PROJECT REPORT

E-COMMERCE SHIPPING

DATA ANALYSIS

INDEX

INTRODUCTION

What is E-Commerce shipping?

E-commerce shipping is the way in which products ordered online are delivered to the location of the buyer. Usually businesses can work with e-commerce logistics providers to understand what works best for their business.

Logistics companies play an important and vital role in the success of the e-commerce business, as customer satisfaction is directly related to the handling of the shipment. The packaging and safe delivery of the product is a factor that keeps the customer happy and ordering more from the e-commerce site. An e-commerce business needs to have a strong and effective strategy in place for the shipping and delivery of goods and services, as this is a competitive differentiator as well in the market.

The most important functions of e-commerce logistics are warehousing and inventory management, order creation and waybill generation, transportation of orders and in-transit storage, handling delivery exceptions and finally reverse logistics. An essential part of transportation management lies in building an efficient supply chain from the six main modes of transportation: road, maritime, air, rail, intermodal, and pipeline. Understanding the strengths and weaknesses of each mode is paramount to building an effective supply chain.

Nowadays, consumers choose who to buy from based on their overall customer experience. Having the lowest price or best product no longer guarantees a sale. The e-commerce sites need to provide them with a faster delivery service to ensure customer satisfaction, which will ultimately boost the growth of the business. All three modes of shipping-land, air, and sea-play a major role in our economy. Each offers benefits that the other mode of transport might not offer. It is up to the Logistics to make a well-informed decision of choosing the right mode of shipping that will be beneficial.

Top E-Commerce sites in the world

1. amazon.com
2. ebay.com
3. rakuten.co.jp
4. aliexpress.com
5. Walmart.com

Top E-Commerce Logistics Companies in Global Market 2021

1. DHL
2. KENCO
3. CLIPPER
4. FEDEX
5. XPO

E-COMMERCE SHIPPING DATA ANALYSIS

Import the Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

Data Extraction

Extract the Ecommerce Shipping Dataset

```
df=pd.read_excel('Project-Ecommerce Shipping Data Analysis (Python).xlsx')
d=df.copy()
```

Ecommerce Shipping DataFrame

```
d
```

```
          ID Warehouse_block Mode_of_Shipment  Customer_care_calls  \
0          1               D           Flight                    4   
1          2               F           Flight                    4   
2          3               A           Flight                    2   
3          4               B           Flight                    3   
4          5               C           Flight                    2   
...      ...             ...              ...                  ...   
10994  10995               A             Ship                    4   
10995  10996               B             Ship                    4   
10996  10997               C             Ship                    5   
10997  10998               F             Ship                    5   
10998  10999               D             Ship                    2   

       Customer_rating  Cost_of_the_Product  Prior_purchases  \
0                    2                  177                3   
1                    5                  216                2   
2                    2                  183                4   
3                    3                  176                4   
4                    2                  184                3   
...                ...                  ...              ...   
10994                1                  252                5   
10995                1                  232                5   
```

```
10996                4                  242                 5
10997                2                  223                 6
10998                5                  155                 5

      Product_importance Gender  Discount_offered  Weight_in_gms  \
0                    low      F                44           1233
1                    low      M                59           3088
2                    low      M                48           3374
3                 medium      M                10           1177
4                 medium      F                46           2484
...                  ...    ...               ...            ...
10994             medium      F                 1           1538
10995             medium      F                 6           1247
10996                low      F                 4           1155
10997             medium      M                 2           1210
10998                low      F                 6           1639

       Reached.on.Time_Y.N
0                        1
1                        1
2                        1
3                        1
4                        1
...                    ...
10994                    1
10995                    0
10996                    0
10997                    0
10998                    0

[10999 rows x 12 columns]
```

DataFrame Shape

```
d.shape
```

```
(10999, 12)
```

DataFrame Columns

```
d.columns
```

```
Index(['ID', 'Warehouse_block', 'Mode_of_Shipment', 'Customer_care_calls',
       'Customer_rating', 'Cost_of_the_Product', 'Prior_purchases',
       'Product_importance', 'Gender', 'Discount_offered', 'Weight_in_gms',
       'Reached.on.Time_Y.N'],
      dtype='object')
```

First Five Rows of the DataFrame

```
d.head()
```

```
   ID Warehouse_block Mode_of_Shipment  Customer_care_calls  Customer_rating  
\
0   1               D           Flight                    4                2   
1   2               F           Flight                    4                5   
2   3               A           Flight                    2                2   
3   4               B           Flight                    3                3   
4   5               C           Flight                    2                2   

   Cost_of_the_Product  Prior_purchases Product_importance Gender  \
0                  177                3                low      F   
1                  216                2                low      M   
2                  183                4                low      M   
3                  176                4             medium      M   
4                  184                3             medium      F   

   Discount_offered  Weight_in_gms  Reached.on.Time_Y.N  
0                44           1233                    1  
1                59           3088                    1  
2                48           3374                    1  
3                10           1177                    1  
4                46           2484                    1  
```

Last Five Rows of the DataFrame

```
d.tail()
```

```
          ID Warehouse_block Mode_of_Shipment  Customer_care_calls  \
10994  10995               A             Ship                    4   
10995  10996               B             Ship                    4   
10996  10997               C             Ship                    5   
10997  10998               F             Ship                    5   
10998  10999               D             Ship                    2   

       Customer_rating  Cost_of_the_Product  Prior_purchases  \
10994                1                  252                5   
10995                1                  232                5   
10996                4                  242                5   
10997                2                  223                6   
10998                5                  155                5   

      Product_importance Gender  Discount_offered  Weight_in_gms  \
10994             medium      F                 1           1538   
10995             medium      F                 6           1247   
10996                low      F                 4           1155   
10997             medium      M                 2           1210   
10998                low      F                 6           1639   

       Reached.on.Time_Y.N  
10994                    1  
10995                    0  
10996                    0  
```

```
10997                   0
10998                   0
```

DataFrame Information

```
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10999 entries, 0 to 10998
Data columns (total 12 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   ID                   10999 non-null  int64 
 1   Warehouse_block      10999 non-null  object
 2   Mode_of_Shipment     10999 non-null  object
 3   Customer_care_calls  10999 non-null  int64 
 4   Customer_rating      10999 non-null  int64 
 5   Cost_of_the_Product  10999 non-null  int64 
 6   Prior_purchases      10999 non-null  int64 
 7   Product_importance   10999 non-null  object
 8   Gender               10999 non-null  object
 9   Discount_offered     10999 non-null  int64 
 10  Weight_in_gms        10999 non-null  int64 
 11  Reached.on.Time_Y.N  10999 non-null  int64 
dtypes: int64(8), object(4)
memory usage: 1.0+ MB
```

Warehouse Blocks

```
np.sort(d['Warehouse_block'].unique())
```

```
array(['A', 'B', 'C', 'D', 'F'], dtype=object)
```

Mode of Shipment

```
np.sort(d['Mode_of_Shipment'].unique())
```

```
array(['Flight', 'Road', 'Ship'], dtype=object)
```

Data Cleansing

Check for Null Values in the entire DataFrame

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

```
0
```

Check for Null Values in the columns

```
d.isnull().sum()
```

```
ID                     0
Warehouse_block        0
Mode_of_Shipment       0
Customer_care_calls    0
Customer_rating        0
```

```
Cost_of_the_Product    0
Prior_purchases        0
Product_importance     0
Gender                 0
Discount_offered       0
Weight_in_gms          0
Reached.on.Time_Y.N    0
dtype: int64
```

