INTRODUCTION TO LINEAR MODELING IN PYTHON

What makes a model linear

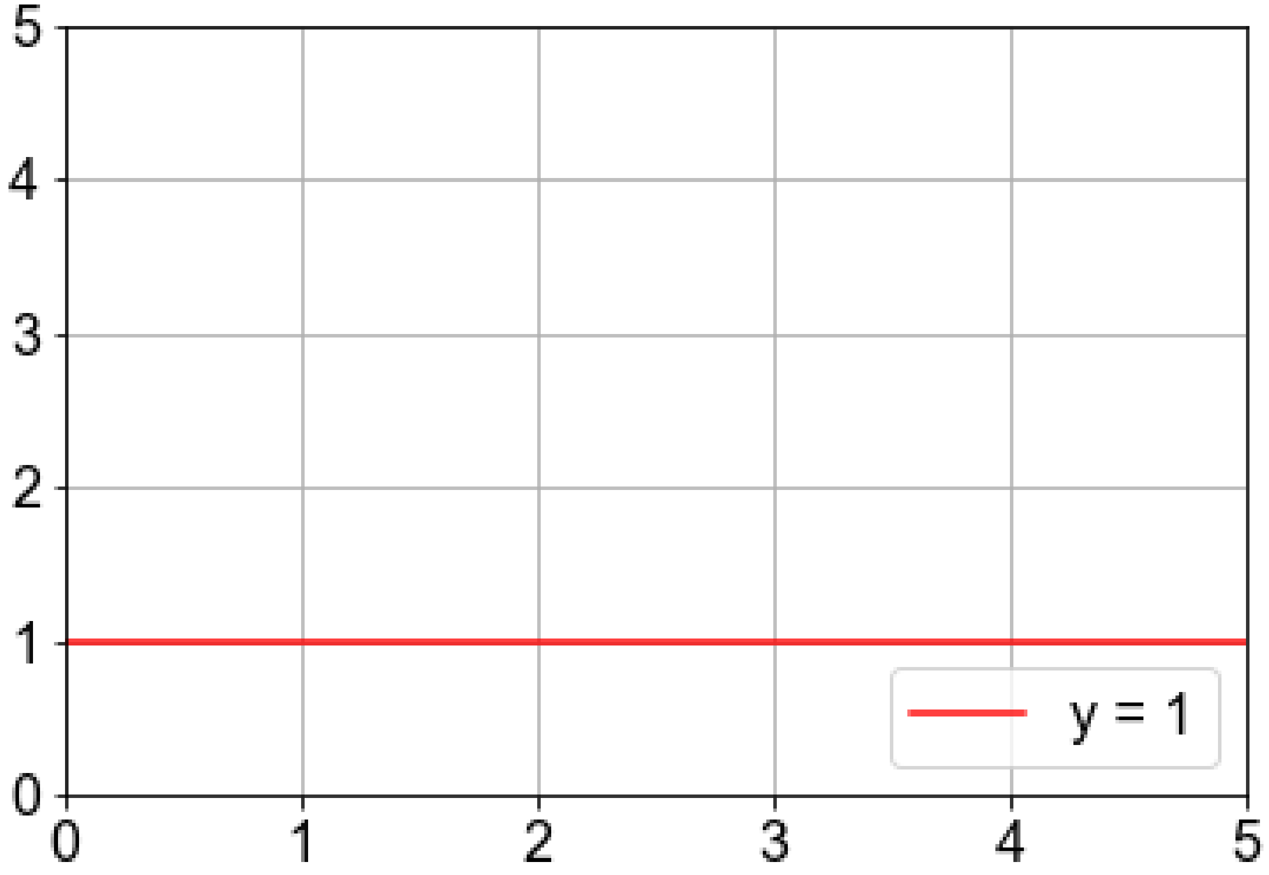
Jason Vestuto
Data Scientist

Taylor Series

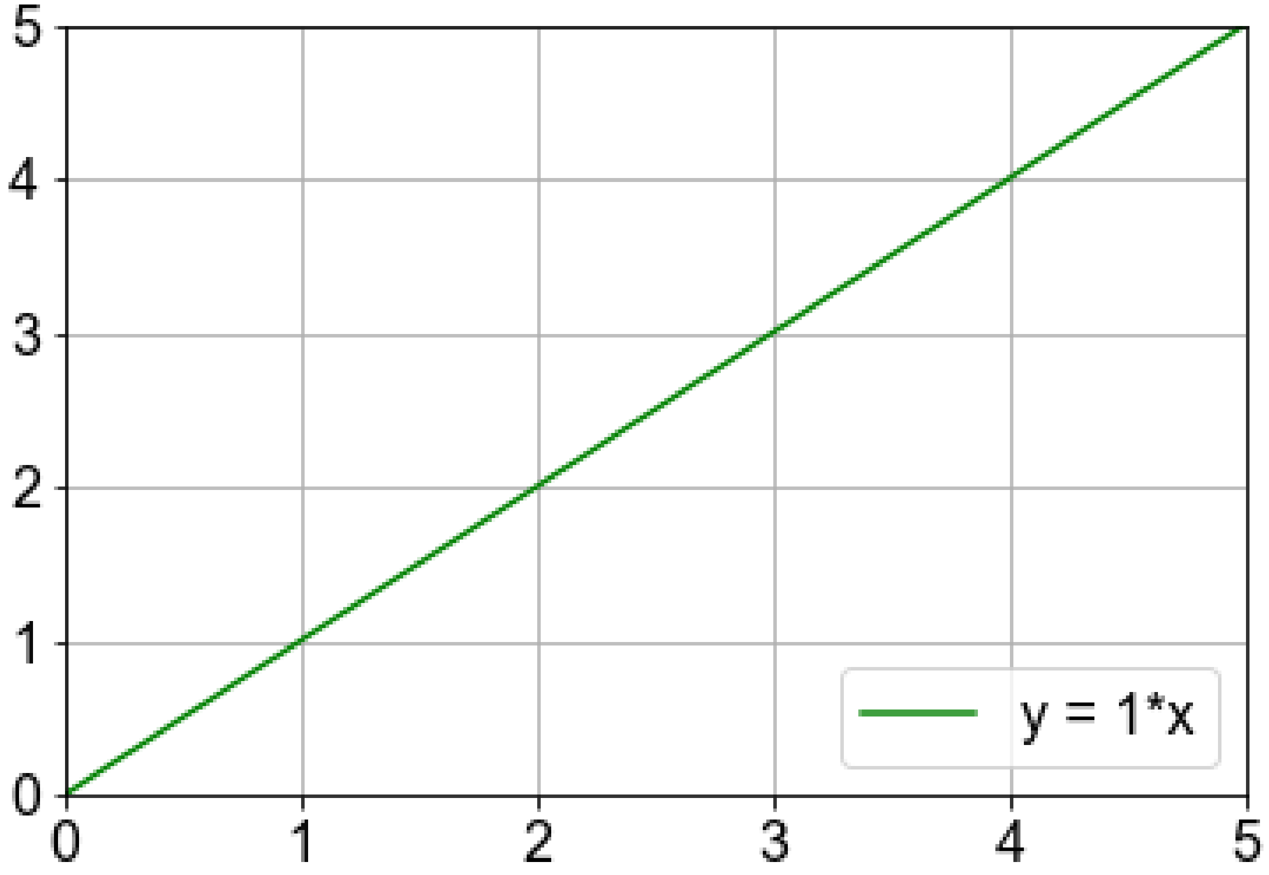
Things to know:

1. approximate any curve
2. polynomial form: `y = a0 + a1*x + a2*x**2 + a3*x**3 + .... + an*x**n`
3. often, first order is enough: `y = a0 + a1*x`

