ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KINH TẾ - LUẬT

FINAL REPORT T

DATA MINING

Name: Đặng Đức Duy

Student ID: K205030798

Ho Chi Minh January - 2023

TOPIC 7: Which promotion was the most effective?

To see all file code and dataset in:
Github: https://github.com/DucDuyDang/Which-promotion-was-the-most-effective-

Table of contents

CHAPTER 1. OVERVIEW OF THESIS

1.1. The reason for choosing the topic:

Marketing today is driven by data-driven research, and customer data can be collected at any point in the buying process. On the other hand, data-driven marketing allows marketers to connect with customers at the right time.

A data-centric marketing team can reach the right audience, build tactics that engage and delight them, and assist sales in converting them into paying customers armed with data. new, accurate and compliant data. We don't have to guess what people want; We just need to know where to look. However, the benefits of using data go beyond simply promoting communication.

Brands can also use data to measure and enhance their strategy in real time. In today's world of B2B marketing, data is the secret weapon.

You can create marketing strategies that meet the individual needs of your target users if you understand their behaviors, goals, problem areas, and obstacles. Data such as users' surfing habits, social media engagement, online purchasing behavior, and other metrics can help you focus your marketing efforts on what succeeds. Therefore, try to gather as much important information as possible about your target market. Any successful marketing strategy will be built around this data.

Data can not only reveal the preferences of the target audience. It can also suggest which advertising channels a brand should use to engage their audience now and in the future.

We can optimize our creative marketing initiatives for maximum impact by gathering data and determining what is most beneficial for your customers. That's why I would like to choose the topic: "Which promotion was the most effective?"

1.2. Topic goal:

Unlike traditional marketing, it always focuses on the needs and wants of customers. And then, use that insight to deliver what customers want to buy. Traditional marketing will learn deeply about the target audience, identify and anticipate customer needs, and finally, devise strategies to provide goods that meet customer wants.

As for using data to determine which marketing campaigns are more suitable for your brand, the benefits of using data are not only to improve communications, but also to support the personalization of the customer experience. and have clearly defined marketing segments. From there we can measure performance and improve new strategies.

By determining which promotion will be most effective for the company's new product, we can come up with appropriate strategies for distribution, using a test-run strategy for all upcoming strategies.

1.3. Tools used:

In this project, I used only one programming language, Python, to create, describe, analyze, and interpret key metrics in the campaign. The Python language puts data into visual context so that patterns, trends, and correlations can be easily represented. Python offers many great graphics libraries that come with a lot of different features. Whether you want to create interactive, live or highly customizable, python has a great library.

Besides, I store data on a raw data set in excel format. Excel makes it easier to calculate indexes and store data.

1.4. Research implications: .

The analysis Marketing Promotion will reply business question like:

- "How did this promotion perform?"
- "Which promotion is referring to the most use?"
- "Why is a particular promotion underperforming?"
- …..

All of the questions will be answered using data from the analysis. This project is built on Python and Pandas fundamentals, such as merging/slicing datasets, groupby(), correcting data types and visualizing results using matplotlib.

1.5. Structure of report:

Chapter 1: Overview of thesis
Chapter 2: Theoretical Basis
Chapter 3: Analysis of user requirements and Data Description
Chapter 4: Data Analysis and Results
Chapter 5: Conclusion

CHAPTER 2. THEORETICAL BASIS

2.1. Overview of Data Analytics:

Data Analytics is the process of examining data sets to find trends and draw conclusions about the information they contain. The purpose of Data Analysis is to extract useful information from data and take the decision based upon the data 8 analysis

.Data Analytics has four basic types: Descriptive analytics, Diagnostic analytics, Predictive Analytics, and Prescriptive analytics

2.1.1. Advantages of Data Analytics in Business:

Personalize the customer experience: Businesses collect customer data from many different channels, including physical retail, e-commerce, and social media. By using data analytics to create comprehensive customer profiles from this data, businesses can gain insights into customer behavior to provide a more personalized experience.

Better decision-making: One of the main benefits of DA is that it improves the decision-making process significantly. Rather than relying on intuition alone, companies are increasingly looking toward data before making decisions. Predictive analytics can suggest what could happen in response to changes to the business, and prescriptive analytics can indicate how the business should react to these changes. Reduce costs: Another great benefit is to reduce costs. With the help of advanced technologies such as predictive analytics, businesses can spot improvement opportunities, trends, and patterns in their data and plan their strategies accordingly. In time, this will help you save money and resources on implementing the wrong strategies. And not just that, by predicting different scenarios such as sales and demand you can also anticipate production and supply

2.1.2. The process of implementing data analysis projects in the enterprise:

The first step is to determine the data requirements or how the data is grouped. Data may be separated by age, demographic, income, or gender. Data values may be numerical or be divided by category.

The second step in data analytics is the process of collecting it. This can be done through a variety of sources such as computers, online sources, cameras, environmental sources, or through personnel. Once the data is collected, it must be organized so it can be analyzed. This may take place on a spreadsheet or other form of software that can take statistical data.

The data is then cleaned up before analysis. This means it is scrubbed and checked to ensure there is no duplication or error, and that it is not incomplete. This step helps correct any errors before it goes on to a data analyst to be analyzed. Data Analysis: Here is where you use data analysis software and other tools to help you interpret and understand the data and arrive at conclusions.

Data Interpretation: Now that you have your results, you need to interpret them and come up with the best courses of action, based on your findings.

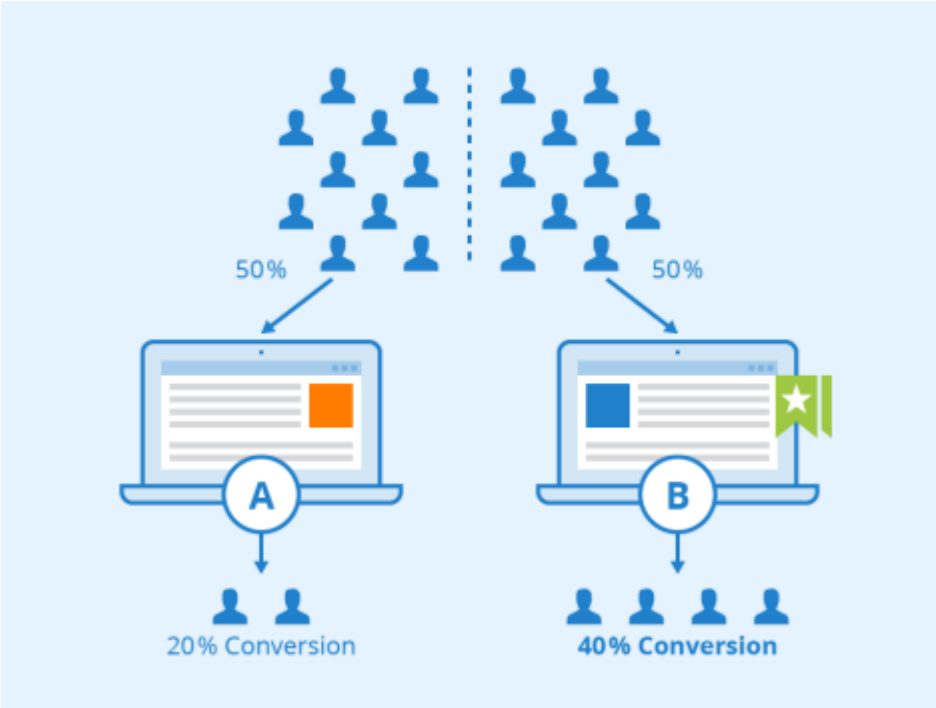
Data Visualization: Data visualization is a fancy way of saying, “graphically show your information in a way that people can read and understand it.” You can use

charts, graphs, maps, bullet points, or a host of other methods. Visualization helps you derive valuable insights by helping you compare datasets and observe relationships.