Data Transformation

Drop the columns

```
d.drop(['ID','Prior_purchases'],axis=1,inplace=True)
d
```

```
      Warehouse_block Mode_of_Shipment  Customer_care_calls  Customer_rating
\
0                   D           Flight                    4                2
1                   F           Flight                    4                5
2                   A           Flight                    2                2
3                   B           Flight                    3                3
4                   C           Flight                    2                2
...               ...              ...                  ...              ...
10994               A             Ship                    4                1
10995               B             Ship                    4                1
10996               C             Ship                    5                4
10997               F             Ship                    5                2
10998               D             Ship                    2                5

       Cost_of_the_Product Product_importance Gender  Discount_offered  \
0                      177                low      F                44
1                      216                low      M                59
2                      183                low      M                48
3                      176             medium      M                10
4                      184             medium      F                46
...                    ...                ...    ...               ...
10994                  252             medium      F                 1
10995                  232             medium      F                 6
10996                  242                low      F                 4
10997                  223             medium      M                 2
10998                  155                low      F                 6

       Weight_in_gms  Reached.on.Time_Y.N
0               1233                    1
1               3088                    1
2               3374                    1
3               1177                    1
4               2484                    1
...              ...                  ...
10994           1538                    1
```

```
10995           1247                    0
10996           1155                    0
10997           1210                    0
10998           1639                    0

[10999 rows x 10 columns]
```

Data sorted by Warehouse block column

```
d=d.sort_values(by='Warehouse_block', ignore_index=1)
d
```

```
      Warehouse_block Mode_of_Shipment  Customer_care_calls  Customer_rating  
\
0                   A             Road                    5                2
1                   A             Ship                    3                4
2                   A             Ship                    3                1
3                   A             Ship                    3                3
4                   A             Road                    4                5
...               ...              ...                  ...              ...
10994               F             Ship                    4                4
10995               F             Road                    4                4
10996               F             Ship                    3                4
10997               F             Ship                    5                5
10998               F           Flight                    6                1

       Cost_of_the_Product Product_importance Gender  Discount_offered  \
0                      249             medium      M                 6   
1                      197             medium      M                 3   
2                      214             medium      F                 2   
3                      193                low      F                60   
4                      204             medium      F                 4   
...                    ...                ...    ...               ...   
10994                  190                low      M                 2   
10995                  266                low      F                 6   
10996                  186             medium      F                 9   
10997                  231             medium      F                10   
10998                  261             medium      M                10   

       Weight_in_gms  Reached.on.Time_Y.N  
0               1298                    0  
1               4890                    1  
2               4958                    0  
3               2581                    1  
4               5476                    0  
...              ...                  ...  
10994           4800                    0  
10995           5758                    1  
10996           5841                    0  
10997           5361                    0  
10998           1290                    1  
```

```
[10999 rows x 10 columns]
```

Descriptive Statistics

Descriptive Statistics Information of the Numerical columns

```
d.describe()
```

```
       Customer_care_calls  Customer_rating  Cost_of_the_Product  \
count         10999.000000     10999.000000         10999.000000   
mean              4.054459         2.990545           210.196836   
std               1.141490         1.413603            48.063272   
min               2.000000         1.000000            96.000000   
25%               3.000000         2.000000           169.000000   
50%               4.000000         3.000000           214.000000   
75%               5.000000         4.000000           251.000000   
max               7.000000         5.000000           310.000000   

       Discount_offered  Weight_in_gms  Reached.on.Time_Y.N  
count      10999.000000   10999.000000         10999.000000  
mean          13.373216    3634.016729             0.596691  
std           16.205527    1635.377251             0.490584  
min            1.000000    1001.000000             0.000000  
25%            4.000000    1839.500000             0.000000  
50%            7.000000    4149.000000             1.000000  
75%           10.000000    5050.000000             1.000000  
max           65.000000    7846.000000             1.000000  
```

Five Random Samples from the Dataset

```
d.sample(5)
```

```
     Warehouse_block Mode_of_Shipment  Customer_care_calls  Customer_rating  
\
3208               B             Ship                    5                1   
3211               B           Flight                    3                3   
5964               D             Ship                    3                1   
8095               F             Road                    4                3   
7797               F             Road                    3                2   

      Cost_of_the_Product Product_importance Gender  Discount_offered  \
3208                  261             medium      M                 3   
3211                  176             medium      M                10   
5964                  247                low      F                 8   
8095                  268             medium      F                23   
7797                  257                low      F                25   

      Weight_in_gms  Reached.on.Time_Y.N  
3208           1644                    1  
3211           4376                    1  
5964           5376                    1  
```

```
8095           2996                       1
7797           2334                       1
```

Number of Unique values in each column

```
d.nunique()
```

```
Warehouse_block           5
Mode_of_Shipment          3
Customer_care_calls       6
Customer_rating           5
Cost_of_the_Product     215
Product_importance        3
Gender                    2
Discount_offered         65
Weight_in_gms          4034
Reached.on.Time_Y.N       2
dtype: int64
```

Correlation

```
d.corr()
```

```
                     Customer_care_calls  Customer_rating  \
Customer_care_calls             1.000000         0.012209
Customer_rating                 0.012209         1.000000
Cost_of_the_Product             0.323182         0.009270
Discount_offered               -0.130750        -0.003124
Weight_in_gms                  -0.276615        -0.001897
Reached.on.Time_Y.N            -0.067126         0.013119

                     Cost_of_the_Product  Discount_offered  Weight_in_gms  \
Customer_care_calls             0.323182         -0.130750      -0.276615
Customer_rating                 0.009270         -0.003124      -0.001897
Cost_of_the_Product             1.000000         -0.138312      -0.132604
Discount_offered               -0.138312          1.000000      -0.376067
Weight_in_gms                  -0.132604         -0.376067       1.000000
Reached.on.Time_Y.N            -0.073587          0.397108      -0.268793

                     Reached.on.Time_Y.N
Customer_care_calls            -0.067126
Customer_rating                 0.013119
Cost_of_the_Product            -0.073587
Discount_offered                0.397108
Weight_in_gms                  -0.268793
Reached.on.Time_Y.N             1.000000
```

Covariance

```
d.cov()
```

```
                     Customer_care_calls  Customer_rating  \
Customer_care_calls             1.302999         0.019700
Customer_rating                 0.019700         1.998274
```

```
Cost_of_the_Product            17.730960        0.629794
Discount_offered               -2.418672       -0.071575
Weight_in_gms                -516.375888       -4.385094
Reached.on.Time_Y.N            -0.037590        0.009098

                     Cost_of_the_Product  Discount_offered  Weight_in_gms  \
Customer_care_calls            17.730960         -2.418672  -5.163759e+02
Customer_rating                 0.629794         -0.071575  -4.385094e+00
Cost_of_the_Product          2310.078091       -107.729679  -1.042289e+04
Discount_offered             -107.729679        262.619108  -9.966577e+03
Weight_in_gms              -10422.887818      -9966.576620   2.674459e+06
Reached.on.Time_Y.N            -1.735119          3.157082  -2.156496e+02

                     Reached.on.Time_Y.N
Customer_care_calls            -0.037590
Customer_rating                 0.009098
Cost_of_the_Product            -1.735119
Discount_offered                3.157082
Weight_in_gms                -215.649645
Reached.on.Time_Y.N             0.240673
```

