DATA ANALYTICS PHASE

In [1]:
```
# Importing Python Neccesory Libraries
import numpy as np
import pandas as pd

# Importing scipy.stats library
import scipy.stats as stats
from statsmodels.stats.outliers_influence import variance_inflation_factor

# Import Data visualization Library
import matplotlib.pyplot as plt
import seaborn as sns

# Import Filter Warning Library
import warnings
warnings.filterwarnings('ignore')

# Import Stats Model Library
import scipy
import statsmodels.api as sm
from statsmodels.formula.api import ols
```

In [2]:
```
# Importing Data using pandas Function

URL = 'https://raw.githubusercontent.com/chandanc5525/SupplyChain_BusinessMo
df = pd.read_csv(URL)
data = pd.read_csv(URL)
df.sample(10)
```

Out[2]:

	Ware_house_ID	WH_Manager_ID	Location_type	WH_capacity_size	zone	WH_regional_z
1169	WH_101169	EID_51169	Rural	Mid	North	Zo
19546	WH_119546	EID_69546	Rural	Mid	West	Zo
8255	WH_108255	EID_58255	Rural	Small	West	Zo
7359	WH_107359	EID_57359	Rural	Large	West	Zo
12747	WH_112747	EID_62747	Rural	Large	North	Zo
8783	WH_108783	EID_58783	Rural	Large	North	Zo
20919	WH_120919	EID_70919	Rural	Mid	West	Zo
13687	WH_113687	EID_63687	Urban	Mid	South	Zo
15687	WH_115687	EID_65687	Rural	Small	West	Zo
20910	WH_120910	EID_70910	Rural	Large	North	Zo

10 rows × 24 columns

In [3]:
```
# Checking Data Information
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22150 entries, 0 to 22149
Data columns (total 24 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Ware_house_ID                 22150 non-null  object
```

```
 1   WH_Manager_ID                 22150 non-null  object
 2   Location_type                 22150 non-null  object
 3   WH_capacity_size              22150 non-null  object
 4   zone                          22150 non-null  object
 5   WH_regional_zone              22150 non-null  object
 6   num_refill_req_l3m            22150 non-null  int64
 7   transport_issue_l1y           22150 non-null  int64
 8   Competitor_in_mkt             22150 non-null  int64
 9   retail_shop_num               22150 non-null  int64
 10  wh_owner_type                 22150 non-null  object
 11  distributor_num               22150 non-null  int64
 12  flood_impacted                22150 non-null  int64
 13  flood_proof                   22150 non-null  int64
 14  electric_supply               22150 non-null  int64
 15  dist_from_hub                 22150 non-null  int64
 16  workers_num                   21273 non-null  float64
 17  wh_est_year                   11605 non-null  float64
 18  storage_issue_reported_l3m    22150 non-null  int64
 19  temp_reg_mach                 22150 non-null  int64
 20  approved_wh_govt_certificate  21345 non-null  object
 21  wh_breakdown_l3m              22150 non-null  int64
 22  govt_check_l3m                22150 non-null  int64
 23  product_wg_ton                22150 non-null  int64
dtypes: float64(2), int64(14), object(8)
memory usage: 4.1+ MB
```

In [4]:
```python
# Columns in the Dataset
df.columns
```

Out[4]:
```
Index(['Ware_house_ID', 'WH_Manager_ID', 'Location_type', 'WH_capacity_siz
e',
       'zone', 'WH_regional_zone', 'num_refill_req_l3m', 'transport_issue_l1
y',
       'Competitor_in_mkt', 'retail_shop_num', 'wh_owner_type',
       'distributor_num', 'flood_impacted', 'flood_proof', 'electric_suppl
y',
       'dist_from_hub', 'workers_num', 'wh_est_year',
       'storage_issue_reported_l3m', 'temp_reg_mach',
       'approved_wh_govt_certificate', 'wh_breakdown_l3m', 'govt_check_l3m',
       'product_wg_ton'],
      dtype='object')
```

In [5]:
```python
# Checking Dataset Description
df.describe(include='all')
```

Out[5]:

	Ware_house_ID	WH_Manager_ID	Location_type	WH_capacity_size	zone	WH_regional_
count	22150	22150	22150	22150	22150	2
unique	22150	22150	2	3	4	
top	WH_100000	EID_50000	Rural	Large	North	Z
freq	1	1	20334	8968	9069	
mean	NaN	NaN	NaN	NaN	NaN	
std	NaN	NaN	NaN	NaN	NaN	
min	NaN	NaN	NaN	NaN	NaN	
25%	NaN	NaN	NaN	NaN	NaN	
50%	NaN	NaN	NaN	NaN	NaN	
75%	NaN	NaN	NaN	NaN	NaN	
max	NaN	NaN	NaN	NaN	NaN	

11 rows × 24 columns

In [6]:

```
# Checking Categorical Information
cat_feature = df[['Location_type','WH_capacity_size','zone','WH_regional_zon
                               'wh_owner_type','approved_wh_govt_certificate
for i in cat_feature:
  print('*'*20)
  print(i)
  print(cat_feature[i].value_counts())
  print('*'*20)
```

```
********************
Location_type
Rural    20334
Urban     1816
Name: Location_type, dtype: int64
********************
********************
WH_capacity_size
Large    8968
Mid      8902
Small    4280
Name: WH_capacity_size, dtype: int64
********************
********************
zone
North    9069
West     7055
South    5644
East      382
Name: zone, dtype: int64
********************
********************
WH_regional_zone
Zone 6    7376
Zone 5    4045
Zone 4    3708
Zone 2    2642
Zone 3    2552
Zone 1    1827
Name: WH_regional_zone, dtype: int64
********************
********************
wh_owner_type
Company Owned    12035
Rented           10115
Name: wh_owner_type, dtype: int64
********************
********************
approved_wh_govt_certificate
C     4859
B+    4321
B     4269
A     4158
A+    3738
Name: approved_wh_govt_certificate, dtype: int64
********************
```

CONVERTING CATEGORICAL COLUMNS INTO NUMERICAL COLUMNS

```
In [7]: df['Location_type'] = df['Location_type'].map({'Rural':1,'Urban':0})
        df['wh_owner_type'] = df['wh_owner_type'].map({'Company Owned':1,'Rented':0}
```

```
In [8]: # Using Label Enconder technique
        from sklearn import preprocessing
        label_encoder = preprocessing.LabelEncoder()
        df['WH_capacity_size']= label_encoder.fit_transform(df['WH_capacity_size'])
        df['zone']= label_encoder.fit_transform(df['zone'])
        df['WH_regional_zone']= label_encoder.fit_transform(df['WH_regional_zone'])
        df['approved_wh_govt_certificate']= label_encoder.fit_transform(df['approved
        df['Ware_house_ID'] = label_encoder.fit_transform(df['Ware_house_ID'])
        df['WH_Manager_ID'] = label_encoder.fit_transform(df['WH_Manager_ID']).astyp
        df['wh_est_year '] = label_encoder.fit_transform(df['wh_est_year'])
```

```
In [9]: # Checking random 10 sample
        df.sample(10)
```

Out[9]:

	Ware_house_ID	WH_Manager_ID	Location_type	WH_capacity_size	zone	WH_regional_zc
14911	14911	14911	1	0	3	
2587	2587	2587	1	0	1	
6502	6502	6502	1	0	1	
1746	1746	1746	1	2	1	
6696	6696	6696	1	1	3	
20855	20855	20855	1	1	2	
3498	3498	3498	1	1	2	
2001	2001	2001	1	0	2	
19243	19243	19243	1	0	1	
3867	3867	3867	0	0	3	

10 rows × 25 columns

```
In [10]: # Checling for Null Values in the Dataset
         df.isnull().sum()
```

```
Out[10]: Ware_house_ID                       0
         WH_Manager_ID                       0
         Location_type                       0
         WH_capacity_size                    0
         zone                                0
         WH_regional_zone                    0
         num_refill_req_l3m                  0
         transport_issue_l1y                 0
         Competitor_in_mkt                   0
         retail_shop_num                     0
         wh_owner_type                       0
         distributor_num                     0
         flood_impacted                      0
         flood_proof                         0
         electric_supply                     0
         dist_from_hub                       0
         workers_num                       877
         wh_est_year                     10545
         storage_issue_reported_l3m          0
         temp_reg_mach                       0
         approved_wh_govt_certificate        0
         wh_breakdown_l3m                    0
```

```
govt_check_l3m                   0
product_wg_ton                   0
wh_est_year\t                    0
dtype: int64
```

In [11]:
```python
# Missing Percentage in Feature Columns Information
percent_missing = df.isnull().sum() * 100 / len(df)
percent_missing
```