2.2. Theory and methods in data analysis:

2.2.1. Theory of using A/B Testing:

A/B Testing (also known as split testing or bucket testing) is a technique of dividing the object to be tested into two versions A and B to compare between the two versions, thereby giving the version that the user wants. more interested in how users interact with each of those instances.



★ ***Why is it important to do A/B testing?***

A/B Testing allows for making careful changes to the user experience while collecting data for results. That is, they can formulate hypotheses and better understand why certain factors in their experience influence user behavior through A/B testing.

★ ***A/B testing process***

In order to perform a proper A/B testing (or any other testing process) process, it is necessary to follow the steps, including the following steps:

❖ Data collection:

Your analytics will often provide a sharp, clear view of where you can start optimizing. It helps you get started with high-traffic areas of your website or app. As this will allow you to collect data faster.

❖ Determine the target:

Your conversion goal is the metric you're using to determine if the variation is more successful than the original.

❖ Generate hypothesis:

Once you've identified your goals, you can start generating A/B testing ideas and hypotheses about why you think they'll be better than the current version.

❖ Create Variations:

Use your A/B Testing software (such as Optimizely). This helps make desired changes to a campaign element

❖ Analyze the results:

When your test is complete, it's time to analyze the results

Your A/B Testing software will pull out the data from the test and show you the difference between how the two versions of your website are performing. And is there a statistically significant difference?

CHAPTER 3. ANALYSIS OF USER REQUIREMENTS AND DATA DESCRIPTION

3.1. Identify and analyze user requirements:

For this report, the dataset WA_Fn-UseC_-Marketing-Campaign-Eff-UseC_-FastF from the statistic platform *"Kaggle"* was used.

Scenario: A fast food chain plans to add a new item to its menu. However, they are still undecided between three possible marketing campaigns for promoting the new product. In order to determine which promotion has the greatest effect on sales, the new item is introduced at locations in several randomly selected markets. A different promotion is used at each location, and the weekly sales of the new item are recorded for the first four weeks.

The end user of this analysis will be a fast food company. The main goal will be to consider what promotion to use in the store and which promotion to use for new items in the menu (Not counting costs - only sales).

In a nutshell, this will be achieved by using A/B testing and creating marketing campaigns that specifically target their profiles. Using historical data on sales, we hope to create the most relevant promotion that is reflective of all customers.

3.2. Identify and analyze user requirements:

In this project, the chosen approach was A/B Testing. Following the Business Understanding, the next step is to proceed to Data Understanding. In this phase, all the features and their correspondent values within the data frame created are analyzed.

3.2.1. Describe the source data:

Step 1: we import libraries like: numpy, pandas, matplotlib, seaborn, scipy,... Then we read the file. Here I have exported the first 10 rows of the dataset by LocationID.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from scipy import stats
```

```
df=pd.read_csv(r"WA_Fn-UseC_-Marketing-Campaign-Eff-UseC_-FastF.csv")
```

```
df.sort_values(by='LocationID').head(10)
```

	MarketID	MarketSize	LocationID	AgeOfStore	Promotion	week	SalesInThousands
0	1	Medium	1	4	3	1	33.73
1	1	Medium	1	4	3	2	35.67
2	1	Medium	1	4	3	3	29.03
3	1	Medium	1	4	3	4	39.25
4	1	Medium	2	5	2	1	27.81
5	1	Medium	2	5	2	2	34.67
6	1	Medium	2	5	2	3	27.98
7	1	Medium	2	5	2	4	27.72
11	1	Medium	3	12	1	4	34.75
10	1	Medium	3	12	1	3	45.49

Picture 1: *Import các thư viện và read file*

We can look at each variable and try to understand their meaning and relevance to this problem. I know this is time-consuming, but it will give us the flavour of our dataset.

Step 2: Describe the columns and rows in the dataset:

General :

```
#SHAPE OF THE DATASET.....
print("Shape of the DataFrame is :",df.shape)
```

```
Shape of the DataFrame is : (548, 7)
```

```
#CHECK THE COLUMNS NAME....
print("Columns in DataFrame is :\n",df.columns)
```

```
Columns in DataFrame is :
 Index(['MarketID', 'MarketSize', 'LocationID', 'AgeOfStore', 'Promotion',
       'week', 'SalesInThousands'],
      dtype='object')
```

Picture 2: *Mô tả dữ liệu*

The description of the dataset: Our data set consists of 548 entries and 7 Series including:

1. *MarketId:* an inhouse tag used to describe market types, we won't be using it
2. *AgeOfStores:* Age of store in years (1–28). The mean age of a store is 8.5 years.
3. *LocationID:* Unique identifier for store location. Each location is identified by a number. The total number of stores is 137.
4. *Promotion:* One of three promotions that were tested (1, 2, 3). We don’t really know the specifics of each promotion.

5. *Sales in Thousands:* Sales amount for a specific LocationID, Promotion and week. The mean amount of sales are 53.5 thousand dollars.
6. *Market size:* there are three types of market size: small, medium and large.
7. *Week:* One of four weeks when the promotions were run (1–4).

CHAPTER 4. DATA ANALYSIS AND RESULTS

4.1 Descriptive analysis and EDA dataset:

4.1.1. Overview

After do check summary information of dataset, we see:

```
: #PRINT THE COMPLETE INFORMATION OF THE DATASET.....
  # Print a Summary of a Dataframe is :"
  df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 548 entries, 0 to 547
Data columns (total 7 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   MarketID          548 non-null    int64
 1   MarketSize        548 non-null    object
 2   LocationID        548 non-null    int64
 3   AgeOfStore        548 non-null    int64
 4   Promotion         548 non-null    int64
 5   week              548 non-null    int64
 6   SalesInThousands  548 non-null    float64
dtypes: float64(1), int64(5), object(1)
memory usage: 30.1+ KB
```

Picture 3: *Overview dataset*

Describe data through the statement: describe(). Column SalesInThousands maximum value is 99.65 and minimum is 17.34 with Sales standard deviation of 16,755 and average sales for 4 week is 50.2

Correct to the original description of the data, there will be 4 promotions, and the age of the Store will be from 1 to 28,…

```
: df.describe()
```

:

	MarketID	LocationID	AgeOfStore	Promotion	week	SalesInThousands
count	548.000000	548.000000	548.000000	548.000000	548.000000	548.000000
mean	5.715328	479.656934	8.503650	2.029197	2.500000	53.466204
std	2.877001	287.973679	6.638345	0.810729	1.119055	16.755216
min	1.000000	1.000000	1.000000	1.000000	1.000000	17.340000
25%	3.000000	216.000000	4.000000	1.000000	1.750000	42.545000
50%	6.000000	504.000000	7.000000	2.000000	2.500000	50.200000
75%	8.000000	708.000000	12.000000	3.000000	3.250000	60.477500
max	10.000000	920.000000	28.000000	3.000000	4.000000	99.650000

Picture 4: *Data Description*

Next, we analyze to see if there are null values in the dataset. Results after running the command line: isnull().sum() is the tuple we have no columns with nulls