Mean

```
d.mean()
```

```
Customer_care_calls       4.054459
Customer_rating           2.990545
Cost_of_the_Product     210.196836
Discount_offered         13.373216
Weight_in_gms          3634.016729
Reached.on.Time_Y.N       0.596691
dtype: float64
```

Data Wrangling

Customer calls received for each Warehouse block

```
d1=df.copy()
d1=d1.pivot_table('Customer_care_calls',columns='Warehouse_block',aggfunc='su
m')
d1
```

```
Warehouse_block         A     B     C     D      F
Customer_care_calls  7402  7369  7451  7434  14939
```

Maximum, minimum price of a product and total products shipped through the various modes of shipment

```
d2=df.copy()
d2=d2.groupby(['Mode_of_Shipment'])['Cost_of_the_Product'].agg(["max","min","
count"])
d2=pd.DataFrame(d2)
d2.rename(columns={'max':'Max Price product','min':'Min Price
product','count':'Total Products shipped'},inplace=True)
d2

                  Max Price product  Min Price product  Total Products 
shipped
Mode_of_Shipment                                                       
Flight                          310                 96                 
1777
Road                            310                 97                 
1760
Ship                            310                 96                 
7462
```

Maximum,minimum price of a product and total products shipped through different Warehouse_blocks

```
d3=df.copy()
d3=d3.groupby(['Warehouse_block'])['Cost_of_the_Product'].agg(["max","min","c
ount"])
d3=pd.DataFrame(d3)
d3.rename(columns={'max':'Max Price product','min':'Min Price 
product','count':'Total Products shipped'},inplace=True)
d3

                 Max Price product  Min Price product  Total Products shipped
Warehouse_block                                                              
A                              310                 96                    1833
B                              310                 96                    1833
C                              310                101                    1833
D                              310                 96                    1834
F                              310                 96                    3666
```

Total products shipped by Warehouse blocks with different shipments

```
grouped=pd.DataFrame(df.groupby(['Mode_of_Shipment','Warehouse_block'])['Cost
_of_the_Product'].count().unstack())
grouped

Warehouse_block      A     B     C     D     F
Mode_of_Shipment                              
Flight             297   296   295   297   592
Road               294   294   294   292   586
Ship              1242  1243  1244  1245  2488
```

Observation: Most of the products has been sent through ship by all the Warehouse blocks

Customer Ratings(1-5) given to Warehouse blocks by Customers

```
pd.crosstab(d['Customer_rating'],d['Warehouse_block'])

Warehouse_block    A    B    C    D    F
Customer_rating                         
```

```
1                 394  371  364  364  742
2                 376  376  362  340  711
3                 345  371  383  390  750
4                 350  348  369  379  743
5                 368  367  355  361  720
```

Products has been delivered by Warehouse Blocks on time or not

```
# a=np.array(["Products not Reached on Time denoted by 0","Products Reached
on Time denoted by 1"],dtype=object)
a=pd.crosstab(d['Reached.on.Time_Y.N'],d['Warehouse_block'])
a.index=["Products not Reached on time ","Products Reached on time"]
a
```

```
Warehouse_block                   A     B     C     D     F
Products not Reached on time    758   729   739   738  1472
Products Reached on time       1075  1104  1094  1096  2194
```

Heaviest weight product shipped among all the blocks

```
a1=pd.DataFrame(d[d.Weight_in_gms==d.Weight_in_gms.max()])
a2=pd.DataFrame(a1[['Warehouse_block','Mode_of_Shipment','Weight_in_gms']])
a2.index=['Heaviest Weight Shipped']
a2
```

```
                        Warehouse_block Mode_of_Shipment  Weight_in_gms
Heaviest Weight Shipped               F             Ship           7846
```

Highest discount offered among all the blocks

```
a1=pd.DataFrame(d[d.Discount_offered==d.Discount_offered.max()])
#a1=a1.reset_index(drop=True)
list1=a1['Warehouse_block'].unique()
print('Warehouse blocks that offered the highest discount of
',d.Discount_offered.max())
for i in list1:
    print(i)
```

```
Warehouse blocks that offered the highest discount of  65
A
B
C
D
F
```

Lowest discount offered among all the blocks

```
a1=pd.DataFrame(d[d.Discount_offered==d.Discount_offered.min()])
#a1=a1.reset_index(drop=True)
list1=a1['Warehouse_block'].unique()
print('Warehouse blocks that offered the lowest discount of
',d.Discount_offered.min())
for i in list1:
    print(i)
```

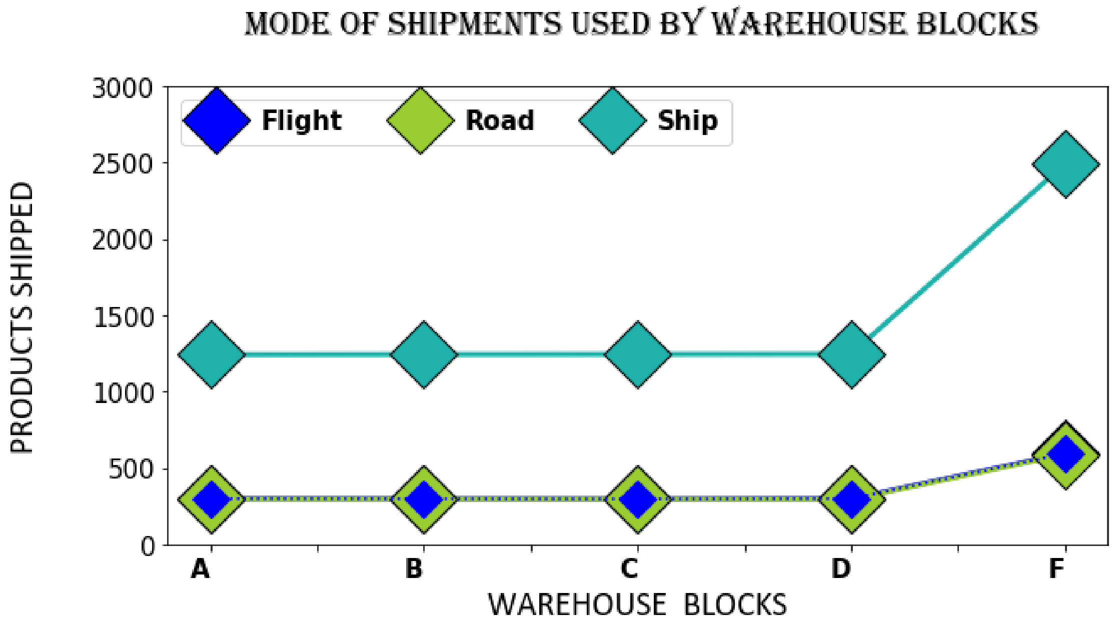
```
Warehouse blocks that offered the lowest discount of  1
A
B
C
D
F
```

DATA VISUALIZATION (Matplotlib, seaborn)