Out[11]:
```
Ware_house_ID                    0.000000
WH_Manager_ID                    0.000000
Location_type                    0.000000
WH_capacity_size                 0.000000
zone                             0.000000
WH_regional_zone                 0.000000
num_refill_req_l3m               0.000000
transport_issue_l1y              0.000000
Competitor_in_mkt                0.000000
retail_shop_num                  0.000000
wh_owner_type                    0.000000
distributor_num                  0.000000
flood_impacted                   0.000000
flood_proof                      0.000000
electric_supply                  0.000000
dist_from_hub                    0.000000
workers_num                      3.959368
wh_est_year                     47.607223
storage_issue_reported_l3m       0.000000
temp_reg_mach                    0.000000
approved_wh_govt_certificate     0.000000
wh_breakdown_l3m                 0.000000
govt_check_l3m                   0.000000
product_wg_ton                   0.000000
wh_est_year\t                    0.000000
dtype: float64
```

In [12]:
```python
df.wh_est_year.mean()
```

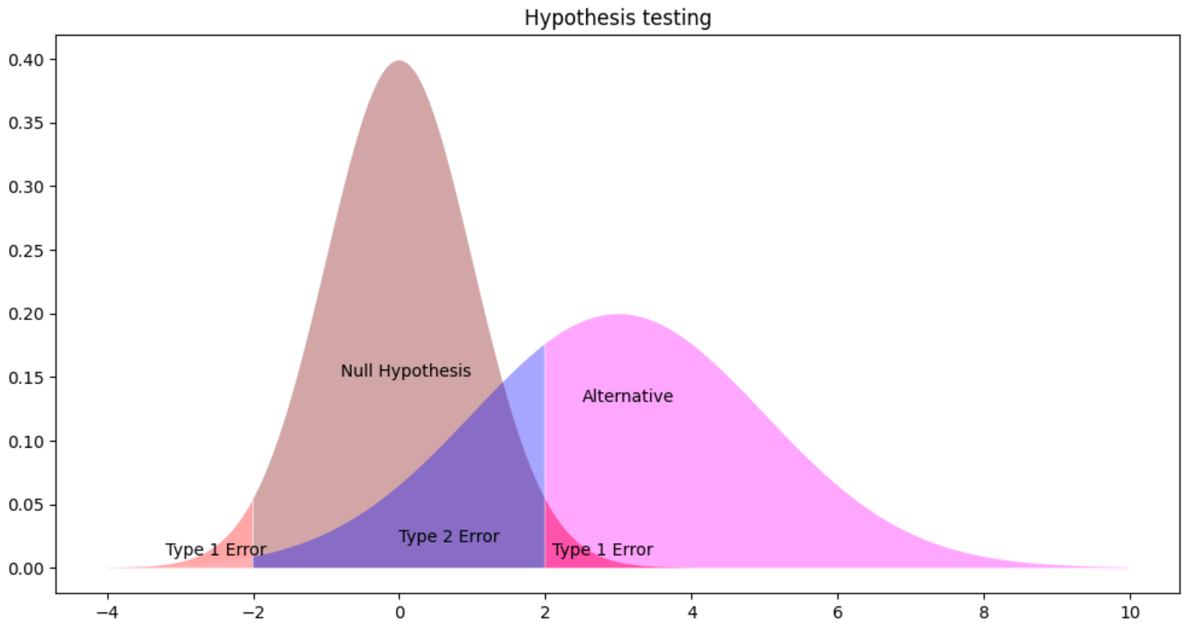
Out[12]:
```
2009.4012063765617
```

In [13]:
```python
df['workers_num'] = df['workers_num'].fillna(28)
```

HYPOTHESIS TESTING

In [14]:
```python
plt.figure(figsize=(12,6))


plt.fill_between(x=np.arange(-4,-2,0.01),
                 y1= stats.norm.pdf(np.arange(-4,-2,0.01)) ,
                 facecolor='red',
                 alpha=0.35)

plt.fill_between(x=np.arange(-2,2,0.01),
                 y1= stats.norm.pdf(np.arange(-2,2,0.01)) ,
                 facecolor='maroon',
                 alpha=0.35)

plt.fill_between(x=np.arange(2,4,0.01),
                 y1= stats.norm.pdf(np.arange(2,4,0.01)) ,
                 facecolor='red',
                 alpha=0.5)
```

```
plt.fill_between(x=np.arange(-4,-2,0.01),
                 y1= stats.norm.pdf(np.arange(-4,-2,0.01),loc=3, scale=2) ,
                 facecolor='pink',
                 alpha=0.35)

plt.fill_between(x=np.arange(-2,2,0.01),
                 y1= stats.norm.pdf(np.arange(-2,2,0.01),loc=3, scale=2) ,
                 facecolor='blue',
                 alpha=0.35)

plt.fill_between(x=np.arange(2,10,0.01),
                 y1= stats.norm.pdf(np.arange(2,10,0.01),loc=3, scale=2),
                 facecolor='magenta',
                 alpha=0.35)
plt.title("Hypothesis testing")
plt.text(x=-0.8, y=0.15, s= "Null Hypothesis")
