute is missing, you can drop the attribute because we don't need to analyst that data anymore.

Question 1
Where do the guests come from?

Answer:
We can get the information by selecting an unique values from country_origin using this syntax **df['country_origin'].unique()**. From that syntax, we know that there are **177 counties** where the guests come from.



```
# 1. Where do the guests come from?
df['country_origin'].unique()
```

```
array(['PRT', 'GBR', 'USA', 'ESP', 'IRL', 'FRA', 'ROU', 'NOR', 'OMN',
       'ARG', 'POL', 'DEU', 'BEL', 'CHE', 'CN', 'GRC', 'ITA', 'NLD',
       'DNK', 'RUS', 'SWE', 'AUS', 'EST', 'CZE', 'BRA', 'FIN', 'MOZ',
       'BWA', 'LUX', 'SVN', 'ALB', 'IND', 'CHN', 'MEX', 'MAR', 'UKR',
       'SMR', 'LVA', 'PRI', 'SRB', 'CHL', 'AUT', 'BLR', 'LTU', 'TUR',
       'ZAF', 'AGO', 'ISR', 'CYM', 'ZMB', 'CPV', 'ZWE', 'DZA', 'KOR',
       'CRI', 'HUN', 'ARE', 'TUN', 'JAM', 'HRV', 'HKG', 'IRN', 'GEO',
       'AND', 'GIB', 'URY', 'JEY', 'CAF', 'CYP', 'COL', 'GGY', 'KWT',
       'NGA', 'MDV', 'VEN', 'SVK', 'FJI', 'KAZ', 'PAK', 'IDN', 'LBN',
       'PHL', 'SEN', 'SYC', 'AZE', 'BHR', 'NZL', 'THA', 'DOM', 'MKD',
       'MYS', 'ARM', 'JPN', 'LKA', 'CUB', 'CMR', 'BIH', 'MUS', 'COM',
       'SUR', 'UGA', 'BGR', 'CIV', 'JOR', 'SYR', 'SGP', 'BDI', 'SAU',
       'VNM', 'PLW', 'QAT', 'EGY', 'PER', 'MLT', 'MWI', 'ECU', 'MDG',
       'ISL', 'UZB', 'NPL', 'BHS', 'MAC', 'TGO', 'TWN', 'DJI', 'STP',
       'KNA', 'ETH', 'IRQ', 'HND', 'RWA', 'KHM', 'MCO', 'BGD', 'IMN',
       'TJK', 'NIC', 'BEN', 'VGB', 'TZA', 'GAB', 'GHA', 'TMP', 'GLP',
       'KEN', 'LIE', 'GNB', 'MNE', 'UMI', 'MYT', 'FRO', 'MMR', 'PAN',
       'BFA', 'LBY', 'MLI', 'NAM', 'BOL', 'PRY', 'BRB', 'ABW', 'AIA',
       'SLV', 'DMA', 'PYF', 'GUY', 'LCA', 'ATA', 'GTM', 'ASM', 'MRT',
       'NCL', 'KIR', 'SDN', 'ATF', 'SLE', 'LAO'], dtype=object)
```

Question 2

How much do guests pay for a room per night?

Answer:

We need to know the meaning of this question, based on that question we need to get the average price every room in every tipe_hotel per night. You can use this syntax to get the information

```
booked = df[(df['is_canceled'] != 1) & df['adr']>0]
pd.pivot_table(booked, index=['hotel_type','assigned_room_type'], values='adr', aggfunc='mean')
```

```
# 2. How much do guests pay for a room per night?
booked = df[(df['is_canceled'] != 1) & df['adr']>0]
pd.pivot_table(booked, index=['hotel_type','assigned_room_type'], values='adr', aggfunc='mean')
```

		adr
hotel_type	assigned_room_type	
City Hotel	A	99.078326
	B	97.518135
	C	110.417879
	D	117.913473
	E	142.225000
	F	178.171620
	G	196.841696
	K	102.107391
Resort Hotel	A	81.309890
	B	105.815517
	C	108.913320
	D	82.425891
	E	102.462598
	F	120.389556
	G	153.179443
	H	161.823956
	I	90.872089

Question 3

How does the price per night vary over the year?

Answer:

Based on this question, we need to find the variance of the price in every night and every year. You can use this syntax to get the variance price in every year

```
booked = df[(df['is_canceled'] != 1) & df['adr']>0]
booked[['arrival_date_year','adr']].groupby(['arrival_date_year']).var()
```

```
# 3. How does the price per night vary over the year?
booked = df[(df['is_canceled'] != 1) & df['adr']>0]
booked[['arrival_date_year','adr']].groupby(['arrival_date_year']).var()
```

arrival_date_year	adr
2015	1771.957729
2016	1982.751061
2017	2661.874713

Question 4

Which are the busiest months?

Answer:

Before we answer this question, we need to deal with the mean of *busiest month*. It means the month where the number of check-out guests in the hotels are maximum. First you need to find the check-out gusts, then count the number of guests in every month. You can use this syntax to answer this question.

```
booked = df[df['is_canceled'] == 0]
pd.pivot_table(booked, index=['arrival_date_year','arrival_date_month'], v
alues='assigned_room_type', aggfunc='count')
```

The busiest month based on the data is **October 2016** with 3689 guest

```
booked = df[df['is_canceled'] == 0]
pd.pivot_table(booked, index=['arrival_date_year','arrival_date_month'], values='assigned_room_type', aggfunc='count'
```

		assigned_room_type
arrival_date_year	arrival_date_month	
2015	August	2291
	December	1947
	July	1517
	November	1854
	October	3225
	September	3020
2016	April	3367
	August	3238
	December	2462
	February	2554
	January	1691
	July	3073
	June	3196
	March	3347
	May	3563
	November	2818
	October	3689
	September	3372
2017	April	3198

Question 5

How long do people stay at the hotel?

Answer:

We just count the average of stay from guests with status check-out. You can use this syntax

```
df_stay = df.copy()
df_stay ['stay_duration'] = df['stays_in_weekend_nights'] + df['stays_in_week_nights']
df_stay[df_stay['is_canceled']==0]['stay_duration'].mean()
```

```
# 5. How long do people stay at the hotels?
df_stay = df.copy()
df_stay ['stay_duration'] = df['stays_in_weekend_nights'] + df['stays_in_week_nights']
```

```
df_stay[df_stay['is_canceled']==0]['stay_duration'].mean()
```

```
3.3930234414495914
```

So the average of guests stay at the hotel **3 until 4 days**

Question 6
Booking by market segment

Answer:
We just need to count the number of reservation with status is_canceled equals 0 (not canceled) by market segment. Cancel means not booking, this is the syntax

```
df[df['is_canceled'] == 0]['market_segment'].value_counts()
```

```
# 6. Bookings by market segment
df[df['is_canceled'] == 0]['market_segment'].value_counts()
```

```
Online TA        35738
Offline TA/TO    15908
Direct           10672
Groups            7714
Corporate         4303
Complementary      646
Aviation           185
Name: market_segment, dtype: int64
```

Question 7
How many booking were cancelled?

Answer:
There are **43017** reservation that canceled. You can use this syntax to get the result

```
df['is_canceled'].value_counts()
df['reservation_status'].value_counts()[1]
```

```
# 7. How many bookings were cancelled?
df['is_canceled'].value_counts()
```

```
0    75166
1    44224
Name: is_canceled, dtype: int64
```

```
df['reservation_status'].value_counts()[1]
```

```
43017
```

Question 8
Which month has the highest number of cancellations?

Answer:
In this case we need the specific answer month and year. First you need to filter which is the canceled reservation. Then count the number of canceled group by month and year. The highest cancellations happened on **May 2017** with 2762 cancellations. You can use this syntax to get the result

```
booked = df[df['is_canceled'] == 1]
pd.pivot_table(booked, index=['arrival_date_year','arrival_date_month'], v
alues='is_canceled', aggfunc='count')
```

If you want get the result by shorting the canceled value, use this syntax

```
booked = df[df['is_canceled'] == 1]
x = pd.pivot_table(booked, index=['arrival_date_year','arrival_date_month'
], values='is_canceled', aggfunc='count')
x.reindex(x['is_canceled'].sort_values(ascending=False).index)
```

```
# 8. Which month has the highest number of cancellations?
booked = df[df['is_canceled'] == 1]
pd.pivot_table(booked, index=['arrival_date_year','arrival_date_month'], values='is_canceled', aggfunc='count')
```

		is_canceled
arrival_date_year	arrival_date_month	
2015	August	1598
	December	973
	July	1259
	November	486
	October	1732
	September	2094
2016	April	2061
	August	1825
	December	1398
	February	1337
	January	557
	July	1499
	June	2096
	March	1477
	May	1915
	November	1636
	October	2514
	September	2022

Visit this link to see the result with its code:
https://colab.research.google.com/drive/1WmHvgwhHNsYtxrvJL9d46cMIIsrc0BS4?usp=sharing

Conclusion

The first, please understand the business process behind the data before you try to analyst the data. Without understand it first, maybe you will be confused about what the data said and make wrong analysis. It looks so simple, but it so powerful as data scientist.