Matplotlib- Lineplot

WAREHOUSE BLOCKS AND MODE OF SHIPMENTS

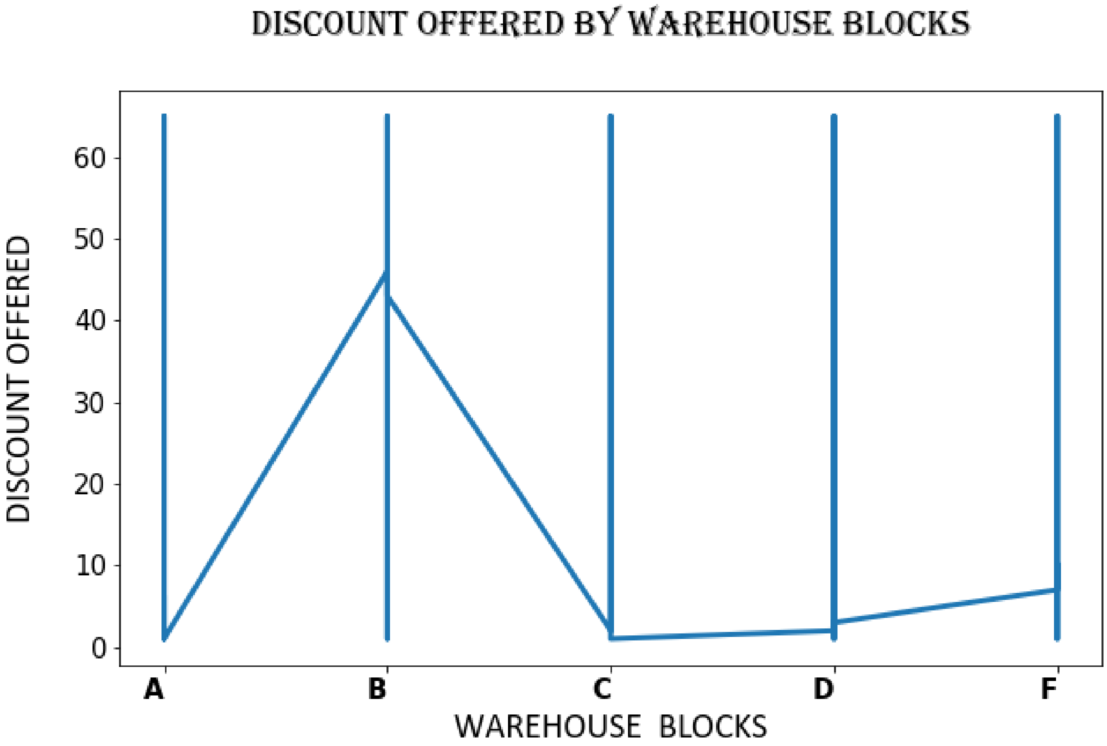
```
grouped=pd.DataFrame(df.groupby(['Warehouse_block','Mode_of_Shipment'])['Cost
_of_the_Product'].count().unstack())
lineplot=grouped.plot(figsize=(10,5),  marker='D',ms=28,mec='k',linestyle='-
',linewidth=3,
                      color=['blue','yellowgreen','lightseagreen'])
grouped['Flight'].plot( marker='D',ms=15,linestyle=':',color='blue')
font1={'family':'Algerian','color':'black','size':20,'fontweight':'bold'}
font2={'family':'Calibri','color':'black','size':20}
plt.title(" MODE OF SHIPMENTS USED BY WAREHOUSE
BLOCKS",fontdict=font1,pad=30)
plt.xlabel("WAREHOUSE  BLOCKS",fontdict=font2,labelpad=5)
plt.ylim(0,3000)
plt.xticks(color='black', fontsize='15',fontweight='bold',
horizontalalignment='right')
plt.yticks(color='black', fontsize='15', horizontalalignment='right')
plt.ylabel("PRODUCTS SHIPPED",fontdict=font2,labelpad=30)
plt.legend(['Flight','Road','Ship'],prop = {'size' :
15,'weight':'bold'},ncol=5,loc='upper left')
plt.show()
```



Observation: The most preferred shipping mode of all the Warehouse blocks is Ship

DISCOUNT OFFERED BY WAREHOUSE BLOCKS

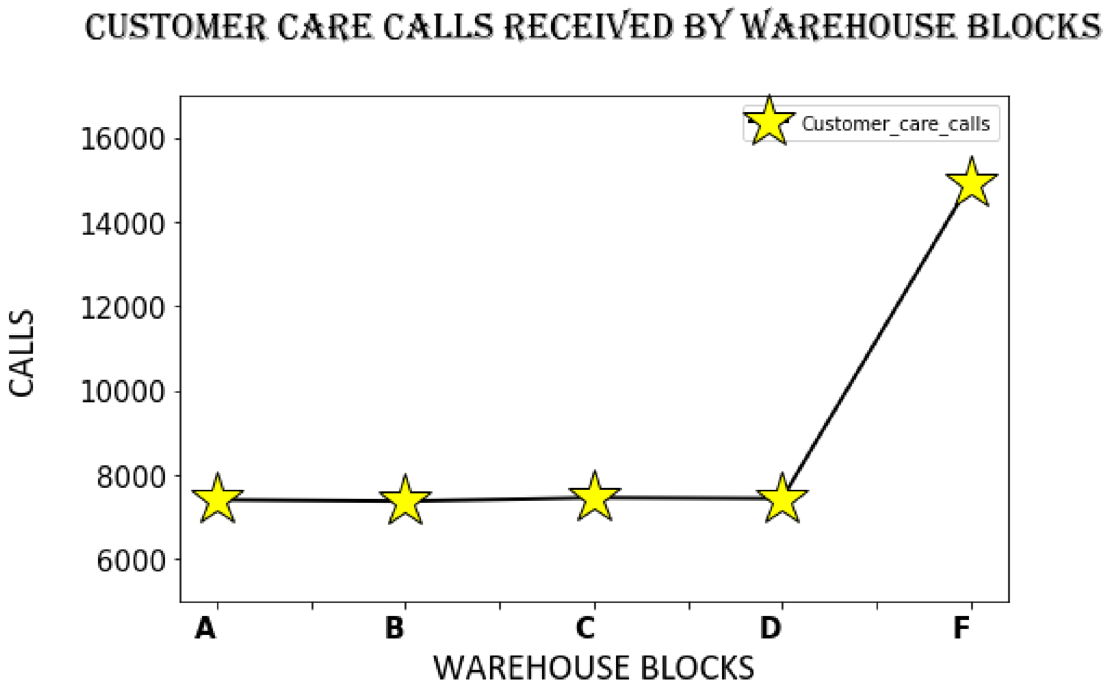
```
plt.figure(figsize=(10,6))
font1={'family':'Algerian','color':'black','size':20,'fontweight':'bold'}
font2={'family':'Calibri','color':'black','size':20}

plt.title("DISCOUNT OFFERED BY WAREHOUSE BLOCKS",fontdict=font1,pad=30)
plt.xlabel("WAREHOUSE  BLOCKS",fontdict=font2,labelpad=5)
plt.xticks(color='black', fontsize='15',fontweight='bold',
horizontalalignment='right')
plt.yticks(color='black', fontsize='15', horizontalalignment='right')
plt.ylabel("DISCOUNT OFFERED",fontdict=font2,labelpad=20)
plt.plot(d['Warehouse_block'],d['Discount_offered'],linewidth=3)
plt.show()
```



Observation: Warehouse blocks A, B, C, D, F gave the highest discounts.

