In [1]:

**import** pandas **as** pd

**import** numpy **as** np

dataset**=**pd.read\_csv("C:/Users/Harshini/Downloads/Pharma\_Monthly\_Sales.csv") df**=**pd.DataFrame(dataset)

df

**Product Location**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Out[1]: |  | | | |
|  |  | **Month** | **Customer** | **Period** |
|  | **0** | March | Mobil | Q1 |
|  | **1** | March | Mobil | Q1 |
|  | **2** | March | Mobil | Q1 |
|  | **3** | March | Mobil | Q1 |
|  | **4**  **... 545** | March  ...  November | Mobil  ...  Cadbury | Q1  ... Q4 |
|  | **546** | November | Cadbury | Q4 |
|  | **547** | November | Cadbury | Q4 |
|  | **548** | November | Cadbury | Q4 |
|  | **549** | November | Cadbury | Q4 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Rep** |  |
| Afinitor | Port | A. | Astra |
| Tab | Harcourt | Chukwu | Zeneca |
| Brinerdine | Port | A. | Astra |
| SCT | Harcourt | Chukwu | Zeneca |
| Coartem | Port | A. | Astra |
| Tab | Harcourt | Chukwu | Zeneca |
| Codiovan | Port | A. | Astra |
| FCT | Harcourt | Chukwu | Zeneca |
| Femara | Port | A. | Astra |
| FCT | Harcourt | Chukwu | Zeneca |
| ... | ... | ... | ... |

**Sales**

**Supplier**

**Warehouse**

**Actual C.Sales**

**Inventory**

**N**

**L.Sales M.Sales**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Locations** |  |  | **Stock** |  | |
| Amuwo | 9.60000 | 2 | 1200.0 | 1.89 | 4.2240 |
| Amuwo | 9.60000 | 0 | 480.0 | 1.89 | 4.2240 |
| Amuwo | 17.60000 | 3 | 1320.0 | 3.51 | 7.7440 |
| Amuwo | 20.90000 | 3 | 960.0 | 4.14 | 9.1960 |
| Amuwo | 4.50000 | 0 | 600.0 | 0.90 | 1.9800 |
| ... | ... | ... | ... | ... | ... |
| Amuwo | 19.18080 | 1 | 120.0 | 1.92 | 2.4420 |
| Amuwo | 18.24768 | 2 | 120.0 | 1.84 | 2.3232 |
| Amuwo | 19.44000 | 3 | 240.0 | 2.00 | 2.4750 |
| Amuwo | 16.41600 | 0 | 420.0 | 1.68 | 2.0900 |
| Amuwo | 23.70816 | 1 | 300.0 | 2.40 | 3.0184 |

**R**

Galvusmet

FCT

Lescol HGC

Kaduna S. Sunday

Kaduna S. Sunday

Sanofi Sanofi

Ritalin Tab Kaduna S. Sunday

Sanofi

Tegretol

Tab

Zaditen SRO FCT

Kaduna S. Sunday

Kaduna S. Sunday

Sanofi Sanofi

550 rows × 17 columns

In [2]:

df.info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 550 entries, 0 to 549

Data columns (total 17 columns):

# Column Non-Null Count Dtype

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 |  | Month |  | 550 | non-null |  | object |
| 1 |  | Customer |  | 550 | non-null |  | object |
| 2 |  | Period |  | 550 | non-null |  | object |
| 3 |  | Product |  | 550 | non-null |  | object |
| 4 |  | Location |  | 550 | non-null |  | object |
| 5 |  | Sales Rep |  | 550 | non-null |  | object |
| 6 |  | Supplier |  | 550 | non-null |  | object |
| 7 |  | Warehouse | Locations | 550 | non-null |  | object |
| 8 |  | Actual |  | 550 | non-null |  | float64 |
| 9 |  | C.Sales |  | 550 | non-null |  | int64 |
| 10 |  | Inventory | Stock | 550 | non-null |  | float64 |
| 11 |  | L.Sales |  | 550 | non-null |  | float64 |
| 12 |  | M.Sales |  | 550 | non-null |  | float64 |
| 13 |  | Number of | Records | 550 | non-null |  | int64 |

1. Received Inventory 550 non-null int64
2. Rep.Sales 550 non-null float64
3. Target 550 non-null float64 dtypes: float64(6), int64(3), object(8)

memory usage: 73.2+ KB

In [3]:

df.describe()

Out[3]:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Actual** | **C.Sales** | **Inventory Stock** | **L.Sales** | **M.Sales** | **Number of Records** | **Received Inventory** | **Rep.Sales** | **Target** |
| **count** | 550.000000 | 550.000000 | 550.000000 | 550.000000 | 550.000000 | 550.0 | 550.000000 | 550.000000 | 550.000000 |
| **mean** | 15.588579 | 2.510909 | 720.443942 | 2.685091 | 3.706096 | 1.0 | 3254.545455 | 5.200000 | 17.944625 |
| **std** | 6.675771 | 1.919811 | 706.185183 | 2.329320 | 1.783619 | 0.0 | 1779.443250 | 2.076603 | 7.716178 |
| **min** | 4.500000 | 0.000000 | 6.586967 | 0.400000 | 0.792000 | 1.0 | 1000.000000 | 1.500000 | 5.000000 |
| **25%** | 11.250000 | 1.000000 | 250.000000 | 1.200000 | 2.414720 | 1.0 | 2000.000000 | 4.200000 | 12.000000 |
| **50%** | 13.824000 | 2.000000 | 500.000000 | 1.890000 | 3.590400 | 1.0 | 3000.000000 | 4.815000 | 17.280000 |
| **75%** | 19.180800 | 3.000000 | 930.000000 | 3.120000 | 4.527600 | 1.0 | 5000.000000 | 6.000000 | 21.600000 |
| **max** | 36.115200 | 10.000000 | 3850.000000 | 13.800000 | 9.196000 | 1.0 | 8000.000000 | 10.580000 | 49.000000 |