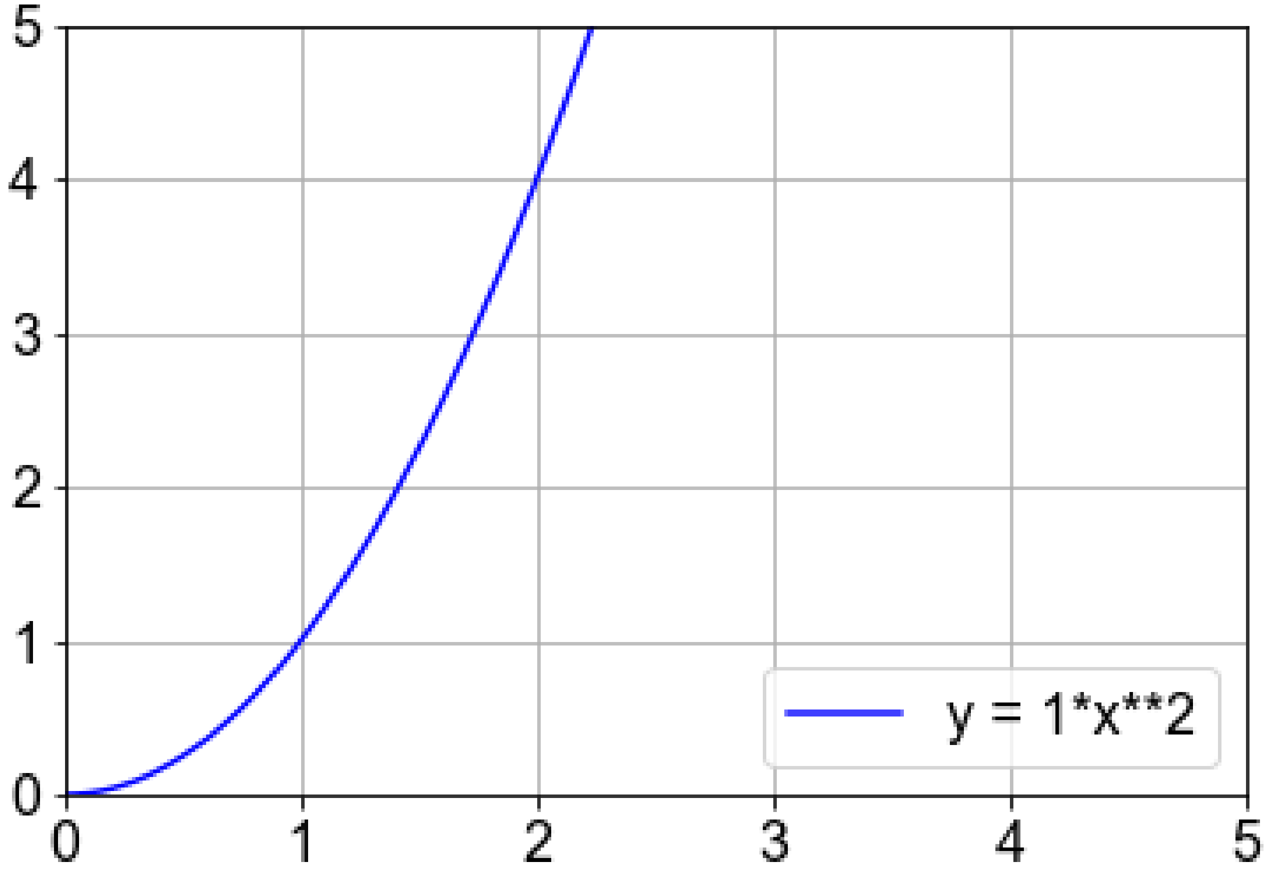
Series Terms: a0=1



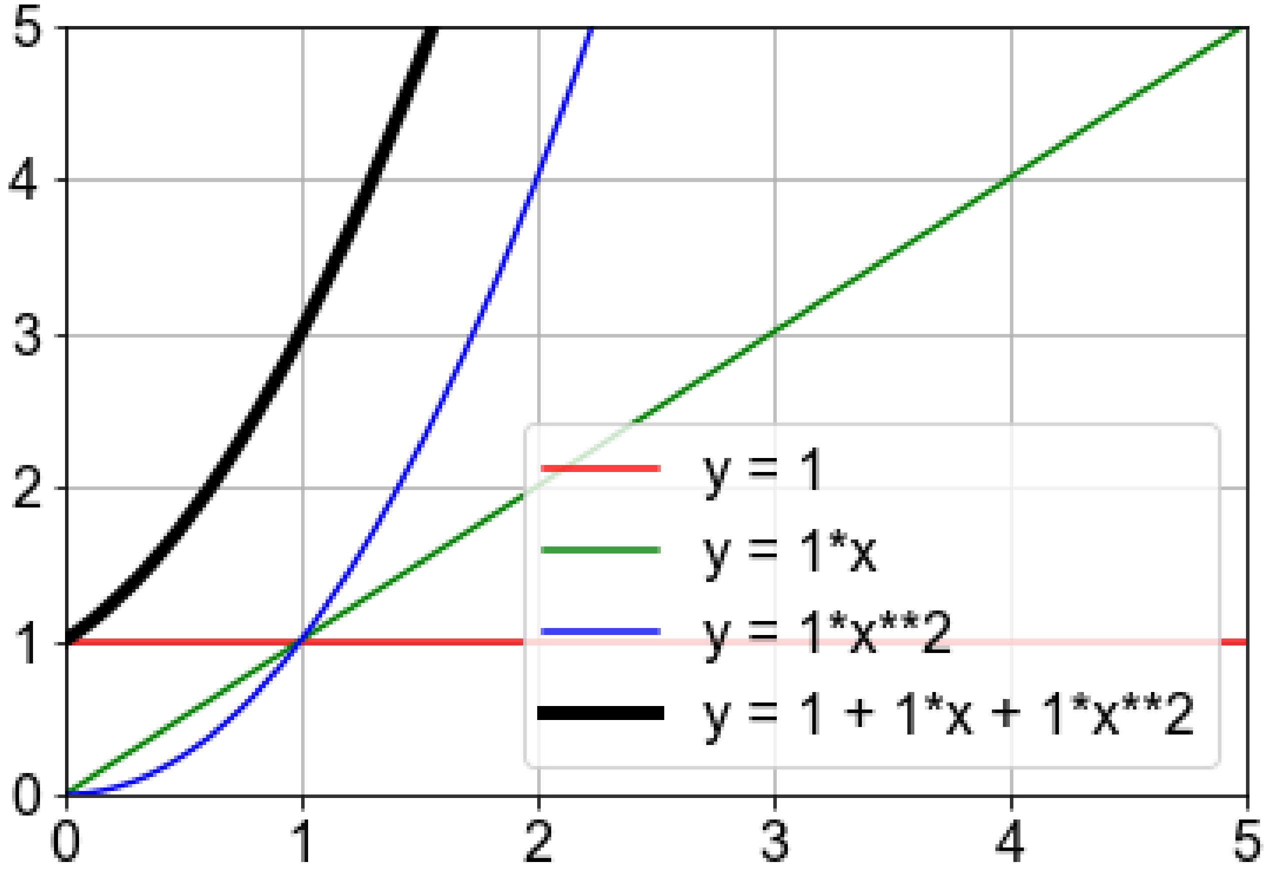
Series Terms: a1=1



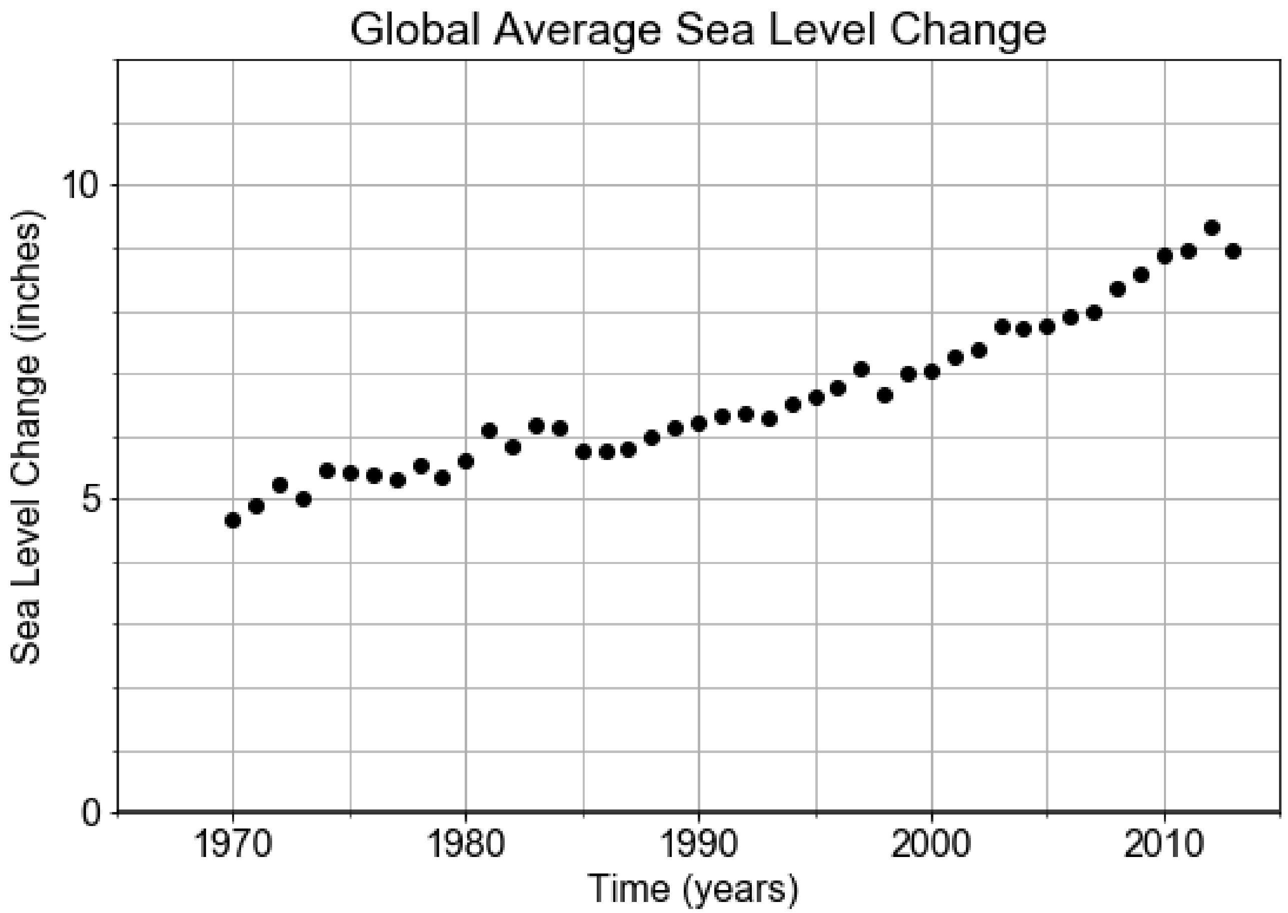
Series Terms: a2=1



Combining all Terms

