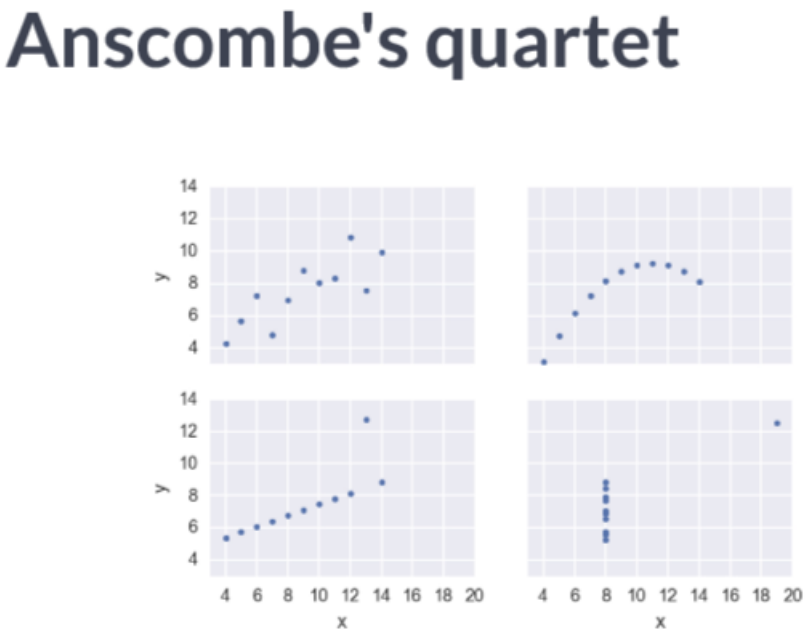
Anscombe's Quartet

The following analysis was excerpted from the Data Camp course "Statistical Thinking in Python"

https://en.wikipedia.org/wiki/Anscombe%27s_quartet



Linear regression on appropriate Anscombe data

For practice, perform a linear regression on the data set from Anscombe's quartet that is most reasonably interpreted with linear regression.

In [7]:

```
import numpy as np
import matplotlib.pyplot as plt
```

In [ ]:

```
x = np.array([10.,  8., 13.,  9., 11., 14.,  6.,  4., 12.,  7.,  5.])
y = np.array([ 8.04,  6.95,  7.58,  8.81,  8.33,  9.96,  7.24,  4.26, 10.84,
        4.82,  5.68])
```

In [18]:

```
# Perform linear regression: a, b
a, b = np.polyfit(x, y, 1)

# Print the slope and intercept
print(a, b)

# Generate theoretical x and y data: x_theor, y_theor
```

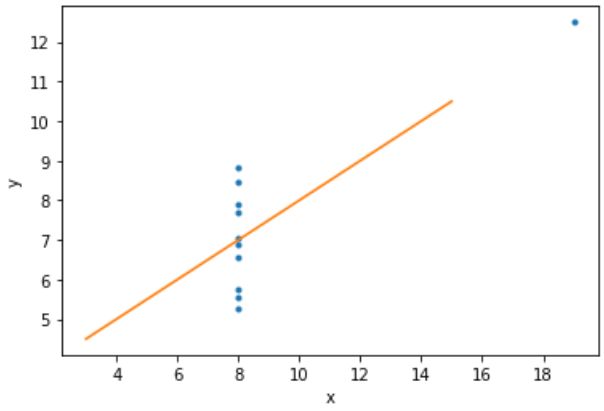
```
x_theor = np.array([3, 15])
y_theor = a * x_theor + b

# Plot the Anscombe data and theoretical line
_ = plt.plot(x, y, marker='.', linestyle='none')
_ = plt.plot(x_theor, y_theor)

# Label the axes
plt.xlabel('x')
plt.ylabel('y')

# Show the plot
plt.show()
```

```
0.49990909090909064 3.0017272727272735
```



Great work! You're getting to be a linear regression pro!

Linear regression on all Anscombe data

Now, to verify that all four of the Anscombe data sets have the same slope and intercept from a linear regression, you will compute the slope and intercept for each set. The data are stored in lists; `anscombe_x = [x1, x2, x3, x4]` and `anscombe_y = [y1, y2, y3, y4]`, where, for example, `x2` and `y2` are the x and y values for the second Anscombe data set.

In [16]:

```
anscombe_x = ([10.,  8., 13.,  9., 11., 14.,  6.,  4., 12.,  7.,  5.],
[10.,  8., 13.,  9., 11., 14.,  6.,  4., 12.,  7.,  5.],
[10.,  8., 13.,  9., 11., 14.,  6.,  4., 12.,  7.,  5.],
[ 8.,  8.,  8.,  8.,  8.,  8.,  8., 19.,  8.,  8.,  8.])


anscombe_y = ([ 8.04,  6.95,  7.58,  8.81,  8.33,  9.96,  7.24,  4.26, 10.84,4.82,  5.68],
[9.14, 8.14, 8.74, 8.77, 9.26, 8.1 , 6.13, 3.1 , 9.13, 7.26, 4.74],
[ 7.46,  6.77, 12.74,  7.11,  7.81,  8.84,  6.08,  5.39,  8.15,6.42,  5.73],
[ 6.58,  5.76,  7.71,  8.84,  8.47,  7.04,  5.25, 12.5 ,  5.56,7.91,  6.89])
```

In [17]:

```
# Iterate through x,y pairs
```

```
for x, y in zip(anscombe_x, anscombe_y):
    # Compute the slope and intercept: a, b
    a, b = np.polyfit(x, y, 1)

    # Print the result
    print('slope:', a, 'intercept:', b)
```

```
slope: 0.5000909090909095 intercept: 3.0000909090909076
slope: 0.5000000000000004 intercept: 3.000909090909089
slope: 0.4997272727272731 intercept: 3.0024545454545453
slope: 0.4999090909090064 intercept: 3.0017272727272735
```