In [4]:

df.shape

Out[4]: (550, 17)

In [5]:

df.columns

Out[5]: Index(['Month', 'Customer', 'Period', 'Product', 'Location', 'Sales Rep', 'Supplier', 'Warehouse Locations', 'Actual', 'C.Sales',

'Inventory Stock', 'L.Sales', 'M.Sales', 'Number of Records', 'Received Inventory', 'Rep.Sales', 'Target'],

dtype='object')

In [6]:

df.dtypes

Out[6]: Month object Customer object

Period object

Product object

Location object

Sales Rep object

Supplier object

Warehouse Locations object Actual float64

C.Sales int64

Inventory Stock float64

L.Sales float64

M.Sales float64

Number of Records int64

Received Inventory int64 Rep.Sales float64

Target float64

dtype: object

In [7]:

df.isnull().sum()

|  |  |  |  |
| --- | --- | --- | --- |
| Out[7]: | Month |  | 0 |
|  | Customer |  | 0 |
|  | Period |  | 0 |
|  | Product |  | 0 |
|  | Location |  | 0 |
|  | Sales Rep |  | 0 |
|  | Supplier |  | 0 |
|  | Warehouse | Locations | 0 |
|  | Actual |  | 0 |
|  | C.Sales |  | 0 |
|  | Inventory | Stock | 0 |
|  | L.Sales |  | 0 |
|  | M.Sales |  | 0 |
|  | Number of | Records | 0 |

Received Inventory 0

Rep.Sales 0

Target 0

dtype: int64

In [8]:

df.duplicated()

|  |  |  |
| --- | --- | --- |
| Out[8]: | 0 | False |
|  | 1 | False |
|  | 2 | False |
|  | 3 | False |
|  | 4 | False |
|  |  | ... |
|  | 545 | False |
|  | 546 | False |
|  | 547 | False |
|  | 548 | False |
|  | 549 | False |

Length: 550, dtype: bool

In [9]:

**Product Location**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Out[9]: |  | **Month** | **Customer** | **Period** |
|  | **0** | March | Mobil | Q1 |
|  | **1** | March | Mobil | Q1 |
|  | **2** | March | Mobil | Q1 |
|  | **3** | March | Mobil | Q1 |
|  | **4** | March | Mobil | Q1 |

**Sales**

|  |  |  |  |
| --- | --- | --- | --- |
| Afinitor | Port | **Rep**  A. | Astra |
| Tab | Harcourt | Chukwu | Zeneca |
| Brinerdine | Port | A. | Astra |
| SCT | Harcourt | Chukwu | Zeneca |
| Coartem | Port | A. | Astra |
| Tab | Harcourt | Chukwu | Zeneca |
| Codiovan | Port | A. | Astra |
| FCT | Harcourt | Chukwu | Zeneca |
| Femara | Port | A. | Astra |
| FCT | Harcourt | Chukwu | Zeneca |

**Supplier**

**Warehouse Actual C.Sales Locations**

**Inventory L.Sales M.Sales Stock**

**Number**

df.head()

**of**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | **Records** |
| Amuwo | 9.6 | 2 | 1200.0 | 1.89 | 4.224 | 1 |
| Amuwo | 9.6 | 0 | 480.0 | 1.89 | 4.224 | 1 |
| Amuwo | 17.6 | 3 | 1320.0 | 3.51 | 7.744 | 1 |
| Amuwo | 20.9 | 3 | 960.0 | 4.14 | 9.196 | 1 |
| Amuwo | 4.5 | 0 | 600.0 | 0.90 | 1.980 | 1 |

In [10]:

df.tail()

Out[10]:

**Month Customer Period Product Location**

**Sales Supplier Rep**

**Warehouse**

**Actual C.Sales**

**Inventory**

**N**

**L.Sales M.Sales**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Locations** |  |  | **Stock** |  | |
| Sanofi | Amuwo | 19.18080 | 1 | 120.0 | 1.92 | 2.4420 |
| Sanofi | Amuwo | 18.24768 | 2 | 120.0 | 1.84 | 2.3232 |
| Sanofi | Amuwo | 19.44000 | 3 | 240.0 | 2.00 | 2.4750 |
| Sanofi | Amuwo | 16.41600 | 0 | 420.0 | 1.68 | 2.0900 |
| Sanofi | Amuwo | 23.70816 | 1 | 300.0 | 2.40 | 3.0184 |

**R**

1. November Cadbury Q4 Galvusmet

FCT

1. November Cadbury Q4 Lescol

HGC

Kaduna

Kaduna

S.

Sunday

S.