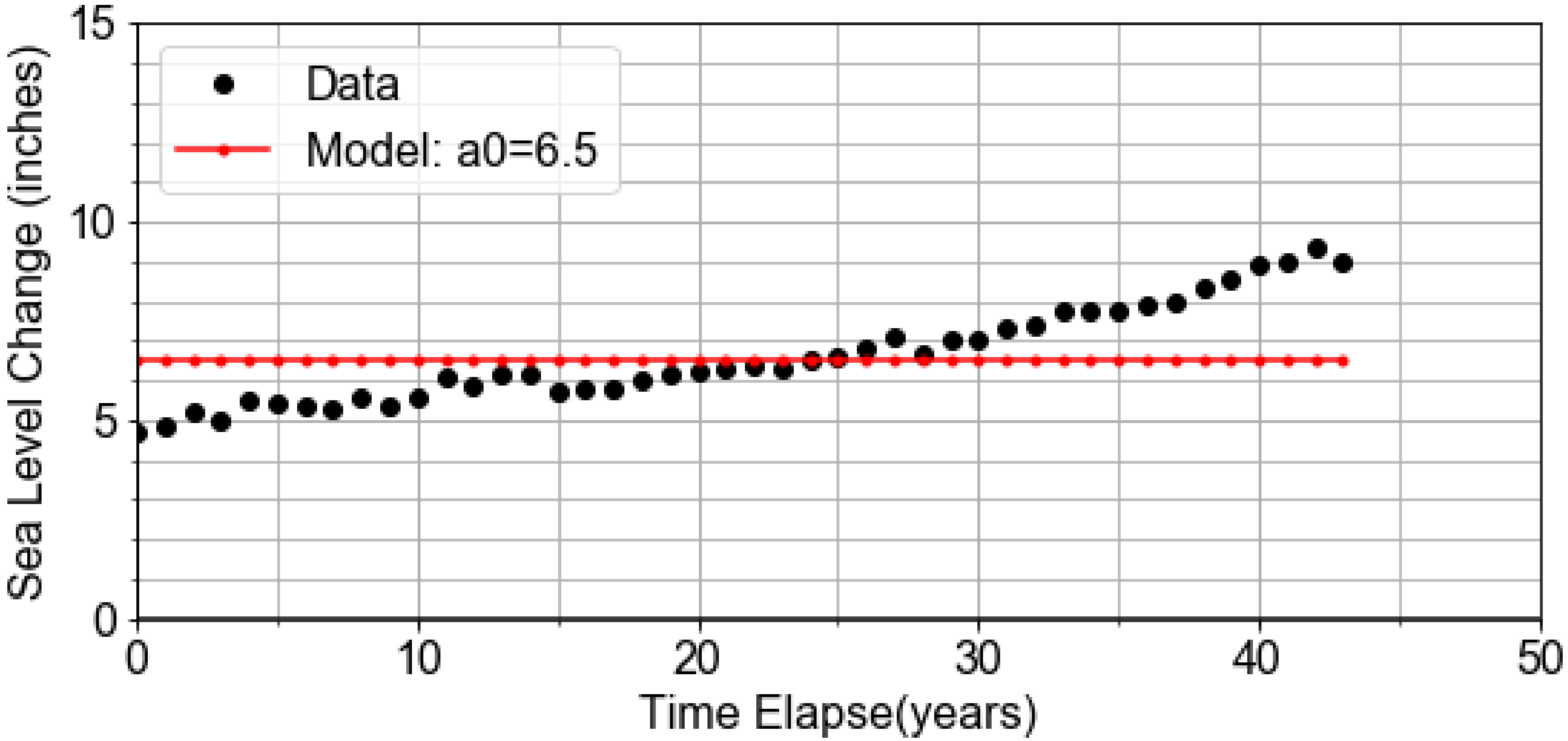


Real Data

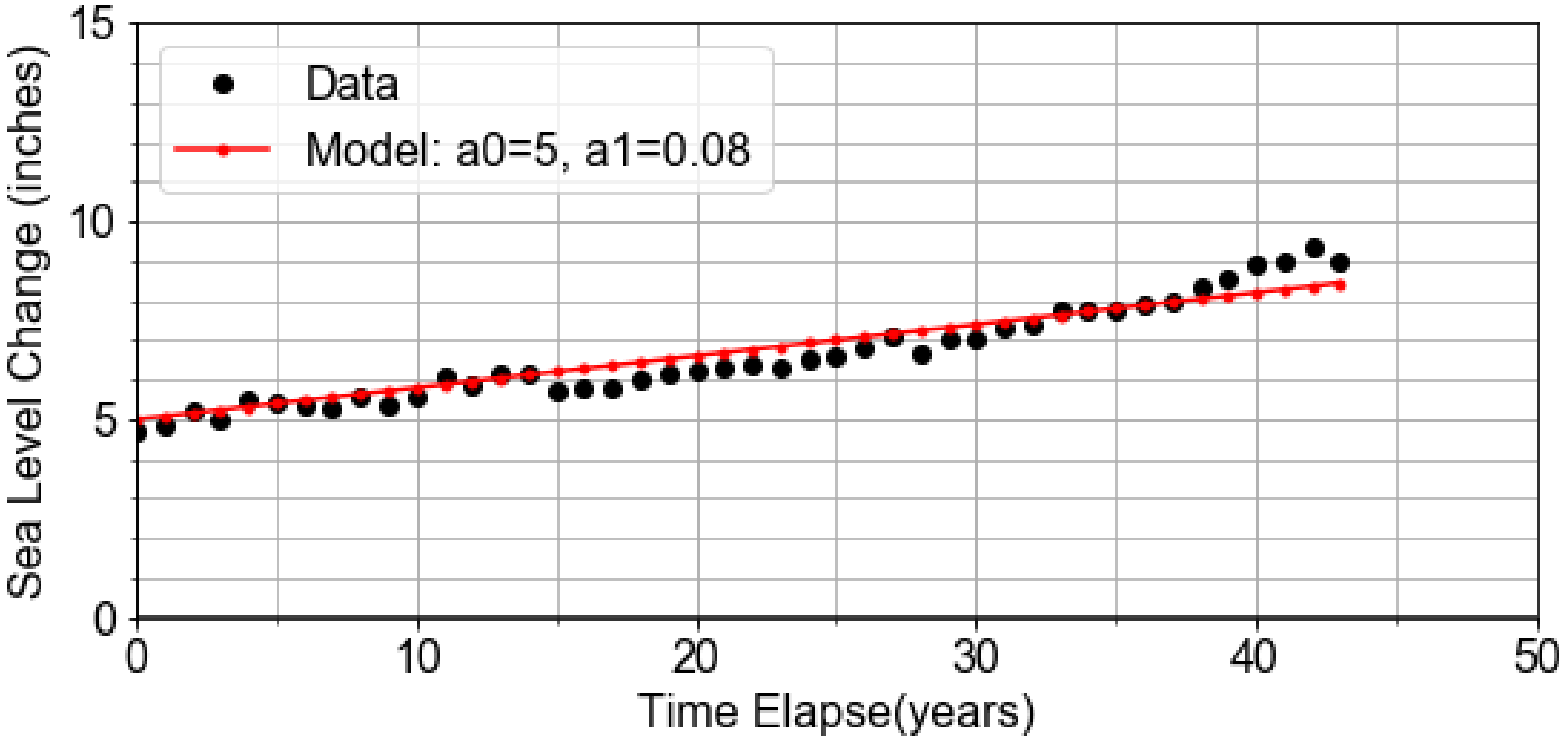


Source: https://www.epa.gov/climate-indicators

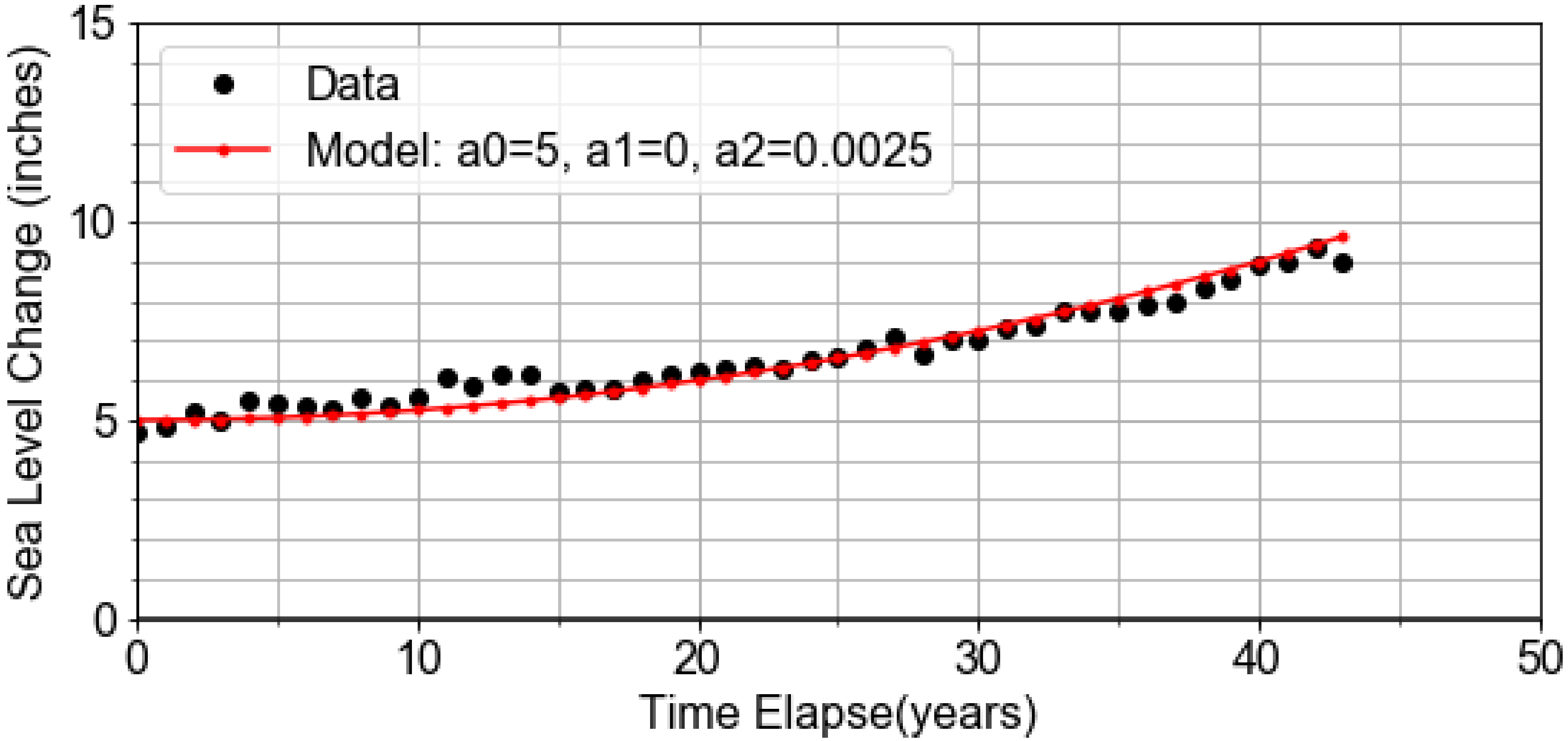
Zeroth Order



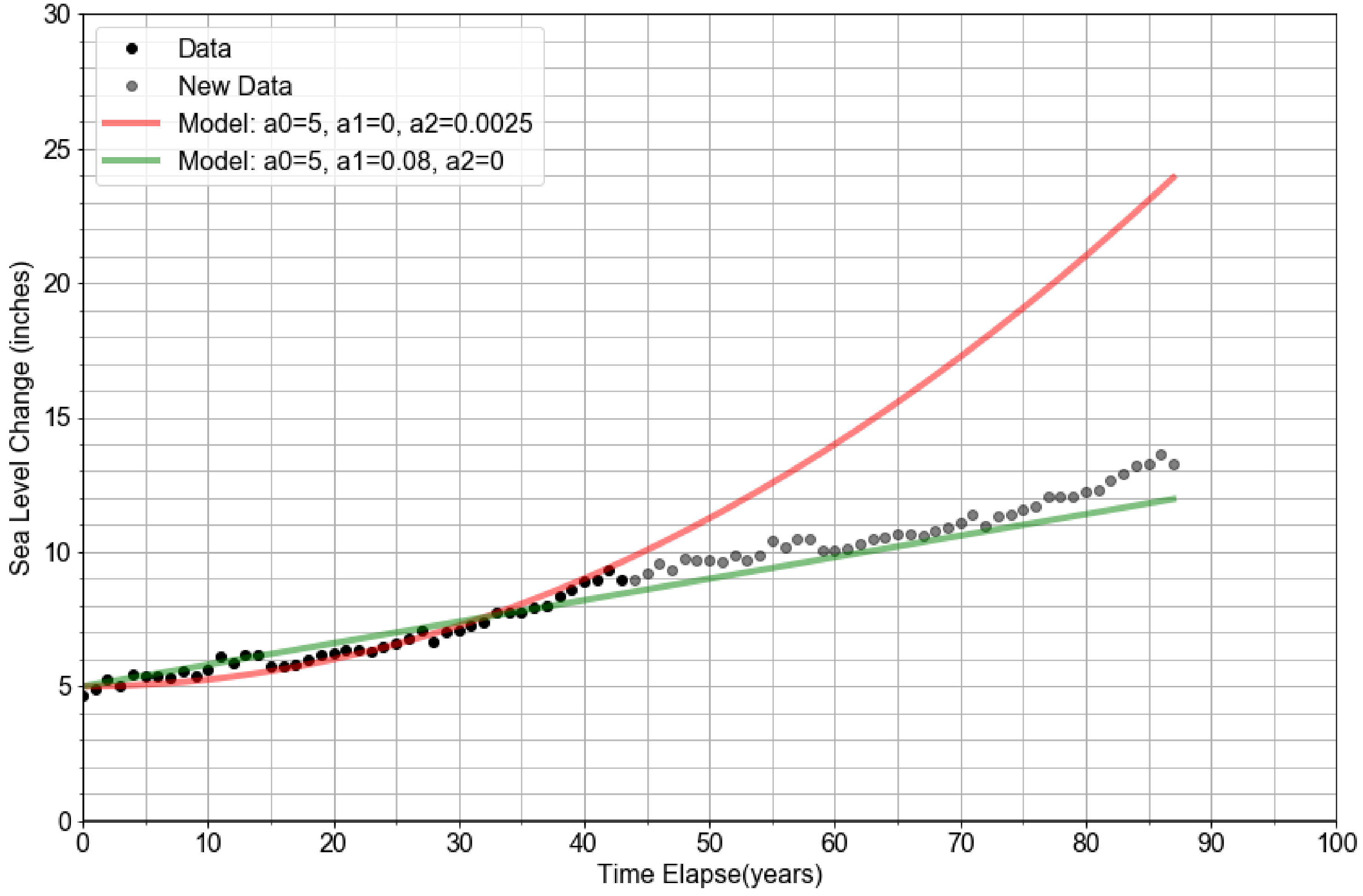
First Order


Data
Model: a0=5, a1=0.08
Sea Level Change (inches)
0
5
10
15
0
10
20
30
40
50
Time Elapse(years)

Higher Order



Over-fitting



INTRODUCTION TO LINEAR MODELING IN PYTHON

Let's practice!