Customer Care calls received by warehouse blocks

```
d1=df.copy()
grouped=pd.DataFrame(d1.groupby(['Warehouse_block'])['Customer_care_calls'].s
um())
lineplot=grouped.plot(figsize=(8,5), marker='*',ms=30,mec='k',linestyle='-
',linewidth=2,mfc='yellow',color='black')
font1={'family':'Algerian','color':'black','size':20,'fontweight':'bold'}
font2={'family':'Calibri','color':'black','size':20}
plt.title("CUSTOMER CARE CALLS RECEIVED BY WAREHOUSE
BLOCKS",fontdict=font1,pad=30)
plt.xlabel("WAREHOUSE BLOCKS",fontdict=font2,labelpad=5)
plt.ylabel("CALLS",fontdict=font2,labelpad=20)
plt.xticks(color='black', fontsize='15',fontweight='bold',
horizontalalignment='right')
plt.yticks(color='black', fontsize='15', horizontalalignment='right')
plt.ylim(5000,17000)
plt.show()
```

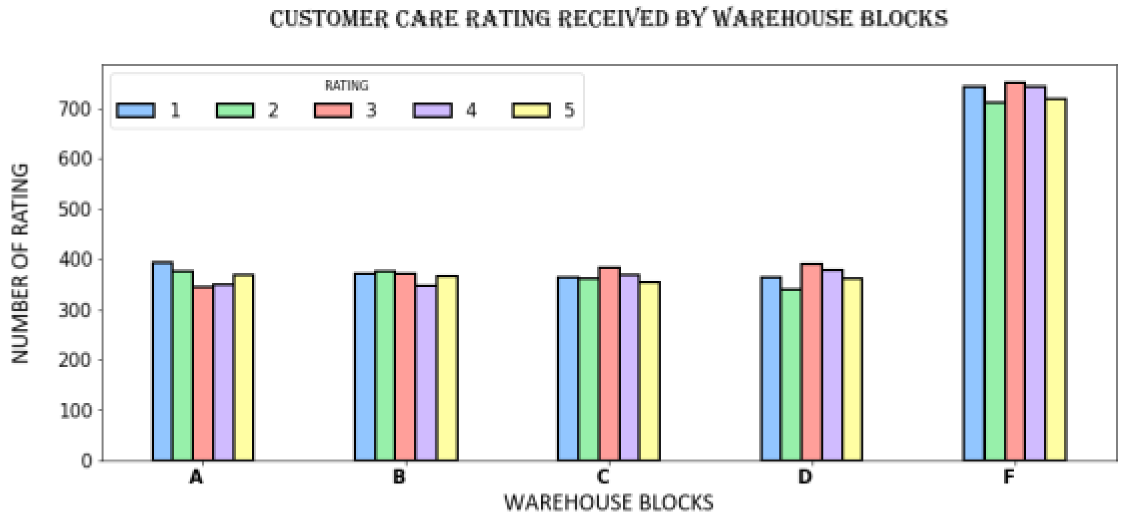


Observation: Warehouse Block 'F' has received the highest Customer Care calls

Bar Plots

Customer ratings 1-5 given to all the warehouse blocks

```
plt.style.use('seaborn-pastel')
ctab=pd.crosstab(d['Warehouse_block'],d['Customer_rating'])
#c=['tomato','skyblue','blue','seagreen','gold',slateblue]
barplot=ctab.plot.bar(figsize=(15,6),edgecolor='black',linewidth=2,width=0.5)
font1={'family':'Algerian','color':'black','size':20,'fontweight':'bold'}
font2={'family':'Calibri','color':'black','size':20}
plt.title("CUSTOMER CARE RATING RECEIVED BY WAREHOUSE
BLOCKS",fontdict=font1,pad=30)
plt.xlabel("WAREHOUSE BLOCKS",fontdict=font2,labelpad=5)
plt.ylabel("NUMBER OF RATING",fontdict=font2,labelpad=20)
plt.xticks(color='black', rotation=360, fontsize='15',fontweight='bold',
horizontalalignment='right')
plt.yticks(color='black', fontsize='15', horizontalalignment='right')
plt.legend(fontsize=15,title='RATING',ncol=5)
plt.show()
```

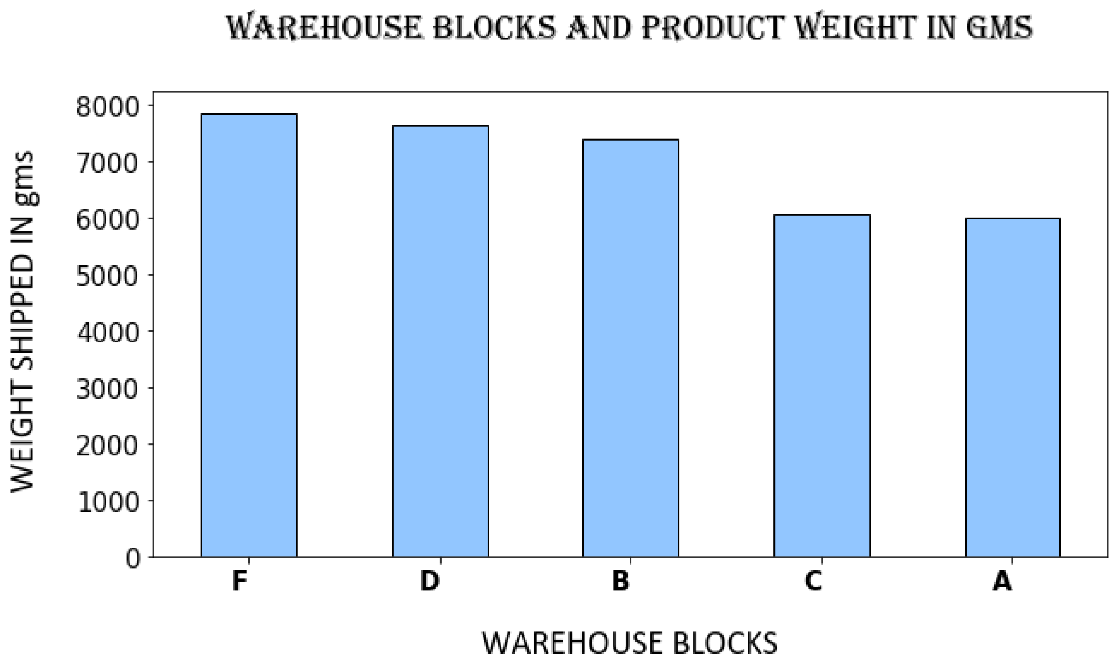


Observations:

1. Warehouse Block 'A' has received the highest rating of '1'
2. Warehouse Block 'B' has received the highest rating of '2'
3. Warehouse Block 'C' has received the highest rating of '3'
4. Warehouse Block 'D' has received the highest rating of '3'
5. Among all the blocks, 'F' has received the highest rating of '1','2','3','4','5'