```
: df.isnull().sum()

: MarketID            0
  MarketSize          0
  LocationID          0
  AgeOfStore          0
  Promotion           0
  week                0
  SalesInThousands    0
  dtype: int64
```

Picture 5: *Check value for null*

```
]: #CHECKING IF ANY NAN IS PRESENT IN COLUMN OR NOT....
   df.isna().any()

]: MarketID            False
   MarketSize          False
   LocationID          False
   AgeOfStore          False
   Promotion           False
   week                False
   SalesInThousands    False
   dtype: bool
```

Check the types of the data. We can see:

- ❖ Types is an object with only columns: Marketsize (Small, Large, Medium)
- ❖ Types is an int consist of columns: MarketID, LocationID, AgeOfStore, Promotion, week
- ❖ Types is a column only float: SalesInThousands

```
: df.dtypes

: MarketID            int64
  MarketSize         object
  LocationID          int64
  AgeOfStore          int64
  Promotion           int64
  week                int64
  SalesInThousands    float64
  dtype: object
```

Picture 6: *Types of dataset*

Use this code below to see clear my script

```
cate = []
for i in df.columns:
    if (df[i].dtypes == "object"):
        cate.append(i)

print(" Object are:",cate)
```

```
 Object are: ['MarketSize']
```

```
Int = []
for i in df.columns:
    if (df[i].dtypes == "int64"):
        Int.append(i)

print(" Integers are:",Int)
```

```
 Integers are: ['MarketID', 'LocationID', 'AgeOfStore', 'Promotion', 'week']
```

```
Float = []
for i in df.columns:
    if (df[i].dtypes == "float64"):
        Float.append(i)

print("Float are:",Float)
```

```
Float are: ['SalesInThousands']
```

Picture 6: *Types of dataset*

Next, we check detecting the duplicates. Columns do not have the same value in all rows, that's why they will all contribute in building the model. So we don't need to remove unwanted features

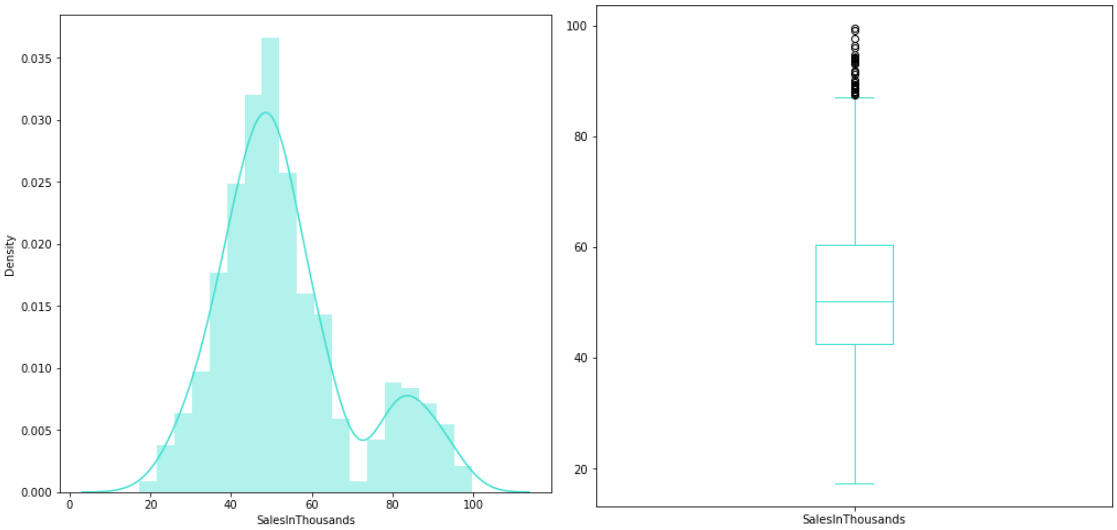
```
#FINDING THE NUMBER OF UNIQUE VALUES PRESENT IN EACH COLUMN...
df.nunique()
```

```
MarketID            10
MarketSize           3
LocationID         137
AgeOfStore          25
Promotion            3
week                 4
SalesInThousands   517
dtype: int64
```

Picture 7: *Unique the dataset*

4.1.2. Analysis on each variables in dataset:

4.1.2.1. SalesInThousand Variable:



Picture 8: *Frequency of SalesInThousand*

Sale has outlier distribution at 65.00 round up. If this outlier is removed, Sales will follow a normal distribution.

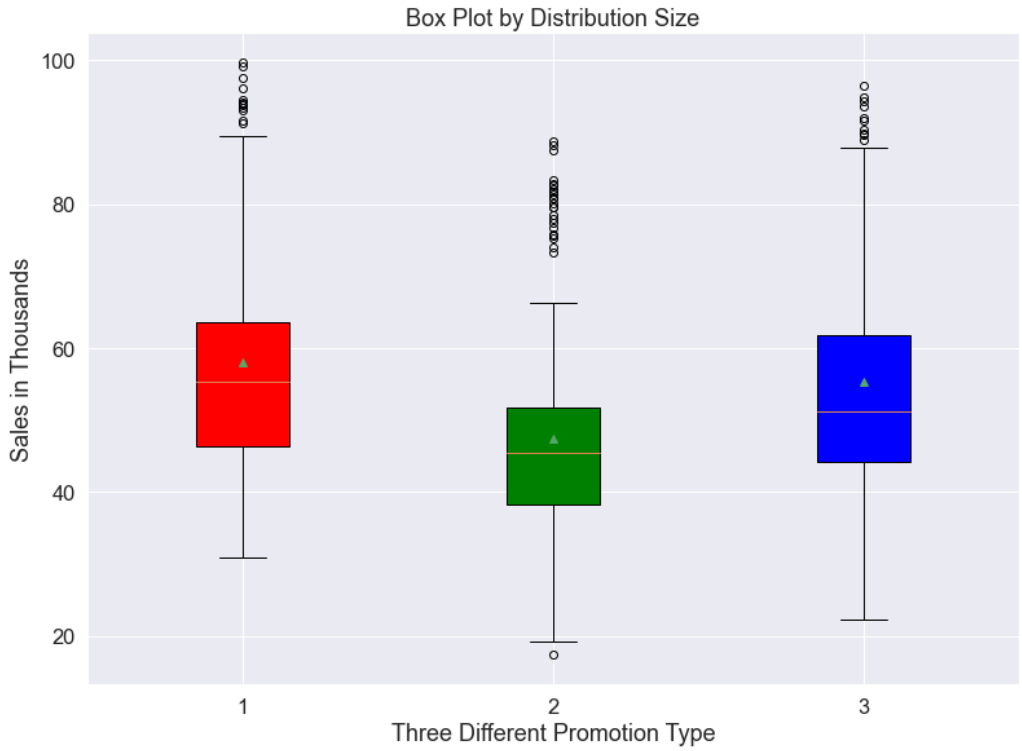
Next, we will find out why there is an outlier in terms of sales like this

```
#Distribution of Sales by Promotion Size
data=[np.array(df[df['Promotion'] ==1]['SalesInThousands']),
      np.array(df[df['Promotion'] ==2]['SalesInThousands'])
      ,np.array(df[df['Promotion'] ==3]['SalesInThousands'])]
```

```
fig = plt.figure(figsize =(10, 7))
ax = fig.add_axes([0, 0, 1,1])
bp=ax.boxplot(data, vert=True, patch_artist=True, showmeans=True)
ax.set_title("Box Plot by Distribution Size")

