



INTRODUCTION TO LINEAR MODELING IN PYTHON

# Introduction to the Course

Jason Vestuto  
Data Scientist



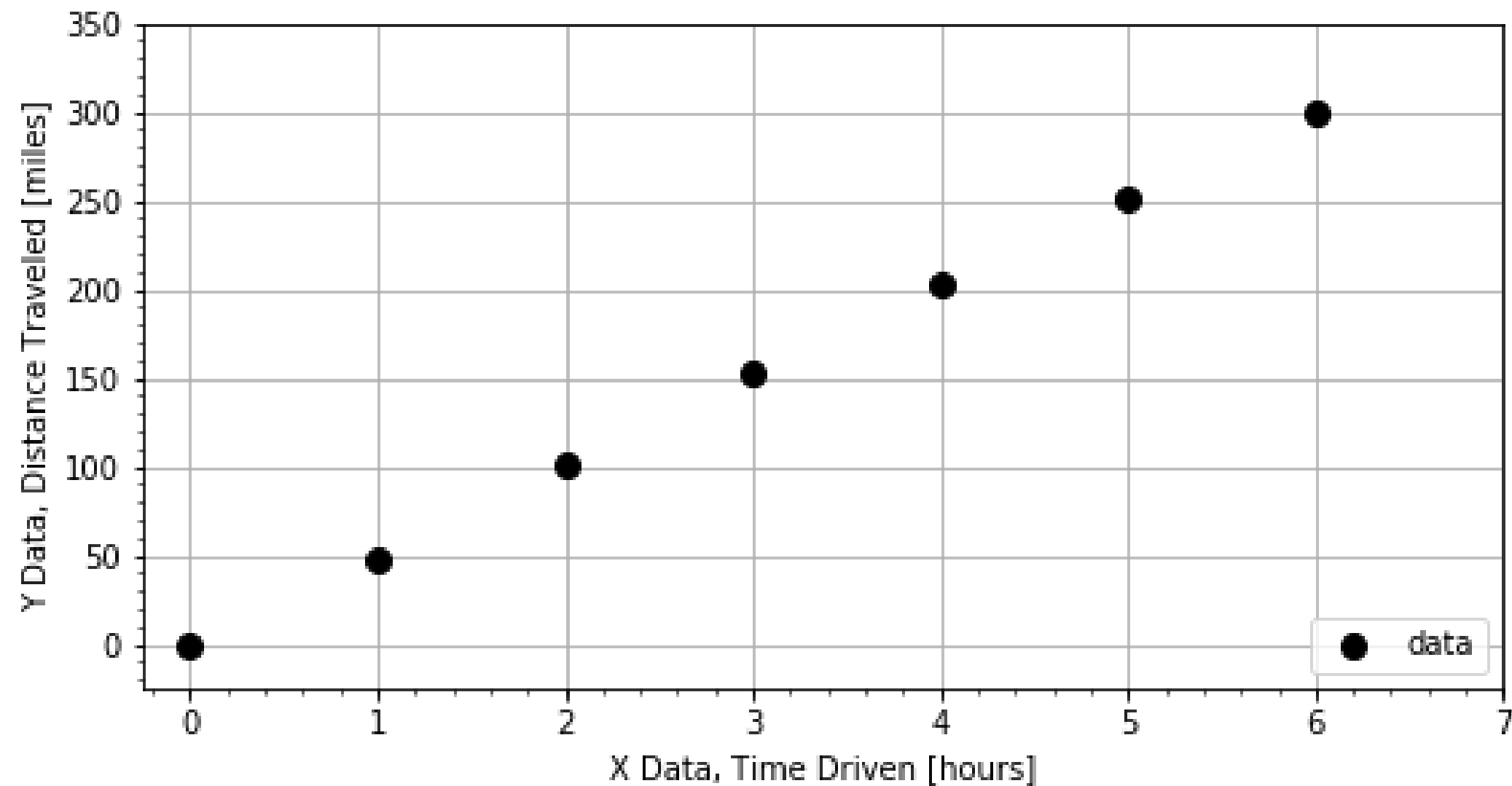
# Introduction to Chapter 1

## Chapter Roadmap:

- Motivating Examples
- Data Visualization
- Descriptive Statistics



