MAY 6, 2021

RETAIL DATA ANALYSIS

WITH SIX SIGMA

SABINA MAMMADOVA
PRODUCTION & OPERATIONS MANAGEMENT
ADA University | Baku 2021

Contents

Overview

It is common for the modern businesses to get valuable insights out of the logs, backed-up data from the past to increase the profit by building and applying best-fit marketing strategies.

The aim of this report is to analyze historical sales data collected from 45 stores located in different regions. 2 datasets are obtained: features and sales where “features” set contains additional data related to store, department, and regional activities over 8K records with 7 columns; “sales” set is the actual data over 420K records with 5 columns. However, around ~101K records are kept which included the data from 11 stores with various departments covering the period February 2010 – October 2012. It is then merged with the “features” dataset and below is the final setup of dataset with 9 columns:

	Store	Dept	Date	Weekly_Sales	IsHoliday	Temperature	Fuel_Price	CPI	Unemployment
0	1	1	12/02/2010	46039.49	True	38.51	2.548	211.242170	8.106
1	1	1	19/02/2010	41595.55	False	39.93	2.514	211.289143	8.106
2	1	1	26/02/2010	19403.54	False	46.63	2.561	211.319643	8.106
3	1	1	05/03/2010	21827.90	False	46.50	2.625	211.350143	8.106
4	1	1	12/03/2010	21043.39	False	57.79	2.667	211.380643	8.106
...	...	...	...	...	...	...	...	...	...
101473	11	27	28/09/2012	1599.02	False	77.67	3.666	226.518093	6.334
101474	11	27	05/10/2012	2224.03	False	73.37	3.617	226.721036	6.034
101475	11	27	12/10/2012	2478.86	False	69.94	3.601	226.923979	6.034
101476	11	27	19/10/2012	2594.25	False	73.77	3.594	226.968844	6.034
101477	11	27	26/10/2012	2638.82	False	74.26	3.506	226.987364	6.034

101478 rows × 9 columns

As population out of this dataset, store #1 is selected which includes over 10K records:

	Store	Dept	Date	Weekly_Sales	IsHoliday	Temperature	Fuel_Price	CPI	Unemployment
0	1	1	12/02/2010	46039.49	True	38.51	2.548	211.242170	8.106
1	1	1	19/02/2010	41595.55	False	39.93	2.514	211.289143	8.106
2	1	1	26/02/2010	19403.54	False	46.63	2.561	211.319643	8.106
3	1	1	05/03/2010	21827.90	False	46.50	2.625	211.350143	8.106
4	1	1	12/03/2010	21043.39	False	57.79	2.667	211.380643	8.106
...	...	...	...	...	...	...	...	...	...
10237	1	99	24/08/2012	0.07	False	77.66	3.620	222.171946	6.908
10238	1	99	31/08/2012	20.06	False	80.49	3.638	222.305480	6.908
10239	1	99	07/09/2012	0.05	True	83.96	3.730	222.439015	6.908
10240	1	99	14/09/2012	0.03	False	74.97	3.717	222.582019	6.908
10241	1	99	05/10/2012	635.00	False	68.55	3.617	223.181477	6.573

10242 rows × 9 columns

Columns:

Store – the store number (id)	*Temperature* – average temperature in the region
Dept – the department number (id)	*Fuel_Price* – cost of fuel in the region
Date – the week	*CPI* – the Consumer Price Index
Weekly_Sales – sales for the given department in the given store ($)	*Unemployment* – unemployment rate for the region
IsHoliday – whether the week is a special holiday week	

Purpose: The target is to check whether the correlation between the weekly sales and the several remaining features exists.

Central Limit Theorem

As a strategy, random sampling is utilized, and negative values are removed. For the application of CLT, 500 samples are taken from the population and the number of observations per sample is 2000. The distribution of records is demonstrated below:

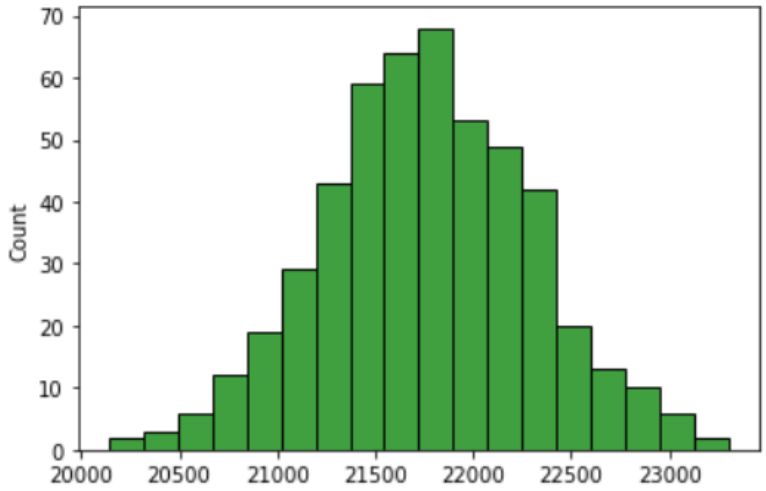
```
means = []

for i in range (0, 500):
    means.append(df.sample(2000).describe().reset_index()['Weekly_Sales'][1])
```

```
import seaborn as sns

sns.histplot(means, color='green')
```

```
<AxesSubplot:ylabel='Count'>
```




```
df['Weekly_Sales'].mean()
```

```
21744.574388383622
```