#Fill with colors
colors=['red', 'green', 'blue']
for patch, color in zip(bp['boxes'], colors):
    patch.set_facecolor(color)


#Adding Horizontal Grid Lines
ax.yaxis.grid(True)
ax.set_xlabel('Three Different Promotion Type')
ax.set_ylabel('Sales in Thousands')
plt.show()
```



Picture 9: *Difference between Promotion Sales*

Comment: The sales outlier mentioned above is due to the difference in promotions: Promotion 1&3 will have sales growth of 85 or more but in small quantities. Promotion 2 has Sales growth of 65 or more in high volume. From 2 sales anomalies at 3 promotion to have outlier from 65 to more

4.1.2.2. MarketSize Variable:

```
]: df['MarketSize'].unique()

]: array(['Medium', 'Small', 'Large'], dtype=object)
```

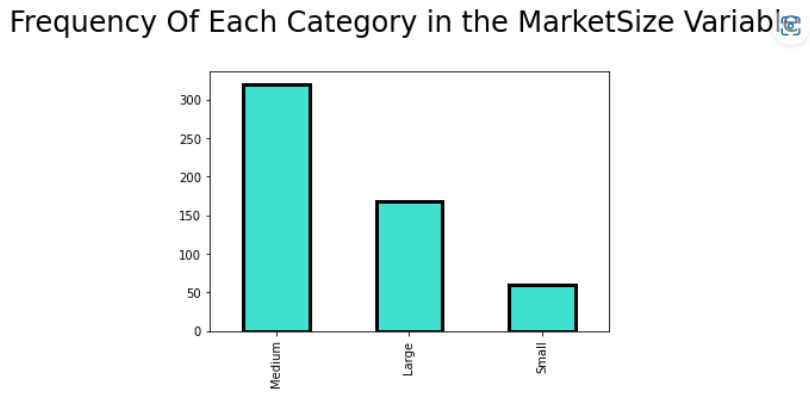
Picture 10: *Values in MarketSize*

```
: #CHECKING NUMBER OF UNIQUE CATEGORIES PRESENT IN THE "Education"
print("Unique categories present in the MarketSize:",df["MarketSize"].value_counts())
print('\n')

#VISUALIZING THE "Education"
df['MarketSize'].value_counts().plot(kind='bar',color = 'turquoise',edgecolor = "black",linewidth = 3)
plt.title("Frequency Of Each Category in the MarketSize Variable \n",fontsize=24)
plt.figure(figsize=(8,8))

Unique categories present in the MarketSize: Medium    320
Large     168
Small      60
Name: MarketSize, dtype: int64
```

```
: <Figure size 576x576 with 0 Axes>
```




```
<Figure size 576x576 with 0 Axes>
```

Picture11: *Frequency of size store in MarketSize*

There is a discrepancy in the data set in the MarketSize variable. In which: With MarketSize, Medium occupies the most (320 rows), Large(168 rows), Small(60 rows). To see if there is an effect between MarketSizes on Sales or not? We proceed to represent it through a Correlation chart. But since MarketSize is type object, we convert object to int by below code:

```
: df['MarketSize'] = df['MarketSize'].replace(['Small'],1)
  df['MarketSize'] = df['MarketSize'].replace(['Medium'],2)
  df['MarketSize'] = df['MarketSize'].replace(['Large'],3)
```

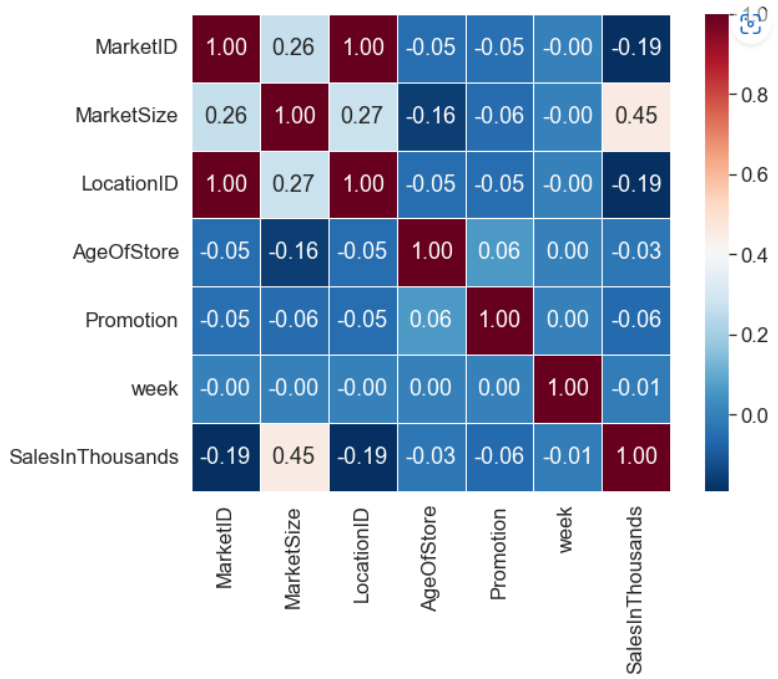
```
: df['MarketSize'].astype(int)
  df
```

:

	MarketID	MarketSize	LocationID	AgeOfStore	Promotion	week	SalesInThousands
0	1	2	1	4	3	1	33.73
1	1	2	1	4	3	2	35.67
2	1	2	1	4	3	3	29.03
3	1	2	1	4	3	4	39.25
4	1	2	2	5	2	1	27.81

Consider the correlation in the dataset:

```
plt.figure(figsize=(10,8))
sns.heatmap(df._get_numeric_data().astype(float).corr(),
            square=True, cmap='RdBu_r', linewidths=.5,
            annot=True, fmt='.2f').figure.tight_layout()
plt.show()
```

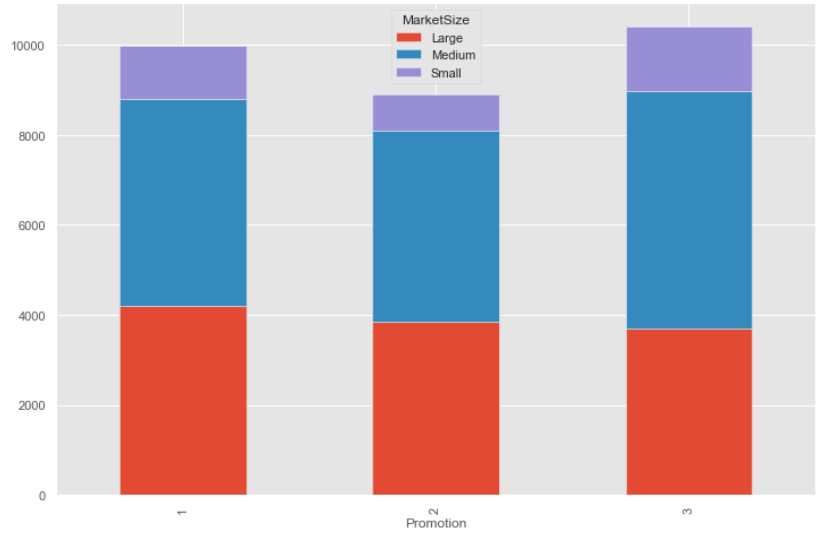


Picture 12: *Correlation of MarketSize with SaleInThousands*

Correlation between Size of Market with Sales is 0.45. From the result, we see that there is an impact on sales from MarketSize.

