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

In [1]:
```
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

In [2]:
```
df=pd.read_excel('Project-Ecommerce  Shipping  Data  Analysis  (Python).xlsx')
d=df.copy()
```

In [3]:
```
#Ecommerce Shipping Dataset
d
```

Out[3]:

	ID	Warehouse_block	Mode_of_Shipment	Customer_care_calls	Customer_rating	Cost_of_the_Product	Prior_purchases	Produc
0	1	D	Flight	4	2	177	3	
1	2	F	Flight	4	5	216	2	
2	3	A	Flight	2	2	183	4	
3	4	B	Flight	3	3	176	4	
4	5	C	Flight	2	2	184	3	
...	...	...	...	...	...	...	...	...
10994	10995	A	Ship	4	1	252	5	
10995	10996	B	Ship	4	1	232	5	
10996	10997	C	Ship	5	4	242	5	
10997	10998	F	Ship	5	2	223	6	
10998	10999	D	Ship	2	5	155	5	

10999 rows × 12 columns

In [4]:
```
d.shape
```

Out[4]: (10999, 12)

In [5]:
```
d.columns
```

Out[5]:
```
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')
```

In [6]:
```
d.head()
```

Out[6]:

	ID	Warehouse_block	Mode_of_Shipment	Customer_care_calls	Customer_rating	Cost_of_the_Product	Prior_purchases	Product_import
0	1	D	Flight	4	2	177	3	
1	2	F	Flight	4	5	216	2	
2	3	A	Flight	2	2	183	4	
3	4	B	Flight	3	3	176	4	me
4	5	C	Flight	2	2	184	3	me

In [7]:
```
d.tail()
```

Out[7]:

	ID	Warehouse_block	Mode_of_Shipment	Customer_care_calls	Customer_rating	Cost_of_the_Product	Prior_purchases	Produc
10994	10995	A	Ship	4	1	252	5	
10995	10996	B	Ship	4	1	232	5	
10996	10997	C	Ship	5	4	242	5	
10997	10998	F	Ship	5	2	223	6	
10998	10999	D	Ship	2	5	155	5	

In [8]:
```
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
```

In [9]:
```
#Warehouse  Blocks
np.sort(d['Warehouse_block'].unique())
```

Out[9]:
```
array(['A', 'B', 'C', 'D', 'F'],  dtype=object)
```

In [10]:
```
#Mode of Shipment
np.sort(d['Mode_of_Shipment'].unique())
```

Out[10]:
```
array(['Flight', 'Road', 'Ship'], dtype=object)
```

Data Cleansing

In [11]:
```
df.isnull().sum().sum()
```

Out[11]:
```
0
```

In [12]:
```
d.isnull().sum()
```

Out[12]:
```
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

In [13]:
```
d.drop(['ID','Prior_purchases'],axis=1,inplace=True)
d
```

Out[13]:

	Warehouse_block	Mode_of_Shipment	Customer_care_calls	Customer_rating	Cost_of_the_Product	Product_importance	Gender	D
0	D	Flight	4	2	177	low	F	
1	F	Flight	4	5	216	low	M	
2	A	Flight	2	2	183	low	M	
3	B	Flight	3	3	176	medium	M	
4	C	Flight	2	2	184	medium	F	
...	...	...	...	...	...	...	...	
10994	A	Ship	4	1	252	medium	F	
10995	B	Ship	4	1	232	medium	F	
10996	C	Ship	5	4	242	low	F	
10997	F	Ship	5	2	223	medium	M	
10998	D	Ship	2	5	155	low	F	

10999 rows × 10 columns

In [14]:
```
#Data sorted by Warehouse block column
d['Warehouse_block']=np.sort(d['Warehouse_block'])
d
```

Out[14]:

	Warehouse_block	Mode_of_Shipment	Customer_care_calls	Customer_rating	Cost_of_the_Product	Product_importance	Gender	D
0	A	Flight	4	2	177	low	F	
1	A	Flight	4	5	216	low	M	
2	A	Flight	2	2	183	low	M	
3	A	Flight	3	3	176	medium	M	
4	A	Flight	2	2	184	medium	F	
...	...	...	...	...	...	...	...	...
10994	F	Ship	4	1	252	medium	F	
10995	F	Ship	4	1	232	medium	F	
10996	F	Ship	5	4	242	low	F	
10997	F	Ship	5	2	223	medium	M	
10998	F	Ship	2	5	155	low	F	

10999 rows × 10 columns

Descriptive Statistics

In [15]:
```
d.describe()
```

Out[15]:

	Customer_care_calls	Customer_rating	Cost_of_the_Product	Discount_offered	Weight_in_gms	Reached.on.Time_Y.N
count	10999.000000	10999.000000	10999.000000	10999.000000	10999.000000	10999.000000
mean	4.054459	2.990545	210.196836	13.373216	3634.016729	0.596691
std	1.141490	1.413603	48.063272	16.205527	1635.377251	0.490584
min	2.000000	1.000000	96.000000	1.000000	1001.000000	0.000000
25%	3.000000	2.000000	169.000000	4.000000	1839.500000	0.000000
50%	4.000000	3.000000	214.000000	7.000000	4149.000000	1.000000
75%	5.000000	4.000000	251.000000	10.000000	5050.000000	1.000000
max	7.000000	5.000000	310.000000	65.000000	7846.000000	1.000000

In [16]:
```
d.sample(5)
```

Out[16]:

	Warehouse_block	Mode_of_Shipment	Customer_care_calls	Customer_rating	Cost_of_the_Product	Product_importance	Gender Dis
8380	F	Ship	5	2	246	low	F
2948	B	Ship	5	2	198	low	F
9183	F	Flight	5	1	247	low	M
3438	B	Flight	3	1	179	high	M
8047	F	Ship	6	5	301	low	M

In [17]:
```
d.nunique()
```

Out[17]:

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

In [18]:
```
#correlation
d.corr()
```

Out[18]:

	Customer_care_calls	Customer_rating	Cost_of_the_Product	Discount_offered	Weight_in_gms	Reached.on.Time_Y.N
Customer_care_calls	1.000000	0.012209	0.323182	-0.130750	-0.276615	-0.067126
Customer_rating	0.012209	1.000000	0.009270	-0.003124	-0.001897	0.013119
Cost_of_the_Product	0.323182	0.009270	1.000000	-0.138312	-0.132604	-0.073587
Discount_offered	-0.130750	-0.003124	-0.138312	1.000000	-0.376067	0.397108
Weight_in_gms	-0.276615	-0.001897	-0.132604	-0.376067	1.000000	-0.268793
Reached.on.Time_Y.N	-0.067126	0.013119	-0.073587	0.397108	-0.268793	1.000000

In [19]:
```
#covariance
d.cov()
```

Out[19]:

	Customer_care_calls	Customer_rating	Cost_of_the_Product	Discount_offered	Weight_in_gms	Reached.on.Time_Y.N
Customer_care_calls	1.302999	0.019700	17.730960	-2.418672	-5.163759e+02	-0.037590
Customer_rating	0.019700	1.998274	0.629794	-0.071575	-4.385094e+00	0.009098
Cost_of_the_Product	17.730960	0.629794	2310.078091	-107.729679	-1.042289e+04	-1.735119
Discount_offered	-2.418672	-0.071575	-107.729679	262.619108	-9.966577e+03	3.157082
Weight_in_gms	-516.375888	-4.385094	-10422.887818	-9966.576620	2.674459e+06	-215.649645
Reached.on.Time_Y.N	-0.037590	0.009098	-1.735119	3.157082	-2.156496e+02	0.240673