From the resulting histogram, it can be inferred that by taking sufficient random samples from the population, the distribution of the sample means will follow normal distribution. Mean of the population is around the same with the middle value in the histogram.

Distribution of Numeric Parameters

To visualize the distribution of numeric parameter – Weekly_Sales, the records greater than 50000 are excluded to have the clear graphical representation; they are the outliers and did not make huge difference in the number of records – dropped to 8973 from 10227:

```
indexes = df[df['Weekly_Sales'] > 50000].index
```

```
df.drop(indexes, inplace = True)
```

```
df
```

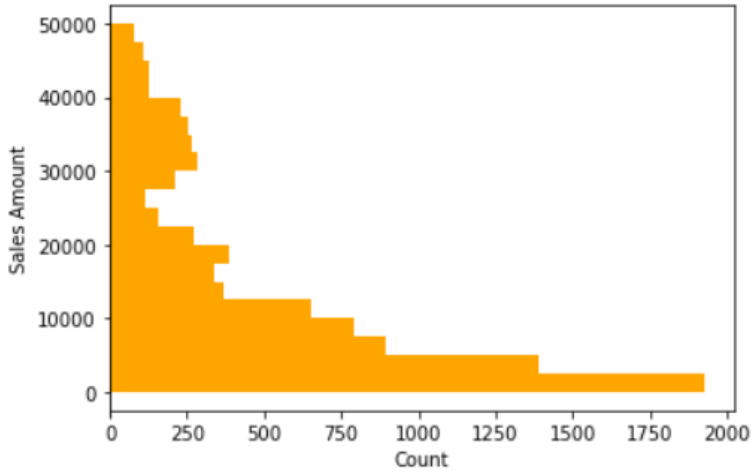
	Store	Dept	Date	Weekly_Sales	IsHoliday	Temperature	Fuel_Price	CPI	Unemployment
0	1	1	12/02/2010	46039.49	True	38.51	2.548	211.242170	8.106
1	1	1	19/02/2010	41595.55	False	39.93	2.514	211.289143	8.106
2	1	1	26/02/2010	19403.54	False	46.63	2.561	211.319643	8.106
3	1	1	05/03/2010	21827.90	False	46.50	2.625	211.350143	8.106
4	1	1	12/03/2010	21043.39	False	57.79	2.667	211.380643	8.106
...	...	...	...	...	...	...	...	...	...
10237	1	99	24/08/2012	0.07	False	77.66	3.620	222.171946	6.908
10238	1	99	31/08/2012	20.06	False	80.49	3.638	222.305481	6.908
10239	1	99	07/09/2012	0.05	True	83.96	3.730	222.439015	6.908
10240	1	99	14/09/2012	0.03	False	74.97	3.717	222.582019	6.908
10241	1	99	05/10/2012	635.00	False	68.55	3.617	223.181477	6.573

8973 rows × 9 columns

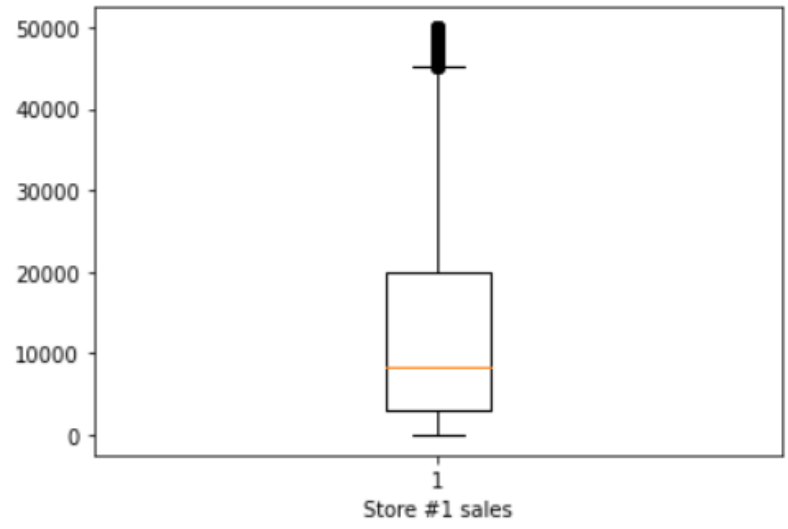
Then, the histogram, boxplot of whole dataset and the boxplot of the weekly sales during 2010 for store #1 is obtained:

```
import matplotlib.pyplot as plt

plt.hist(df['Weekly_Sales'], bins = 20, orientation = 'horizontal', color = 'orange')
plt.xlabel('Count')
plt.ylabel('Sales Amount')
plt.show()
```




```
plt.boxplot(df['Weekly_Sales'])
plt.xlabel('Store #1 sales')
plt.show()
```



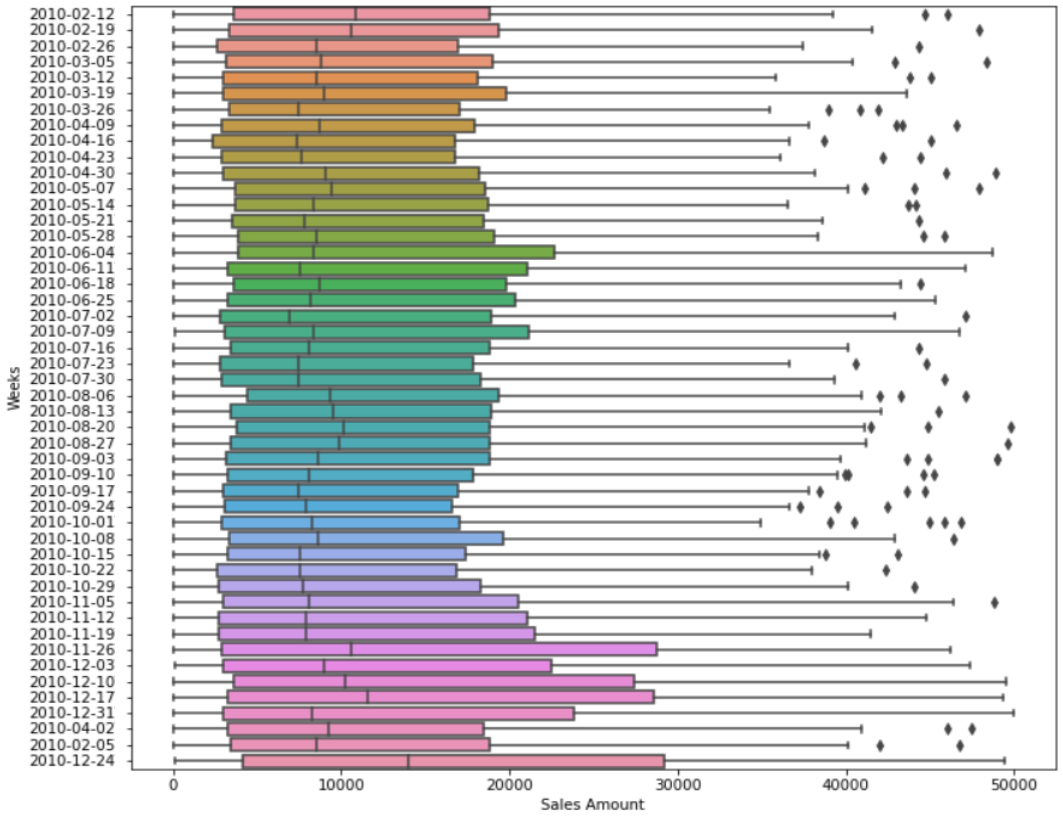
From the histogram, number of occurrences in the 10k ranges can be analyzed. For instance, as the sales amount per range increases, the number of sales significantly decreases. From the boxplot, several statistical values such that the distribution of sales amounts based on upper and lower quartile points, minimum and maximum observations, the median and the outliers can be demonstrated.

From the box plot of 2010, the increase/decrease in the sales rates, the rate of change in the medians among different times of the year in all departments of the store can be observed. For example, close to the Christmas (the last box plot), the sales rate, most importantly the median is significantly higher compared to the first week of December (2010-12-03->10):

```
Sales10 = df[(df['Date'] < '2011-01-01')]
```

```
import seaborn as sns

plt.figure(figsize=(11, 10))
sns.boxplot(x='Weekly_Sales',
            y= 'Date',
            data=Sales10)
plt.xlabel('Sales Amount')
plt.ylabel('Weeks')
```



Process Capability

The chosen process for measuring the capability is the customers' adaptabilities to the changed prices in the goods and services of the store per year. For this, "CPI" column is used which stands for Consumer Price Index measuring "the average change overtime in the prices paid by urban consumers for a market basket of consumer goods and services". It also allows to compare the variance in the value of suggested service that a dollar may buy through different dates. When the price increase, it influences the purchasing power of the consumer's dollar, as a result, CPI decreases.

In 6 Sigma, Cp and Cpk measures are used for process capability tests such that they show how consistent and close the observations are to the average performance. The larger the value of Cp and Cpk, the better the performance of the process is accepted.

Below are the corresponding values for Cp and Cpk per year and Z-scores:

```
import statistics as stat

std = stat.stdev(df['CPI'])
mu = df['CPI'].mean()
```

```
mean10 = Sales10['CPI'].mean()
lsl = min(Sales10['CPI'])
usl = max(Sales10['CPI'])
cp = (usl - lsl) / 6 * std
z_score = (mean10 - mu) / std
cpk = z_score / 3
```

```
mean11 = Sales11['CPI'].mean()
lsl = min(Sales11['CPI'])
usl = max(Sales11['CPI'])
cp = (usl - lsl) / 6 * std
z_score = (mean11 - mu) / std
cpk = z_score / 3
```

```
mean12 = Sales12['CPI'].mean()
lsl = min(Sales12['CPI'])
usl = max(Sales12['CPI'])
cp = (usl - lsl) / 6 * std
z_score = (mean12 - mu) / std
cpk = z_score / 3
```

```
z_score
-1.0786229968410193
```

```
z_score
-0.5580279953968574
```

```
z_score
1.3103246921758178
```

```
cp
1.2036509927739467
```

```
cp
6.6437092152441934
```

```
cp
2.6940064779502984
```

```
cpk
-0.3595409989470064
```

```
cpk
-0.1860093317989525
```

```
cpk
0.43677489739193925
```

2010 ***2011*** ***2012***

According to the Z-scores, the space that falls behind the specification are per year is the following:

Year	% (rounded to 2 digits after decimal)
2010	0.28
2011	0.58
2012	0.19

From the produced results, it can be implied that from year to year, Cpk values tend to increase – it assists the process capability and the specification for the last year is significantly lower than the remaining indicators which means the CPI values of customers gradually falls within the specification limits.

Regression Analysis

For the regression analysis, "<50000" condition on the dataset is kept because of the outliers. The purpose is to check the relationship between independent variables – CPI, Unemployment, and dependent variable – Weekly_Sales. As a technique, Multiple Linear Regression is applied to test whether there is a correlation between the regressors and the response. The train and test sets are kept under 70/30 proportion; below the implementation, the scatter plots and R^2 values for train and test sets are shown:

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

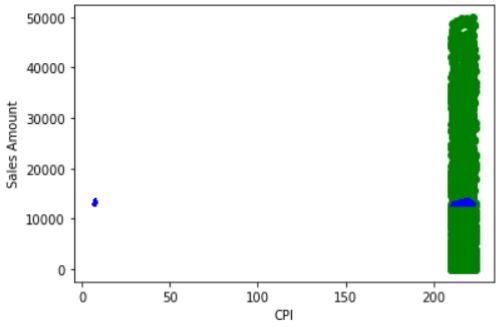
```
X = df[['CPI', 'Unemployment']]
y = df['Weekly_Sales']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 0)
```

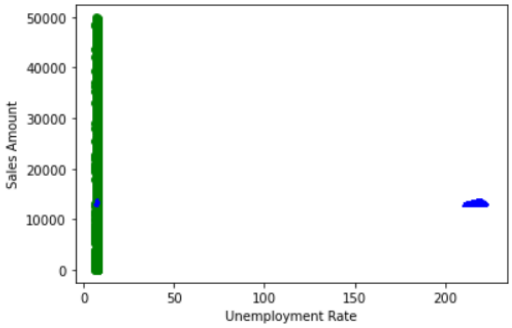
```
rgr = LinearRegression()
rgr.fit(X_train, y_train)

y_pred = rgr.predict(X_test)
```

```
plt.scatter(X_test.iloc[:,0].values, y_test, color = 'green')
plt.plot(X_train, rgr.predict(X_train), color = 'blue')
plt.xlabel('CPI')
plt.ylabel('Sales Amount')
plt.show()
```




```
plt.scatter(X_test.iloc[:,1].values, y_test, color = 'green')
plt.plot(X_train, rgr.predict(X_train), color = 'blue')
plt.xlabel('Unemployment Rate')
plt.ylabel('Sales Amount')
plt.show()
```




```
from sklearn.metrics import r2_score

r2 = r2_score(y_train, rgr.predict(X_train))
r2
```

```
0.0006394904683084679
```

```
r3 = r2_score(y_test, rgr.predict(X_test))
r3
```

```
-0.0012899274536584127
```

As it can be inferred from the plots and the coefficients of determination from the train and test sets, there is no correlation between the weekly sales and 2 independent variables.

References

Consumer Price Index. (November 25, 2020). U.S. Bureau of Labor Statistics. Retrieved from https://www.bls.gov/cpi/questions-and-answers.htm#Question_1