Sunday

1. November Cadbury Q4 Ritalin Tab Kaduna

S.

Sunday

1. November Cadbury Q4 Tegretol

Tab

1. November Cadbury Q4 Zaditen

SRO FCT

Kaduna Kaduna

S.

Sunday

S.

Sunday

In [48]:

df['Sales'] **=** df['C.Sales'] **+** df['M.Sales'] df

**Product Location**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Out[48]: |  | | | |
|  |  | **Month** | **Customer** | **Period** |
|  | **0** | March | Mobil | Q1 |
|  | **1** | March | Mobil | Q1 |
|  | **2** | March | Mobil | Q1 |
|  | **3** | March | Mobil | Q1 |
|  | **4**  **... 545** | March  ...  November | Mobil  ...  Cadbury | Q1  ... Q4 |
|  | **546** | November | Cadbury | Q4 |
|  | **547** | November | Cadbury | Q4 |
|  | **548** | November | Cadbury | Q4 |
|  | **549** | November | Cadbury | Q4 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Rep** |  |
| Afinitor | Port | A. | Astra |
| Tab | Harcourt | Chukwu | Zeneca |
| Brinerdine | Port | A. | Astra |
| SCT | Harcourt | Chukwu | Zeneca |
| Coartem | Port | A. | Astra |
| Tab | Harcourt | Chukwu | Zeneca |
| Codiovan | Port | A. | Astra |
| FCT | Harcourt | Chukwu | Zeneca |
| Femara | Port | A. | Astra |
| FCT | Harcourt | Chukwu | Zeneca |
| ... | ... | ... | ... |

**Sales**

**Supplier**

**Warehouse**

**Actual C.Sales**

**Inventory**

**N**

**L.Sales M.Sales**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Locations** |  |  | **Stock** |  | |
| Amuwo | 9.60000 | 2 | 1200.0 | 1.89 | 4.2240 |
| Amuwo | 9.60000 | 0 | 480.0 | 1.89 | 4.2240 |
| Amuwo | 17.60000 | 3 | 1320.0 | 3.51 | 7.7440 |
| Amuwo | 20.90000 | 3 | 960.0 | 4.14 | 9.1960 |
| Amuwo | 4.50000 | 0 | 600.0 | 0.90 | 1.9800 |
| ... | ... | ... | ... | ... | ... |
| Amuwo | 19.18080 | 1 | 120.0 | 1.92 | 2.4420 |
| Amuwo | 18.24768 | 2 | 120.0 | 1.84 | 2.3232 |
| Amuwo | 19.44000 | 3 | 240.0 | 2.00 | 2.4750 |
| Amuwo | 16.41600 | 0 | 420.0 | 1.68 | 2.0900 |
| Amuwo | 23.70816 | 1 | 300.0 | 2.40 | 3.0184 |

**R**

Galvusmet

FCT

Lescol HGC

Kaduna S. Sunday

Kaduna S. Sunday

Sanofi Sanofi

Ritalin Tab Kaduna S. Sunday

Sanofi

Tegretol

Tab

Zaditen SRO FCT

Kaduna S. Sunday

Kaduna S. Sunday

Sanofi Sanofi

1. rows × 18 columns

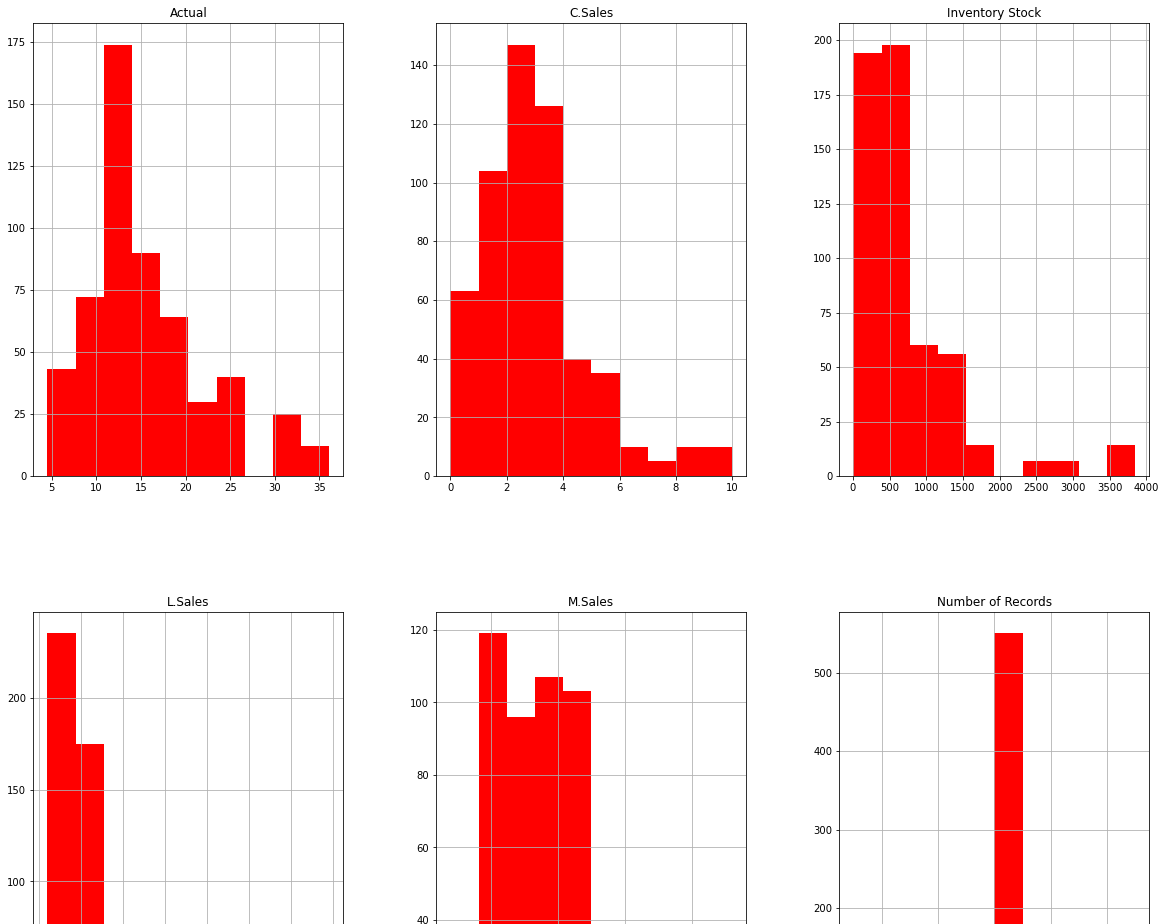
In [2]:

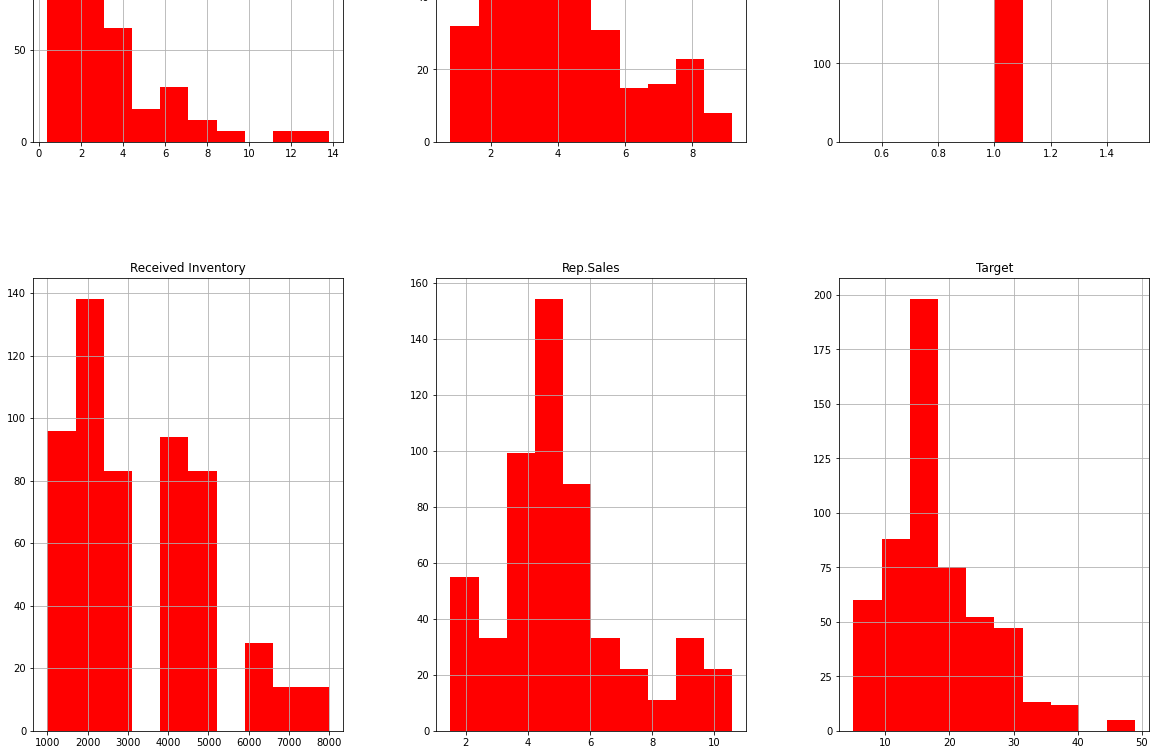
**import** matplotlib.pyplot **as** plt

**import** seaborn **as** sns

In [13]:

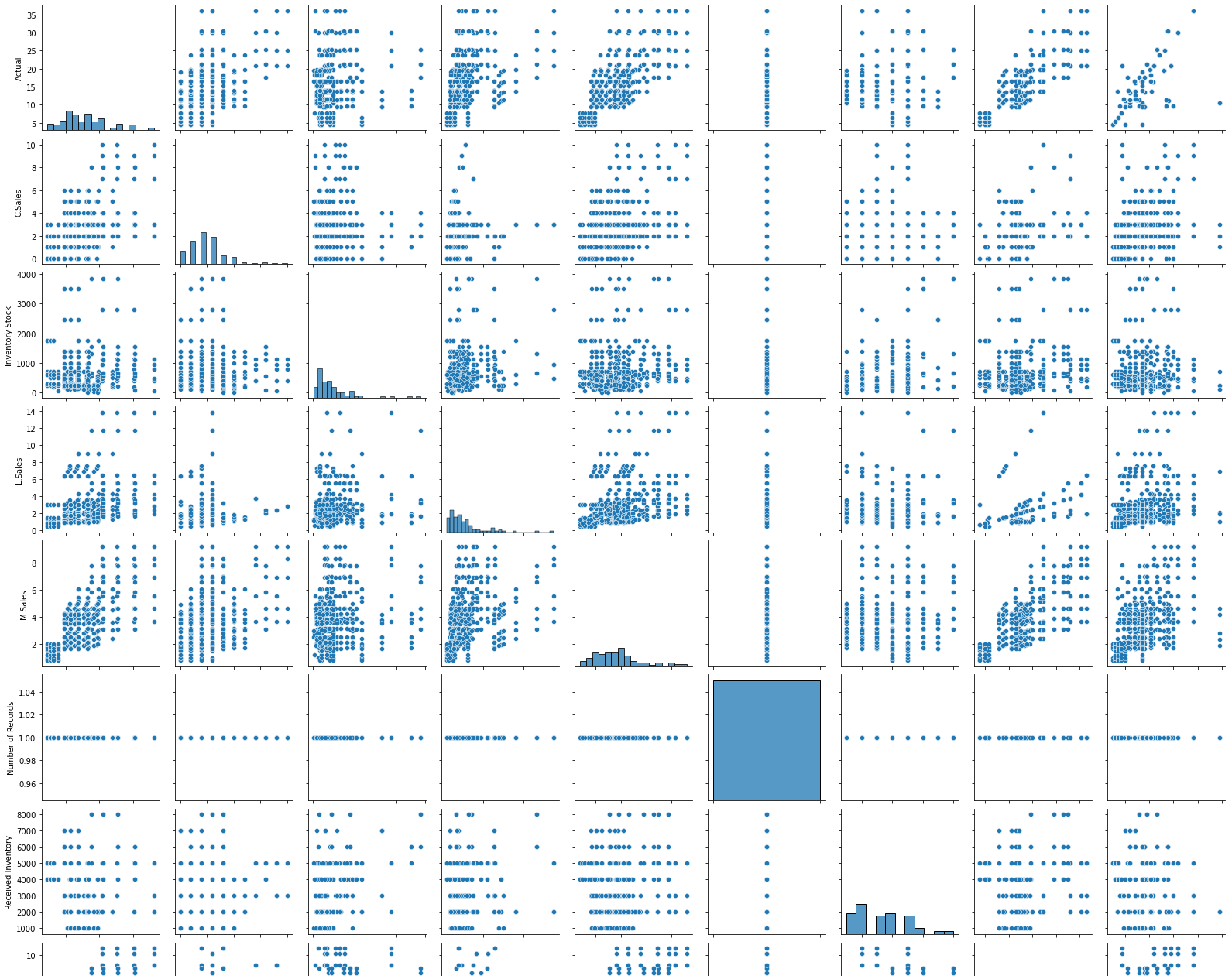
df.hist(figsize**=**(20,30),color**=**'r') plt.show()

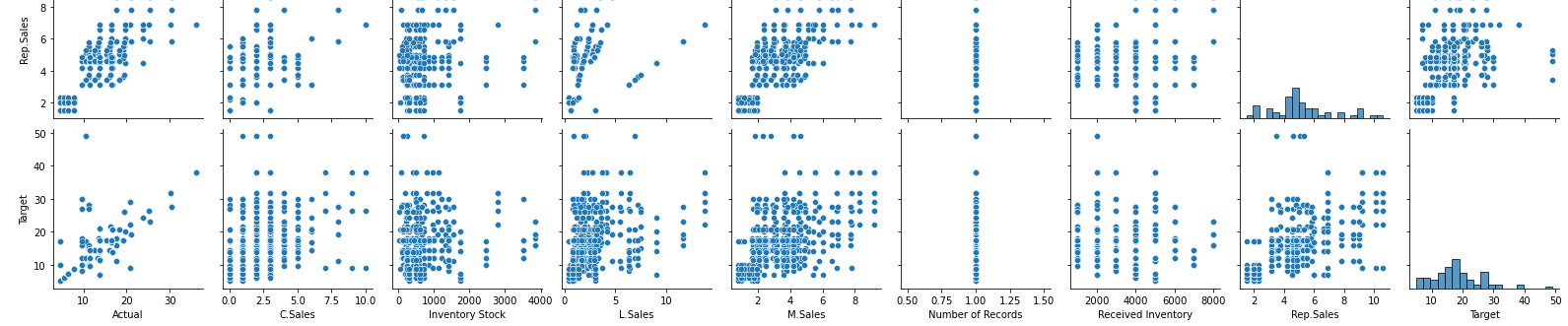




In [14]:

sns.pairplot(df) plt.show()

