Advertising Sales Channel Prediction Project

Problem Definition:

When a company enters a market, the distribution strategy and channel it uses are keys to its success in the market, as well as market know-how and customer knowledge and understanding. Because an effective distribution strategy under efficient supply-chain management opens doors for attaining competitive advantage and strong brand equity in the market, it is a component of the marketing mix that cannot be ignored.

The distribution strategy and the channel design have to be right the first time. The case study of Sales channel includes the detailed study of TV, radio and newspaper channel. The predict the total sales generated from all the sales channel.

Data Analysis

Load Dataset: - The first step is to define the functions and classes we intend to use in this project.

I use Numpy and Pandas and seaborn library

```
import pandas as pd
```

```
import numpy as np
```

```
import seaborn as sns
```

Importing Dataset

```
df=pd.read_csv("Advertising.csv")
```

```
df
```

	Unnamed: 0	TV	radio	newspaper	sales
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5

Advertising Sales Channel Prediction Project

	Unnamed: 0	TV	radio	newspaper	sales
4	5	180.8	10.8	58.4	12.9
...	...	...	...	...	...
195	196	38.2	3.7	13.8	7.6
196	197	94.2	4.9	8.1	9.7
197	198	177.0	9.3	6.4	12.8
198	199	283.6	42.0	66.2	25.5
199	200	232.1	8.6	8.7	13.4

200 rows × 5 columns

The data set has total 200 rows and 5 columns.

In this dataset the target variable is “sales” and input variables are “TV”,”radio” and “newspaper”.

Now I checked is there any null value bye using following code

df.isnull().sum()

```
Unnamed: 0    0
TV            0
radio         0
newspaper     0
sales         0
dtype: int64
```

so there is no null value.

Now I checked what is the type of variables

df.dtypes

```
Unnamed: 0      int64
TV            float64
radio         float64
newspaper     float64
sales         float64
dtype: object
```

Advertising Sales Channel Prediction Project

EDA Concluding Remark

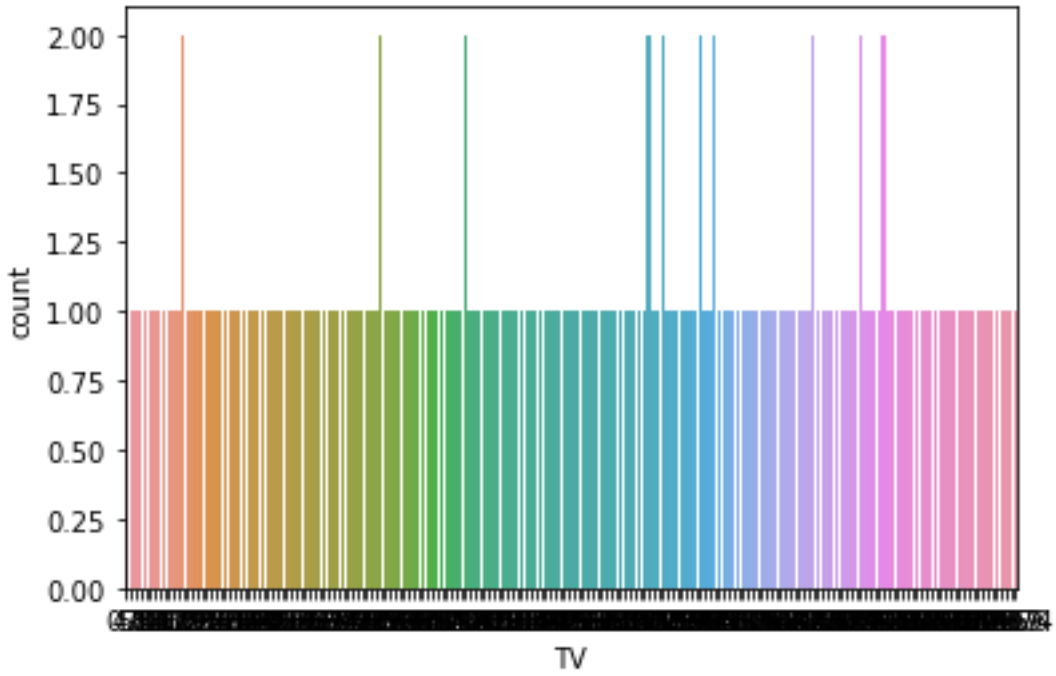
Visualization of the Data:

```
df_visualization=df[['TV','radio','newspaper','sales']].copy()

ax = sns.countplot(x="TV", data=df_visualization)

print(df_visualization["TV"].value_counts())
```

```
197.6    2
237.4    2
177.0    2
76.4     2
222.4    2
        ..
18.8     1
19.4     1
26.8     1
139.2    1
44.5     1
Name: TV, Length: 190, dtype: int64
```



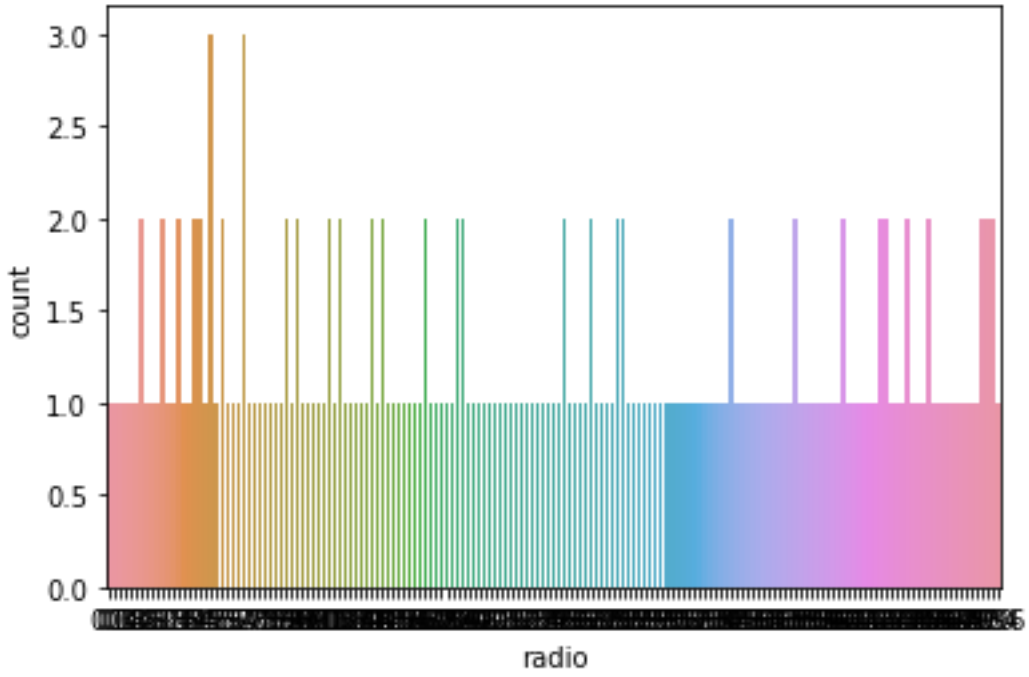

```
ax = sns.countplot(x="radio", data=df_visualization)

print(df_visualization["radio"].value_counts())
```

```
5.7      3
4.1      3
26.7     2
```

Advertising Sales Channel Prediction Project

```
18.1    2
43.0    2
       ..
37.6    1
26.9    1
40.6    1
34.3    1
24.0    1
Name: radio, Length: 167, dtype: int64
```

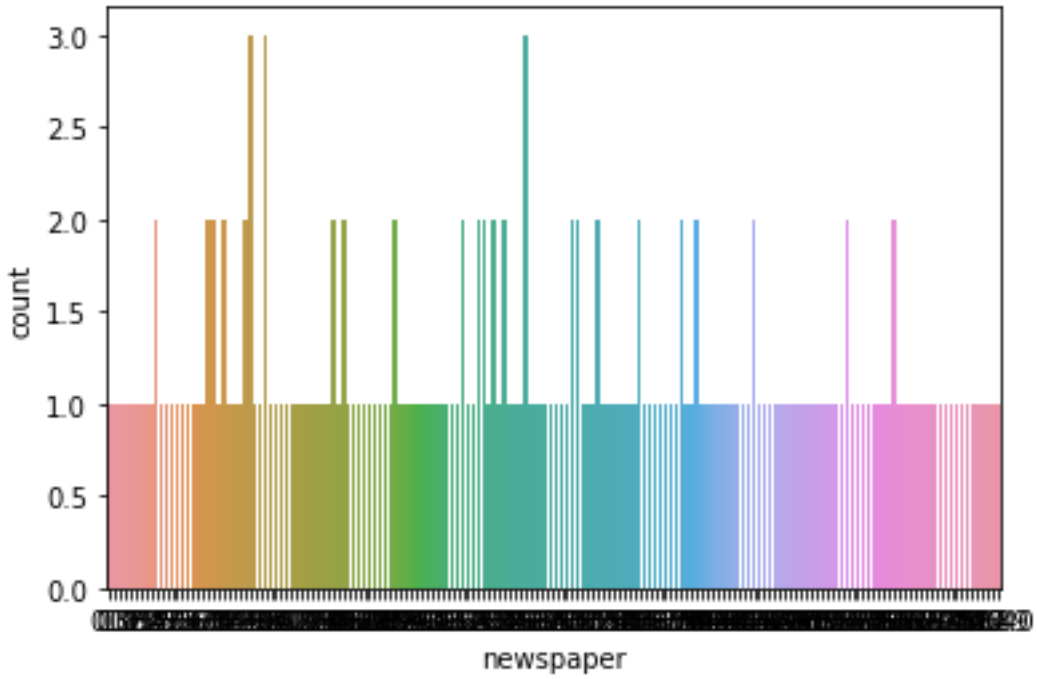



```
ax = sns.countplot(x="newspaper", data=df_visualization)

print(df_visualization["newspaper"].value_counts())
```

```
8.7     3
25.6    3
9.3     3
14.2    2
45.1    2
       ..
43.1    1
49.3    1
31.3    1
44.3    1
58.5    1
Name: newspaper, Length: 172, dtype: int64
```

Advertising Sales Channel Prediction Project

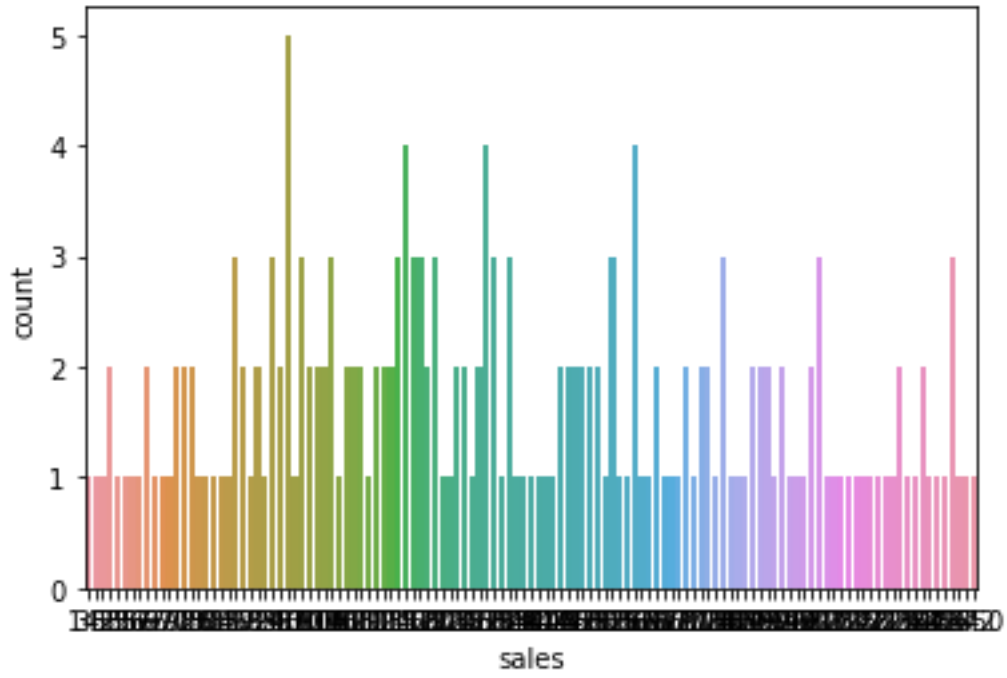



```
ax = sns.countplot(x="sales", data=df_visualization)
```

```
print(df_visualization["sales"].value_counts())
```

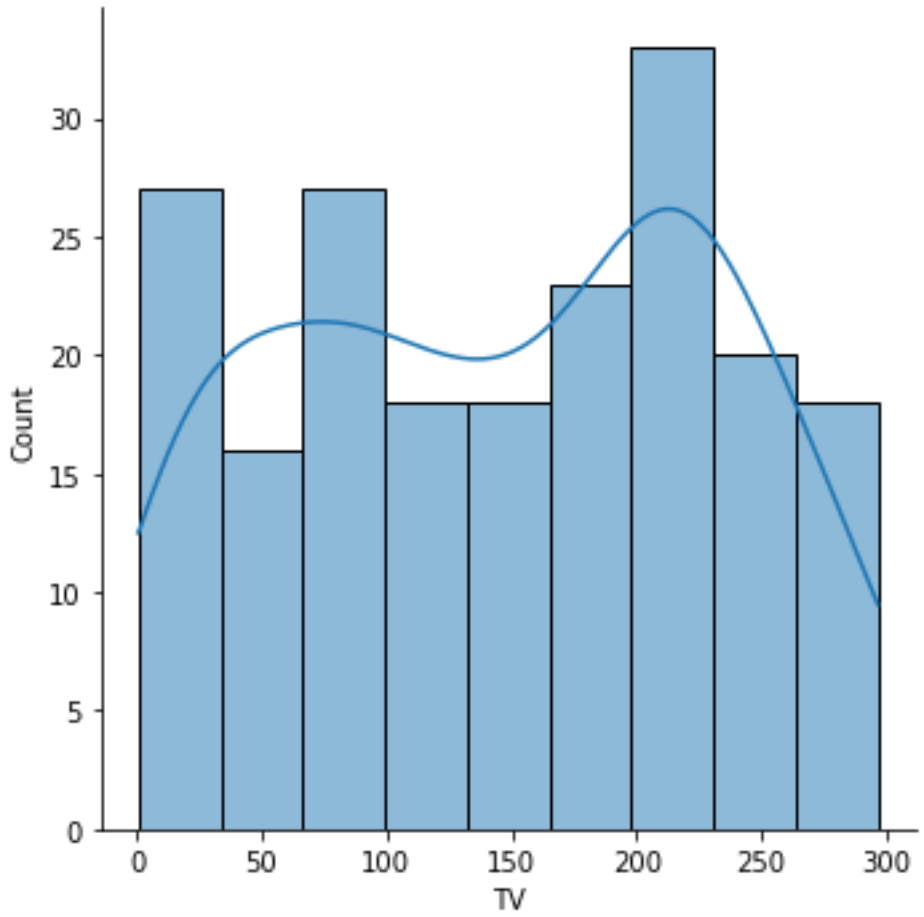
```
9.7     5
12.9    4
11.7    4
15.9    4
25.4    3
       ..
15.7    1
14.2    1
11.2    1
19.4    1
18.5    1
Name: sales, Length: 121, dtype: int64
```

Advertising Sales Channel Prediction Project




```
sns.displot(df_visualization['TV'], kde=True)
```

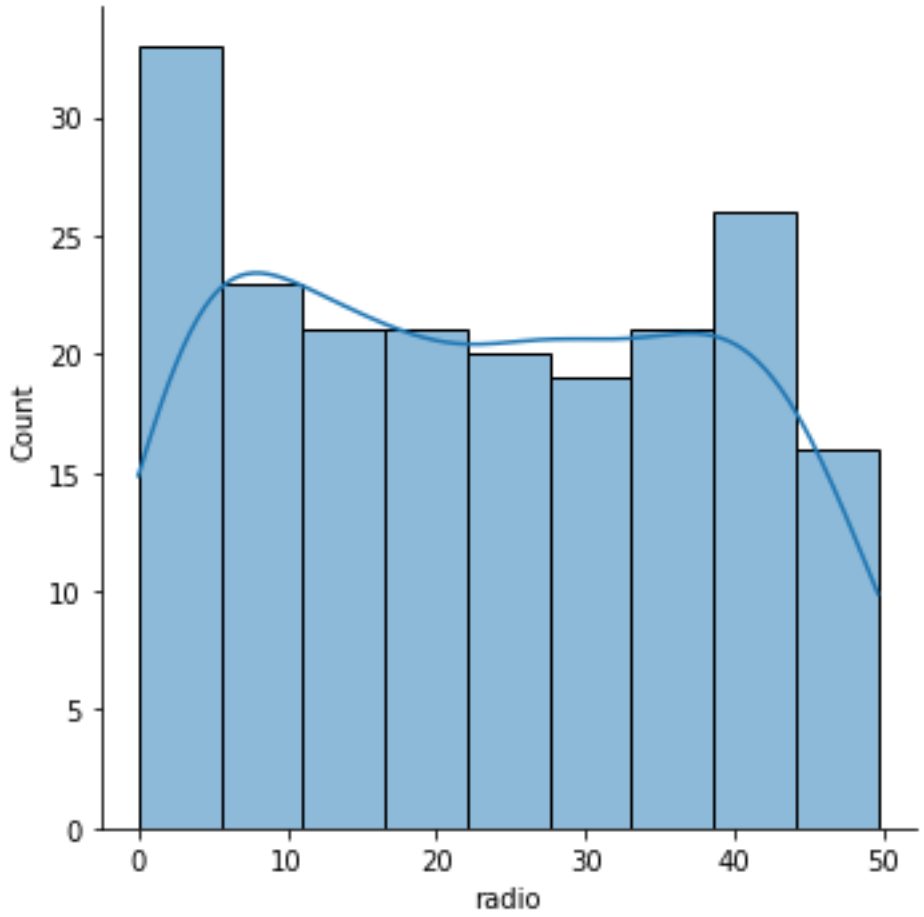
```
<seaborn.axisgrid.FacetGrid at 0x17435e50ee0>
```




```
sns.displot(df_visualization['radio'], kde=True)
```

```
<seaborn.axisgrid.FacetGrid at 0x17436040c10>
```