plt.text(x=2.5, y=0.13, s= "Alternative")
plt.text(x=2.1, y=0.01, s= "Type 1 Error")
plt.text(x=-3.2, y=0.01, s= "Type 1 Error")
plt.text(x=0, y=0.02, s= "Type 2 Error");
```



Hypothesis Testing for Categorical to Numerical Columns

In [15]:
```
f_statistic, p_value = stats.f_oneway(df['Ware_house_ID'],df['product_wg_ton

# Output the results
print('stat=%.2f, p=%.20f' %(f_statistic,p_value))
```

```
stat=15257.34, p=0.00000000000000000000
```

In [16]:
```
f_statistic, p_value = stats.f_oneway(df['WH_Manager_ID'],df['product_wg_ton

# Output the results
print('stat=%.2f, p=%.20f' %(f_statistic,p_value))
```

```
stat=15257.34, p=0.00000000000000000000
```

In [17]:
```
t,p = stats.ttest_ind(data[data['Location_type']=='Rural']['product_wg_ton']
print('t :'+ str(round(t,2)))
print('t=%.2f, p=%.20f' %(t,p))
```

```
t :-11.07
t=-11.07, p=0.00000000000000000000
```

In [18]:
```
f_statistic, p_value = stats.f_oneway(df['WH_capacity_size'],df['product_wg_

# Output the results
print('stat=%.2f, p=%.20f' %(f_statistic,p_value))
```
```
stat=79934.19, p=0.00000000000000000000
```

In [19]:
```
f_statistic, p_value = stats.f_oneway(df['zone'],df['product_wg_ton'])

# Output the results
print('stat=%.2f, p=%.20f' %(f_statistic,p_value))
```
```
stat=79926.32, p=0.00000000000000000000
```

In [20]:
```
f_statistic, p_value = stats.f_oneway(df['WH_regional_zone'],df['product_wg_

# Output the results
print('stat=%.2f, p=%.20f' %(f_statistic,p_value))
```
```
stat=79916.39, p=0.00000000000000000000
```

In [21]:
```
t,p = stats.ttest_ind(data[data['wh_owner_type']=='Rented']['product_wg_ton'
print('t :'+ str(round(t,2)))
print('t=%.2f, p=%.20f' %(t,p))
```
```
t :0.51
t=0.51, p=0.61158875451007932433
```

In [22]:
```
f_statistic, p_value = stats.f_oneway(df['approved_wh_govt_certificate'],df[

# Output the results
print('stat=%.2f, p=%.20f' %(f_statistic,p_value))
```
```
stat=79923.98, p=0.00000000000000000000
```

Hypothesis Testing for Numerical to Numerical Columns

In [23]:
```
R,P = stats.pearsonr(df['num_refill_req_l3m'],df['product_wg_ton'])
print('R :'+ str(round(R,2)))
print('P :'+ str(round(P,2)))
print('stat=%.2f, p=%.20f' %(R,P))
```
```
R :0.0
P :0.92
stat=0.00, p=0.91989267492983683994
```

In [24]:
```
R,P = stats.pearsonr(df['transport_issue_l1y'],df['product_wg_ton'])
print('R :'+ str(round(R,2)))
print('P :'+ str(round(P,2)))
print('stat=%.2f, p=%.20f' %(R,P))
```
```
R :-0.18
P :0.0
stat=-0.18, p=0.00000000000000000000
```