# Example Trip Data





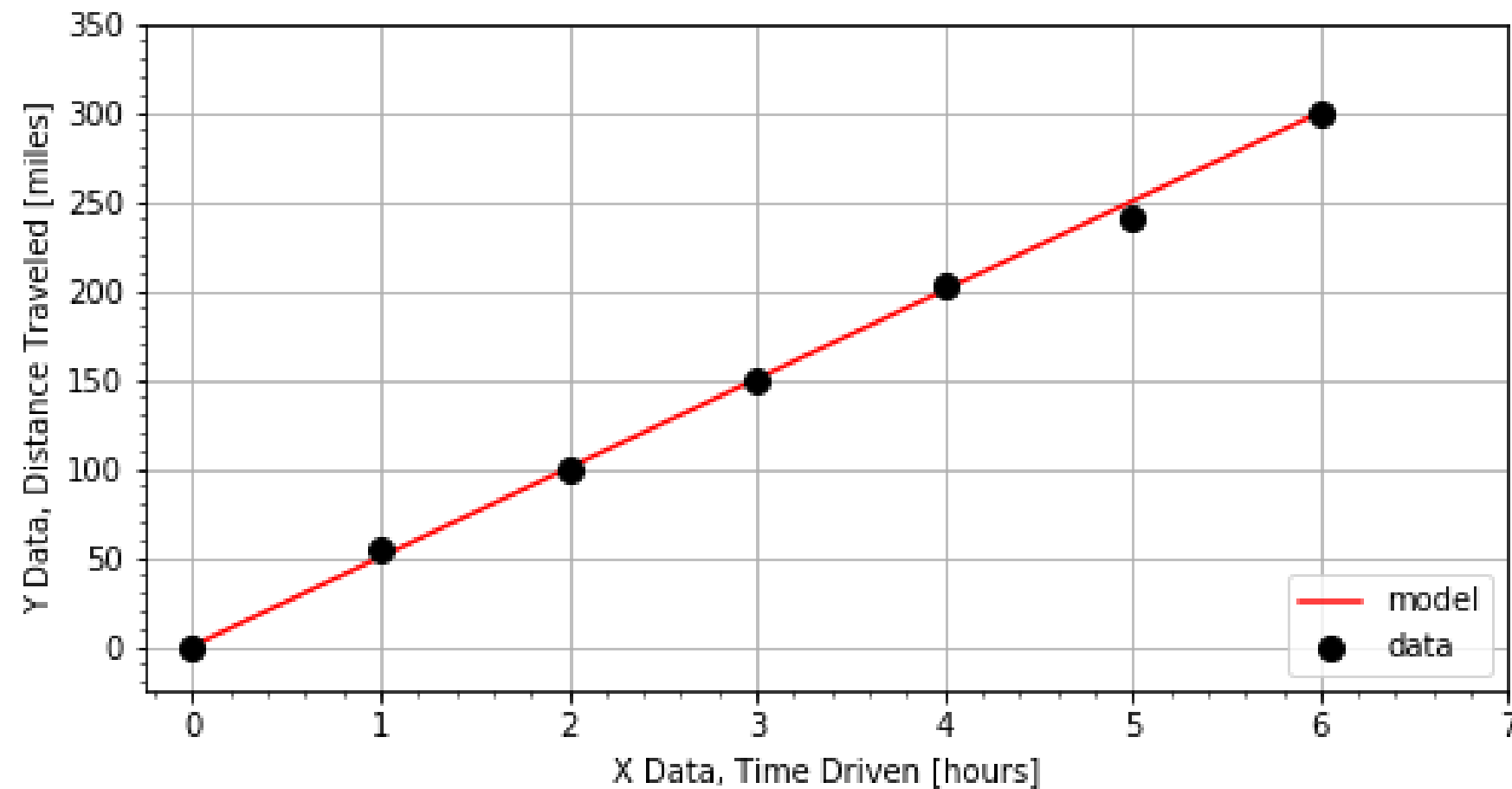
# Models as Descriptions

```
# range of y data  
y_range = np.max(y) - np.min(y) = 300 - 0 = 300 # miles
```

```
# range of x data  
x_range = np.max(x) - np.min(x) = 6 - 0 = 6 # hours
```

```
# estimating the speed  
mph = y_range / x_range = 300 / 6 = 50
```