In [20]:
```
#mean
d.mean()
```

Out[20]:

```
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

In [21]:
```
#Customer calls received for each Warehouse block
d1=df.copy()
d1=d1.pivot_table('Customer_care_calls',columns='Warehouse_block',aggfunc='sum')
d1
```

Out[21]:

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

In [22]:
```
#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
d2
```

Out[22]:

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

In [23]:
```
#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","count"])
d3=pd.DataFrame(d3)
d3.rename(columns={'max':'Max Price product','min':'Min Price product','count':'Total Products shipped'},inplace
d3
```

Out[23]:

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

In [24]:
```
#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
#Conclusion: Most of the products has been sent through ship by all the Warehouse blocks
```

Out[24]:

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

In [25]:
```
#Customer Ratings(1-5) given to Warehouse blocks by Customers
pd.crosstab(d['Customer_rating'],d['Warehouse_block'])
```

Out[25]:

Warehouse_block	A	B	C	D	F
Customer_rating					
1	369	362	356	367	781
2	375	374	342	347	727
3	383	360	394	372	730
4	347	375	381	369	717
5	359	362	360	379	711

In [26]:
```
#Whether 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
```

Out[26]:

Warehouse_block	A	B	C	D	F
Products not Reached on time	0	293	1039	1024	2080
Products Reached on time	1833	1540	794	810	1586

In [27]:
```
#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
```

Out[27]:

	Warehouse_block	Mode_of_Shipment	Weight_in_gms
Heaviest Weight Shipped	A	Ship	7846

In [28]:
```
#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
```

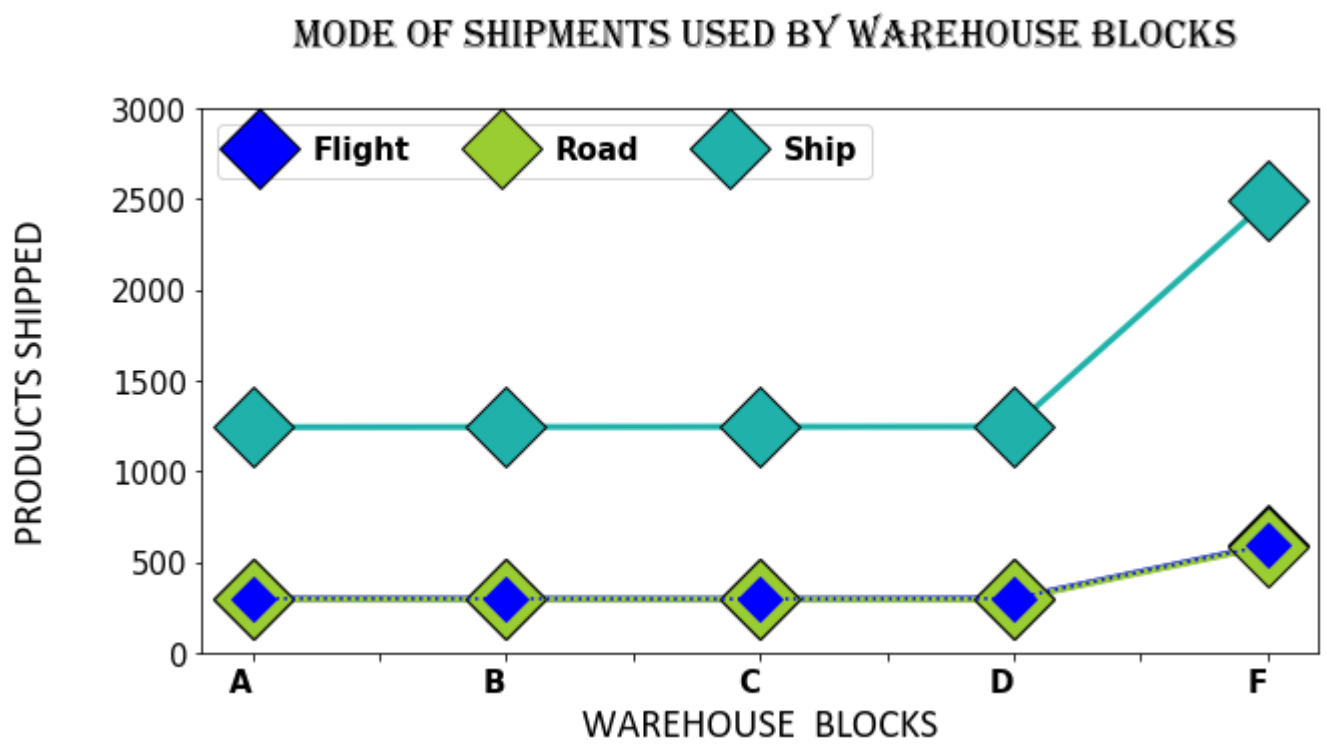
In [29]:
```
#Lowest discount offered among all the blocks
a1=pd.DataFrame(d[d.Discount_offered==d.Discount_offered.min()])
#a1=a1.reset_index(drop=True)
list1=a1['Warehouse_block'].unique()
print('Warehouse blocks that offered the lowest discount of ',d.Discount_offered.min())
for i in list1:
    print(i)
```

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

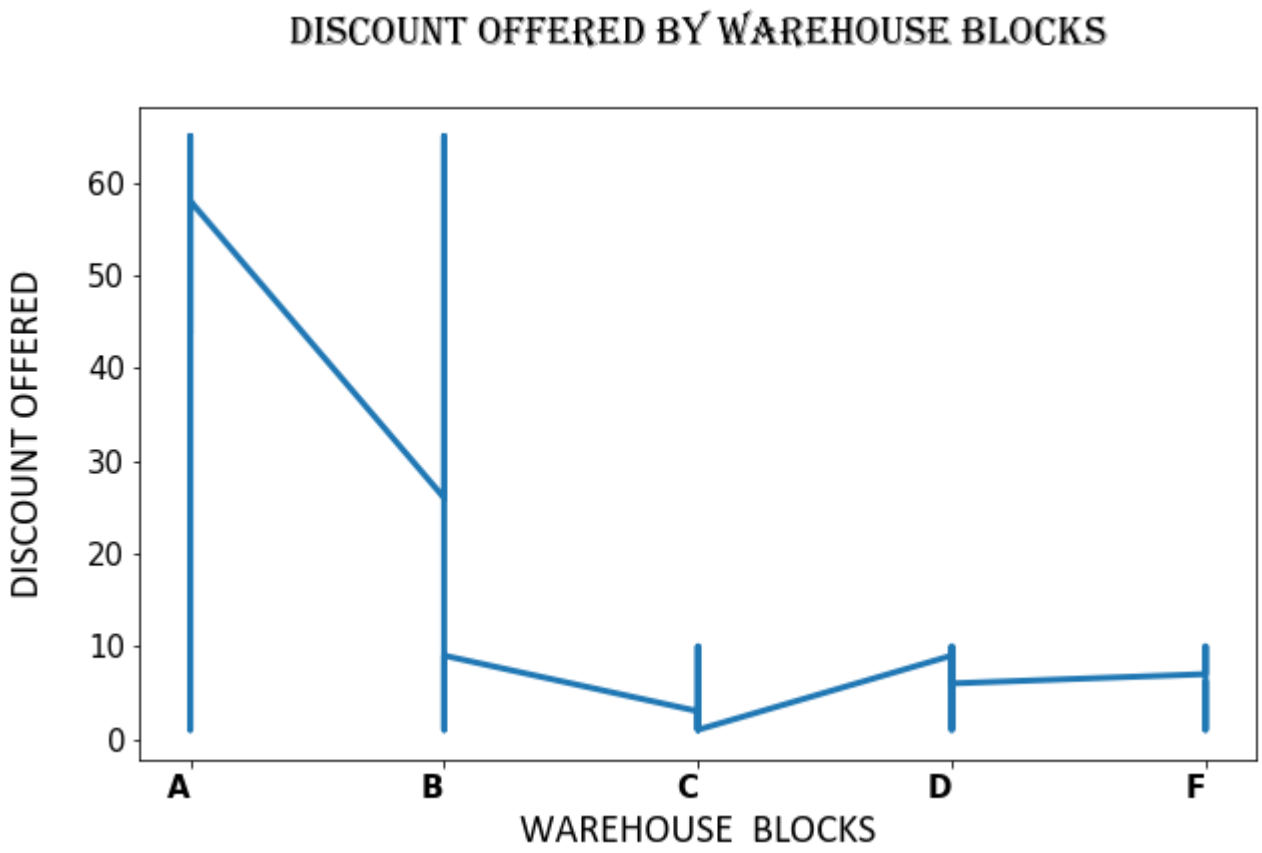
DATA VISUALIZATION (Matplotlib, seaborn)

Matplotlib- Lineplot

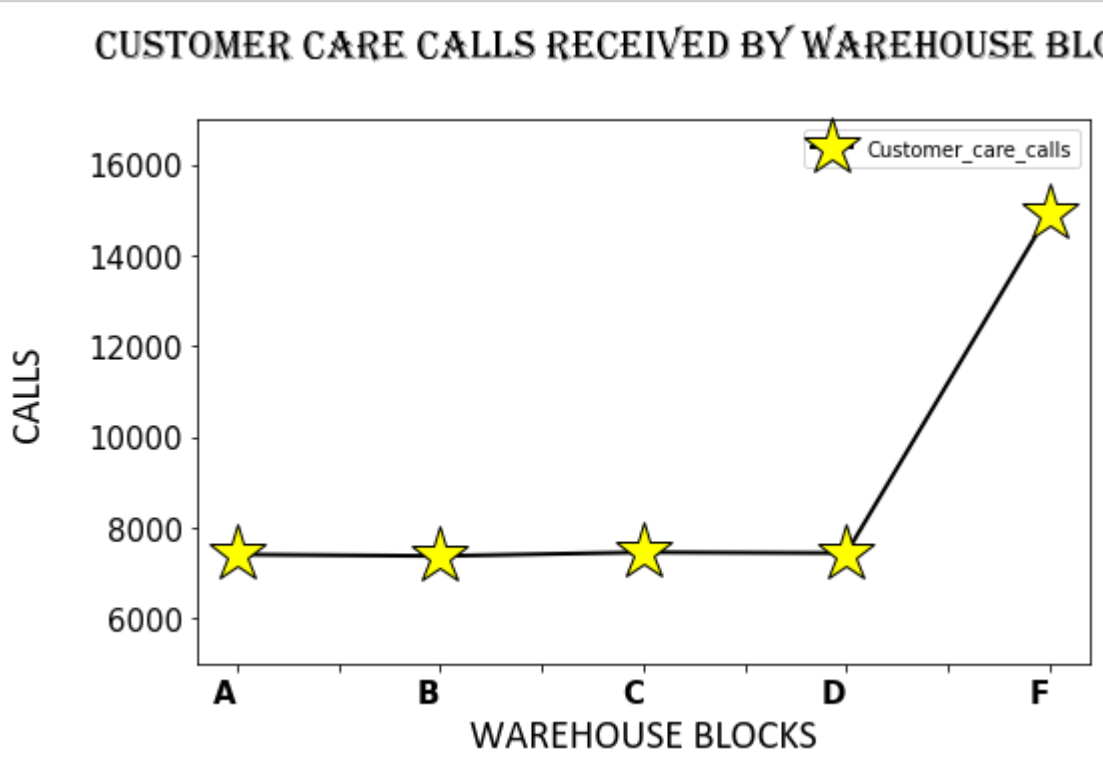
In [30]:
```
#WAREHOUSE BLOCKS AND MODE OF SHIPMENTS
grouped=pd.DataFrame(df.groupby(['Warehouse_block','Mode_of_Shipment'])['Cost_of_the_Product'].count().unstack()
lineplot=grouped.plot(figsize=(10,5),   marker='D',ms=28,mec='k',linestyle='-',linewidth=3,
                      color=['blue','yellowgreen','lightseagreen'])
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()
#Conclusion: The most preferred shipping mode of all the Warehouse blocks is Ship
```



In [31]:
```
#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()

#Conclusion:Warehouse blocks A and B gave the highest discounts.
```

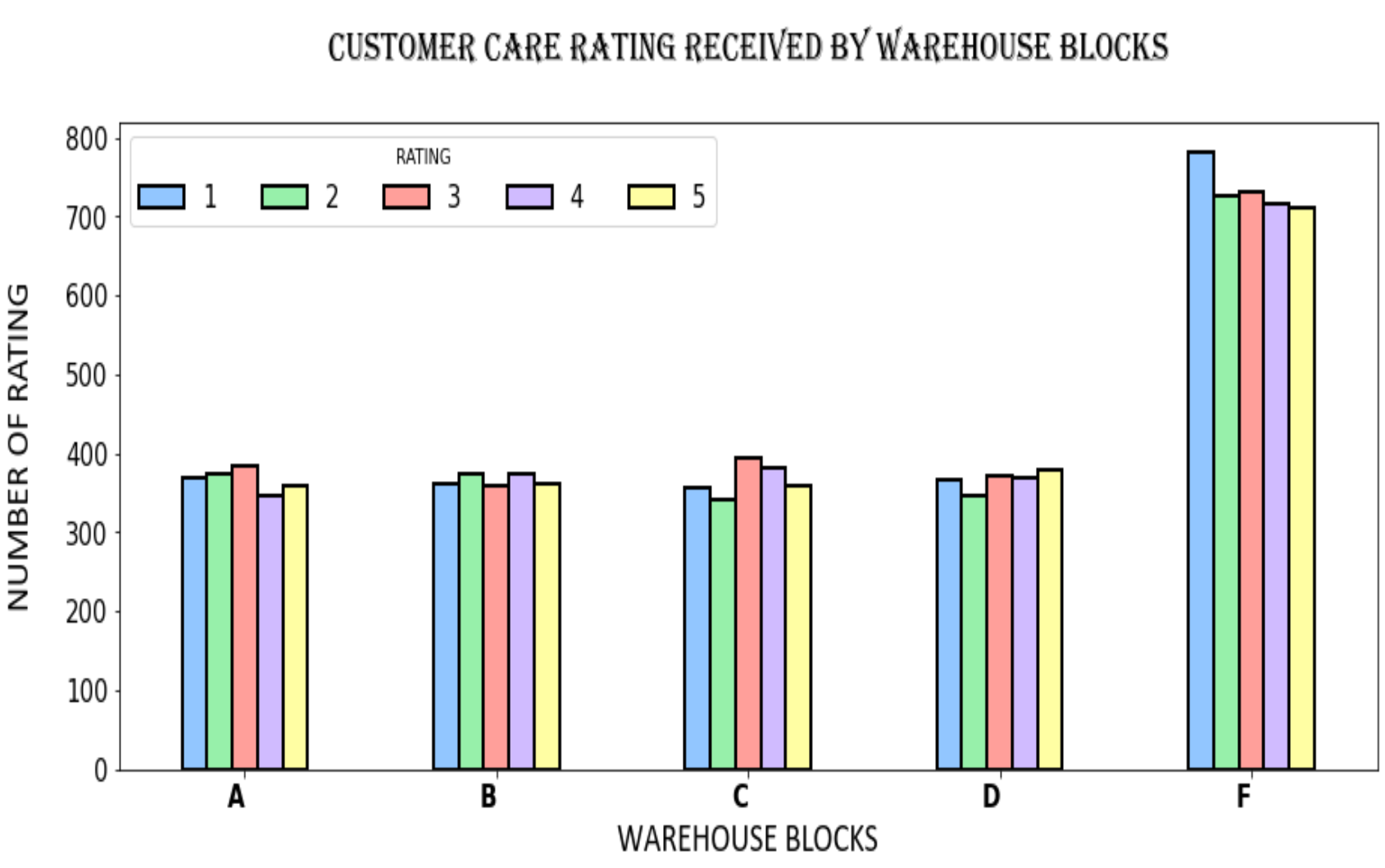


In [32]:
```
#Customer Care calls received by warehouse blocks
d1=df.copy()
grouped=pd.DataFrame(d1.groupby(['Warehouse_block'])['Customer_care_calls'].sum())
lineplot=grouped.plot(figsize=(8,5), marker='*',ms=30,mec='k',linestyle='-',linewidth=2,mfc='yellow',color='blac
font1={'family':'Algerian','color':'black','size':20,'fontweight':'bold'}
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()
#Conclusion: Warehouse Block 'F' has received the highest Customer Care calls
```

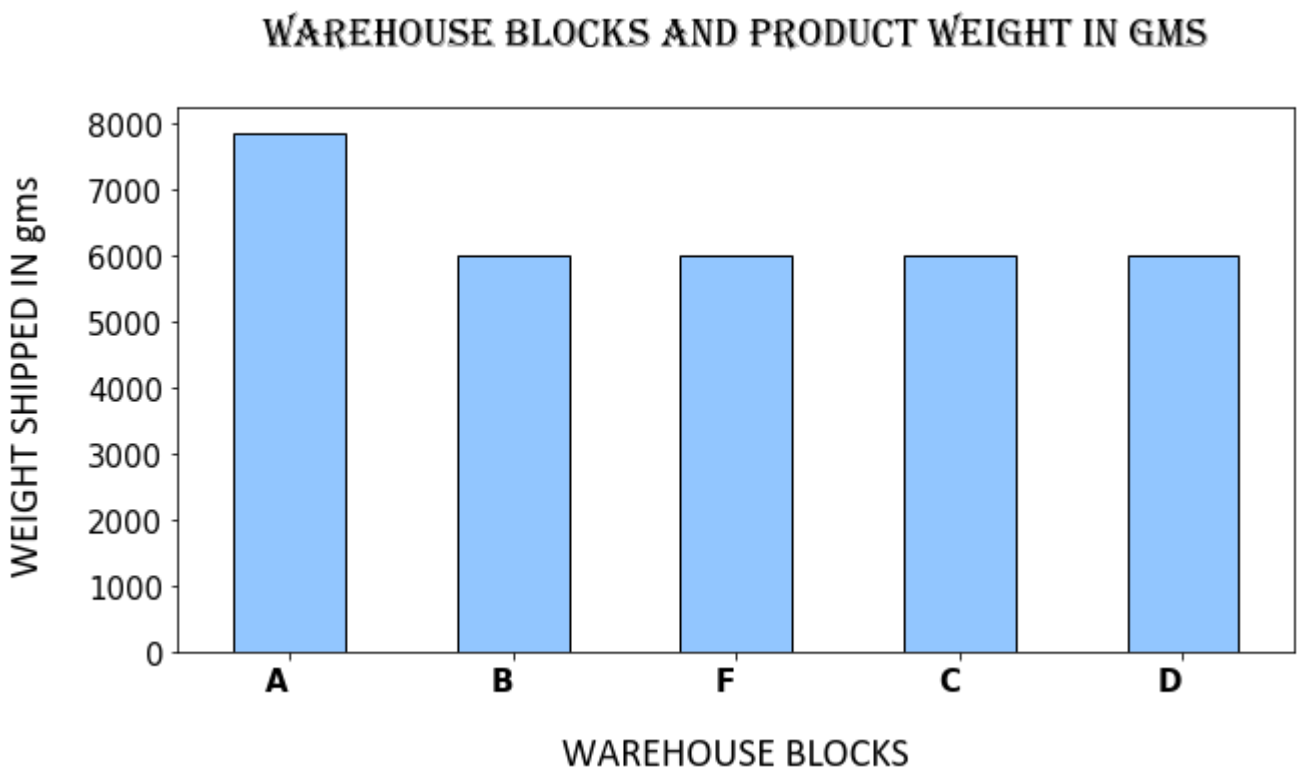


Bar plots

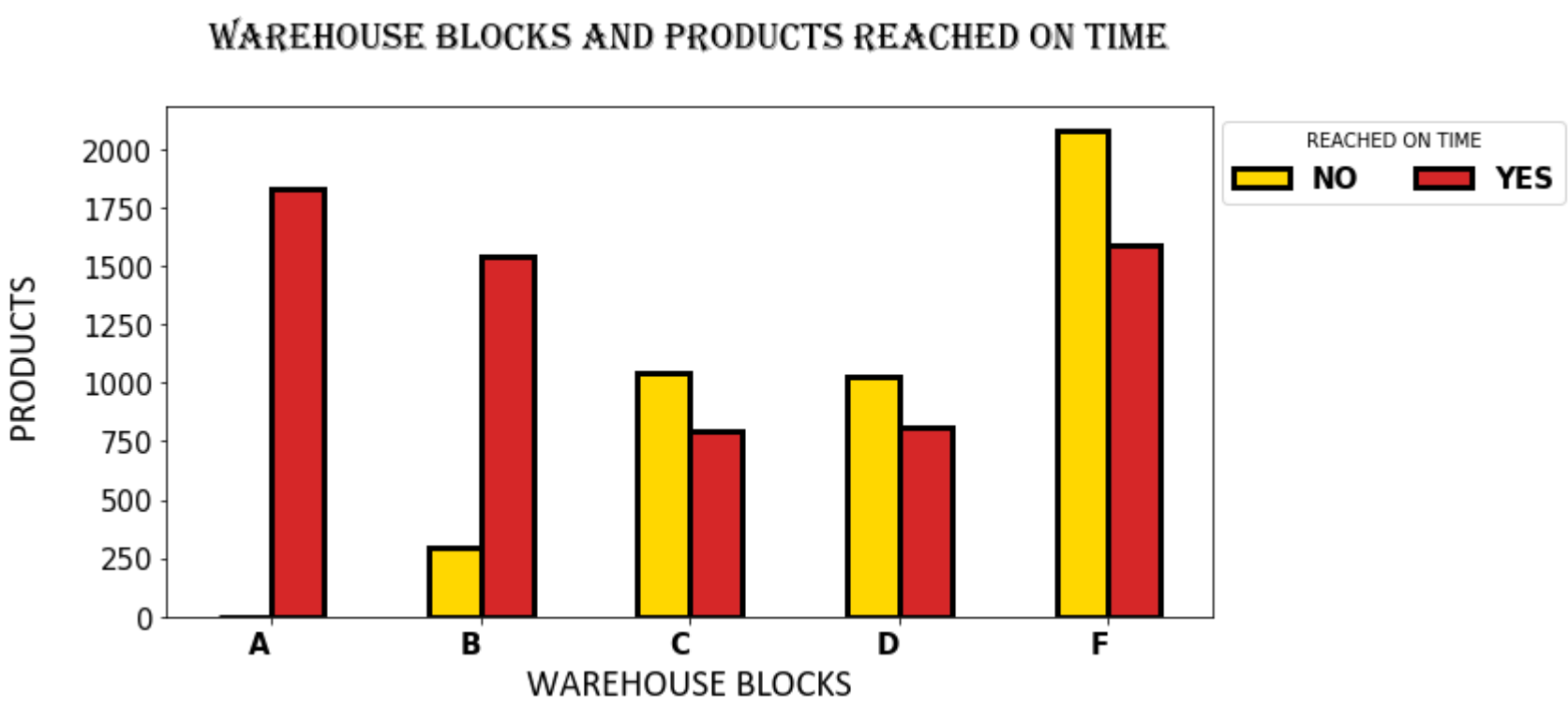
```
#Customer ratings 1-5 given to all the warehouse blocks
plt.style.use('seaborn-pastel')
ctab=pd.crosstab(d['Warehouse_block'],d['Customer_rating'])
#c=['tomato','skyblue','blue','seagreen','gold',slateblue]
barplot=ctab.plot.bar(figsize=(15,6),edgecolor='black',linewidth=2,width=0.5)
font1={'family':'Algerian','color':'black','size':20,'fontweight':'bold'}
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()

#Conclusion:
#Warehouse Block 'A' and 'B' has received the highest rating of '3'
#Warehouse Block 'B' has received the highest rating of '2' and '4'
#Warehouse Block 'D' has received the highest rating of '5'
#Warehouse Block 'F' has received the highest rating of '1'
#Among all the blocks, 'F' has received the highest rating of '1','2','3','4','5'
```



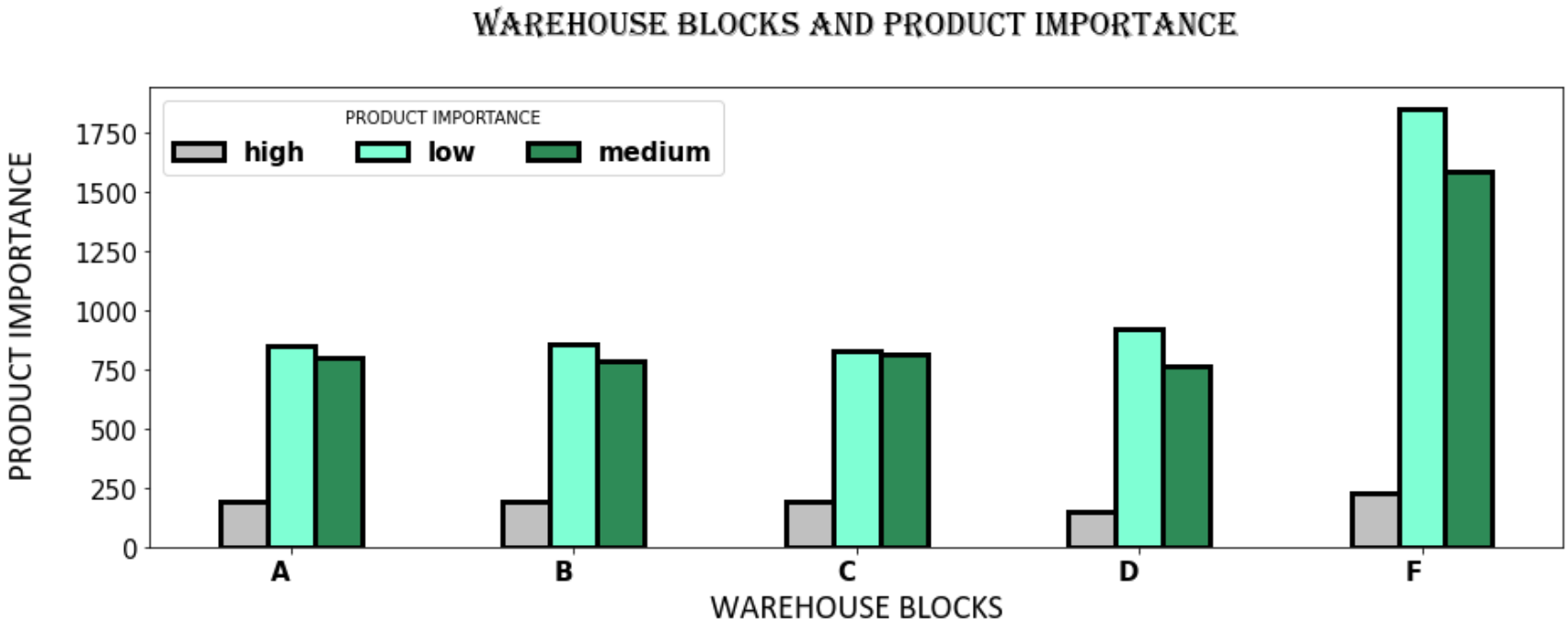
In [34]:
```
#Warehouse blocks and different weights of products shipped
plt.figure(figsize=(10,5))

warehouse=d.groupby(['Warehouse_block'])
#warehouse.max().sort_values(by="Weight_in_gms",ascending=False)["Weight_in_gms"].plot.bar(edgecolor='k',linewid
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()
#Conclusion:Warehouse Block A has shipped a product that has the heaviest weight among all the products.
#Remaining warehouse blocks have shipped equal weights.
```



In [35]:
```
#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()

#Conclusion: Warehouse block A has delivered all the products on time.
#Warehouse block F has not delivered most of the products on time.
```

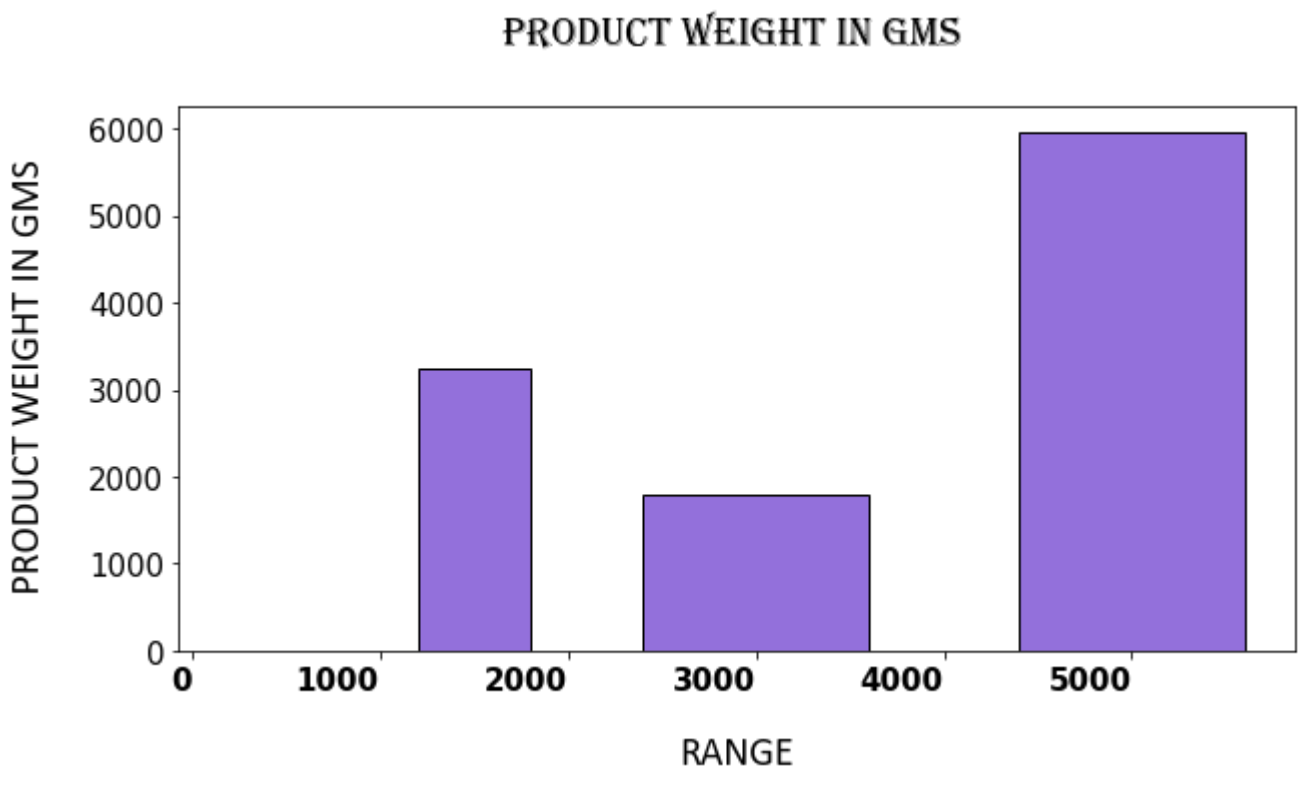


In [36]:
```
#low,Medium,High importance products delivered by all the warehouse blocks
ctab=pd.crosstab(d['Warehouse_block'],d['Product_importance'])
barplot=ctab.plot.bar(figsize = (15,5),edgecolor='black',color=['silver', 'aquamarine', 'seagreen'],linewidth=3)
plt.title("WAREHOUSE BLOCKS AND PRODUCT 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()
#Conclusion:Warehouse block A,B,D,F has shipped most of the products of low importance.
#Warehouse block C has shipped most of the products of low and medium 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 w
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()
#Conclusion: Heavy products were shipped by most of the blocks
```



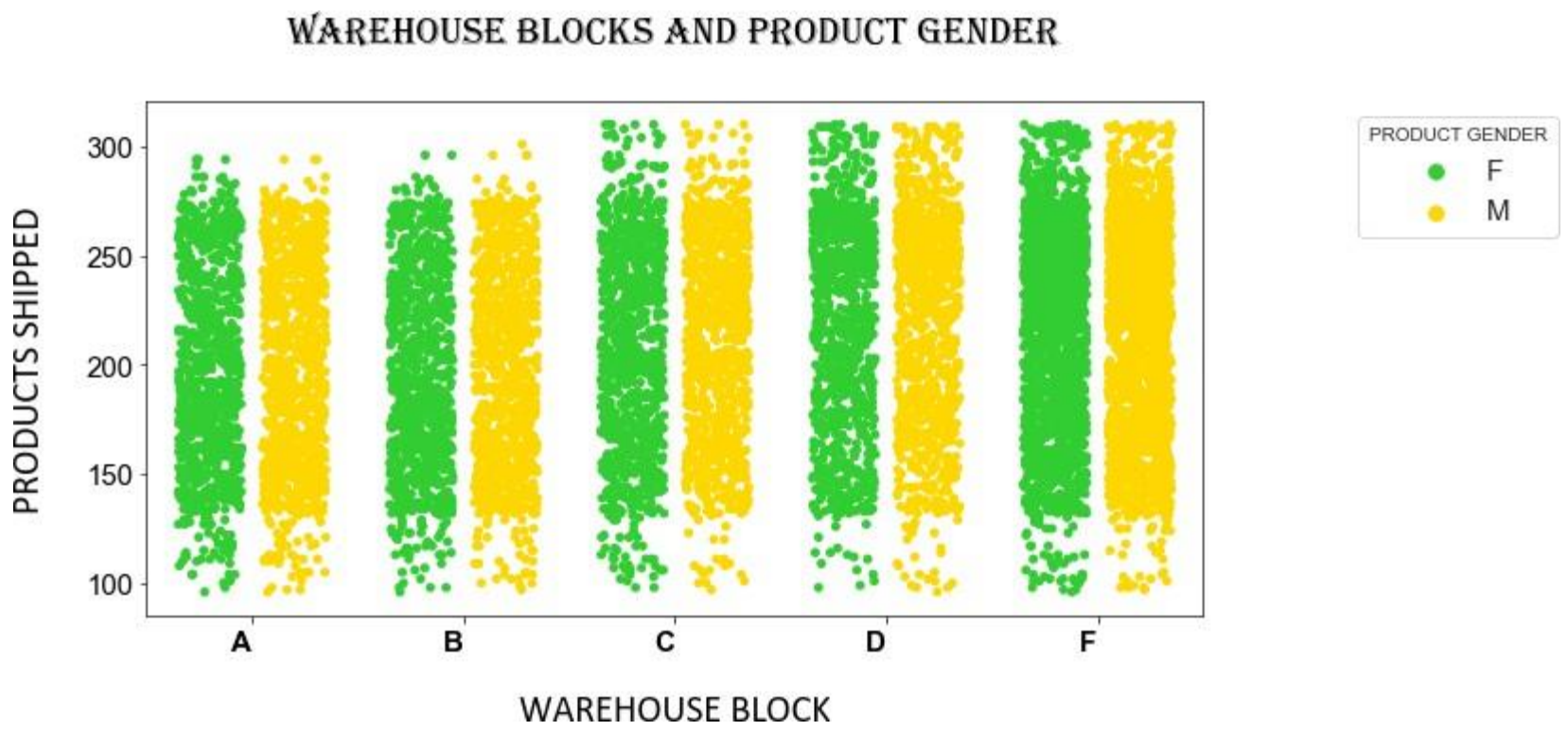
Seaborn

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()
#Conclusion: From the graph we can see that the density of ship mode is more.
#Most of the Warehouse blocks have used Ship mode to deliver the products.
#Few of the heavy products weighing more than 7000gms was shipped through Ship mode.
```

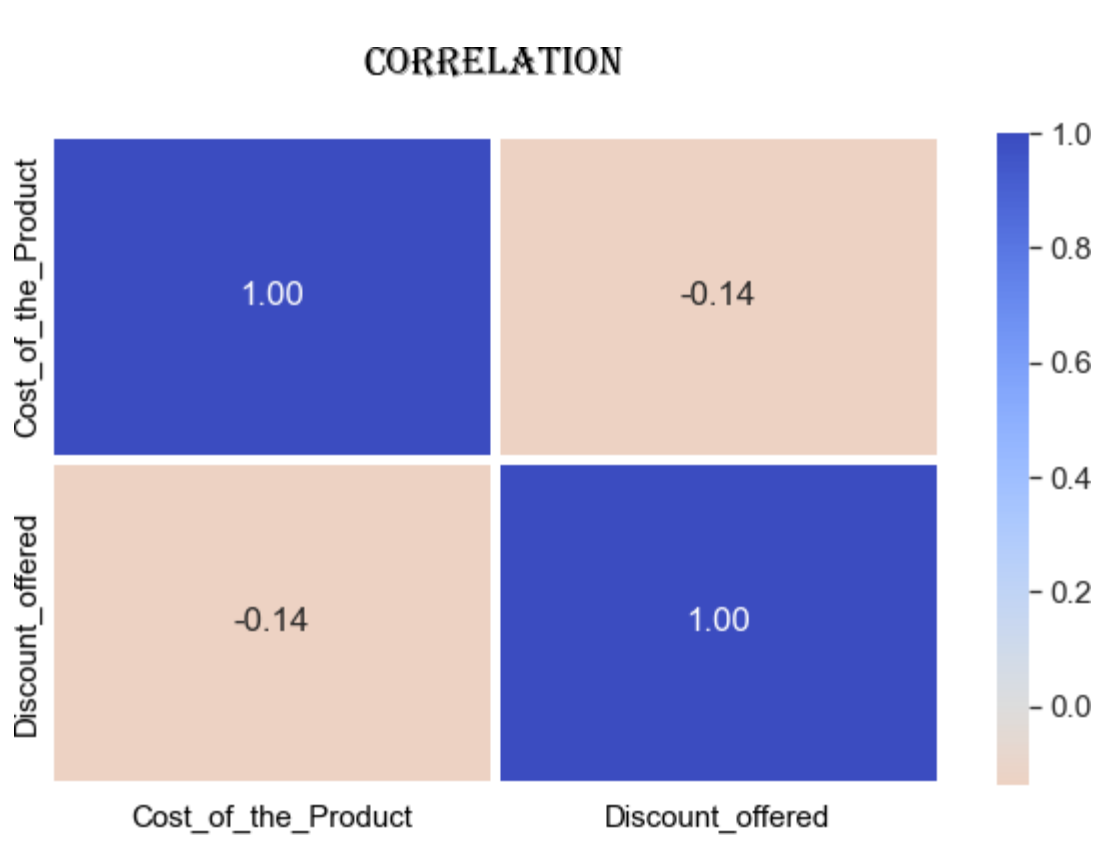
STRIPPLOT

```
#STRIP PLOT
#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()
```



HEATMAP

```
#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()
```



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. We can also observe that Warehouse blocks A,B delivered the products on time and Warehouse blocks C,D,F, did not deliver most of the products on time.
3. Customer care calls was received more by the Warehouse Block F.
4. 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/