In [25]:
```
R,P = stats.pearsonr(df['Competitor_in_mkt'],df['product_wg_ton'])
print('R :'+ str(round(R,2)))
print('P :'+ str(round(P,2)))
print('stat=%.2f, p=%.20f' %(R,P))
```
```
R :0.01
P :0.28
stat=0.01, p=0.27848681841281763827
```

In [26]:
```
R,P = stats.pearsonr(df['retail_shop_num'],df['product_wg_ton'])
print('R :'+ str(round(R,2)))
print('P :'+ str(round(P,2)))
print('stat=%.2f, p=%.20f' %(R,P))
```

```
R :-0.01
P :0.31
stat=-0.01, p=0.31259007105613251243
```

In [27]:
```
R,P = stats.pearsonr(df['distributor_num'],df['product_wg_ton'])
print('R :'+ str(round(R,2)))
print('P :'+ str(round(P,2)))
print('stat=%.2f, p=%.20f' %(R,P))
```

```
R :0.01
P :0.44
stat=0.01, p=0.44404971522041036813
```

In [28]:
```
R,P = stats.pearsonr(df['flood_impacted'],df['product_wg_ton'])
print('R :'+ str(round(R,2)))
print('P :'+ str(round(P,2)))
print('stat=%.2f, p=%.20f' %(R,P))
```

```
R :-0.01
P :0.45
stat=-0.01, p=0.45184983951362300836
```

In [29]:
```
R,P = stats.pearsonr(df['flood_proof'],df['product_wg_ton'])
print('R :'+ str(round(R,2)))
print('P :'+ str(round(P,2)))
print('stat=%.2f, p=%.20f' %(R,P))
```

```
R :-0.0
P :0.6
stat=-0.00, p=0.59554197309092182788
```

In [30]:
```
R,P = stats.pearsonr(df['electric_supply'],df['product_wg_ton'])
print('R :'+ str(round(R,2)))
print('P :'+ str(round(P,2)))
print('stat=%.2f, p=%.20f' %(R,P))
```

```
R :-0.0
P :0.91
stat=-0.00, p=0.91402026048922280221
```

In [31]:
```
R,P = stats.pearsonr(df['dist_from_hub'],df['product_wg_ton'])
print('R :'+ str(round(R,2)))
print('P :'+ str(round(P,2)))
print('stat=%.2f, p=%.20f' %(R,P))
```

```
R :-0.01
P :0.37
stat=-0.01, p=0.36871402505469297317
```

In [32]:
```
R,P = stats.pearsonr(df['workers_num'],df['product_wg_ton'])
print('R :'+ str(round(R,2)))
print('P :'+ str(round(P,2)))
print('stat=%.2f, p=%.20f' %(R,P))
```

```
R :-0.01
P :0.27
stat=-0.01, p=0.26520196589743239013
```

In [33]:
```
R,P = stats.pearsonr(df['storage_issue_reported_l3m'],df['product_wg_ton'])
print('R :'+ str(round(R,2)))
```

```
print('P :'+ str(round(P,2)))
print('stat=%.2f, p=%.20f' %(R,P))
```

```
R :0.99
P :0.0
stat=0.99, p=0.00000000000000000000
```

In [34]:
```
R,P = stats.pearsonr(df['temp_reg_mach'],df['product_wg_ton'])
print('R :'+ str(round(R,2)))
print('P :'+ str(round(P,2)))
print('stat=%.2f, p=%.20f' %(R,P))
```

```
R :0.1
P :0.0
stat=0.10, p=0.00000000000000000000
```

In [35]:
```
R,P = stats.pearsonr(df['wh_breakdown_l3m'],df['product_wg_ton'])
print('R :'+ str(round(R,2)))
print('P :'+ str(round(P,2)))
print('stat=%.2f, p=%.20f' %(R,P))
```

```
R :0.34
P :0.0
stat=0.34, p=0.00000000000000000000
```

In [36]:
```
R,P = stats.pearsonr(df['govt_check_l3m'],df['product_wg_ton'])
print('R :'+ str(round(R,2)))
print('P :'+ str(round(P,2)))
print('stat=%.2f, p=%.20f' %(R,P))
```

```
R :-0.01
P :0.07
stat=-0.01, p=0.07467192869333899585
```