INTRODUCTION TO LINEAR MODELING IN PYTHON

Interpreting Slope and Intercept

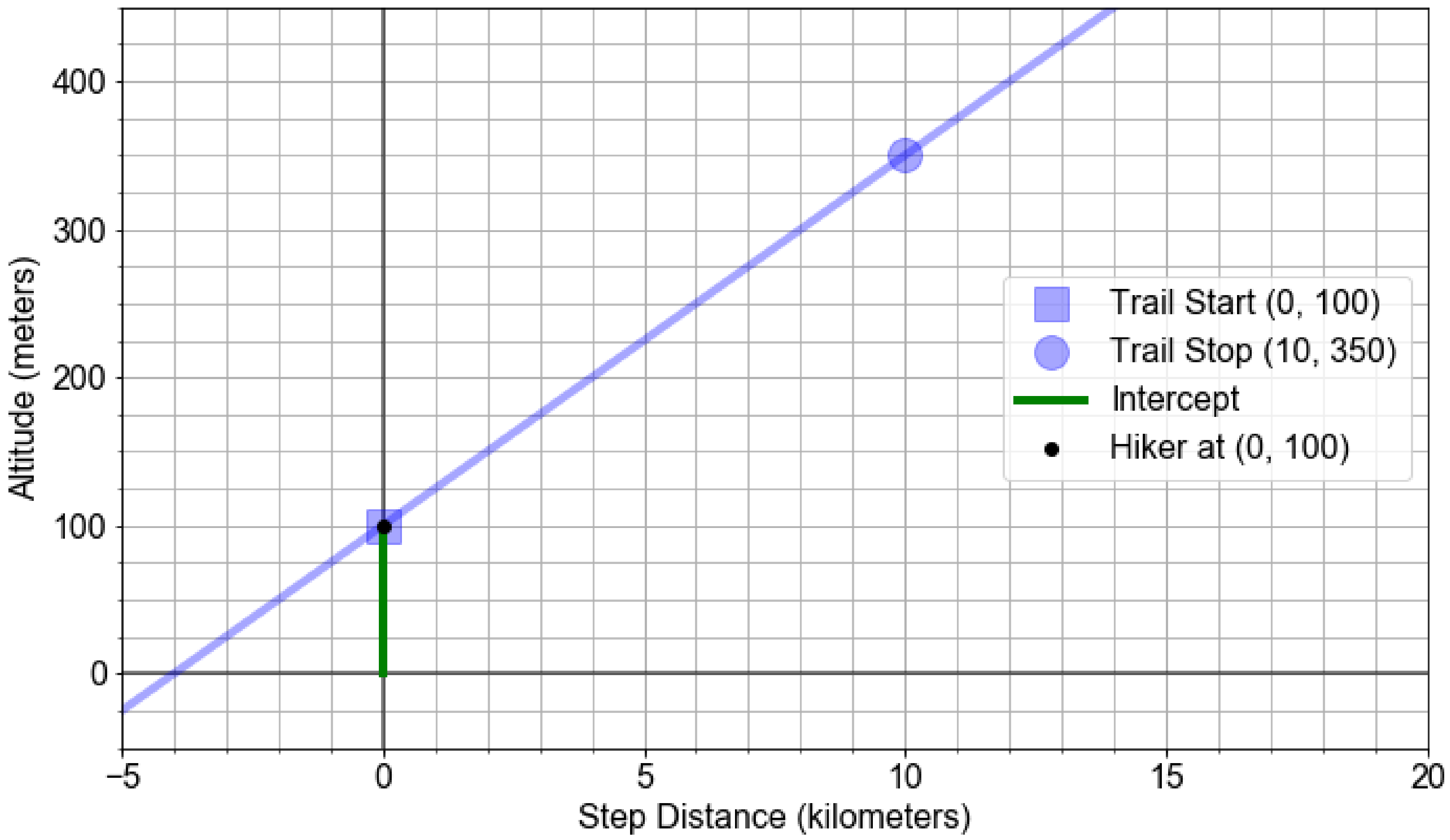
Jason Vestuto
Data Scientist

Reminder: Terminology

Review:

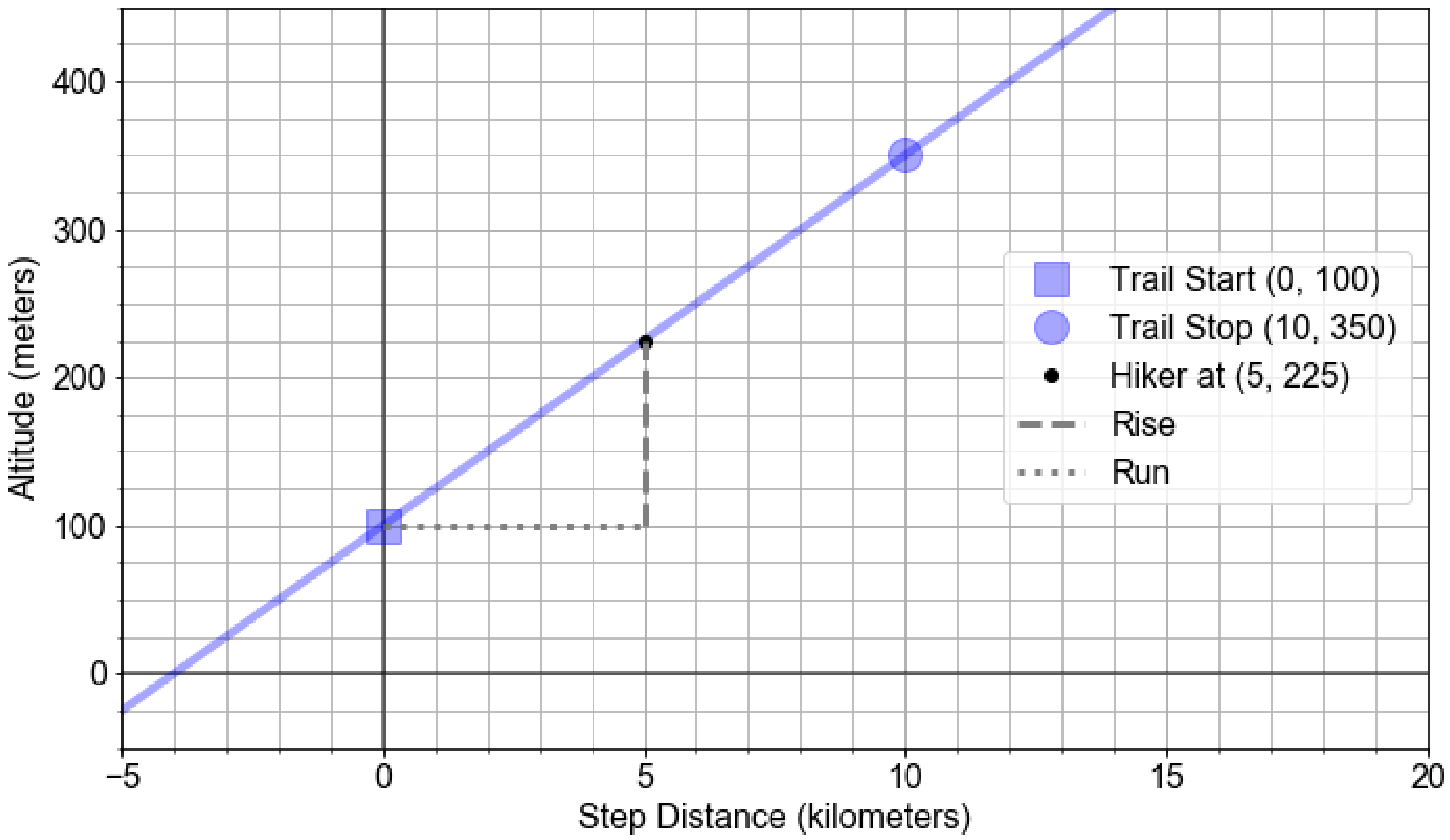
- `y = a0 + a1*x`
- `x` = independent variable, e.g. time
- `y` = dependent variable, e.g. distance traveled
- `xp = 10; yp = a0 + a1*xp`, "model prediction"