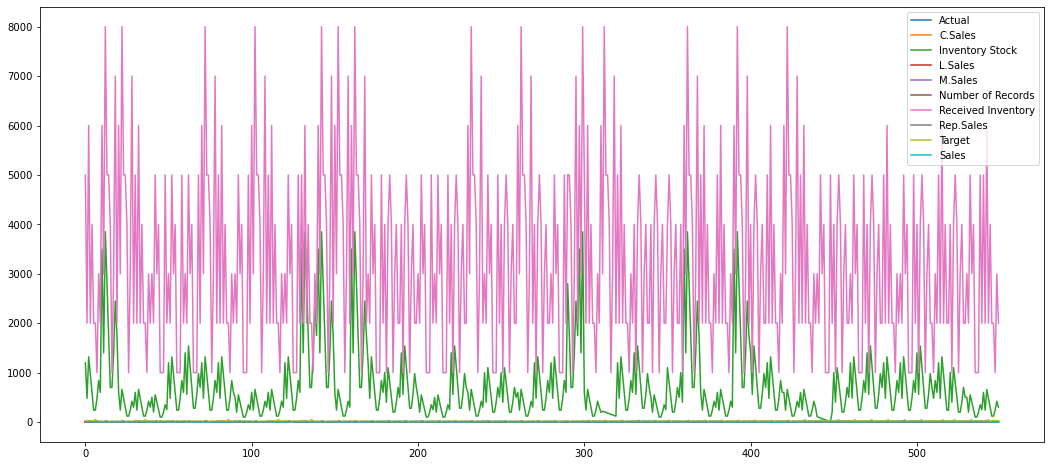




In [52]:

**import** matplotlib.pyplot **as** plt df.plot(figsize**=**(18, 8))

plt.show()



In [15]:

corr**=**df.corr() corr

Out[15]: **Actual C.Sales Inventory Stock**

**L.Sales M.Sales Number of Records**

**Received Inventory**

**Rep.Sales Target**

**Actual** 1.000000 0.427882 0.123447 0.383440 0.675535 NaN 0.071937 0.797727 0.645664

**C.Sales** 0.427882 1.000000 -0.019680 -0.020000 0.339860 NaN -0.028007 0.340277 0.263212

**Inventory Stock** 0.123447 -0.019680 1.000000 0.134041 0.143435 NaN 0.489744 0.205145 0.057460

**L.Sales** 0.383440 -0.020000 0.134041 1.000000 0.336746 NaN 0.081496 0.251889 0.269045

**M.Sales** 0.675535 0.339860 0.143435 0.336746 1.000000 NaN 0.101103 0.716406 0.442714

**Number of**

**Records** NaN NaN NaN NaN NaN NaN NaN NaN NaN

**Received**

**Inventory** 0.071937 -0.028007 0.489744 0.081496 0.101103 NaN 1.000000 0.128267 -0.053876

**Rep.Sales** 0.797727 0.340277 0.205145 0.251889 0.716406 NaN 0.128267 1.000000 0.542991

**Target** 0.645664 0.263212 0.057460 0.269045 0.442714 NaN -0.053876 0.542991 1.000000

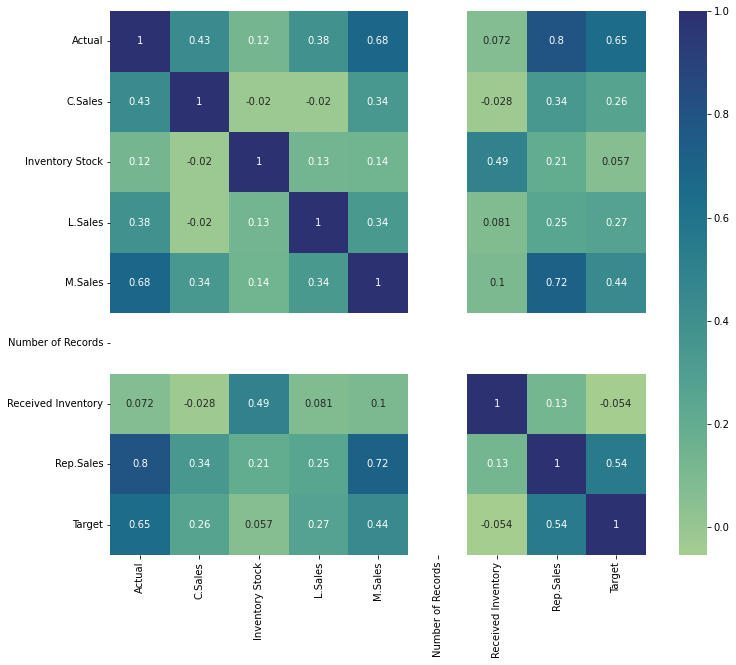
In [16]:

plt.figure(figsize**=**(12,10))

sns.heatmap(corr,annot**=True**,cmap**=**'crest')