# Visualizing a Model





# Model Predictions

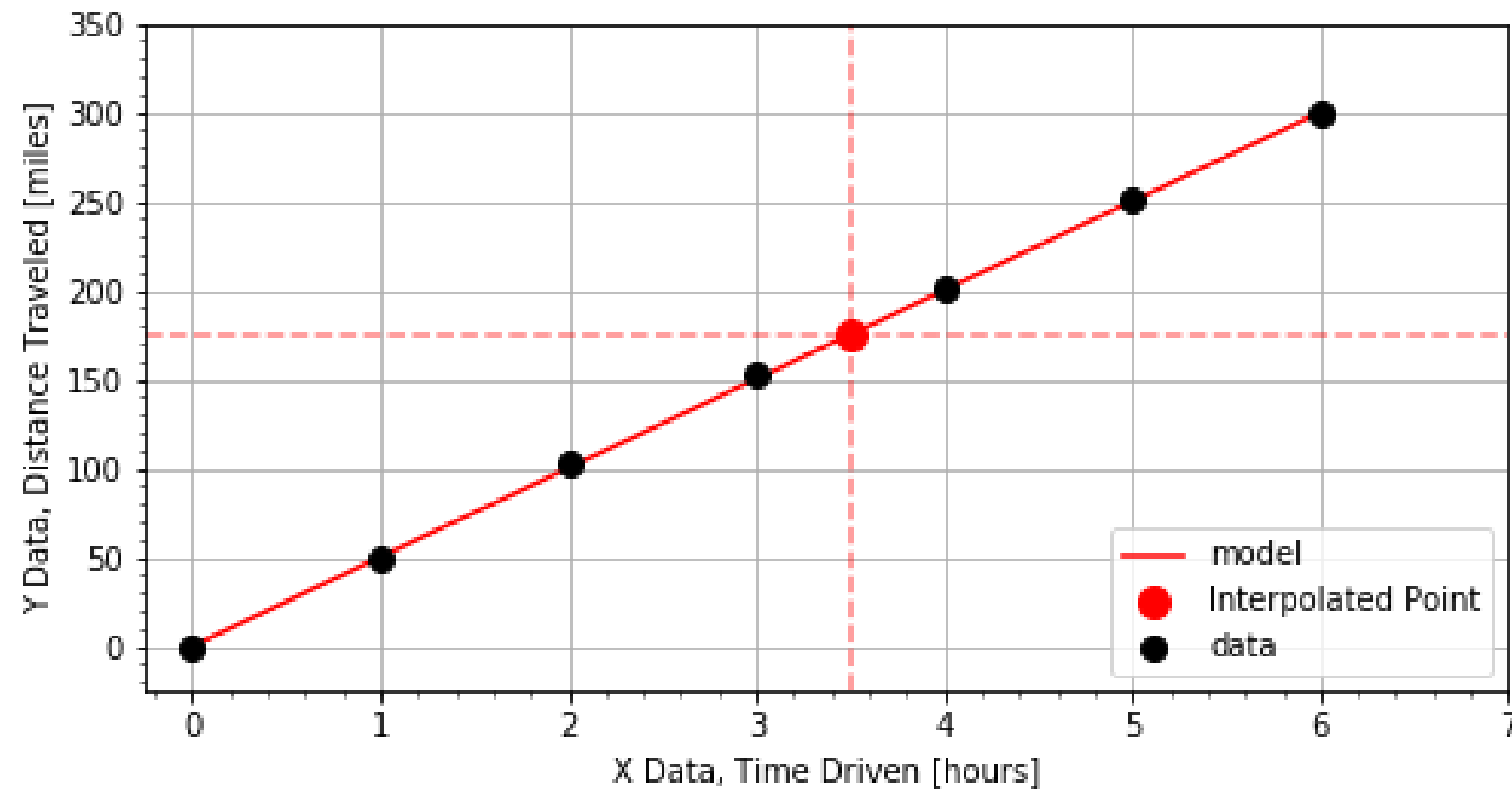
```
# model as python expression  
miles = 50*hours
```

```
# model predicts distance is 300 miles at 6 hours  
time = 6  
distance = 50 * time = 50 * 6 = 300
```

```
def model(time):  
    return 50*time  
  
predicted_distance = model(time=10)
```

