In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df = pd.read_csv("Advertising.csv")
```

In [3]:
```
df
```

Out[3]:

	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

200 rows × 5 columns

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Unnamed: 0   200 non-null    int64  
 1   TV           200 non-null    float64
 2   Radio        200 non-null    float64
 3   Newspaper    200 non-null    float64
 4   Sales        200 non-null    float64
dtypes: float64(4), int64(1)
memory usage: 7.9 KB
```

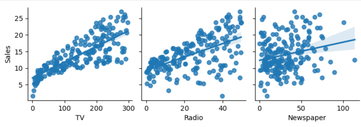
In [5]:
```
df.describe()
```

Out[5]:

	Unnamed: 0	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000	200.000000
mean	100.500000	147.042500	23.264000	30.554000	14.022500
std	57.879185	85.854236	14.846809	21.778621	5.217457
min	1.000000	0.700000	0.000000	0.300000	1.600000
25%	50.750000	74.375000	9.975000	12.750000	10.375000
50%	100.500000	149.750000	22.900000	25.750000	12.900000
75%	150.250000	218.825000	36.525000	45.100000	17.400000
max	200.000000	296.400000	49.600000	114.000000	27.000000

In [6]:
```
# MISSING VALUES

df = df.dropna(axis=0,how='any')
```

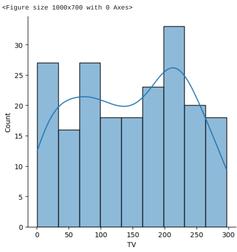
In [7]:
```
# DATA VISUALIZATION

sns.pairplot(df, x_vars=["TV", "Radio", "Newspaper"], y_vars="Sales", kind="reg")
plt.show()
```



In [8]:
```
plt.figure(figsize=(10,7))
sns.displot(df['TV'],kde=True)
```

Out[8]:
```
<seaborn.axisgrid.FacetGrid at 0x1687ba190>
```

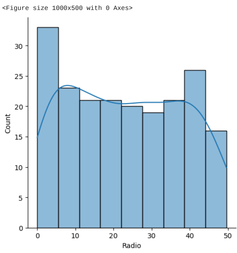
<Figure size 1000x700 with 0 Axes>



In [9]:
```
plt.figure(figsize=(10,5))
sns.displot(df['Radio'],kde=True)
```

Out[9]:
```
<seaborn.axisgrid.FacetGrid at 0x1688394d0>
```

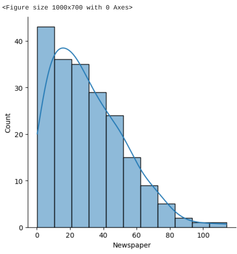
<Figure size 1000x500 with 0 Axes>



In [10]:
```
plt.figure(figsize=(10,7))
sns.displot(df['Newspaper'],kde=True)
```

Out[10]:
```
<seaborn.axisgrid.FacetGrid at 0x168b2b390>
```

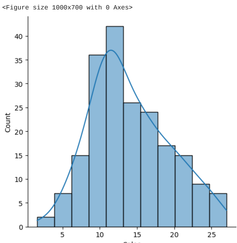
<Figure size 1000x700 with 0 Axes>



In [11]:
```
plt.figure(figsize=(10,7))
sns.displot(df['Sales'],kde=True)
```

Out[11]:
```
<seaborn.axisgrid.FacetGrid at 0x168bacc90>
```

<Figure size 1000x700 with 0 Axes>



In [12]:
```
# TRAIN_TEST_SPLIT

from sklearn.model_selection import train_test_split
X = df.drop(columns= ['Sales'])
Y = df['Sales']
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.25, random_state = 0)
```

In [13]:
```
print(X_train.shape)
print(X_test.shape)
print(Y_train.shape)
print(Y_test.shape)
```

```
(150, 4)
(50, 4)
(150,)
(50,)
```

In [14]:
```
# SALES LIST

y=df['Sales']
```

In [15]:
```
Y
```

Out[15]:
```
0      22.1
1      10.4
2       9.3
3      18.5
4      12.9
       ... 
195     7.6
196     9.7
197    12.8
198    25.5
199    13.4
Name: Sales, Length: 200, dtype: float64
```

In [16]:
```
# FIT MODEL

from sklearn.linear_model import LinearRegression
model= LinearRegression()
```

In [17]:
```
model.fit(X_train,Y_train)
```

Out[17]:
```
LinearRegression
LinearRegression()
```

In [18]:
```
# PREDICITIONS

y_predictions=model.predict(X_test)
```

In [19]:
```
y_predictions
```

Out[19]:
```
array([10.24207365,  7.29622961,  6.96911428, 24.13701112, 11.83126729,
        6.37501198, 13.27313357, 14.86981237, 11.17602518, 16.16675966,
       23.84627168,  9.876649  , 10.18031251, 15.29184583, 11.65932058,
       12.33256504, 18.76092755, 10.77301233, 16.20072024, 17.29059635,
       24.00755606,  9.39879619, 15.2550586 , 12.28045557,  5.78167907,
       15.19147624, 12.05439043, 20.75810519, 13.24067648,  9.28565297,
       13.42492203, 21.64276489, 17.97478881, 21.12653564,  6.86281034,
        5.97661237,  7.89802255, 13.27995489, 14.84476667,  6.2183205 ,
       12.1845611 ,  9.28457076, 15.22112327, 16.24213394, 16.9940899 ,
       13.42657395,  3.92670248, 12.45466279, 15.8015056 ,  8.59524665])
```

In [20]:
```
# MODEL ACCURACY FOR MEAN ABSOLUTE ERROR, ROOT MEAN SQUARE ERROR AND R-SQUARED

from sklearn import metrics
print('MAE:',metrics.mean_absolute_error(y_predictions,Y_test))
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_predictions,Y_test)))
print('R-Squared',metrics.r2_score(y_predictions,Y_test))
```

```
MAE: 1.338220395467825
RMSE: 2.026068613474676
R-Squared 0.8311834811337795
```

In [ ]:
```
                                    #Best Fit - LinearRegression
    #THUS, HERE SALES PREDICTION ADVERTISING.csv, WE HAVE USED LINEARREGRESSION MACHINE LEARNING MODEL
        #FOUND THE ACCURACY IN MEAN ABSOLUTE ERROR, ROOT MEAN SQUARE ERROR AND R-SQUARED
```