In [37]:
```
Result = {
'Feature Comparison' : ['Ware_house_ID vs product_wg_ton', 'WH_Manager_ID vs
                'WH_capacity_size vs product_wg_ton','zone vs product_wg_ton
                'num_refill_req_l3m vs product_wg_ton', 'transport_issue_l1y
                'retail_shop_num vs product_wg_ton', 'wh_owner_type vs produ
                'flood_impacted vs product_wg_ton', 'flood_proof vs product_
                'dist_from_hub vs product_wg_ton', 'workers_num vs product_w
                'temp_reg_mach vs product_wg_ton','approved_wh_govt_certific
                'govt_check_l3m vs product_wg_ton'],

'Statistics Used' :       ['Anova Test','Anova Test','T Test','Anova Test',
                 'Pearson Correlation','Pearson Correlation','Pearson Correl
                 'Pearson Correlation','Pearson Correlation','Pearson Correl
                 'Pearson Correlation','Pearson Correlation','Anova Test','P

'P-value':      [0.00000000,0.00000000,0.00000000,0.00000000,0.00000000,0.00
                0.27848,0.31259,0.61158,0.44404,0.45184,0.59554,0.91402,0.36
                0.00000000,0.00000000,0.00000000,0.07467],


'Null Hypothesis' :         ['There is significance association between war
                 'There is significance association between WH_Manager_ID an
                 'In location Type Column, Rural locations are significant t
                 'Warehouse Capacity are significant to Product weight',
                 'Zones significantly affecting product weight',
                 'Warehouse Regional locations zone are significant to Produ
                 'Refill Request has some significance with product weight i
                 'Transport issues reported has some significance with produ
                 'No.of Competitors in Market has some significance with pro
                 'No.of Retailors shops has some significance with product w
                 'Warehouse Owner type has some significance with product we
```

```
                 'Distributor Number has some significance with product weig
                 'Flood Impacted Regions has some significance with product
                 'Flood Proof Regions has some significance with product wei
                 'Electric supply with power backup facilities has some sign
                 'Distance from Hub has some significance with product weigh
                 'No of workers in Warehouse has some significance with prod
                 'Storage issues has some significance with product weight',
                 'Temp reg_match has some significance with product weight',
                 'Approved Govt_Certificates has some significance with prod
                 'Warehouse Breakdown has some significance with product wei
                 'Government inspection checks has some significance with pr




],

'Alternate Hypothesis' :          ['There is no significance association bet
                 'There is no significance association between WH_Manager_ID
                 'In Location Type Column, Rural locations are not significa
                 'Warehouse Capacity are not significant to Product weight',
                 'Zones are not significantly affecting product weight',
                 'Warehouse Regional locations zone are not significant to P
                 'Refill Request has no significance with product weight in
                 'Transport issues reported has no significance with product
                 'No.of Competitors in Market has no significance with produ
                 'No.of Retailors shops has no significance with product wei
                 'Warehouse Owner type has no significance with product weig
                 'Distributor Number has no significance with product weight
                 'Flood Impacted Regions has no significance with product we
                 'Flood Proof Regions has no significance with product weigh
                 'Electric supply with power backup facilities has no signif
                 'Distance from Hub has no significance with product weight'
                 'No of workers in Warehouse has no significance with produc
                 'Storage issues has no significance with product weight',
                 'Temp reg_match has no significance with product weight',
                 'Approved Govt_Certificates has no significance with produc
                 'Warehouse Breakdown has no significance with product weigh
                 'Government inspection checks has no significance with prod


                ],

'Conclusion' :  ['There is no significance association between warehouse_ID
                 'There is no significance association between WH_Manager_ID
                 'In Location Type Column, Rural locations are not significa
                 'Warehouse Capacity are not significant to Product weight',
                 'Zones are not significantly affecting product weight',
                 'Warehouse Regional locations zone are not significant to P
                 'Refill Request has some significance with product weight i
                 'Transport issues reported has no significance with product
                 'No.of Competitors in Market has some significance with pro
                 'No.of Retailors shops has some significance with product w
                 'Warehouse Owner type has some significance with product we
                 'Distributor Number has some significance with product weig
                 'Flood Impacted Regions has some significance with product
                 'Flood Proof Regions has some significance with product wei
                 'Electric supply with power backup facilities has some sign
                 'Distance from Hub has some significance with product weigh
                 'No of workers in Warehouse has some significance with prod
                 'Storage issues has some significance with product weight',
                 'Temp reg_match has some significance with product weight',
                 'Approved Govt_Certificates has no significance with produc
                 'Warehouse Breakdown has some significance with product wei
```

```
                'Government inspection checks has some significance with pr

]
}

result_df = pd.DataFrame(Result)
```