```
df.groupby(['Promotion', 'MarketSize']).sum()['SalesInThousands'].unstack('MarketSize').plot(
    kind='bar',
    figsize=(12,8),
    grid=True,
    stacked=True
)
ax.set_ylabel('Sales (in Thousands)')
ax.set_title('breakdowns of market sizes across different promotions')

plt.show()
```



Picture 13: *Correlation Size to Sales*

Comment: For small stores, promotion 2 is not effective so sales are lower than Size Large and Medium

Comment: For medium stores, promotion 1 and 3 will be more effective than promotion 2, so the sales between promotion 1 and 3 are equal and higher than promotion2.

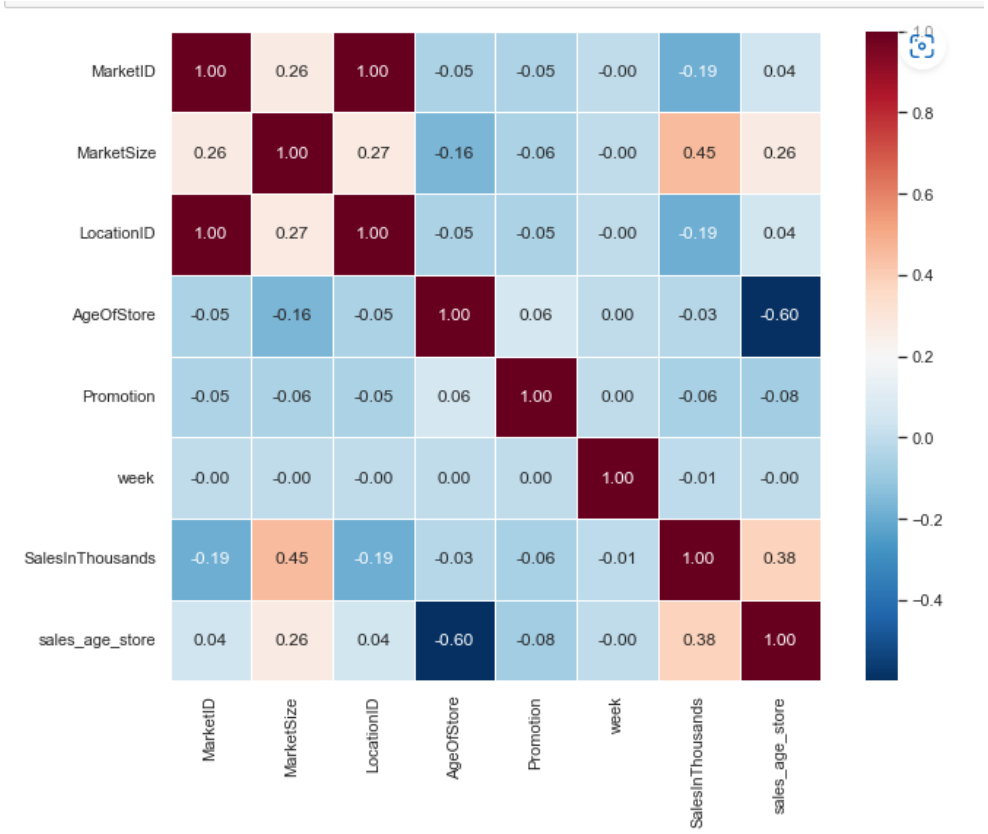
Comment: For Large stores, promotion 1 is the most effective (in the order of 1 2 3), so sales promotion 1 is higher than promotion 2 and 3.

4.1.2.3. AgeOfStore Variable:

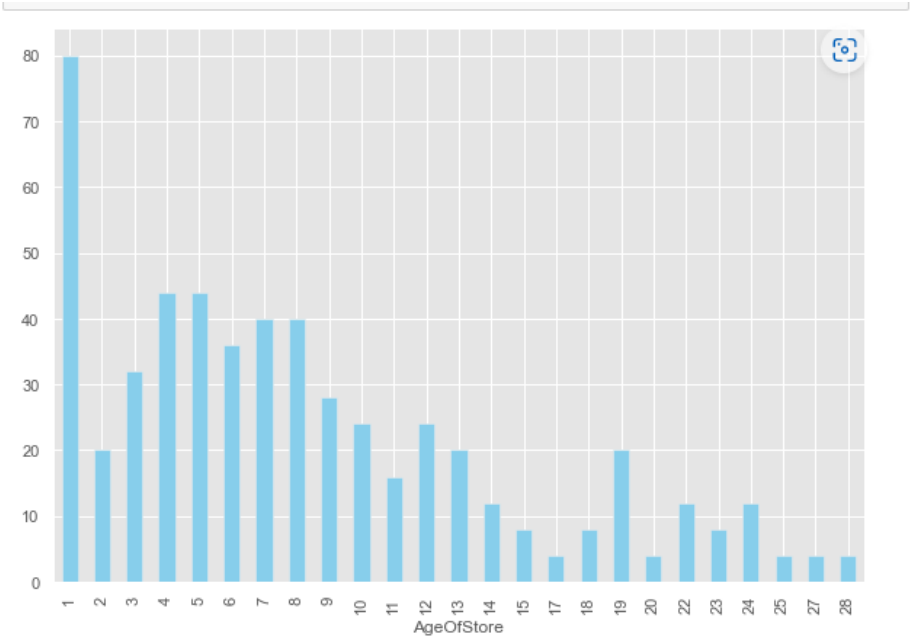
Create a new variable - Sales/(Age of store) to see if there is a relation with promotion and Market Size. Correlation between Age of Store with Sales is 0.26.