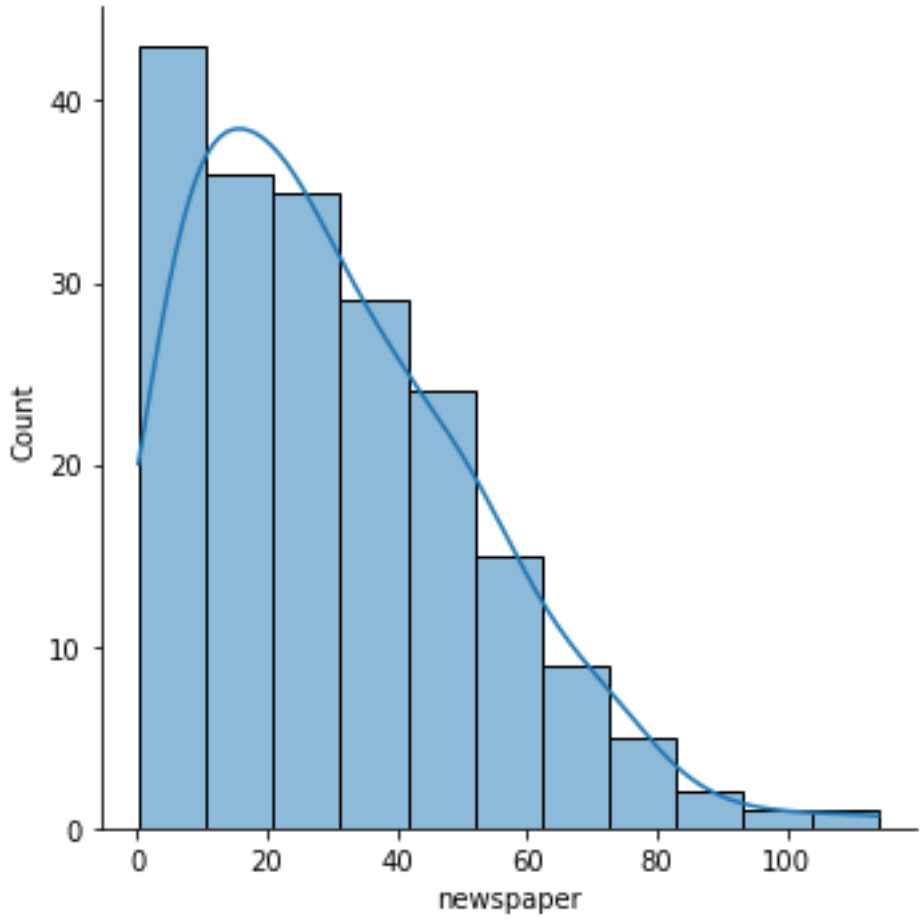
Advertising Sales Channel Prediction Project




```
sns.displot(df_visualization['newspaper'], kde=True)
```

```
<seaborn.axisgrid.FacetGrid at 0x174357fcca0>
```

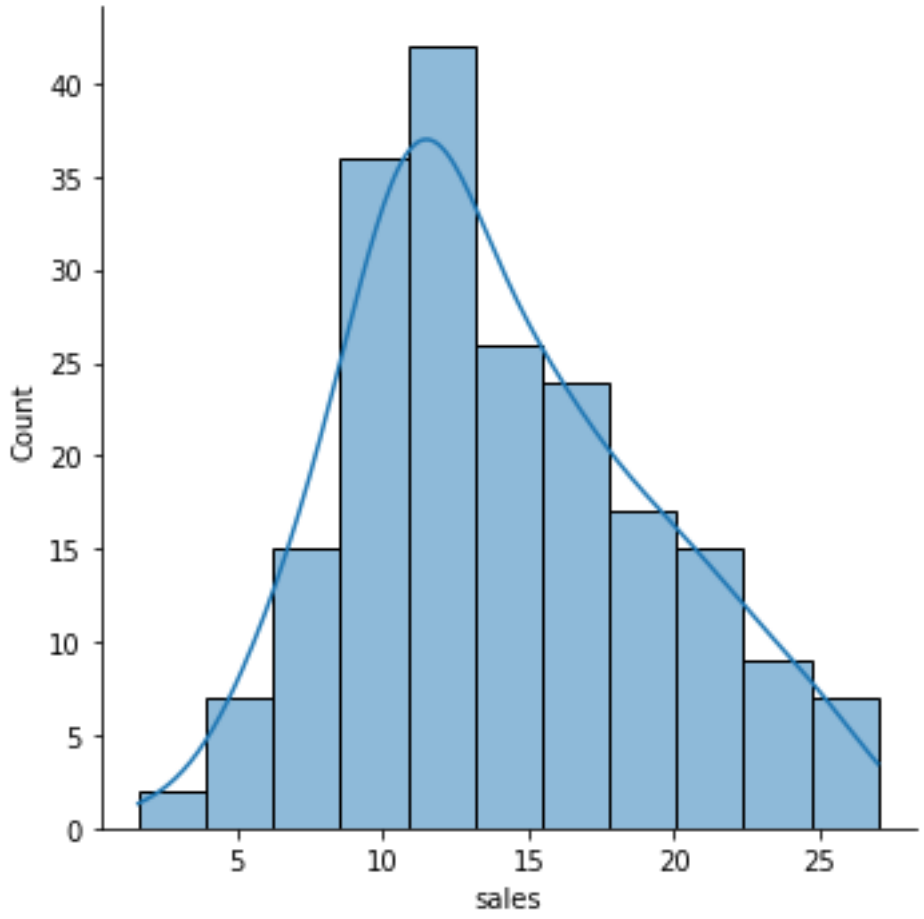
Advertising Sales Channel Prediction Project




```
sns.displot(df_visualization['sales'], kde=True)
```

```
<seaborn.axisgrid.FacetGrid at 0x174362aa6d0>
```

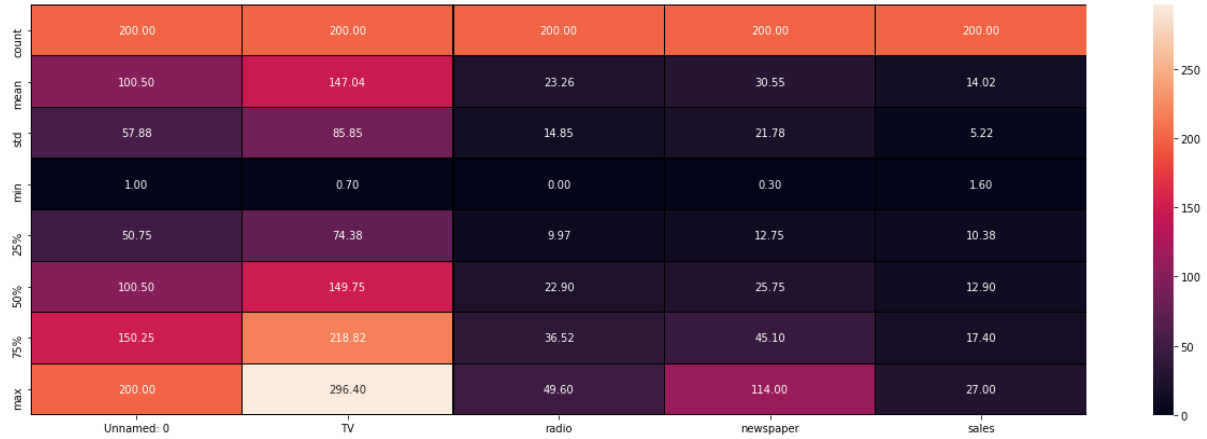
Advertising Sales Channel Prediction Project



Visualization of data set is done dis plot and count plot created

Heat map of Describe

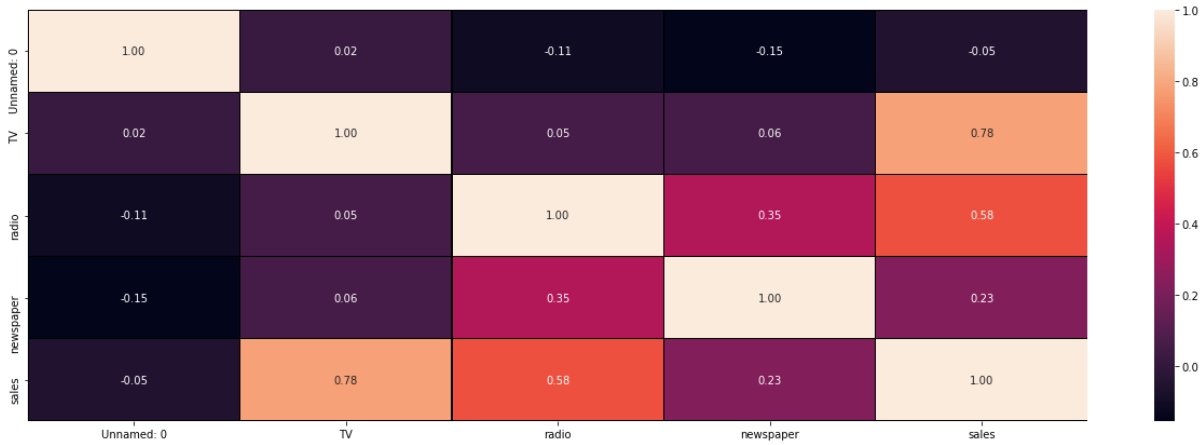
```
<AxesSubplot:>
```



Heat map corelation

```
<AxesSubplot:>
```

Advertising Sales Channel Prediction Project

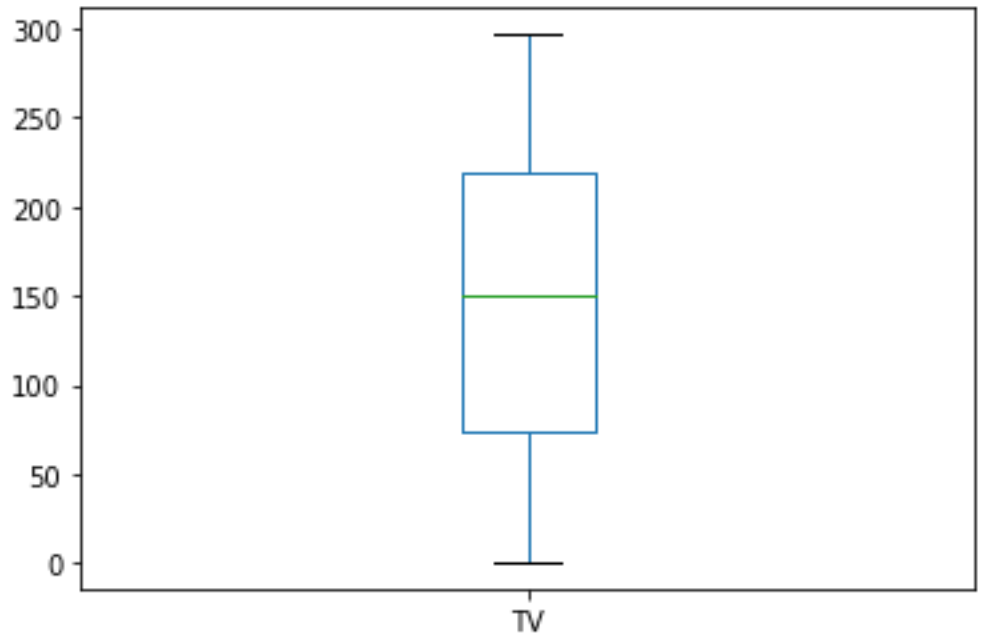


Pre-Processing Pipeline.

Outlier Checking

```
df['TV'].plot.box()
```

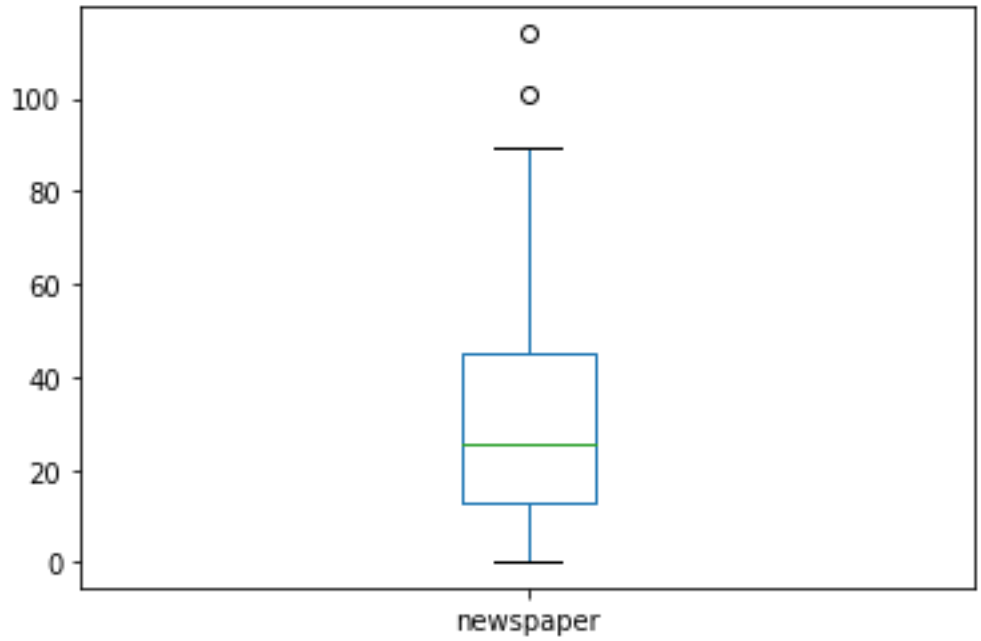
```
<AxesSubplot:>
```




```
df['newspaper'].plot.box()
```

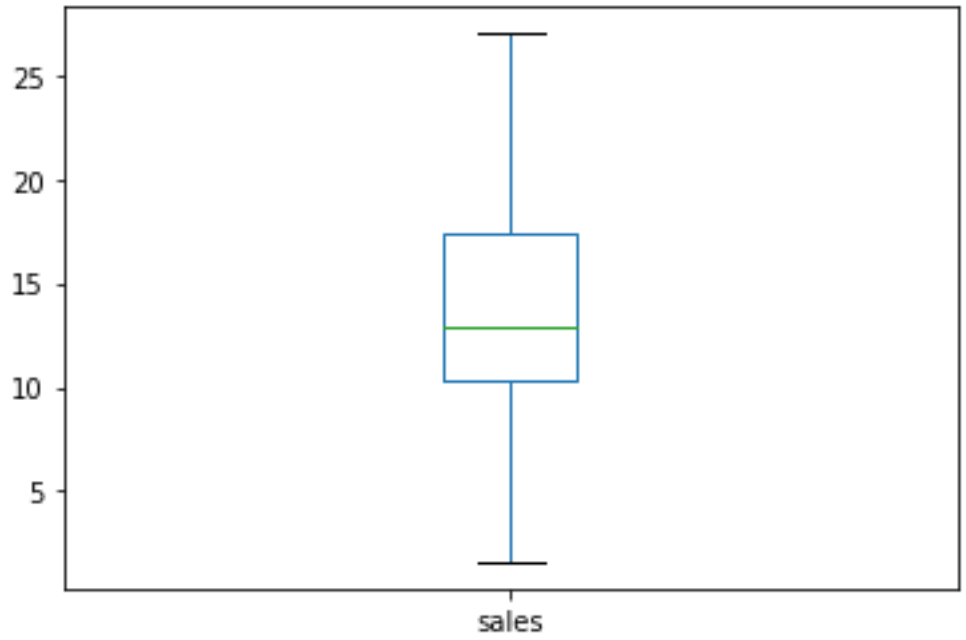
```
<AxesSubplot:>
```

Advertising Sales Channel Prediction Project



df['sales'].plot.box()

```
<AxesSubplot:>
```



Considering the outlier removal¶

```
from scipy.stats import zscore
import numpy as np
z=np.abs(zscore(df))
threshold=3
np.where(z>3)
```

Advertising Sales Channel Prediction Project

```
df_new_z=df[(z<3).all(axis=1)]
```

```
df_new_z
```

```
(array([ 16, 101], dtype=int64), array([3, 3], dtype=int64))
```

	Unnamed: 0	TV	radio	newspaper	sales
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9
...	...	...	...	...	...
195	196	38.2	3.7	13.8	7.6
196	197	94.2	4.9	8.1	9.7
197	198	177.0	9.3	6.4	12.8
198	199	283.6	42.0	66.2	25.5
199	200	232.1	8.6	8.7	13.4

198 rows × 5 columns

```
data_loss=((200-198)/200)*100
```

```
data_loss
```

1

Outlier is removed and as data loss is less than 1 we can go ahead

Separating the columns into features and target

```
features=df.drop("sales",axis=1)
```

```
target=df["sales"]
```

Target column is sales

Advertising Sales Channel Prediction Project

Building Machine Learning Models.

```
pred_test=lr.predict(features_test)

print(r2_score(target_test,pred_test))
```

```
0.8989530886907349
```

```
cv_score=cross_val_score(lr,features,target,cv=5)

cv_mean=cv_score.mean()

cv_mean
```

```
0.8827160786305974
```

```
parameters = {'alpha':[.0001, .001, .01, .1, 1, 10],'random_state':list(range(0,10))}

ls = Lasso()

clf = GridSearchCV(ls,parameters)

clf.fit(features_train,target_train)


print(clf.best_params_)
```

```
{'alpha': 1, 'random_state': 0}
```

```
ls = Lasso(alpha=1,random_state=0)

ls.fit(features_train,target_train)

ls.score(features_train,target_train)

pred_ls = ls.predict(features_test)


lss = r2_score(target_test,pred_ls)

lss
```

```
0.899358362447154
```

```
cv_score=cross_val_score(ls,features,target,cv=5)

cv_mean=cv_score.mean()

cv_mean
```

0.8844838672207616

Advertising Sales Channel Prediction Project

```
parameters = {'alpha':[.0001, .001, .01, .1, 1, 10],'random_state':list(range(0,10))}

rd = Ridge()

clf = GridSearchCV(rd,parameters)

clf.fit(features_train,target_train)


print(clf.best_params_)

{'alpha': 10, 'random_state': 0}
rd = Ridge(alpha=1,random_state=0)

rd.fit(features_train,target_train)

rd.score(features_train,target_train)

pred_rd = rd.predict(features_test)


rds = r2_score(target_test,pred_rd)

rds

0.8989527832410423
cv_score=cross_val_score(rd,features,target,cv=5)

cv_mean=cv_score.mean()

cv_mean

0.8827165526421137

We use linear regressor, lasso and ridge these 3 regressor techniques h
ere.
Now we are considering ensemble techniques.
```

Ensemble technique: ¶

```
rf= RandomForestRegressor(criterion="mse",max_features="auto")

rf.fit(features_train, target_train)

rf.score(features_train, target_train)

pred_decision = rf.predict(features_test)


rfs = r2_score(target_test,pred_decision)

print('R2 Score:',rfs*100)
```

Advertising Sales Channel Prediction Project

```
rfscore = cross_val_score(rf,features,target,cv=5)

rfc = rfscore.mean()

print('Cross Val Score:',rfc*100)
```

```
R2 Score: 98.02086937309616
Cross Val Score: 97.2329678861501
```

We are getting model r2 score and cross validation as 98.02% and 97.21% which shows our model is performing extremely well¶

Conclusion:¶

```
loaded_model = pickle.load(open('bb.pkl', 'rb'))

result = loaded_model.score(features_test, target_test)

print(result)
```

```
0.9802086937309616
conclusion=pd.DataFrame([loaded_model.predict(features_test)[:],pred_decision[:]],index=["Predicted","Orginal"])
```

	0	1	2	3	4	5	6	7	8	9	...	30	31	32	33	34	35	36	37	38	39	
Predicted	18.8833	16.7997	10.552	17.5555	15.6554	6.854	16.999	15.4388	12.7633	19.9332	...	...	11.1163	19.6665	21.4	11.9332	14.01	11.3116	7.0554	15.5594	22.3999	16.1777
Orginal	18.8833	16.7997	10.552	17.5555	15.6554	6.854	16.999	15.4388	12.7633	19.9332	...	...	11.1163	19.6665	21.4	11.9332	14.01	11.3116	7.0554	15.5594	22.3999	16.1777

2 rows × 40 columns

So the model is done.

Request is given to predict sales , now as per my model it is giving 98% accuracy , I have input in different media tv , radio and newspaper and all the given some deferent data regarding sales and I analysis all those things and I create a data frame which is showing up.

Advertising Sales Channel Prediction Project

This model is good to go.