In [38]:
```
result_df
```

Out[38]:

	Feature Comparison	Statistics Used	P-value	Null Hypothesis	Alternate Hypothesis	Concl
0	Ware_house_ID vs product_wg_ton	Anova Test	0.00000	There is significance association between ware...	There is no significance association between w...	There signifi assoc betwee
1	WH_Manager_ID vs product_wg_ton	Anova Test	0.00000	There is significance association between WH_M...	There is no significance association between W...	There signifi assoc betwee
2	Location_type vs product_wg_ton	T Test	0.00000	In location Type Column, Rural locations are s...	In Location Type Column, Rural locations are n...	In Location Column, locations a
3	WH_capacity_size vs product_wg_ton	Anova Test	0.00000	Warehouse Capacity are significant to Product ...	Warehouse Capacity are not significant to Prod...	Warel Capacity a signific F
4	zone vs product_wg_ton	Anova Test	0.00000	Zones significantly affecting product weight	Zones are not significantly affecting product ...	Zones a signifi affecting pr
5	WH_regional_zone vs product_wg_ton	Anova Test	0.00000	Warehouse Regional locations zone are signific...	Warehouse Regional locations zone are not sign...	Warel Re locations are not
6	num_refill_req_l3m vs product_wg_ton	Pearson Correlation	0.91989	Refill Request has some significance with prod...	Refill Request has no significance with produc...	Refill Re has significanc p
7	transport_issue_l1y sv product_wg_ton	Pearson Correlation	0.00000	Transport issues reported has some significanc...	Transport issues reported has no significance ...	Transport i reported h significar
8	Competitor_in_mkt vs product_wg_ton	Pearson Correlation	0.27848	No.of Competitors in Market has some significa...	No.of Competitors in Market has no significanc...	Competit Market has sign
9	retail_shop_num vs product_wg_ton	Pearson Correlation	0.31259	No.of Retailors shops has some significance wi...	No.of Retailors shops has no significance with...	No.of Ret shops has significanc
10	wh_owner_type vs product_wg_ton	T Test	0.61158	Warehouse Owner type has some significance wit...	Warehouse Owner type has no significance with ...	Warel Owner typ significance
11	distributor_num vs product_wg_ton	Pearson Correlation	0.44404	Distributor Number has some	Distributor Number has no significance with pr...	Distr Numbe

				significance with ...		significanc
12	flood_impacted vs product_wg_ton	Pearson Correlation	0.45184	Flood Impacted Regions has some significance w...	Flood Impacted Regions has no significance wit...	Flood Imp Regior significanc
13	flood_proof vs product_wg_ton	Pearson Correlation	0.59554	Flood Proof Regions has some significance with...	Flood Proof Regions has no significance with p...	Flood Regior signifi
14	electric_supply vs product_wg_ton	Pearson Correlation	0.91402	Electric supply with power backup facilities h...	Electric supply with power backup facilities h...	Electric s with p backup fac
15	dist_from_hub vs product_wg_ton	Pearson Correlation	0.36871	Distance from Hub has some significance with p...	Distance from Hub has no significance with pro...	Distance Hub has significanc
16	workers_num vs product_wg_ton	Pearson Correlation	0.26520	No of workers in Warehouse has some significan...	No of workers in Warehouse has no significance...	No of work Warehous signifi
17	storage_issue_reported_l3m vs product_wg_ton	Pearson Correlation	0.00000	Storage issues has some significance with prod...	Storage issues has no significance with produc...	Storage i has significanc p
18	temp_reg_mach vs product_wg_ton	Pearson Correlation	0.00000	Temp reg_match has some significance with prod...	Temp reg_match has no significance with produc...	Temp reg_ has significanc p
19	approved_wh_govt_certificate vs product_wg_ton	Anova Test	0.00000	Approved Govt_Certificates has some significan...	Approved Govt_Certificates has no significance...	App Govt_Certif h significa
20	wh_breakdown_l3m vs product_wg_ton	Pearson Correlation	0.00000	Warehouse Breakdown has some significance with...	Warehouse Breakdown has no significance with p...	Warel Breakdow signifi
21	govt_check_l3m vs product_wg_ton	Pearson Correlation	0.07467	Government inspection checks has some signific...	Government inspection checks has no significan...	Goverr insp check some sig

```
In [ ]:
```