Here are some recommended packages, not all are required and depends on your solution.

In [2]:
```
# imports
import pandas as pd
import seaborn as sns
import statsmodels.formula.api as smf
from sklearn.linear_model import LinearRegression
from sklearn import metrics
from sklearn.model_selection import train_test_split
import numpy as np

# allow plots to appear directly in the notebook
%matplotlib inline
```

Questions

You are a consultant for a company that sells widgets. They have historical data on their sales on their investments in advertising in various media outlets, including TV, radio, and newspapers. On the basis of this data, how should they be spending their advertising money in the future?

Your analysis should answer the following questions:

Is there a relationship between ads and sales?

```
For TV and Radio there appears to be a relationship that as spending on ads goes up
so do sales.
For Newspaper there is hardly any correlation in the data.
```

How strong is that relationship?

```
TV's relationship is fairly strong (R-squared of .612), Radio's is very strong (R-s
quared of .332).
Newspapers is essentially non existant (R-squared of .052)
```

Which ad types contribute to sales?

```
TV contributes the most, then radio and finally newspapers.
```

What is the effect of each ad type on sales?

```
TV and radio ads will increase sales where a newspaper ad may or may not increase s
ales.
```

Given ad spending in a particular market, can sales be predicted?

```
If there is good historical data, sales could be predicted for TV ads but radio and
newspaper are more suspect.
```

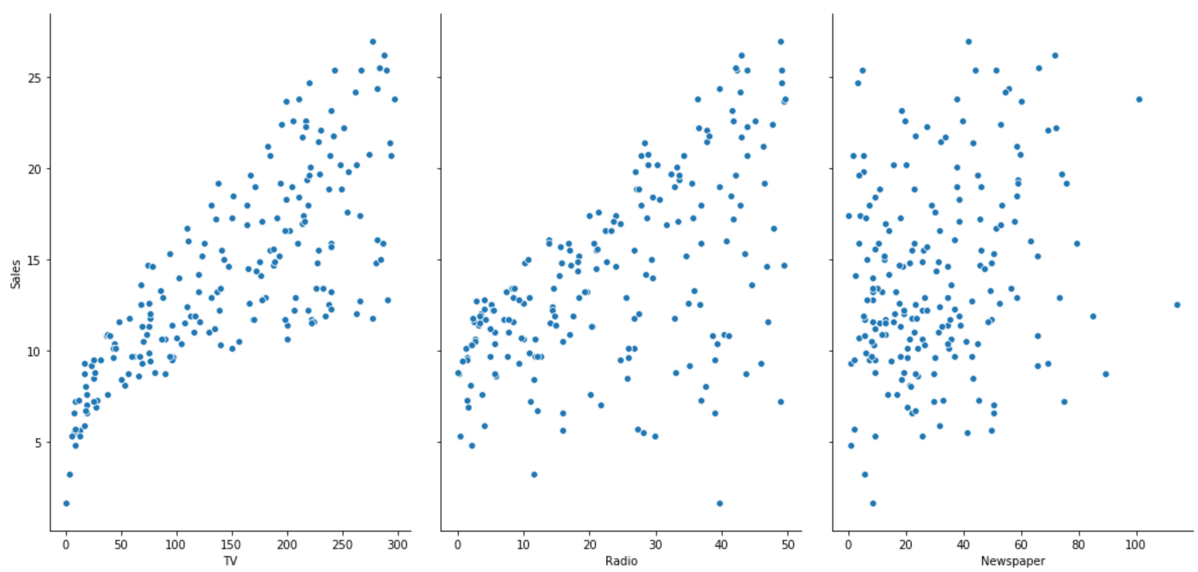
In [3]:
```
# read data into a DataFrame, this is money spent on different medias
data = pd.read_csv('https://raw.githubusercontent.com/lneisenman/isl/master/da
ta/Advertising.csv', index_col=0)
data.head()
```

Out[3]:

	TV	Radio	Newspaper	Sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9

In [4]:
```
# visualize the relationship between the features and the response using scatt
erplots
sns.pairplot(data, x_vars=['TV','Radio','Newspaper'], y_vars='Sales', height=7
, aspect=0.7)
```

Out[4]: <seaborn.axisgrid.PairGrid at 0x214b3079940>



In the lecture, we covered how to perform a linear regression model. We did not however explore how "good" this model is. The task below will have you identifying ways to evaluate a linear regression model.

Machine learning focuses on what the model predicts. If you would like to dive into the meaning of fit parameters within the model, other tools are available, including the Statsmodels Python package. Take some time to look at this [package (https://www.statsmodels.org/stable/regression.html)](https://www.statsmodels.org/stable/regression.html) and also an [example of evaluating a linear regression (https://www.statsmodels.org/stable/examples/notebooks/generated/gls.html)](https://www.statsmodels.org/stable/examples/notebooks/generated/gls.html).

Similar to Scikit-learn, one can calculate the intercept and coefficient for a linear fit for a set of data.

In [5]:
```
TV_model = smf.ols(formula='Sales ~ TV', data=data).fit()
TV_int = round(TV_model.params.Intercept,3)
TV_slope = round(TV_model.params.TV,3)
print(TV_model.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                  Sales   R-squared:                       0.61
2
Model:                            OLS   Adj. R-squared:                  0.61
0
Method:                 Least Squares   F-statistic:                     312.
1
Date:                Mon, 07 Oct 2019   Prob (F-statistic):           1.47e-4
2
Time:                        15:10:55   Log-Likelihood:                -519.0
5
No. Observations:                 200   AIC:                             104
2.
Df Residuals:                     198   BIC:                             104
9.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
=
                 coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
Intercept      7.0326      0.458     15.360      0.000       6.130       7.93
5
TV             0.0475      0.003     17.668      0.000       0.042       0.05
3
==============================================================================
=
Omnibus:                        0.531   Durbin-Watson:                   1.93
5
Prob(Omnibus):                  0.767   Jarque-Bera (JB):                0.66
9
Skew:                          -0.089   Prob(JB):                        0.71
6
Kurtosis:                       2.779   Cond. No.                         33
8.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [17]:
```
Radio_model = smf.ols(formula='Sales ~ Radio', data=data).fit()
Radio_int = round(Radio_model.params.Intercept,3)
Radio_slope = round(Radio_model.params.Radio,3)
print(Radio_model.summary())
```

```
                            OLS Regression Results
=============================================================================
=
Dep. Variable:                  Sales   R-squared:                       0.33
2
Model:                            OLS   Adj. R-squared:                  0.32
9
Method:                 Least Squares   F-statistic:                     98.4
2
Date:                Mon, 07 Oct 2019   Prob (F-statistic):           4.35e-1
9
Time:                        15:14:20   Log-Likelihood:               -573.3
4
No. Observations:                 200   AIC:                             115
1.
Df Residuals:                     198   BIC:                             115
7.
Df Model:                           1
Covariance Type:            nonrobust
=============================================================================
=
                 coef    std err          t      P>|t|      [0.025      0.97
5]
-----------------------------------------------------------------------------
-
Intercept      9.3116      0.563     16.542      0.000       8.202      10.42
2
Radio          0.2025      0.020      9.921      0.000       0.162       0.24
3
=============================================================================
=
Omnibus:                       19.358   Durbin-Watson:                   1.94
6
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               21.91
0
Skew:                          -0.764   Prob(JB):                     1.75e-0
5
Kurtosis:                       3.544   Cond. No.                         51.
4
=============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [16]:
```
Print_model = smf.ols(formula='Sales ~ Newspaper', data=data).fit()
Print_int = round(Print_model.params.Intercept,3)
Print_slope = round(Print_model.params.Newspaper,3)
print(Print_model.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                  Sales   R-squared:                       0.05
2
Model:                            OLS   Adj. R-squared:                  0.04
7
Method:                 Least Squares   F-statistic:                     10.8
9
Date:                Mon, 07 Oct 2019   Prob (F-statistic):            0.0011
5
Time:                        15:14:04   Log-Likelihood:                -608.3
4
No. Observations:                 200   AIC:                             122
1.
Df Residuals:                     198   BIC:                             122
7.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
=
                 coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
Intercept     12.3514      0.621     19.876      0.000      11.126      13.57
7
Newspaper      0.0547      0.017      3.300      0.001       0.022       0.08
7
==============================================================================
=
Omnibus:                        6.231   Durbin-Watson:                   1.98
3
Prob(Omnibus):                  0.044   Jarque-Bera (JB):                5.48
3
Skew:                           0.330   Prob(JB):                       0.064
5
Kurtosis:                       2.527   Cond. No.                         64.
7
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

A confidence interval can be used to describe a linear model. How would you calculate the confidence interval of this model and what does this confidence interval mean?

In [8]:
```
TV_CI = TV_model.conf_int()
TV_Diff_int = round((TV_int - TV_CI.iloc[0][0]),3)
TV_Diff_slope = round((TV_slope - TV_CI.iloc[1][0]),3)
print("TV Intercept Confidence Interval is " + str(TV_int) + " +/- " + str(TV_Diff_int) + ".")
print("TV Slope Confidence Interval is " + str(TV_slope) + " +/- " + str(TV_Diff_slope) + ".")
```