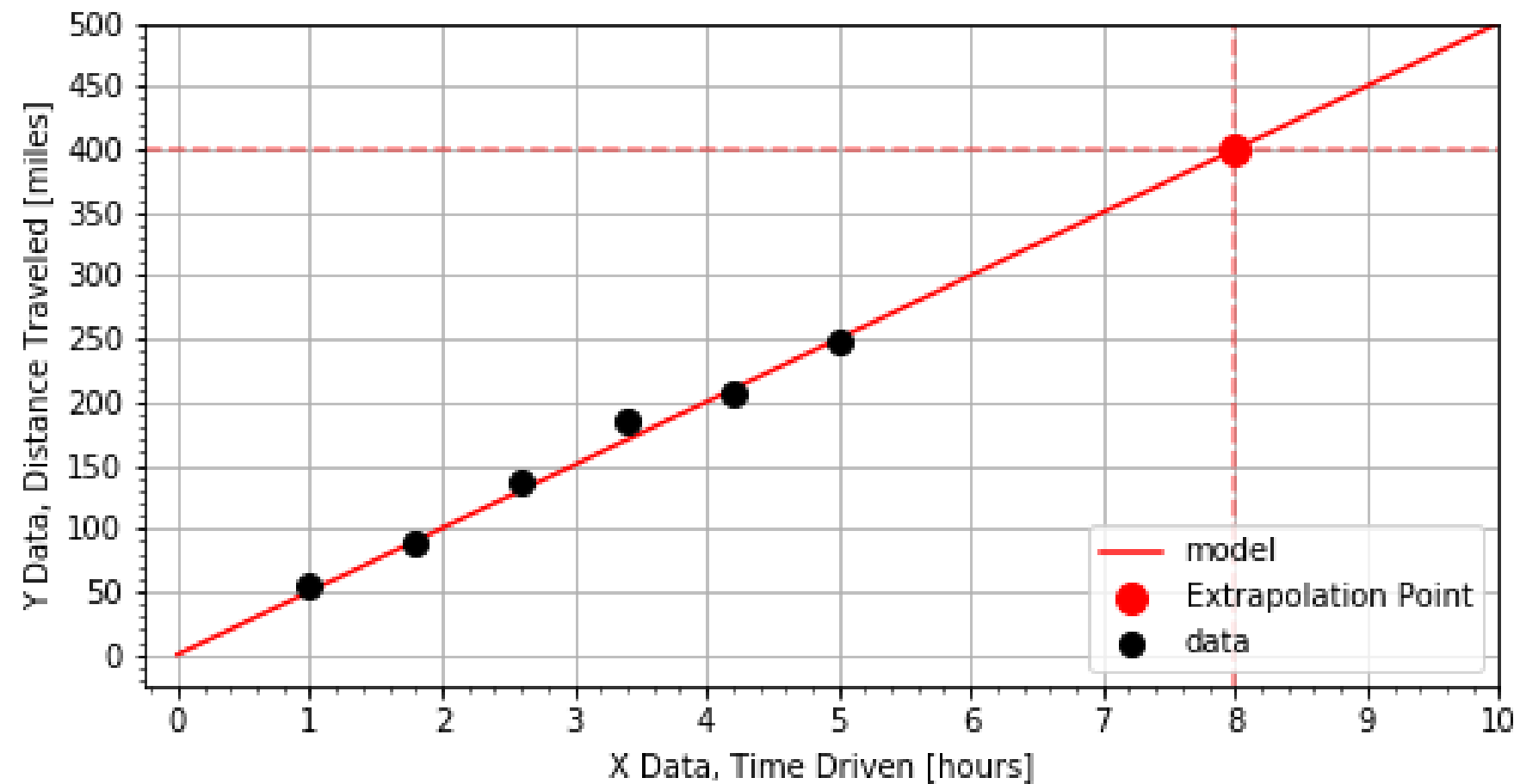


# Interpolation





# Extrapolation







## INTRODUCTION TO LINEAR MODELING IN PYTHON

**Let's practice!**



INTRODUCTION TO LINEAR MODELING IN PYTHON

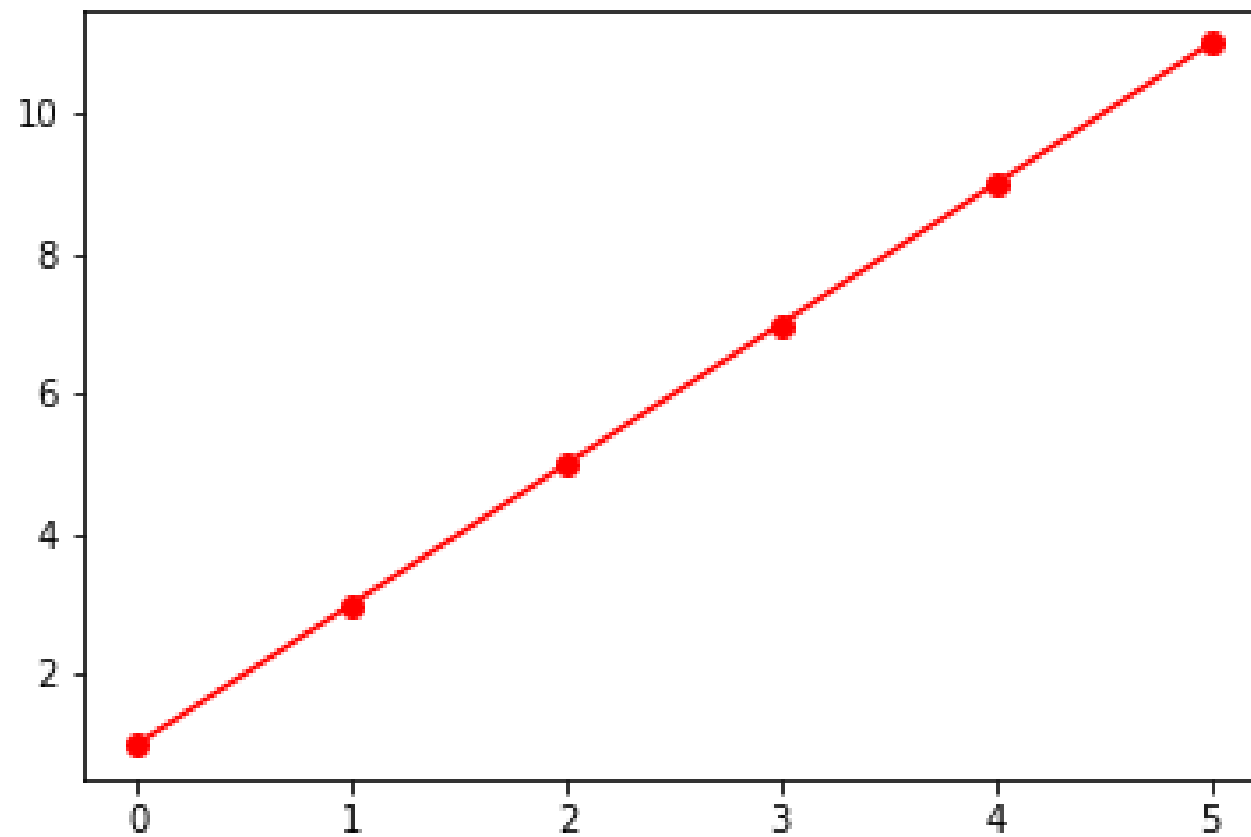
# Visualizing Linear Relationships

Jason Vestuto  
Data Scientist



# Quick Plots

```
import matplotlib.pyplot as plt  
  
plt.plot(x, y, 'r-o')  
  
plt.show()
```





# Object Interface

```
# Import the pyplot module
import matplotlib.pyplot as plt
```

```
# Create figure and axis objects
fig, axis = plt.subplots()
```

```
# prepare initial style options
options = dict(marker='o', color='blue')
```