Intercept



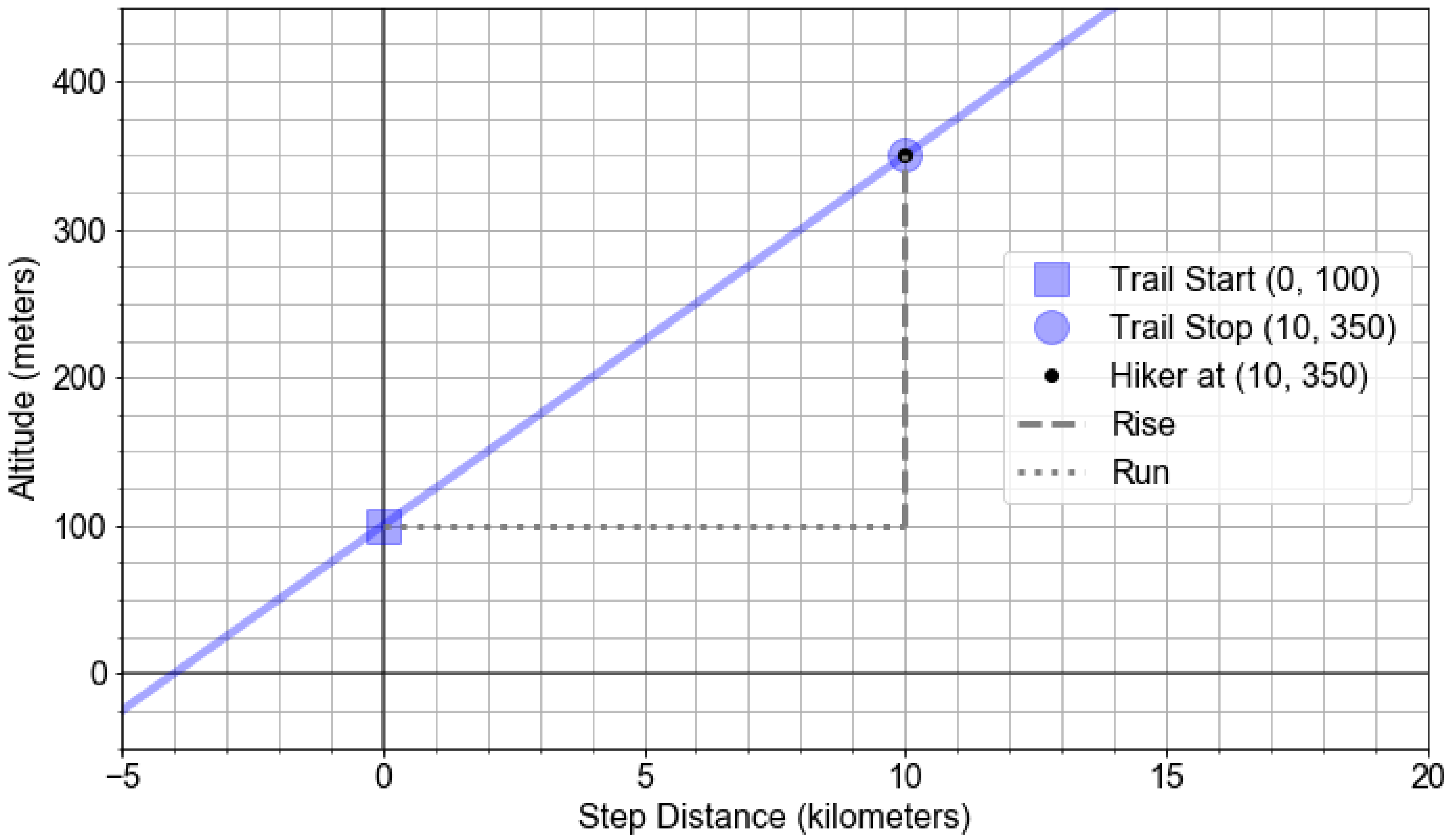
```
x0 = 0
print(y(x0))
    100
```

Slope



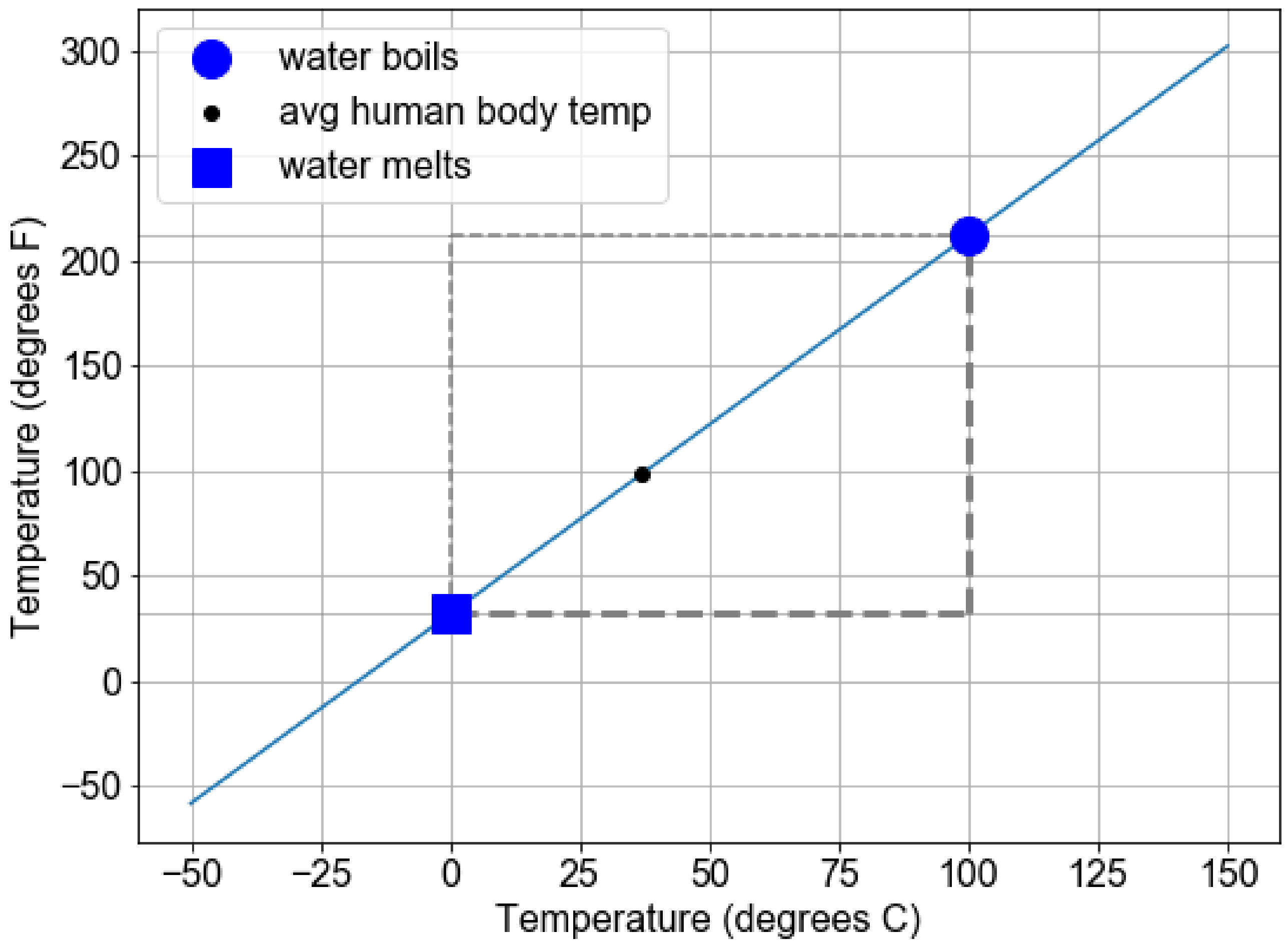
```
slope = (225 - 100) / (5 - 0)
print(slope)
    25
```

Average Slope



```
slope = (350 - 100) / (10 - 0)
print(slope)
    25
```

Rescaling versus Dependency



```
slope = (212-32)/(100-0) # 180/100 = 9/5
intercept = 32
```

