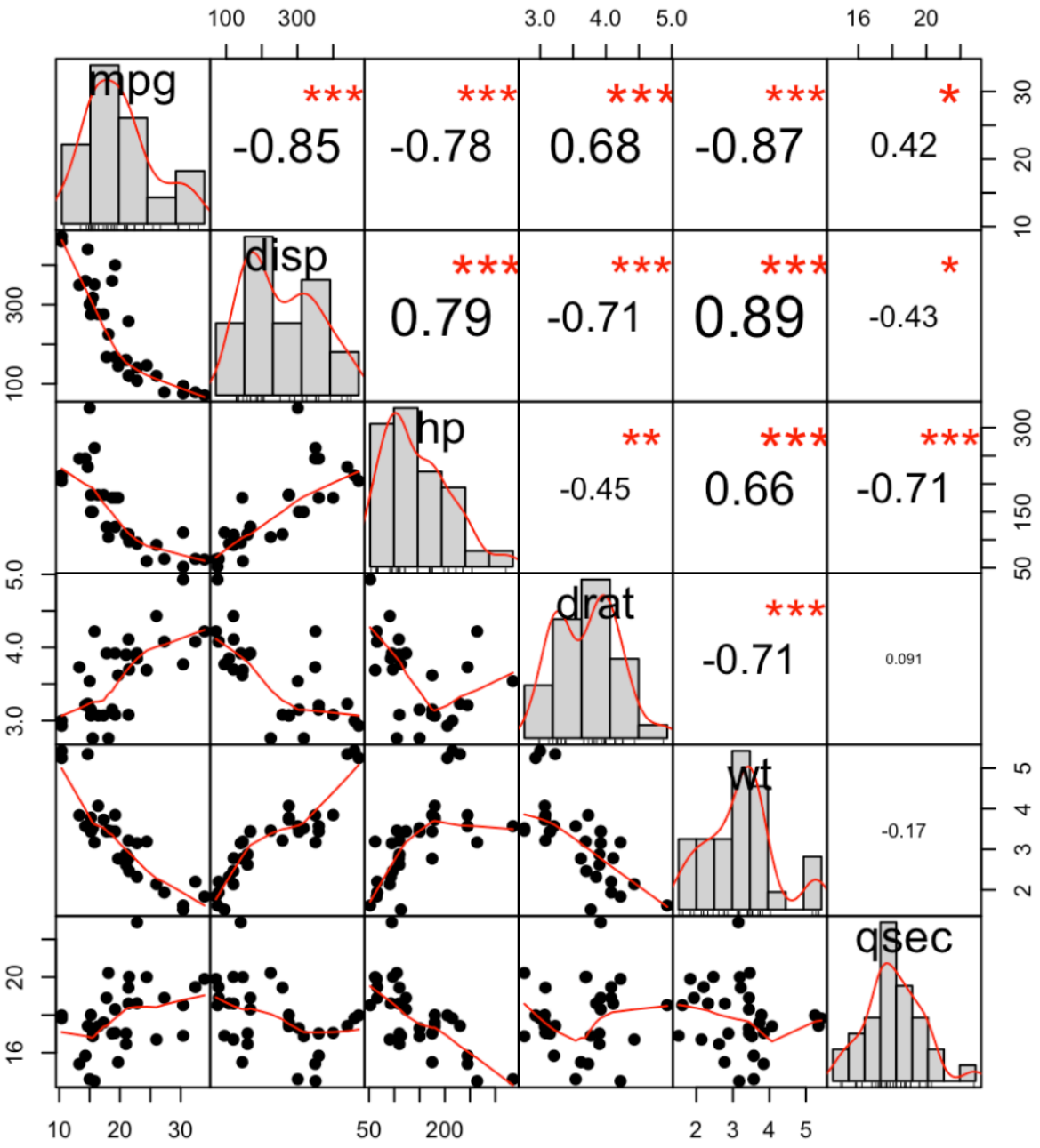
Multi-axis figure homework

In this homework, you'll make a simulated data set with a relationship between x and y, then make a "correlation plot" of the data.

A correlation plot of data values is a matrix of subplots with one row and one column corresponding to each variable. The main diagonal has histograms of the variable corresponding to that row and column. In one type of correlation plot, the lower triangle of the matrix has plots of the variables against one another, and the upper triangles shows the correlations between the variables.

Like in the image in corrPlot.png



, which shows an example.

Make your data

Make some y vs. x data like in the last homework. It can be straight line based (like y = m*x * b * random noise), or it can be something more exotic (like y = x^3 + random noise).

Play around with the amount of noise, the number of data points, etc., plotting as you go until you have data that you are happy with.