Warehouse blocks and different weights of products shipped

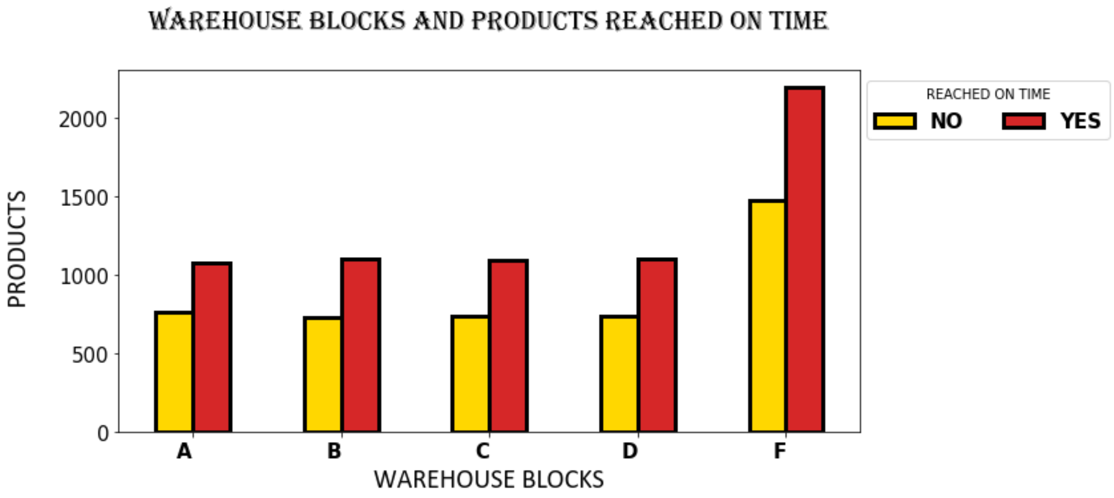
```
plt.figure(figsize=(10,5))

warehouse=d.groupby(['Warehouse_block'])
#warehouse.max().sort_values(by="Weight_in_gms",ascending=False)["Weight_in_g
ms"].plot.bar(edgecolor='k',linewidth=2)
weight=warehouse.max().sort_values(by="Weight_in_gms",ascending=False)
weight["Weight_in_gms"].plot.bar(edgecolor='k')
font1={'family':'Algerian','color':'black','size':20,'fontweight':'bold'}
font2={'family':'Calibri','color':'black','size':20}
plt.title("WAREHOUSE BLOCKS AND PRODUCT WEIGHT IN GMS",fontdict=font1,pad=30)
plt.xlabel("WAREHOUSE BLOCKS",fontdict=font2,labelpad=20)
plt.ylabel(" WEIGHT SHIPPED IN gms",fontdict=font2,labelpad=20)
plt.xticks(color='black', rotation=360, fontsize='15',fontweight='bold',
horizontalalignment='right')
plt.yticks(color='black', fontsize='15', horizontalalignment='right')

plt.show()
```



Observation: Warehouse Block A has shipped a product that has the heaviest weight among all the products.

Products reached on time is denoted by 1 and products not reached on time is denoted by 0

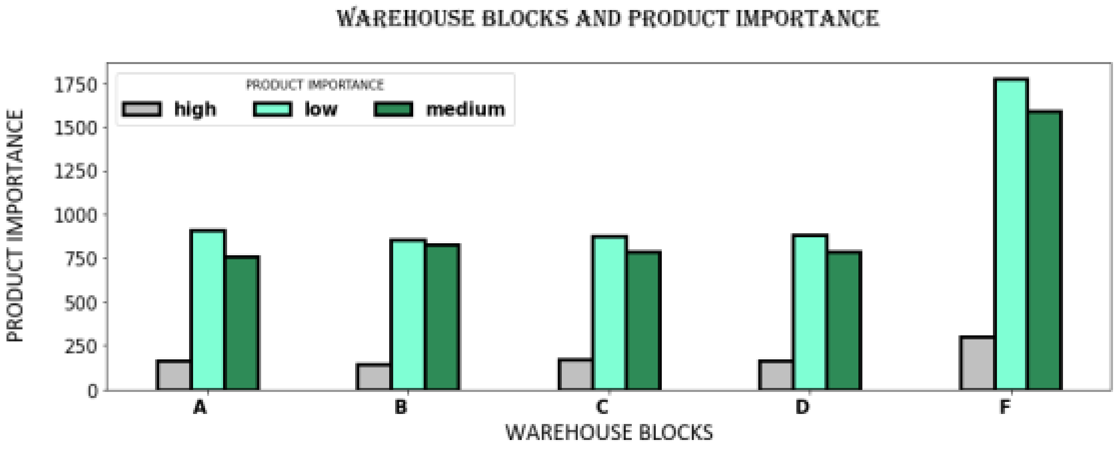
```
ctab=pd.crosstab(d['Warehouse_block'],d['Reached.on.Time_Y.N'])
barplot=ctab.plot.bar(figsize =
(10,5),edgecolor='black',color=['gold','tab:red'],linewidth=3)
plt.title("WAREHOUSE BLOCKS AND PRODUCTS REACHED ON
TIME",fontdict=font1,pad=30)
plt.xlabel("WAREHOUSE BLOCKS",fontdict=font2,labelpad=5)
plt.ylabel(" PRODUCTS",fontdict=font2,labelpad=20)
plt.xticks(color='black', rotation=360, fontsize='15',fontweight='bold',
horizontalalignment='right')
plt.yticks(color='black', fontsize='15', horizontalalignment='right')
legend_drawn_flag = True
plt.legend(['NO','YES'],title='REACHED ON TIME',bbox_to_anchor =(1.35,
1),loc='upper right',ncol=2,
           prop = {'size' : 15,'weight':'bold'},frameon=legend_drawn_flag)
plt.show()
```



Observation: Warehouse blocks A,B,C,D,F has delivered most of the products on time.

Low,Medium,High importance products delivered by all the warehouse blocks

```
ctab=pd.crosstab(d['Warehouse_block'],d['Product_importance'])
barplot=ctab.plot.bar(figsize = (15,5),edgecolor='black',color=['silver',
'aquamarine', 'seagreen'],linewidth=3)
plt.title("WAREHOUSE BLOCKS AND PRODUCT IMPORTANCE",fontdict=font1,pad=30)
plt.xlabel("WAREHOUSE BLOCKS",fontdict=font2,labelpad=5)
plt.ylabel(" PRODUCT IMPORTANCE",fontdict=font2,labelpad=20)
plt.xticks(color='black', rotation=360, fontsize='15',fontweight='bold',
horizontalalignment='right')
plt.yticks(color='black', fontsize='15', horizontalalignment='right')
plt.legend(fontsize=15,ncol=3,title='PRODUCT IMPORTANCE',prop = {'size' :
15,'weight':'bold'})
plt.show()
```

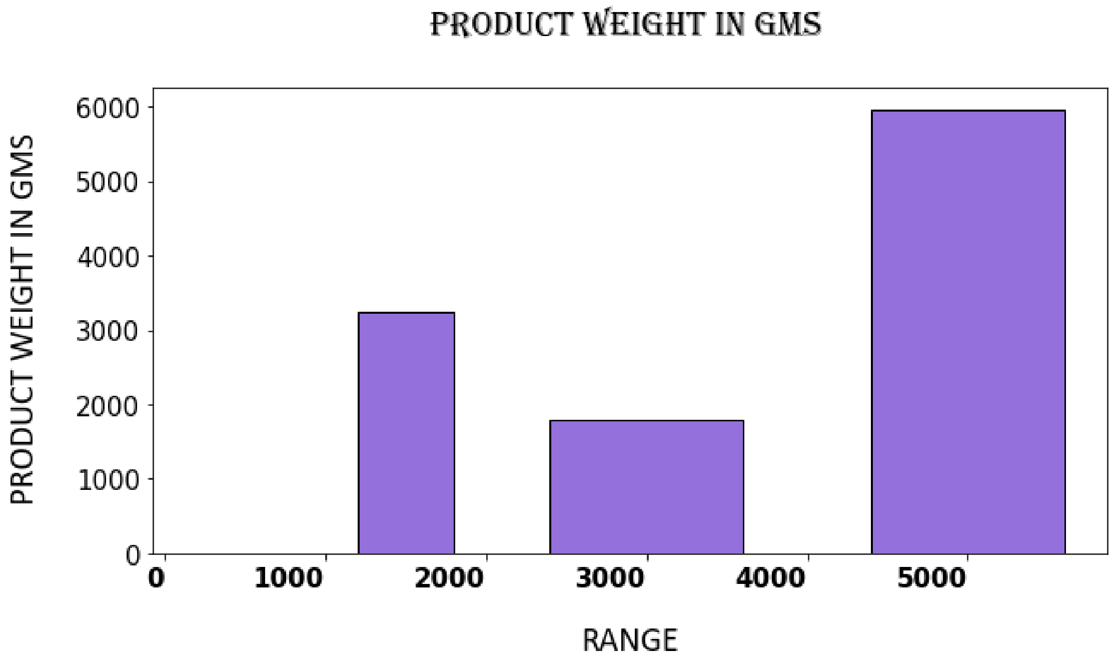


Observation: Warehouse blocks A,B,C,D,F has shipped most of the products of low importance.

Histogram

Product weight in gms shipped by all the Warehouse blocks

```
plt.figure(figsize=(10,5))
bins=[0,1000,2000,4000,6000]
plt.hist(d['Weight_in_gms'],bins,histtype='bar',rwidth=0.6,color='mediumpurple',edgecolor='black')#rectangular width
plt.title("PRODUCT WEIGHT IN GMS ",fontdict=font1,pad=30)
plt.xlabel("RANGE",fontdict=font2,labelpad=20)
plt.ylabel(" PRODUCT WEIGHT IN GMS",fontdict=font2,labelpad=20)
plt.xticks(color='black', rotation=360, fontsize='15',fontweight='bold', horizontalalignment='right')
plt.yticks(color='black', fontsize='15', horizontalalignment='right')
plt.show()
```



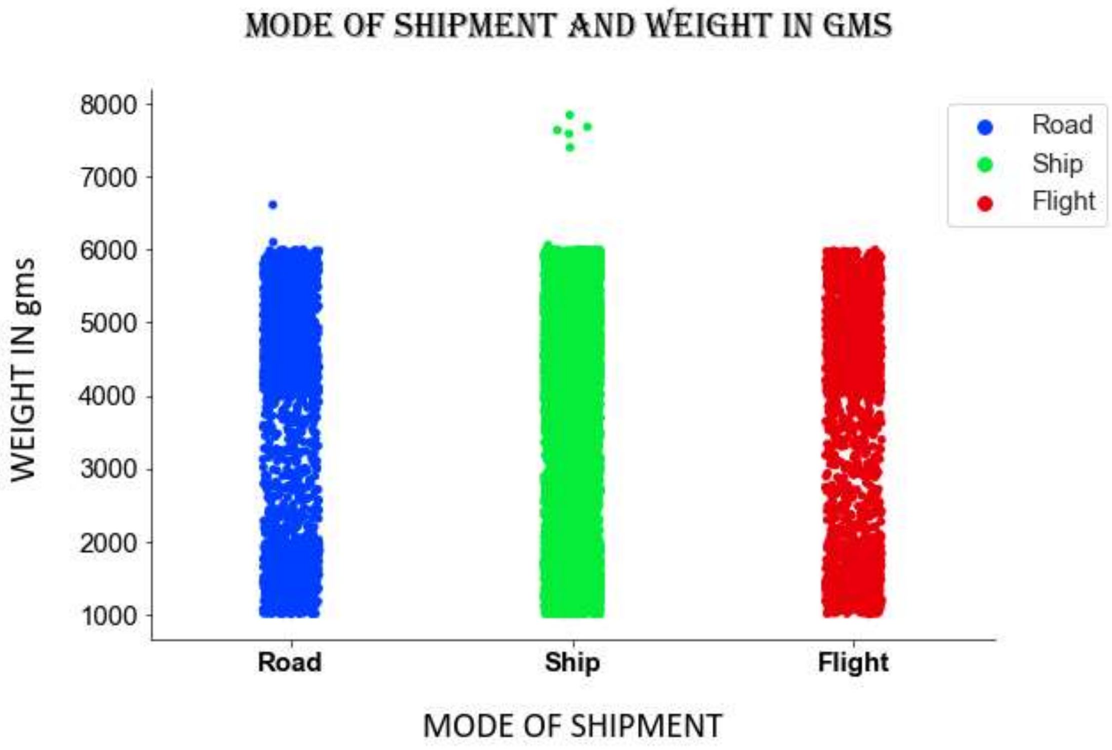
Observation: Heavy products were shipped by most of the blocks

Seaborn

Catplot

Mode of Shipment and Weight in gms

```
import seaborn as sns
plt.style.use('seaborn-bright')
sns.set_style("ticks")
plt.style.use('seaborn-bright')
sns.catplot(y="Weight_in_gms",x='Mode_of_Shipment',hue='Mode_of_Shipment',data=d,height=5,aspect=15/10)
plt.title("MODE OF SHIPMENT AND WEIGHT IN gms ",fontdict=font1,pad=30)
plt.xlabel("MODE OF SHIPMENT",fontdict=font2,labelpad=20)
plt.ylabel("WEIGHT IN gms ",fontdict=font2,labelpad=20)
plt.xticks(color='black',fontsize='15',fontweight='bold')
plt.yticks(color='black', fontsize='15')
plt.legend(fontsize=15,bbox_to_anchor =(1.15, 1),loc='upper right')
plt.show()
```



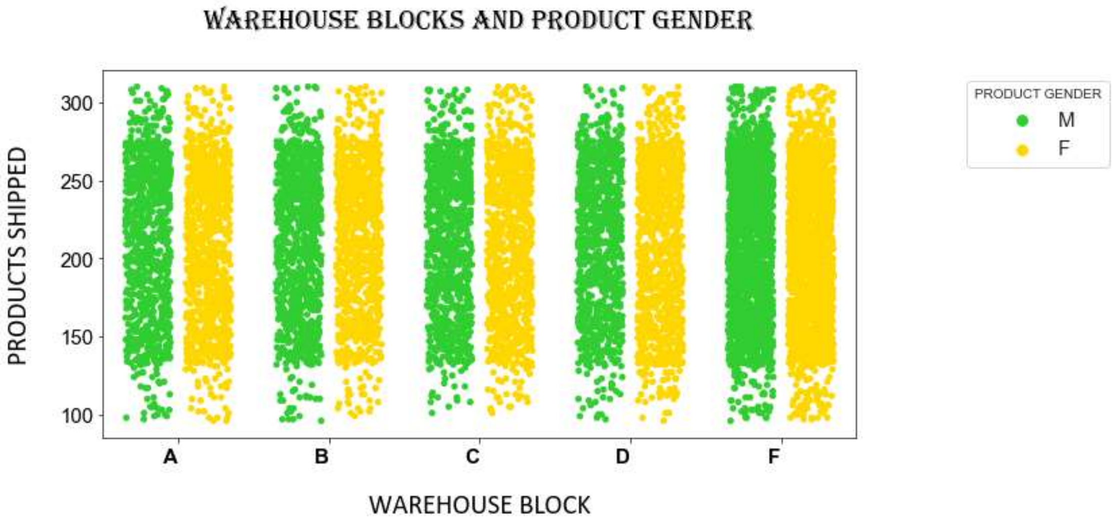
Observations: From the graph we can see that the density of ship mode is more.

1. Most of the Warehouse blocks have used Ship mode to deliver the products.
2. Few of the heavy products weighing more than 7000gms was shipped through Ship mode.

Striplot

WAREHOUSE BLOCKS AND PRODUCT GENDER

```
custom_palette = ["limegreen","gold"]
sns.set_palette(custom_palette)
plt.figure(figsize=(10,5))
sns.stripplot(x='Warehouse_block',y='Cost_of_the_Product',hue="Gender",jitter
=0.3,dodge=True, data=d)
plt.title("WAREHOUSE BLOCKS AND PRODUCT GENDER",fontdict=font1,pad=30)
plt.xlabel("WAREHOUSE BLOCK",fontdict=font2,labelpad=20)
plt.ylabel("PRODUCTS SHIPPED",fontdict=font2,labelpad=20)
plt.xticks(color='black',fontsize='15',fontweight='bold',
horizontalalignment='right')
plt.yticks(color='black', fontsize='15', horizontalalignment='right')
plt.legend(fontsize=15,bbox_to_anchor =(1.35, 1),loc='upper
right',frameon=True,title='PRODUCT GENDER')
plt.show()
```

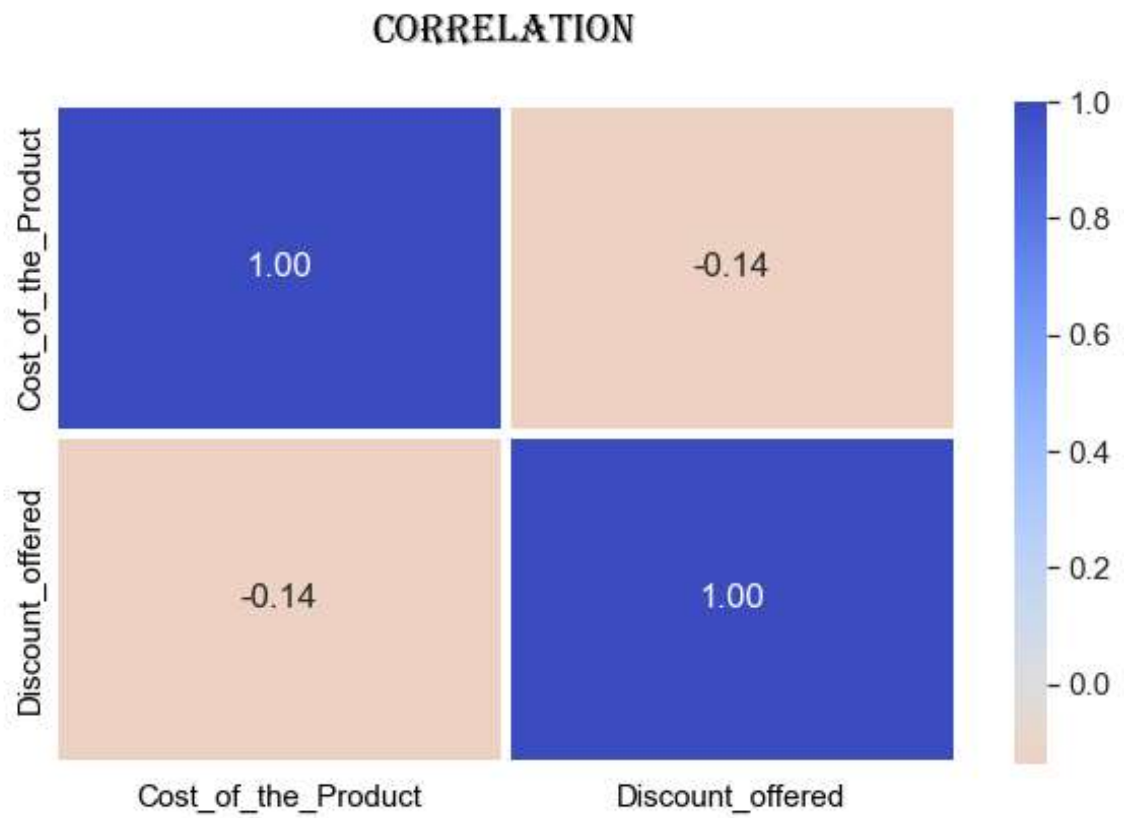


Observation: All the Warehouse blocks has shipped approximately equal products of Female and Male Gender.

Heatmap

Correlation between the cost of the product and discount

```
plt.figure(figsize=(10,6))
sns.set(font_scale=1.4)
n_data=['Cost_of_the_Product','Discount_offered']
sns.heatmap(d[n_data].corr(),annot=True,fmt='.2f',cmap='coolwarm_r',center=0,
linewidths=5)
plt.title("Correlation",fontdict=font1,pad=30)
plt.xticks(color='black', horizontalalignment='center', fontsize=15)
plt.yticks(color='black', horizontalalignment='center',fontsize=15)
plt.show()
```



Observation: Cost of the product and discount offered is negatively correlated.

Report

```
from pandas_profiling import ProfileReport

k=ProfileReport(d)
k
```

{"model_id":"c2a4f5fd4f184f959d1299e2c012b511","version_major":2,"version_minor":0}

```
<IPython.core.display.HTML object>
```

```
k.to_file('Report.html')
```

SUMMARY

The Analysis of the E-Commerce Shipping Statistics

1. The Warehouse blocks prefer Ship mode as the shipping method for most of the products.
2. Customer care calls was received more by the Warehouse Block F.
3. High customer rating as well as low customer rating was given to Warehouse block F.

Improvement 1: Change in Mode of Shipment Required

We can see most of the Warehouse blocks did not deliver the product on time because the mode of shipping was through Ship.

Although ships are capable of carrying much bigger loads than other transportation methods, shipping takes much longer.

It is not usually the preferred shipping method for businesses that rely on speedy delivery. The shipping mode should be changed to other modes of transport like Road or Flight, so that the products reaches the customer on time. According to a recent study, 98% of consumers are likely to order again from a website if the delivery experience went well.

Improvement 2: Customer Care Required

We can observe that Warehouse Block F has received more calls from the customers. It has also received a high customer rating of 1 out of 5. This clearly shows that the customers are not happy with the products delivered. Feedback has to be taken from the customer and make necessary changes in order to improve the overall customer experience. More Customer service executives has to be assigned to the Warehouse block F.

Conclusion:

Logistics is a major pillar of the e-commerce customer experience. Optimizing logistics is an important factor for the success of any brand. In order to retain the customers and attract new ones, it is important to optimize the processes at different levels: delivery, packaging, returns, customer service.

REFERENCES

https://www.kaggle.com/
https://matplotlib.org/
https://seaborn.pydata.org/