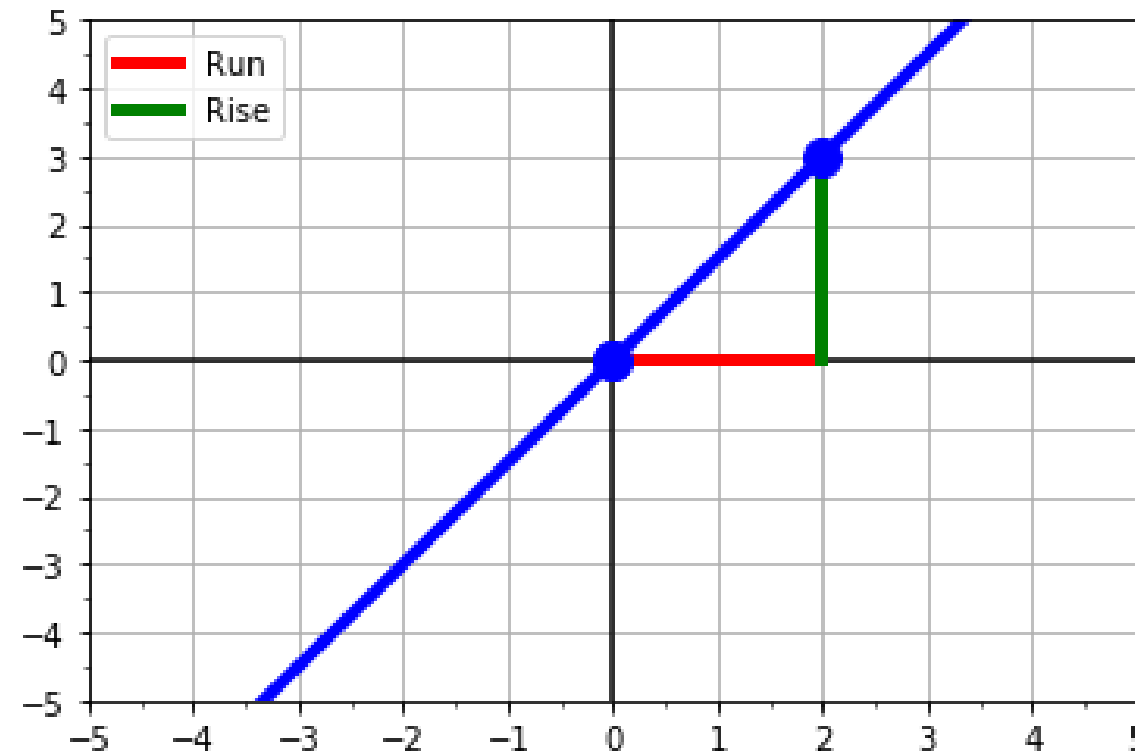
```
# call the plot method on the axis object
line = axis.plot(x, y, **options)
```

```
# modify the axis object with set methods
_ = axis.set_ylabel('Times')
_ = axis.set_xlabel('Distances')
```

```
# display figure
plt.show()
```

# Visualizing Linear Data

- two points:
  - $(x_1, y_1) = (0, 0)$
  - $(x_2, y_2) = (2, 3)$
- change in x and y:
  - $dy = (y_2 - y_1) = 3 - 0$
  - $dx = (x_2 - x_1) = 2 - 0$
- slope = rise-over-run
  - $\text{slope} = dy/dx = 3/2$
- intercept:
  - when  $x=0$ ,  $y=?$