Out[16]: <AxesSubplot:>





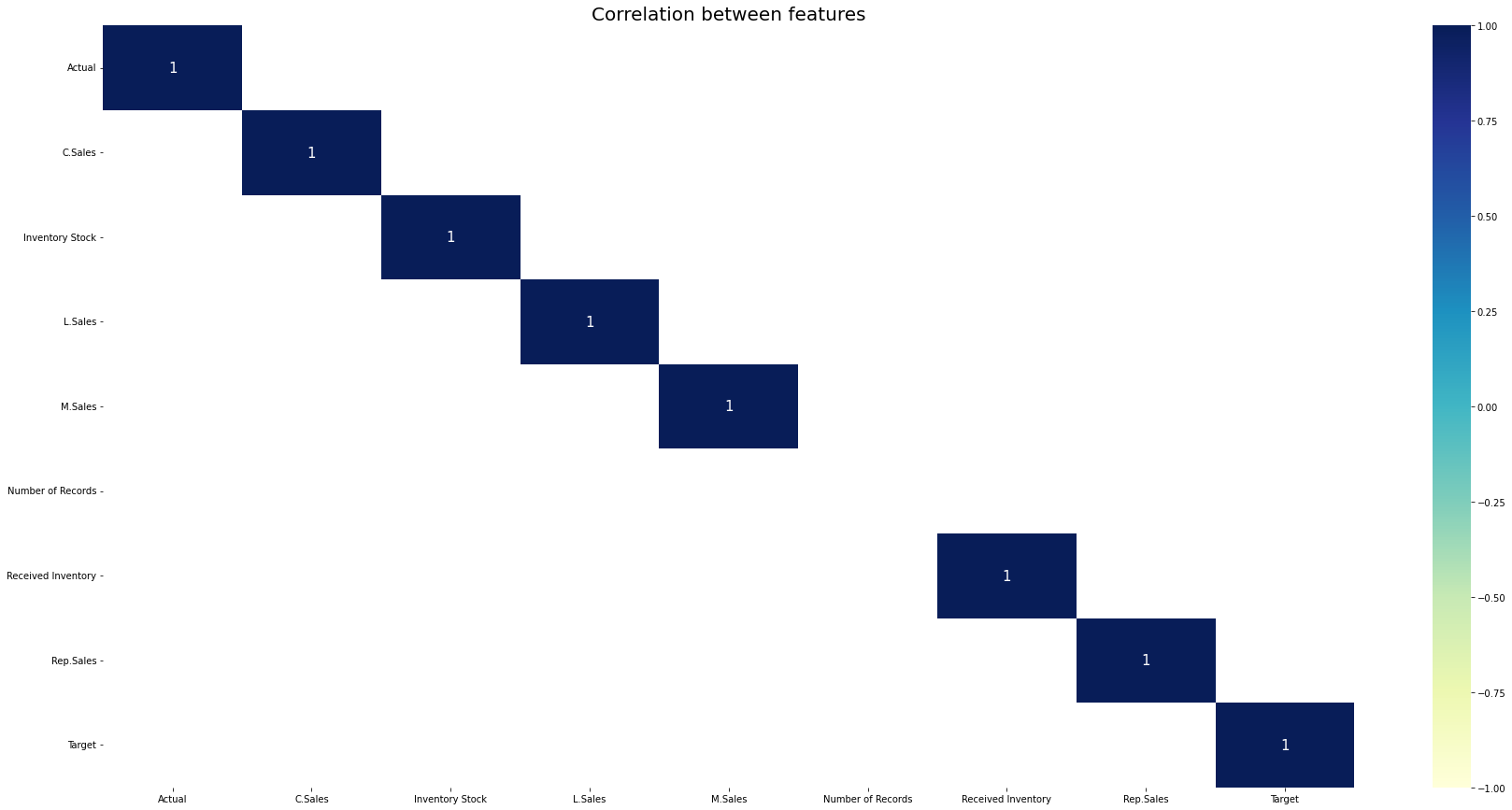


In [17]:

plt.figure(figsize**=**(30, 15))

sns.heatmap(corr[(corr **>=** 0.8) **|** (corr **<= -**0.9)], cmap**=**'YlGnBu', vmax**=**1.0, vmin**=-**1.0, annot**=True**, annot\_kws**=**{"size": 15})

plt.title('Correlation between features', fontsize**=**20) plt.show()

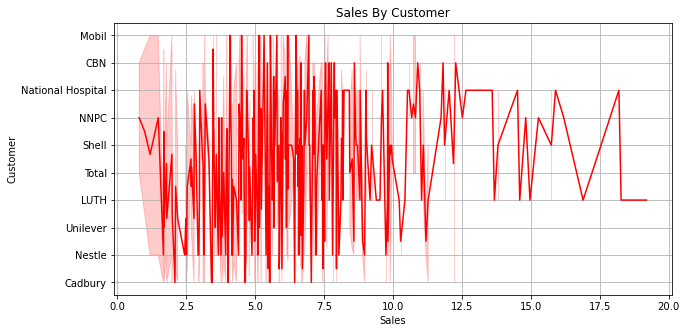


In [32]:

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

sns.lineplot(data**=**df, x**=**'Sales', y**=**'Customer',color**=**'r') plt.title("Sales By Customer")

plt.grid()

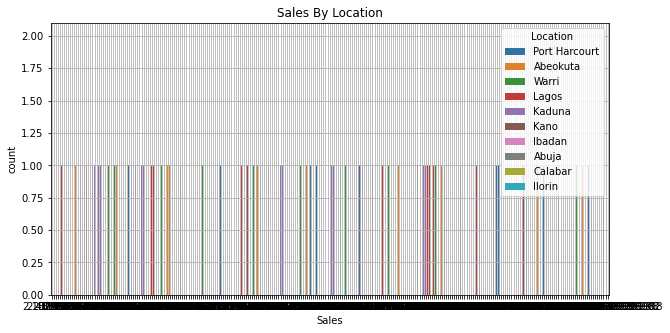


In [34]:

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

sns.countplot(data**=**df, x**=**'Sales', hue**=**'Location') plt.title("Sales By Location")

plt.grid()