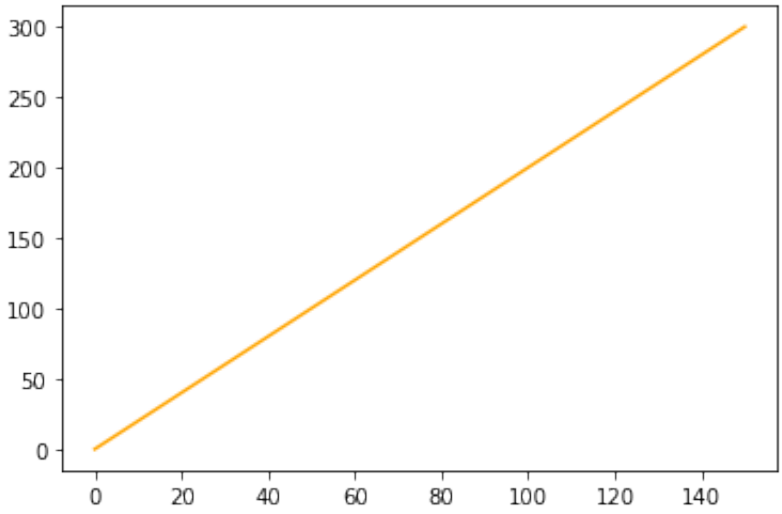
Numpy has a handy function to compute the ***corr***elation ***coef***icient.

In [1]:
```
# Library
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
# X-value
x = np.linspace(0, 150)
y = 0 + 2*x              # Y-value (slope = 2)

# Graph
plt.plot(x,y, color = 'orange')
```

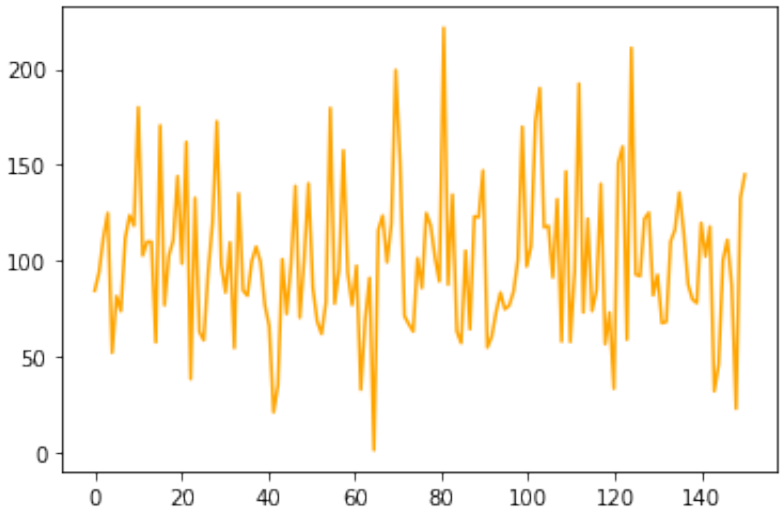
Out[2]:
```
[<matplotlib.lines.Line2D at 0x7fae105d3cd0>]
```



In [3]:
```
# Random Y Noise
x_noise = np.linspace(0, 150, 150)
z_noise = (np.random.randn(len(x_noise)) * 0.4)* 100
y_noise = (((np.cos(x_noise)) + z_noise))  + 100

# Both graph together
plt.plot(x_noise,y_noise, color = 'orange')
```

Out[3]:
```
[<matplotlib.lines.Line2D at 0x7fae10708430>]
```



Make your figure

Because you have 2 variables, your figure will be a 2x2 matrix of subplots, with two histograms down the main diagonal, a scatterplot in the lower left, and a correlation value in the plot in the upper right.

But feel free to make it your own. For example, the upper right cell could have a faint re-plot of the data with the correlation value shown on top of it.

In [4]:
```
# Add correlation and scatter plot
```

In [5]:
```
cor = np.corrcoef(x_noise,y_noise)[1,0]
cor.round(2)
```

Out[5]:
```
0.01
```

In [6]:

```
fig = plt.figure()
fig.set_facecolor('#EDDAC4')

# Histrogram with x noise
yhist_ax = plt.subplot(2,2,1)
plt.hist(x, bins = 20, color = '#FC0000', alpha = 0.4)
plt.title('Histrogram with X Noise')

# Correlation
plt.subplot(2,2,2)
plt.scatter(x_noise,z_noise, color = '#6ED2F5', alpha = 0.4, s = 3
plt.plot(cor)
plt.text(80, 50, cor.round(2))
plt.text(5,50,'Correlation = ')
plt.title('Correlation between X and Y Noise')

# Scatterplot
plt.subplot(2,2,3)
plt.scatter(x_noise,z_noise, color = '#F1FC00', alpha = 0.6, s = 3
plt.title('Scatterplot between Y and X Noise')

# Histrogram with Z noise
yhist_ax = plt.subplot(2,2,4)
plt.hist(y_noise, bins = 20, color = '#39FC00', alpha = 0.4)
plt.title('Histrogram with Y Noise')

plt.tight_layout()

```