## INTRODUCTION TO LINEAR MODELING IN PYTHON

**Let's practice!**



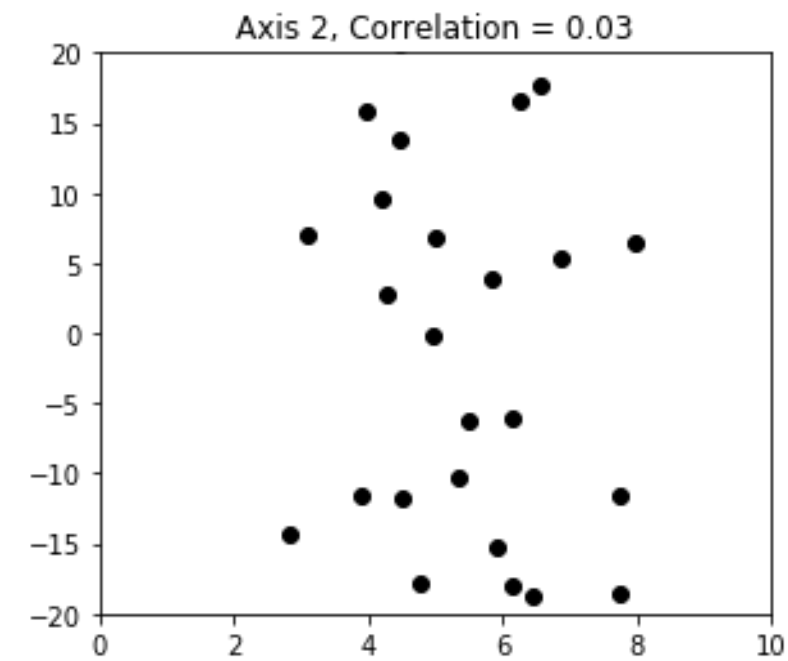
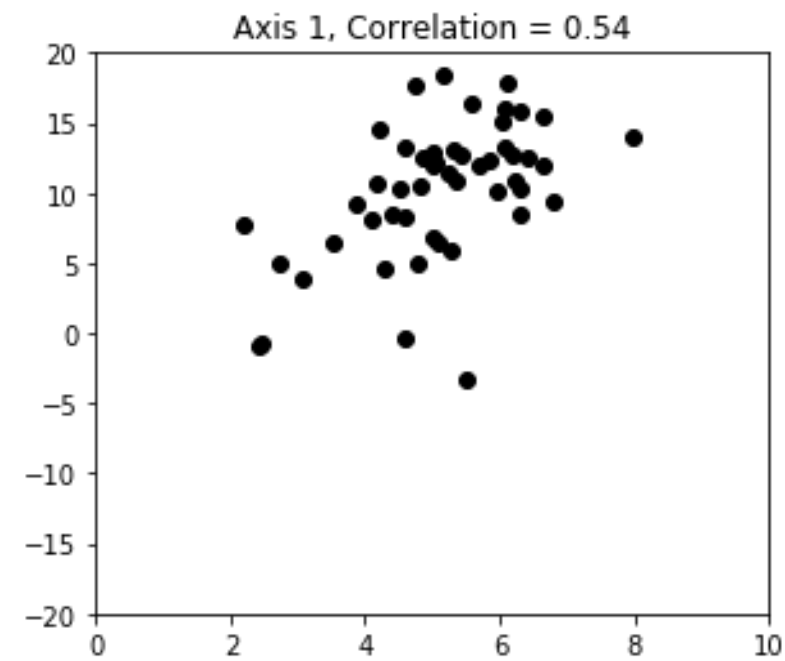
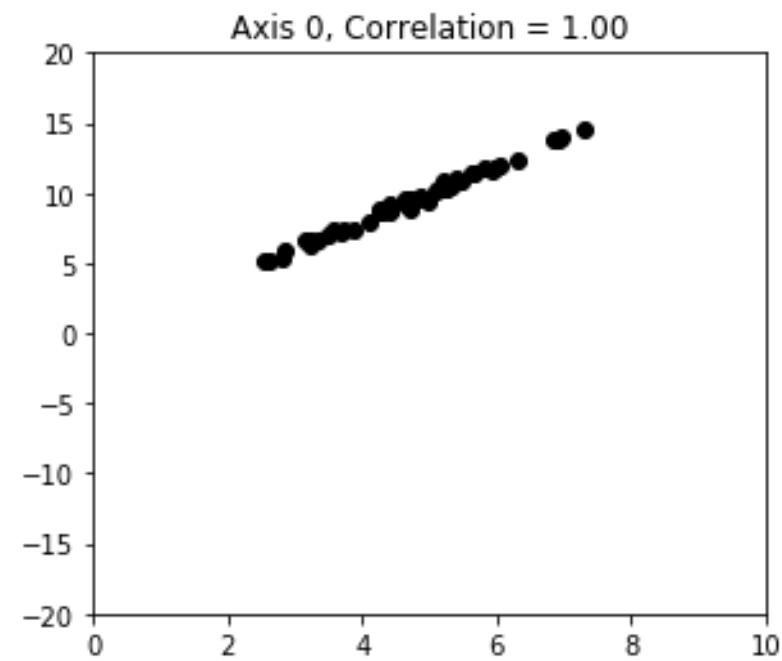
## INTRODUCTION TO LINEAR MODELING IN PYTHON

# Quantifying Linear Relationships

Jason Vestuto  
Data Scientist



# Pre-Visualization







# Review of Single Variable Statistics

```
# Mean
mean = sum(x)/len(x)
```

```
# Deviation, sometimes called "centering"
dx = x - np.mean(x)
```

```
# Variance
variance = np.mean(dx*dx)
```

```
# Standard Deviation
stdev = np.sqrt(variance)
```



# Covariance

```
# deviations of two variables  
dx = x - np.mean(x)  
dy = y - np.mean(y)
```

```
# co-vary means to vary together  
deviation_products = dx*dy
```

```
# covariance as the mean  
covariance = np.mean(dx*dy)
```