INTRODUCTION TO LINEAR MODELING IN PYTHON

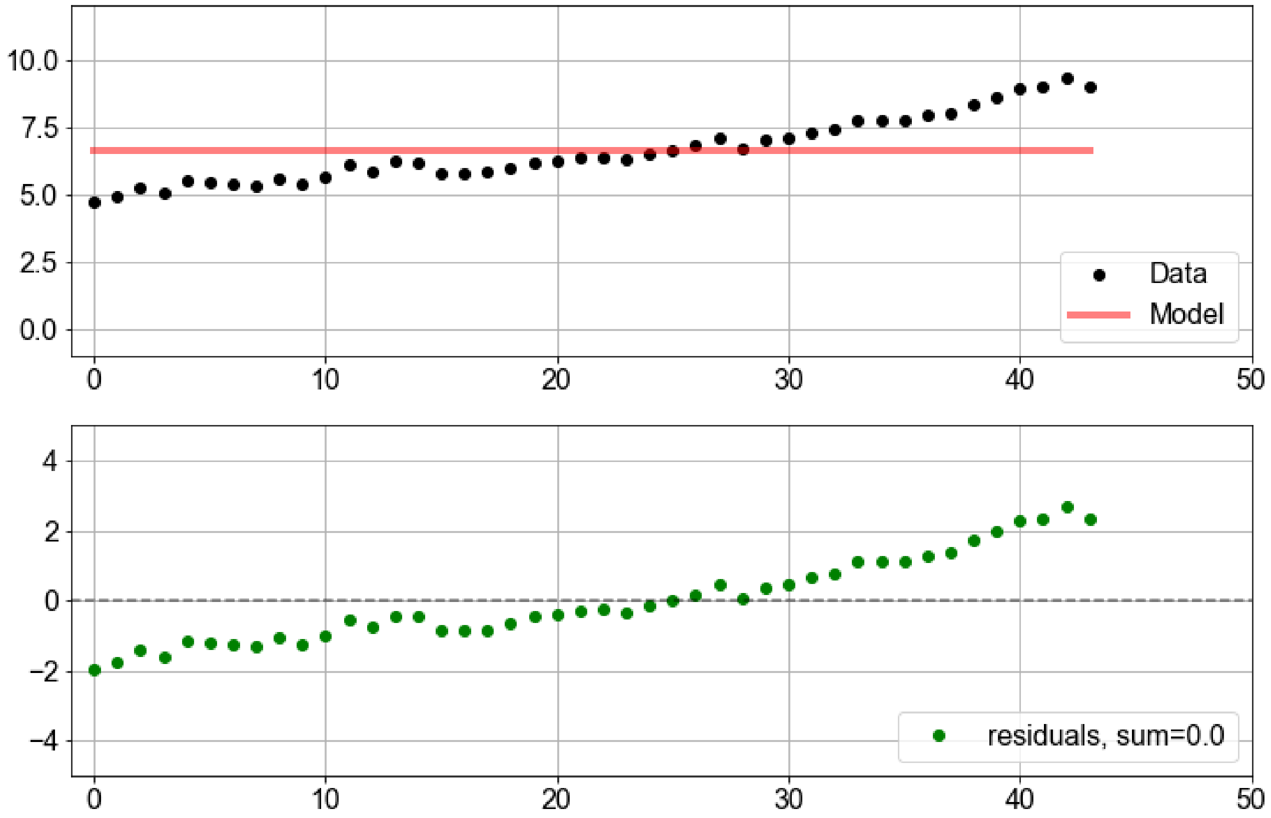
Let's practice!

INTRODUCTION TO LINEAR MODELING IN PYTHON

Model Optimization

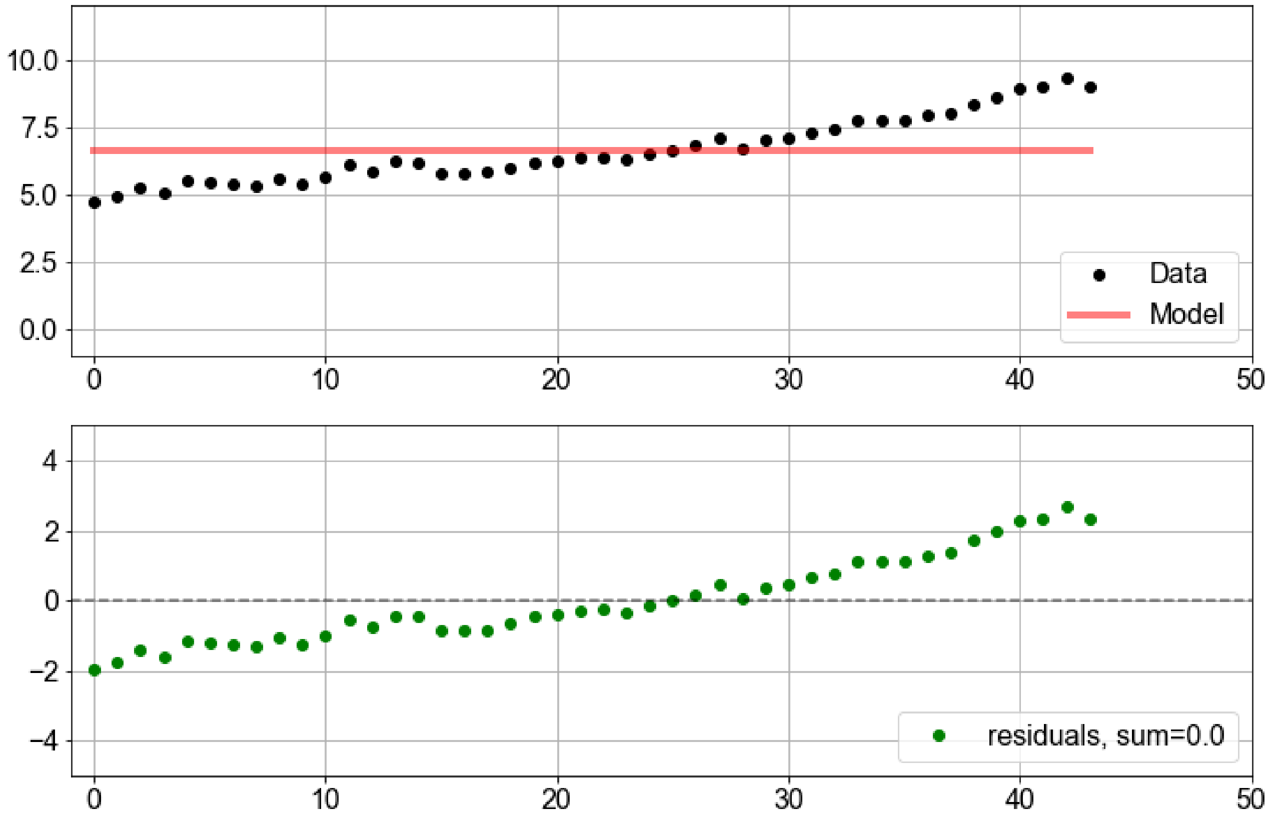
Jason Vestuto
Data Scientist

Residuals



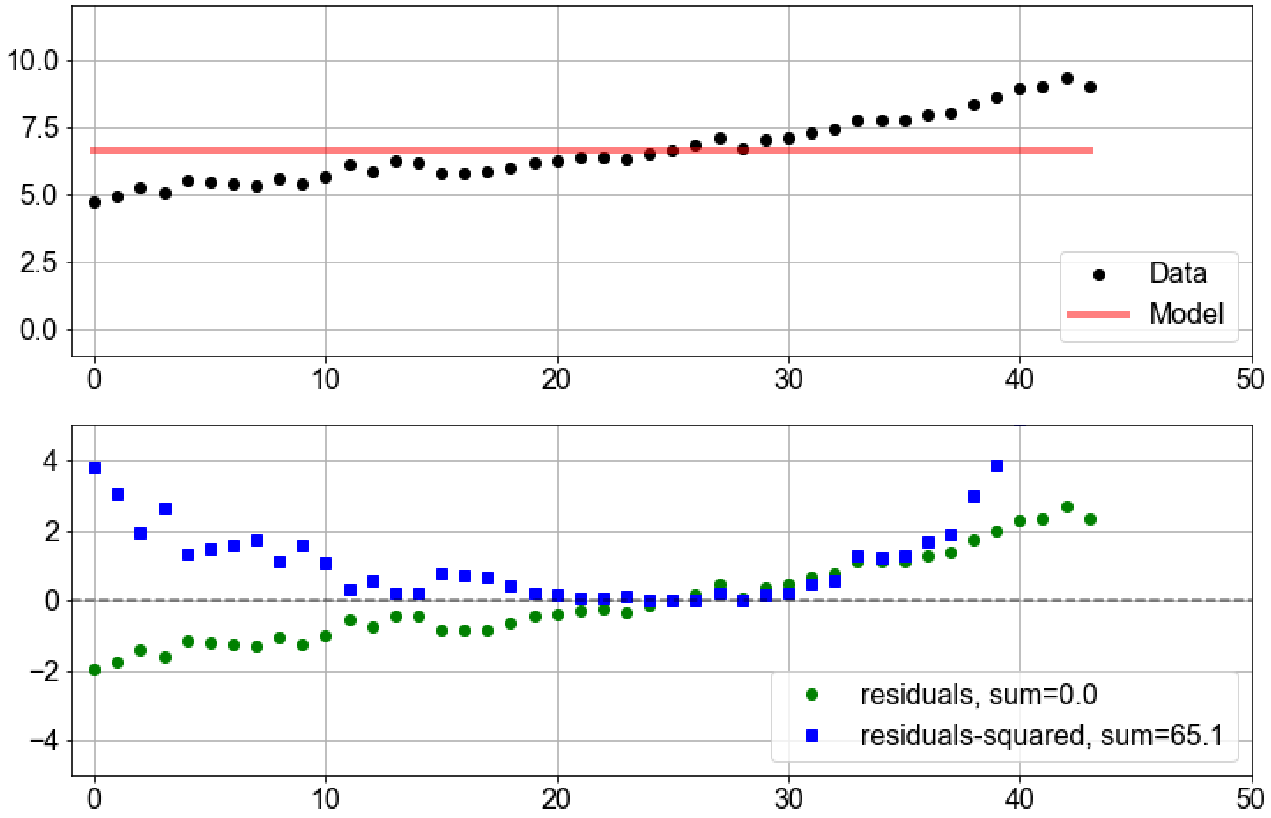
```
residuals = y_model - y_data
len(residuals) == len(y_data)
    True
```