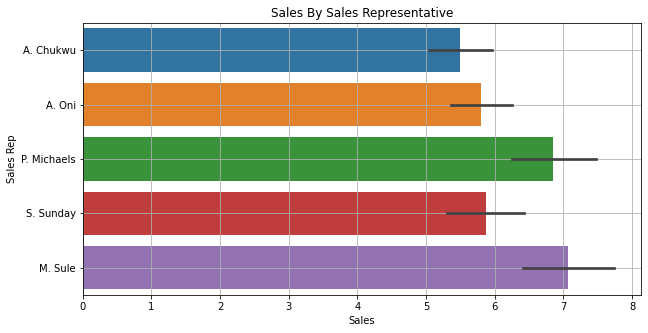


In [51]:

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

sns.barplot(data**=**df, x**=**'Sales',y**=**'Sales Rep') plt.title("Sales By Sales Representative")

plt.grid()

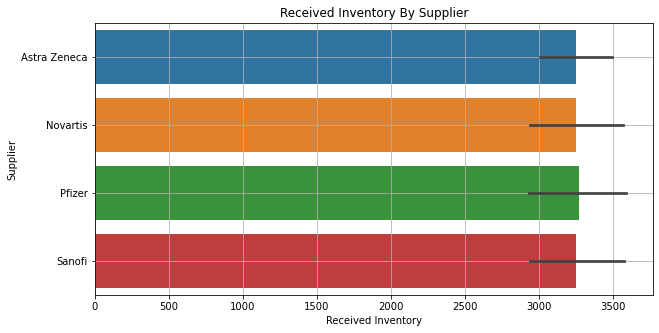


In [93]:

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

sns.barplot(data**=**df, x**=**'Received Inventory',y**=**'Supplier') plt.title("Received Inventory By Supplier")

plt.grid()



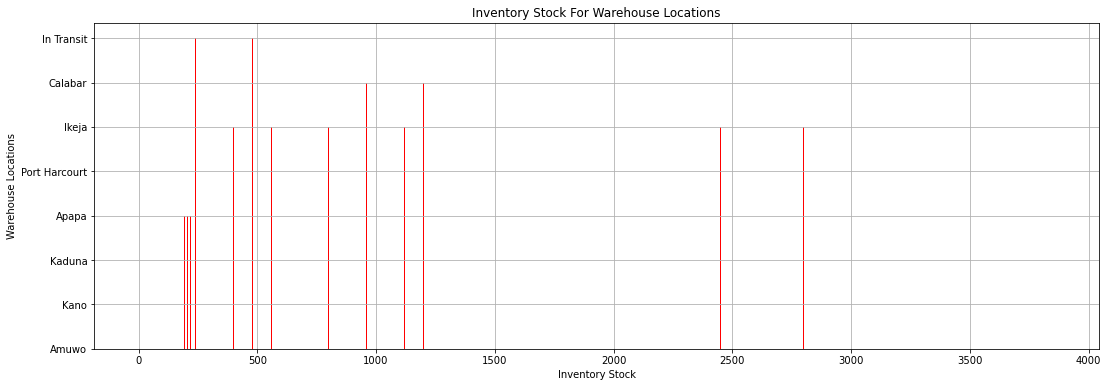
In [81]:

plt.figure(figsize**=**(18, 6))

plt.title("Inventory Stock For Warehouse Locations") plt.grid()

plt.bar(df['Inventory Stock'] , df['Warehouse Locations'], color**=**'r') plt.xlabel("Inventory Stock")

plt.ylabel("Warehouse Locations") plt.show()

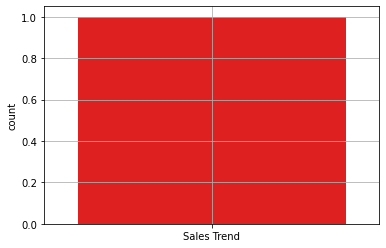


In [66]:

**from** warnings **import** filterwarnings filterwarnings("ignore")

In [82]:

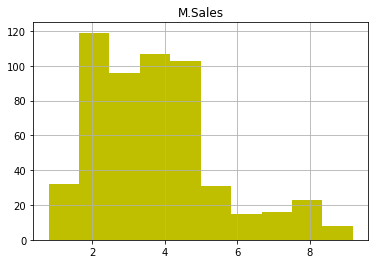
sns.countplot(["Sales Trend"],color**=**'r') plt.grid()



In [75]:

df.hist(["M.Sales"],color**=**'y')

Out[75]: array([[<AxesSubplot:title={'center':'M.Sales'}>]], dtype=object)

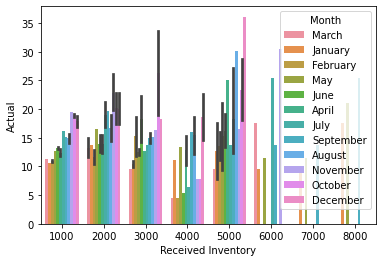


In [8]:

**import** seaborn **as** sns

sns.barplot(x**=**'Received Inventory', y**=**'Actual', hue**=**'Month', data**=**df, saturation**=**0.8)

Out[8]: <AxesSubplot:xlabel='Received Inventory', ylabel='Actual'>



In [ ]: