logistic_regression

April 13, 2020

1 linear regression

[1]:

```
# imports
import pandas as pd
import matplotlib.pyplot as plt

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

1.1 Example: Advertising Data

[2]:

```
# read data into a DataFrame
data = pd.read_csv('http://www-bcf.usc.edu/~gareth/ISL/Advertising.csv',␣
 ↪index_col=0)
data.head()
```

[2]:

```
      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
```

What are the **features**? - TV: advertising dollars spent on TV for a single product in a given market (in thousands of dollars) - Radio: advertising dollars spent on Radio - Newspaper: advertising dollars spent on Newspaper

What is the **response**? - Sales: sales of a single product in a given market (in thousands of widgets)

[3]:

```
# print the shape of the DataFrame
data.shape
```

[3]:

```
(200, 4)
```

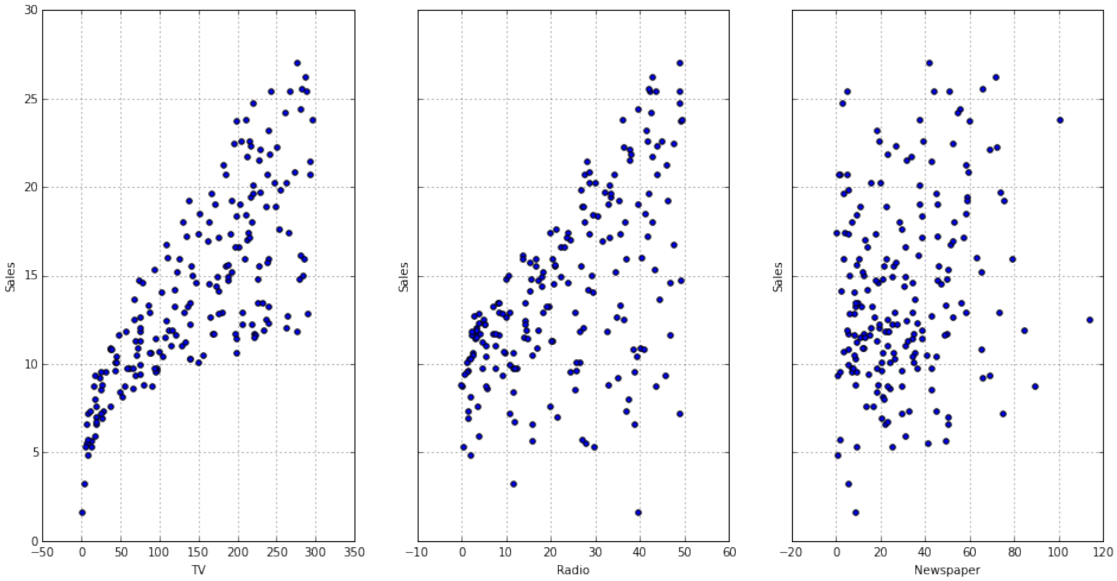
There are 200 **observations**, and thus 200 markets in the dataset.

[4]:

```
# visualize the relationship between the features and the response using␣
 ↪scatterplots
fig, axs = plt.subplots(1, 3, sharey=True)
data.plot(kind='scatter', x='TV', y='Sales', ax=axs[0], figsize=(16, 8))
data.plot(kind='scatter', x='Radio', y='Sales', ax=axs[1])
data.plot(kind='scatter', x='Newspaper', y='Sales', ax=axs[2])
```

[4]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xc1a3908>
```



Let's use **Statsmodels** to estimate the model coefficients for the advertising data:

[5]:

```
# this is the standard import if you're using "formula notation" (similar to R)
import statsmodels.formula.api as smf

# create a fitted model in one line
lm = smf.ols(formula='Sales ~ TV', data=data).fit()

# print the coefficients
lm.params
```

[5]:

```
Intercept    7.032594
TV           0.047537
dtype: float64
```

1.2 Interpreting Model Coefficients

1.3 Using the Model for Prediction

Let's say that there was a new market where the TV advertising spend was **$50,000**. What would we predict for the Sales in that market?

$$y = \beta_0 + \beta_1 x$$
$$y = 7.032594 + 0.047537 \times 50$$

[6]:
```
# manually calculate the prediction
7.032594 + 0.047537*50
```

[6]:
```
9.409444
```

Thus, we would predict Sales of **9,409 widgets** in that market.

Of course, we can also use Statsmodels to make the prediction:

[7]:
```
# you have to create a DataFrame since the Statsmodels formula interface␣
 ↪expects it
X_new = pd.DataFrame({'TV': [50]})
X_new.head()
```

[7]:
```
   TV
0  50
```

[8]:
```
# use the model to make predictions on a new value
lm.predict(X_new)
```

[8]:
```
array([ 9.40942557])
```

1.4 Plotting the Least Squares Line

Let's make predictions for the **smallest and largest observed values of x**, and then use the predicted values to plot the least squares line:

[9]:
```
# create a DataFrame with the minimum and maximum values of TV
X_new = pd.DataFrame({'TV': [data.TV.min(), data.TV.max()]})
X_new.head()
```

[9]:
```
      TV
0    0.7
1  296.4
```

[10]:

```
# make predictions for those x values and store them
preds = lm.predict(X_new)
preds
```

[10]:

```
array([  7.0658692 ,  21.12245377])
```

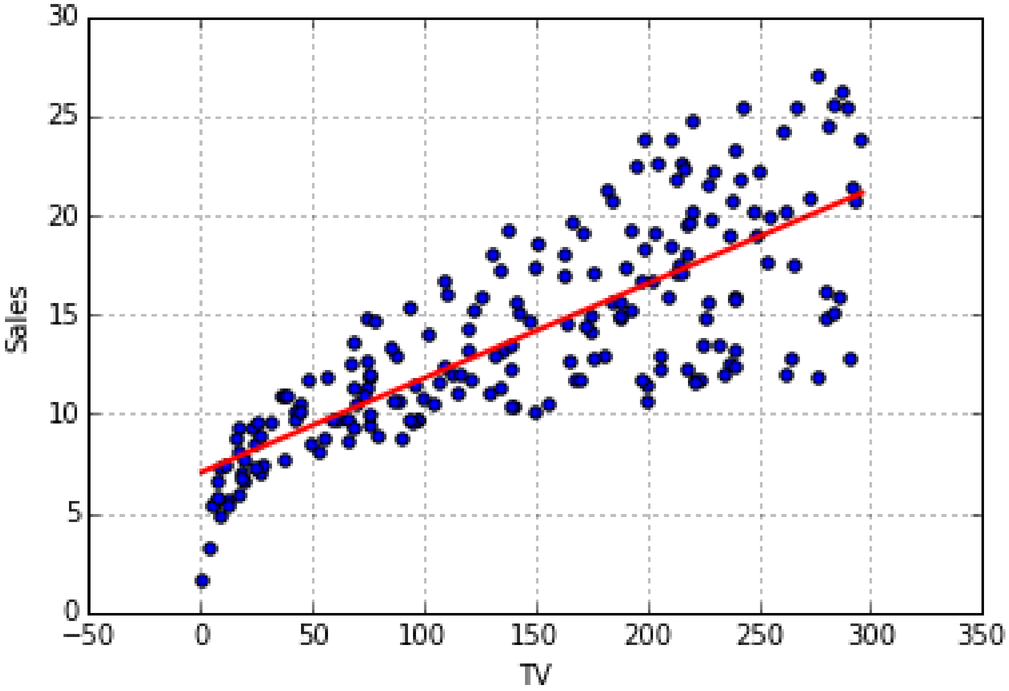
[11]:

```
# first, plot the observed data
data.plot(kind='scatter', x='TV', y='Sales')

# then, plot the least squares line
plt.plot(X_new, preds, c='red', linewidth=2)
```

[11]:

```
[<matplotlib.lines.Line2D at 0x14625128>]
```



1.5 Confidence in our Model

[12]:

```
# print the confidence intervals for the model coefficients
lm.conf_int()
```

[12]:

```
                  0         1
Intercept  6.129719  7.935468
TV         0.042231  0.052843
```