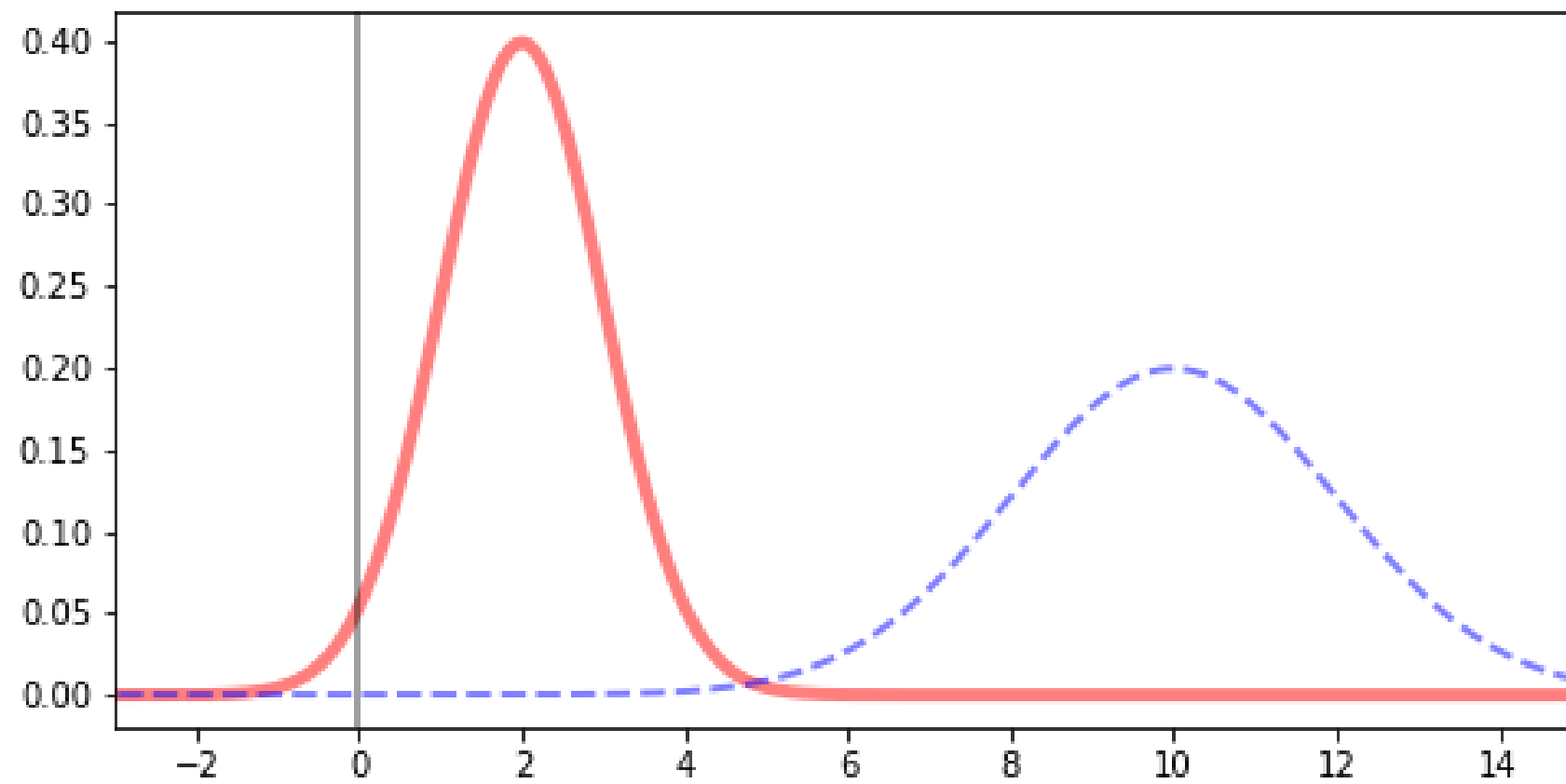
# Correlation

```
# divide deviations by standard deviation  
zx = dx/np.std(x)  
zy = dy/np.std(y)
```

```
# mean of the normalize deviations  
correlation = np.mean(zx*zy)
```

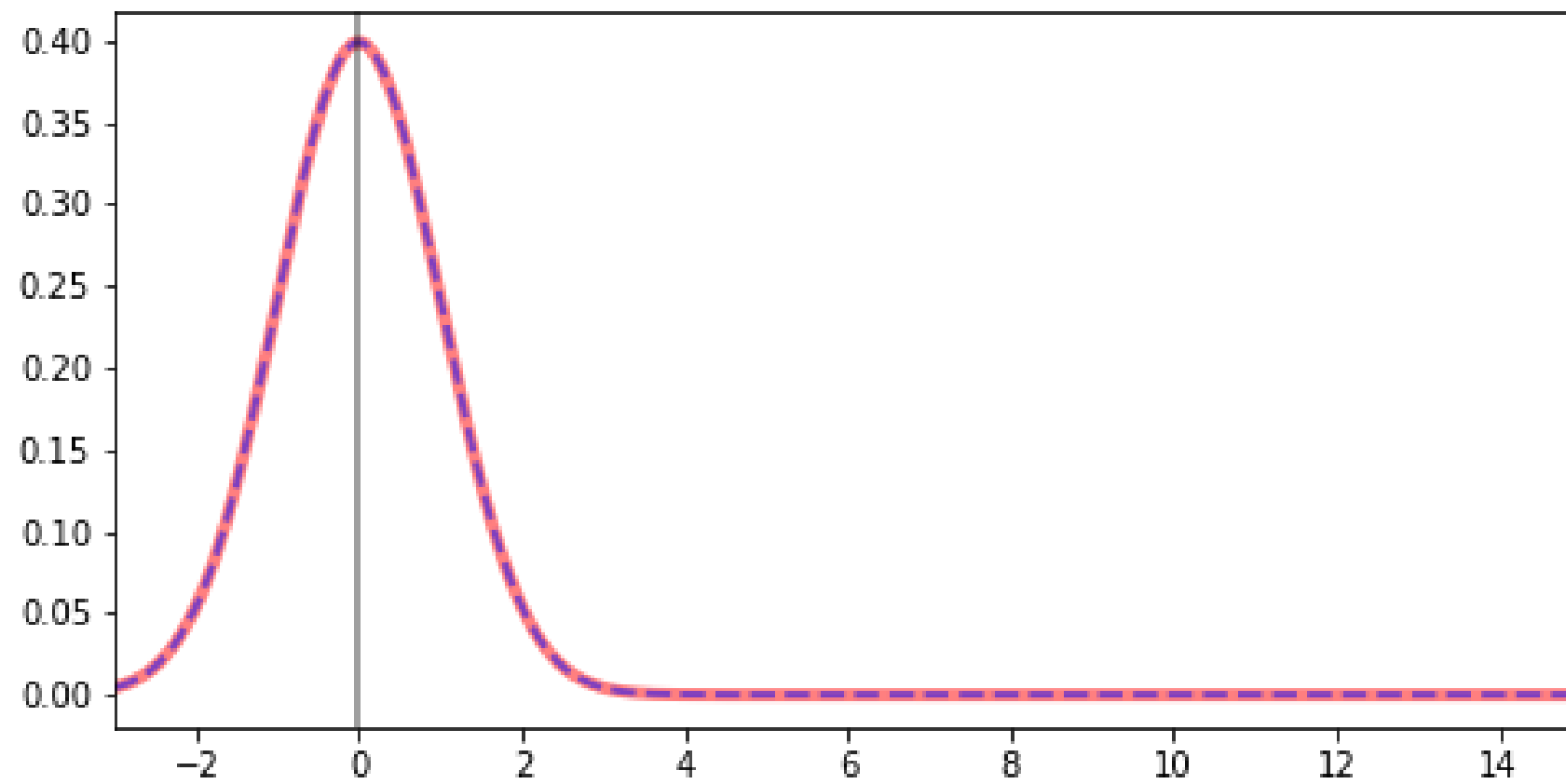


# Normalization: Before



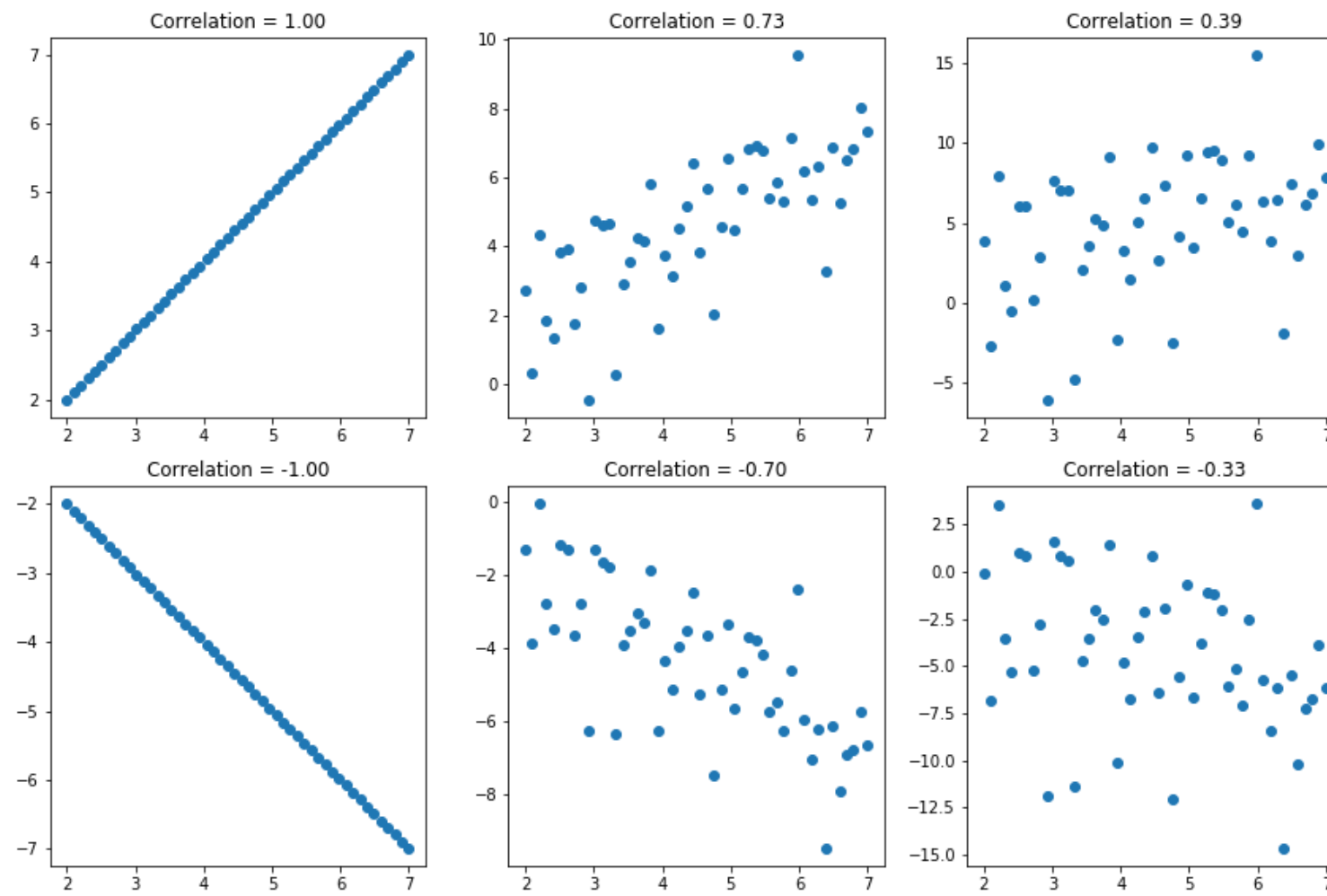


# Normalization: After



# Magnitude versus Direction

- Correlation values: -1 to +1



- Two Parts: Magnitude (1 to 0) versus Sign (+ or -)



## INTRODUCTION TO LINEAR MODELING IN PYTHON

**Let's practice!**