```
TV Intercept Confidence Interval is 7.033 +/- 0.903.
TV Slope Confidence Interval is 0.048 +/- 0.006.
```

In [9]:
```
R_CI = Radio_model.conf_int()
R_Diff_int = round((Radio_int - R_CI.iloc[0][0]),3)
R_Diff_slope = round((Radio_slope - R_CI.iloc[1][0]),3)
print("Radio Intercept Confidence Interval is " + str(Radio_int) + " +/- " + str(R_Diff_int) + ".")
print("Radio Slope Confidence Interval is " + str(Radio_slope) + " +/- " + str(R_Diff_slope) + ".")
```

```
Radio Intercept Confidence Interval is 9.312 +/- 1.11.
Radio Slope Confidence Interval is 0.202 +/- 0.04.
```

In [10]:
```
P_CI = Print_model.conf_int()
P_Diff_int = round((Print_int - P_CI.iloc[0][0]),3)
P_Diff_slope = round((Print_slope - P_CI.iloc[1][0]),3)
print("TV Intercept Confidence Interval is " + str(Print_int) + " +/- " + str(P_Diff_int) + ".")
print("TV Slope Confidence Interval is " + str(Print_slope) + " +/- " + str(P_Diff_slope) + ".")
```

```
TV Intercept Confidence Interval is 12.351 +/- 1.225.
TV Slope Confidence Interval is 0.055 +/- 0.033.
```

Other metrics that are used to describe the appropriateness of a model is a p-value. How would you calculate the p-value and r-squared values of the model? What do these values mean?

In [11]:
```
TV_pval = TV_model.pvalues
R_pval = Radio_model.pvalues
P_pval = Print_model.pvalues
print(TV_pval)
print(R_pval)
print(P_pval)
print()
print('Based on the very low P-Value, we can conlude that our models are accur
ate')
```

```
Intercept    1.406300e-35
TV           1.467390e-42
dtype: float64
Intercept    3.561071e-39
Radio        4.354966e-19
dtype: float64
Intercept    4.713507e-49
Newspaper    1.148196e-03
dtype: float64

Based on the very low P-Value, we can conlude that our models are accurate
```

In [12]:
```
TV_r2 = round(TV_model.rsquared,3)
R_r2 = round(Radio_model.rsquared,3)
P_r2 = round(Print_model.rsquared,3)
print(str(TV_r2) + " - this model appears to have okay correlation to the dat
a")
print(str(R_r2) + " - this model appears to have not very good correlation to
 the data")
print(str(P_r2) + " - this model appears to have poor correlation to the data"
)
```

```
0.612 - this model appears to have okay correlation to the data
0.332 - this model appears to have not very good correlation to the data
0.052 - this model appears to have poor correlation to the data
```