```
df['sales_age_store']=df['SalesInThousands']/df['AgeOfStore']
df.head(5)
```

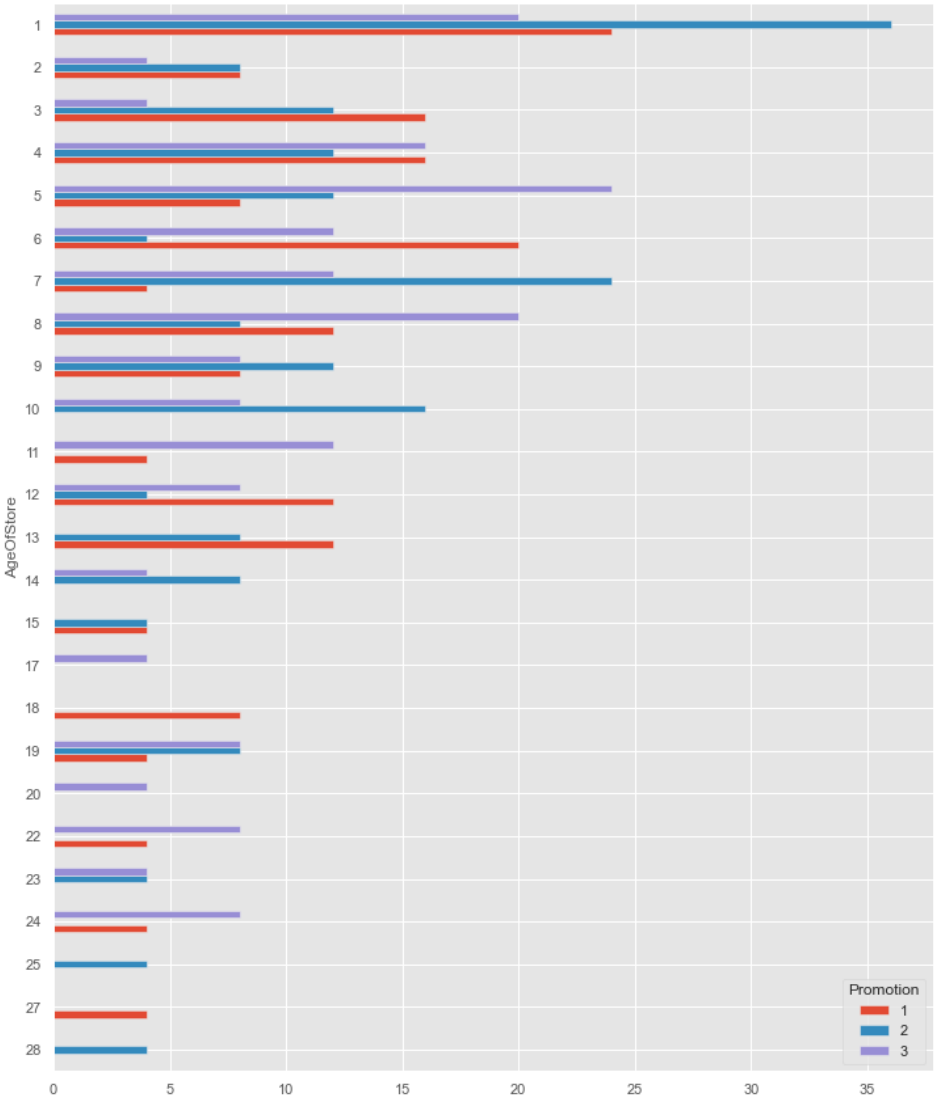
	MarketID	MarketSize	LocationID	AgeOfStore	Promotion	week	SalesInThousands	sales_age_store
0	1	2	1	4	3	1	33.73	8.4325
1	1	2	1	4	3	2	35.67	8.9175
2	1	2	1	4	3	3	29.03	7.2575
3	1	2	1	4	3	4	39.25	9.8125
4	1	2	2	5	2	1	27.81	5.5620



Picture 14: *Correlation between Age of Store with Sales*



Picture 15: *Age of Store*



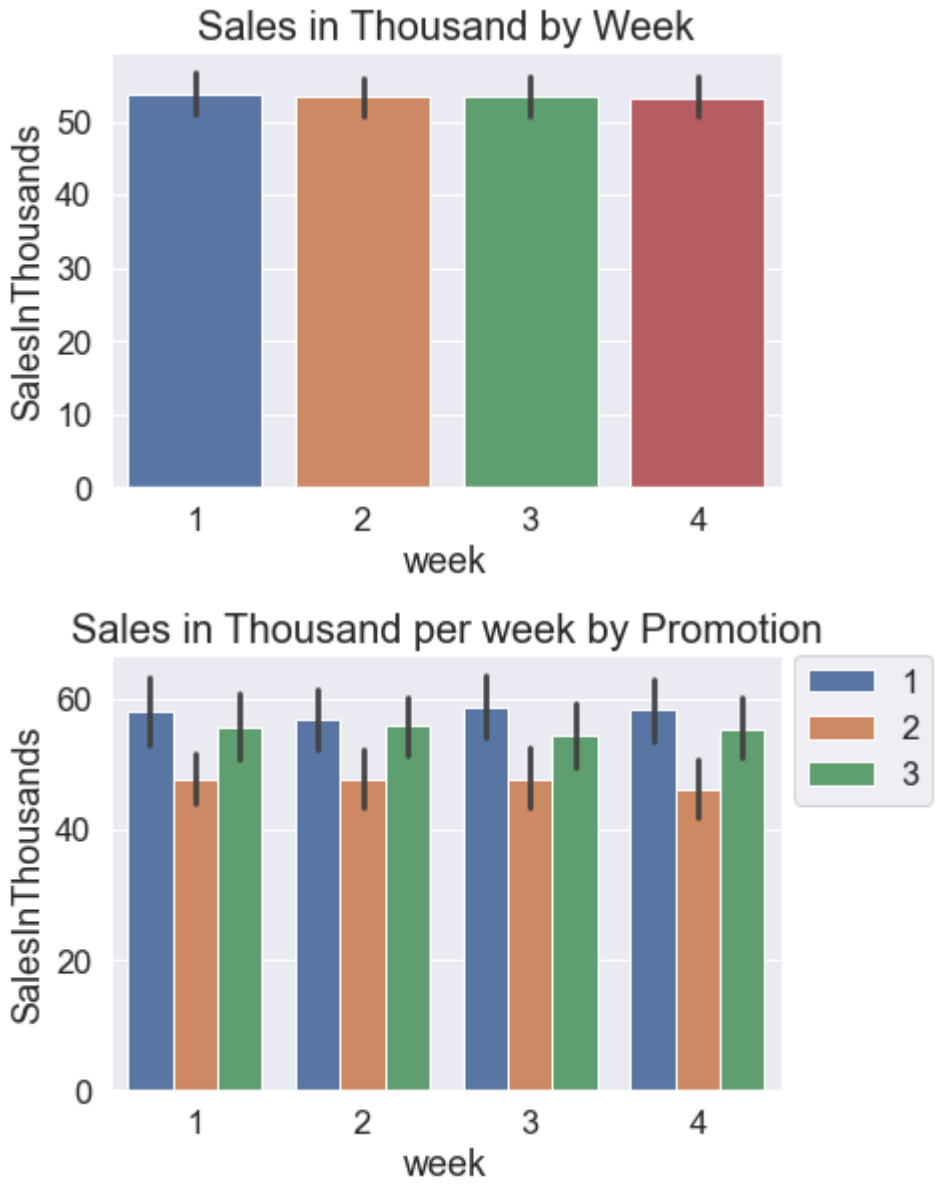
Picture 16: *Distribution of promotions according to store age*

In the 1-year-old stores, promotion2 was tested the most (36 stores). Judging from the sales analysis above, promotion 2 has the lowest sales coming from running at these stores too much, the 1-year-old store is little known, so sales will not be high.

There are no stores with ages 16, 21, 26.

4.1.2.4. Week Variable:

Sales in 4 weeks are balanced but have a difference by Promotion.



Picture 17: *Week by SalesInThousands*

Promotion 2 have Sales lowest promotion 1&3 all 4 weeks. Promotion 1 have Sales largest (but in week 2: Sales by promotion1 lower than others)

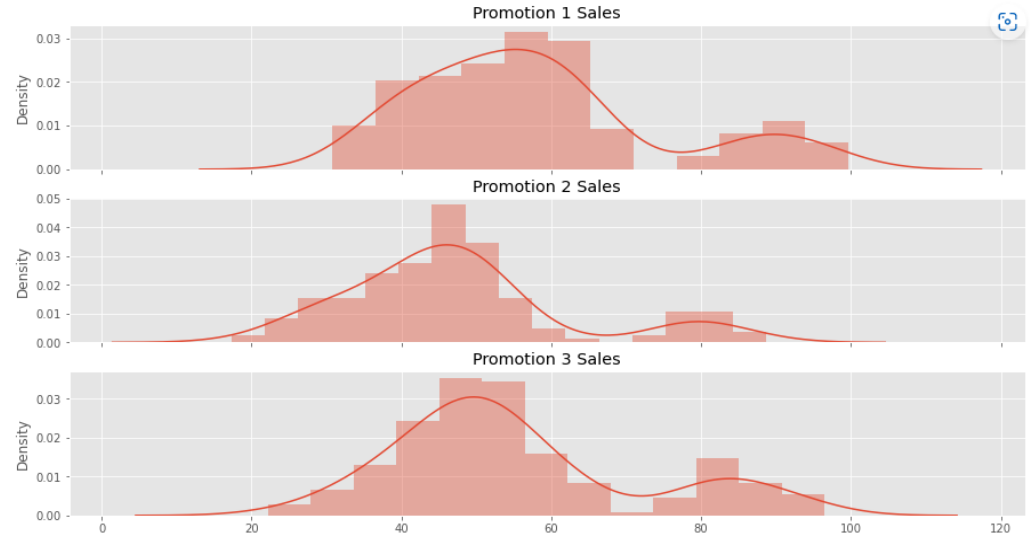
4.2. Model to solve the problem:

4.2.1. Overview distribution SaleInThousand:

We have a bimodal distribution:

```
fig,ax=plt.subplots(3,1,figsize=(15,8),sharex=True)

for i in [1,2,3]:
    sns.distplot(df.loc[df['Promotion']==i,'SalesInThousands'],ax=ax[i-1])
    ax[i-1].set_title('Promotion {} Sales'.format(i))
    ax[i-1].set_xlabel ("")
```

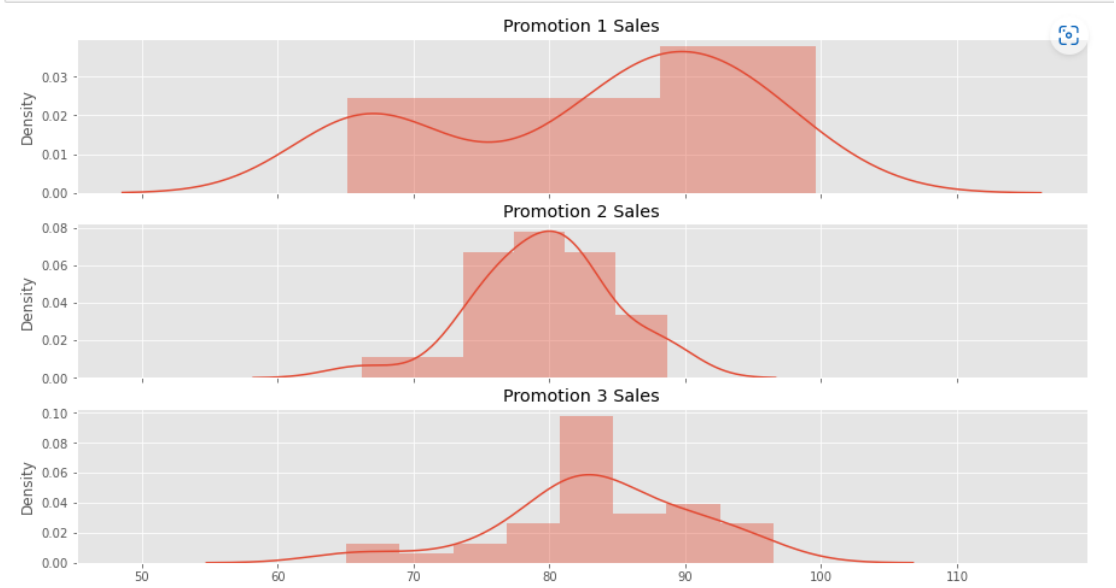


Picture 18 : *Distribution Promotion Sales*

- Case of promotion1: Distribution sales have an outlier value of ~75 or more.
- Case of promotion2: Distribution sales have an outlier value of ~65 or more.
- Case of promotion3: Distribution sales have an outlier value of ~75 or more.

➢ Collecting from sales of 3 promotions, we divide sales into 2 cases. Sales greater than 65 and less than 65

Next, we checked the distribution of > 65 groups to see clearly.



Picture 19: *Distribution Promotion Sales > 65*

Distribution of <= 65 sales group:

```
fig,ax=plt.subplots(3,1,figsize=(15,8),sharex=True)

for i in [1,2,3]:
    sns.distplot(df.loc[(df['Promotion']==i) & (df['SalesInThousands']<=65),'SalesInThousands'],ax=ax[i-1])
    ax[i-1].set_title('Promotion {} Sales'.format(i))
    ax[i-1].set_xlabel ("")
```

Picture 20 : *Distribution Promotion Sales < 65*

T-test requires normality but here we have a bimodal distribution, so will split into <= 65 & > 65 groups.

Next, We do count the observations that have sales > 65.

```
for i in [1,2,3]:
    print("Count of obs with >65 SALES promo_{}: {}".format(i,len(df[(df['SalesInThousands']>65) & (df['Promotion']==i)])))
    ###check how many obs in upper distribution
```

```
Count of obs with >65 SALES promo_1: 39
Count of obs with >65 SALES promo_2: 24
Count of obs with >65 SALES promo_3: 39
```

We have sufficient observation in all 3 promotion > 65 sales, thus we can conduct a separate t-test for >65 group

The number of stores for promotion 1 & 3 is 39 stores, which is larger than the number of stores implementing promotion 2 is 24 stores.

Now it is time for significance tests. To investigate which campaign was most successful we use the t-test. What exactly does a t-test do? Simply put, it looks to see if the mean values of two groups differ significantly. Let's take a look at the average sales figures broken down by campaign.

➢ Next, we calculate means sales promotion:

- *Means of Sales <65 group:*

```
: df[df["SalesInThousands"]<=65].groupby('Promotion')['SalesInThousands'].mean()

: Promotion
  1    50.744361
  2    42.608049
  3    47.983691
  Name: SalesInThousands, dtype: float64
```

- *Means of Sales >65 group:*

```
df[df["SalesInThousands"]>65].groupby('Promotion')['SalesInThousands'].mean()

Promotion
1    83.180256
2    79.592083
3    83.562821
Name: SalesInThousands, dtype: float64
```

Picture 21 : *Means Sale of all Promotion*

The mean values are different. But are these differences also significant? This question can be answered with a t-test. For this we use the t-test function from the library scipy.

There are two important statistics in a t-Test, the t-value and the p-value:

The t-value measures the degree of difference relative to the variation in the data. The larger the t-value is, the more difference there is between two groups.

On the other hand the p-value measures the probability that the results would occur by chance. The smaller the p-value is, the more statistically significant difference there will be between the two groups.

4.2.2. Assumption Control:

The common assumptions made when doing a t-test include normality of data distribution and equality of variance in standard deviation.

The samples must have been taken from different populations (that might have the same mean or not, but they surely have different variances). What is the rationale behind judging differences in mean values? Just as a side note: often the difference in variances under H0 can be explained by inhomogeneities within one of the groups.

a. Normality Test for <=65 group

H0: The data is normally distributed

H1: The data is not normally distributed

```
from scipy.stats import shapiro
print ("Promotion 1 & <= 65: {}".format(shapiro(df.loc[(df['Promotion'] == 1)
                                                       & (df['SalesInThousands']<=65), 'SalesInThousands'])))
print ("Promotion 2 & <= 65: {}".format(shapiro(df.loc[(df['Promotion'] == 2)
                                                       & (df['SalesInThousands']<=65), 'SalesInThousands'])))
print ("Promotion 3 & <= 65: {}".format(shapiro(df.loc[(df['Promotion'] == 3)
                                                       & (df['SalesInThousands']<=65), 'SalesInThousands'])))
```

```
Promotion 1 & <= 65: ShapiroResult(statistic=0.958892285823822, pvalue=0.0004932560259476304)
Promotion 2 & <= 65: ShapiroResult(statistic=0.9712842702865601, pvalue=0.0017379027558490634)
Promotion 3 & <= 65: ShapiroResult(statistic=0.981341660022735 6, pvalue=0.04059775173664093)
```

Comment: *All 3 groups pass normality test*

b. Normality Test for >65 group

H0: The data is normally distributed

H1: The data is not normally distributed

```
from scipy.stats import shapiro
print ("Promotion 1 & > 65: {}".format(shapiro(df.loc[(df['Promotion'] == 1)
                                                       & (df['SalesInThousands']>65), 'SalesInThousands'])))
print ("Promotion 2 & > 65: {}".format(shapiro(df.loc[(df['Promotion'] == 2)
                                                       & (df['SalesInThousands']>65), 'SalesInThousands'])))
print ("Promotion 3 & > 65: {}".format(shapiro(df.loc[(df['Promotion'] == 3)
                                                       & (df['SalesInThousands']>65), 'SalesInThousands'])))
```

```
Promotion 1 & > 65: ShapiroResult(statistic=0.8731706738471985, pvalue=0.0004092589661013335)
Promotion 2 & > 65: ShapiroResult(statistic=0.9646238088607788, pvalue=0.5380414724349976)
Promotion 3 & > 65: ShapiroResult(statistic=0.9593083262443542, pvalue=0.16909459233283997)
```

Comment: 2 groups fail normality test, so testing can't be done for these 3 groups

H0: The variances are equal(homogenous)

H1: The variances are unequal(non-homogenous)

```
#promo 1 & promo 2
stats.levene(df.loc[(df['Promotion'] == 1) & (df['SalesInThousands']<=65),'SalesInThousands']
             , df.loc[(df['Promotion'] == 2) & (df['SalesInThousands']<=65), 'SalesInThousands'])
```

```
LeveneResult(statistic=0.08101138968625114, pvalue=0.776131155039429)
```

```
#promo 1 & promo 3
stats.levene(df.loc[(df['Promotion'] == 1) & (df['SalesInThousands']<=65),'SalesInThousands']
             , df.loc[(df['Promotion'] == 3) & (df['SalesInThousands']<=65), 'SalesInThousands'])
```

```
LeveneResult(statistic=0.6698505571326162, pvalue=0.41379969879676226)
```

Hình : Checking Same Variance Assumption

Comments:

- Promo 1 & 3 have diff variances, so will do unequal variances t-test for them
- Promo 1 & 2 have diff variances, so will do unequal variances t-test for them

4.2.3. A/B Testing for <=65 group

a. Comparing Promotion 1 vs Promotion 2:

```
t, p= stats.ttest_ind(df.loc[(df['Promotion'] == 1) & (df['SalesInThousands']<=65), 'SalesInThousands'],
    df.loc[(df['Promotion'] == 2) & (df['SalesInThousands']<=65), 'SalesInThousands'],
    equal_var=False)
print("t-value = " +str(t))
print("p-value = " +str(p))
```

```
t-value = 7.824776742399254
p-value = 9.879819174248633e-14
```

```
df[df["SalesInThousands"]<=65].groupby('Promotion')['SalesInThousands'].mean()
```

```
Promotion
1    50.744361
2    42.608049
3    47.983691
Name: SalesInThousands, dtype: float64
```

Comment: P < 0.05 thus statistically significant difference in sales of Promo 1 & 2, promo 1 should be preferred

Our p-Value is close to 0 which means that there is good evidence to *reject the null hypothesis*. Meaning that there is a statistical difference between the two groups. Our threshold rejectings the Null is usually less than 0.05.

Furthermore, the t-test shows that the marketing performances for these two groups are significantly different and that first marketing campaigns outperforms second marketing campaigns.

b. Comparing Promotion 1 vs Promotion 3

```
t, p= stats.ttest_ind(df.loc[(df['Promotion'] == 1) & (df['SalesInThousands']<=65), 'SalesInThousands'],
    df.loc[(df['Promotion'] == 3) & (df['SalesInThousands']<=65), 'SalesInThousands'],
    equal_var=False)
print("t-value = " +str(t))
print("p-value = " +str(p))
```

```
t-value = 2.6332470611279164
p-value = 0.008932765144586902
```

Comment: P < 0.05 thus statistically significant difference in sales of Promo 1 & 3, promo 1 should be preferred. However, if we run a t-test between the promotion group 1 and promotion group 3, we see different results

We note that the average sales from first marketing promotion is higher than those from third marketing campaigns. But, running a t-test between these two groups, gives us a t-value of 2.633 and a p-value of 0.089. The computed **p-value is a lot higher than 0.05**, past the threshold for statistical significance.

c. Comparing Promotion 2 vs Promotion 3:

```
t, p= stats.ttest_ind(df.loc[(df['Promotion'] == 2) & (df['SalesInThousands']<=65), 'SalesInThousands'],
    df.loc[(df['Promotion'] == 3) & (df['SalesInThousands']<=65), 'SalesInThousands'],
    equal_var=False)
print("t-value = " +str(t))
print("p-value = " +str(p))
```

```
t-value = -5.308277452910551
p-value = 2.1146060264994 46e-07
```

```
df[df["SalesInThousands"]<=65].groupby('Promotion')['SalesInThousands'].mean()
```

```
Promotion
1    50.744361
2    42.608049
3    47.983691
Name: SalesInThousands, dtype: float64
```

Picture :

Comment: P<0.05 thus statistically significant difference in sales of Promo 2 & 3, promo 3 should be preferred

Based on the average sales, the second marketing promotion is higher than those from the third marketing promotion. If we run the t-test and between these two promotions, results show that the p-value (2.1146 - 07) is lower than 0.05 threshold, which concludes that it **rejects null hypotheses**. In other words, there's a statistical difference between second marketing campaigns and third marketing campaigns.

CHAPTER 5. CONCLUSION

As we can see from the p-value, the average value of the sales figures in promotion 1 and 3 do not differ significantly. But the difference between promotion 1 and 2 does as well as promotion 2 vs. 3. If you look at the corresponding t-value, you can say that promotion 1 and 3 were better than promotion 2.

Of all 3 promotions 1st is the best one and 2nd should be stopped & replaced with 1 or 3. Out of 1 or 3 depending on the cost of promotion selection can be made, if promo 1 & 3 cost the same then 1 is the way to go and if the cost of 1 is so high that it removes the gains from extra sales then promo 3 is to be used.

Since a lot depends on the decisions of the marketing department, it is worthwhile, especially for far-reaching decisions, to carry out extensive A/B tests to get the desired results.

REFERENCES

https://viblo.asia/p/ab-testing-V3m5WXmbKO7

https://www.kaggle.com/code/rucheiitr/a-b-testing-fast-food-marketing-campaign/notebook

https://www.kaggle.com/code/kingabzpro/alcoholic-drinks-in-russia-and-design-promotional#Categorizing-Clusters

https://www.javatpoint.com/aggregation-in-data-mining

https://khamdb.com/ad-tech/when-and-when-not-to-a-b-test/

https://khamdb.com/ad-tech/how-to-predict-the-success-of-your-marketing-campaign/