Residuals Summed



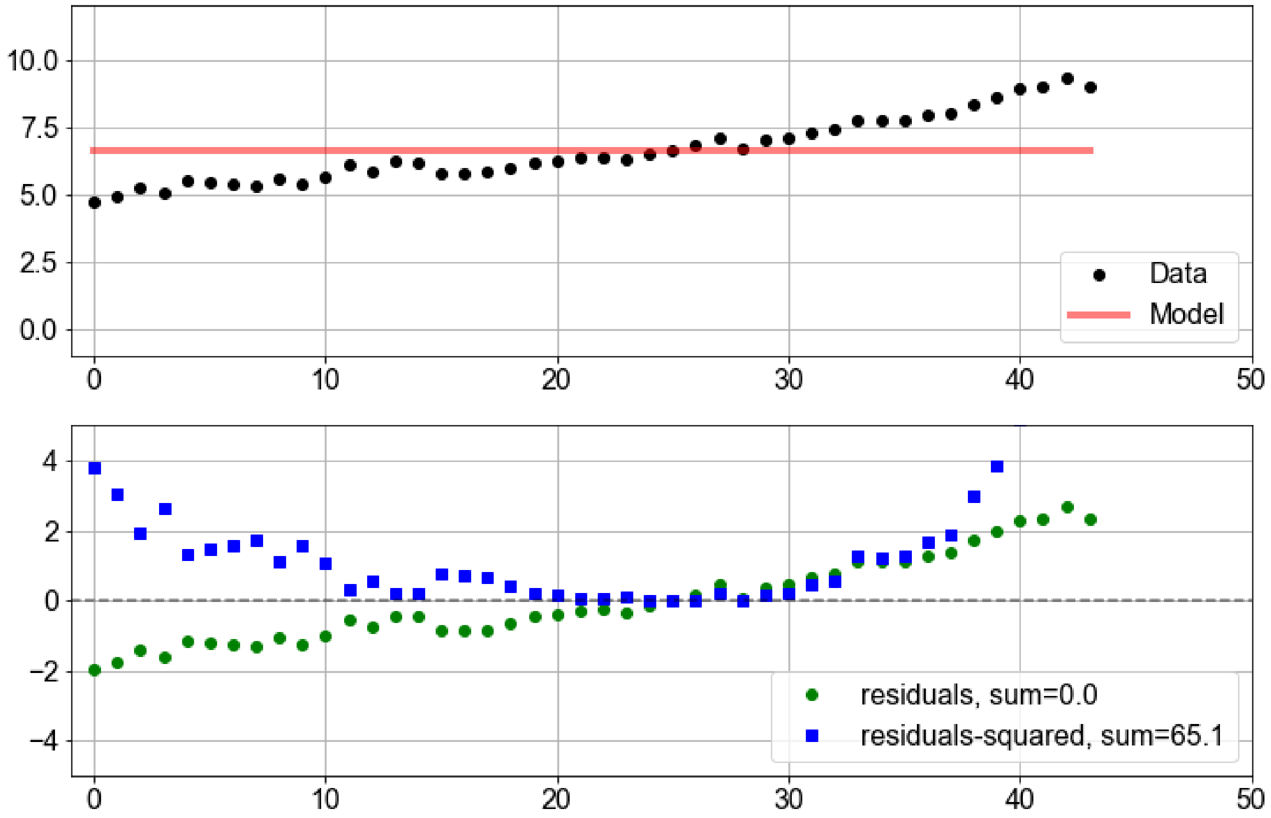
```
residuals = y_model - y_data
print( np.sum(residuals) )
    0.0
```

Residuals Squared



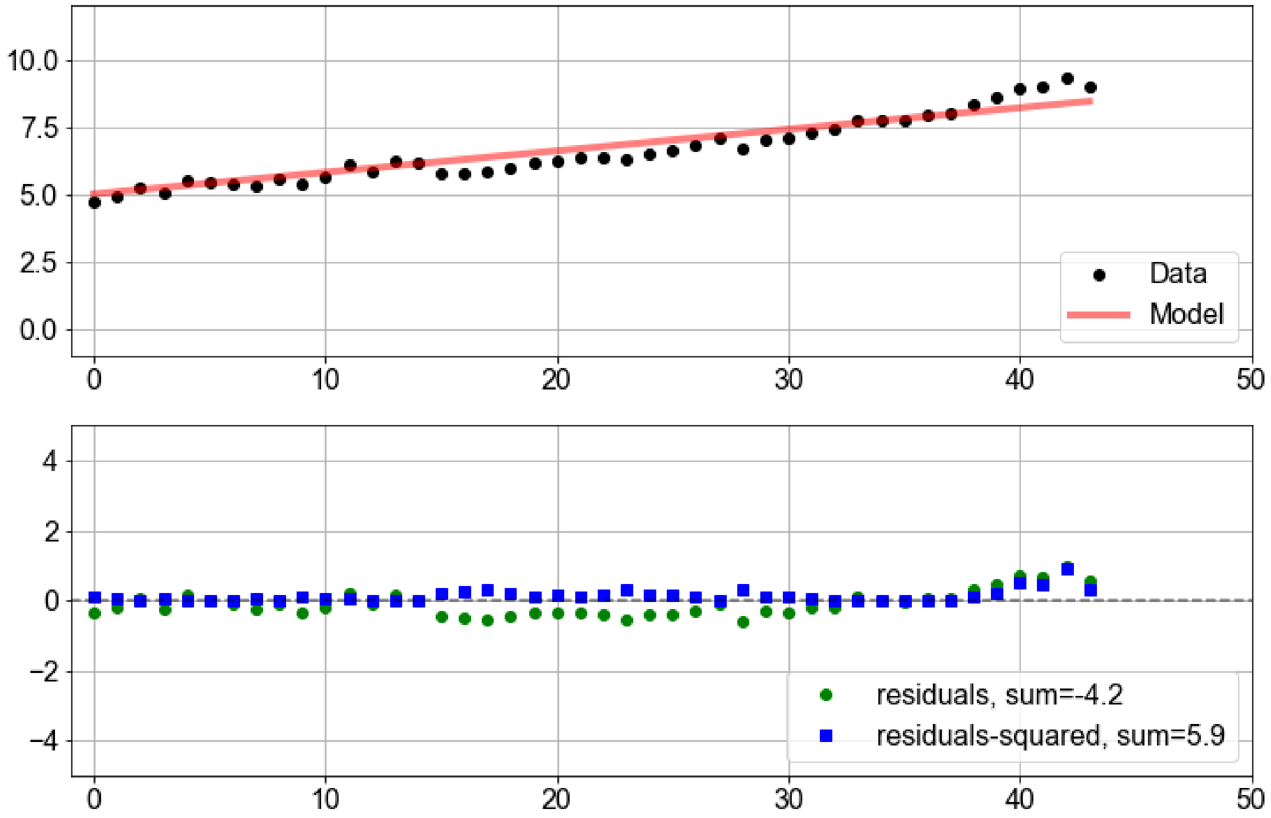
```
residuals_squared = np.square(y_model - y_data)

print( np.sum(residuals_squared) )
    65.1
```

RSS



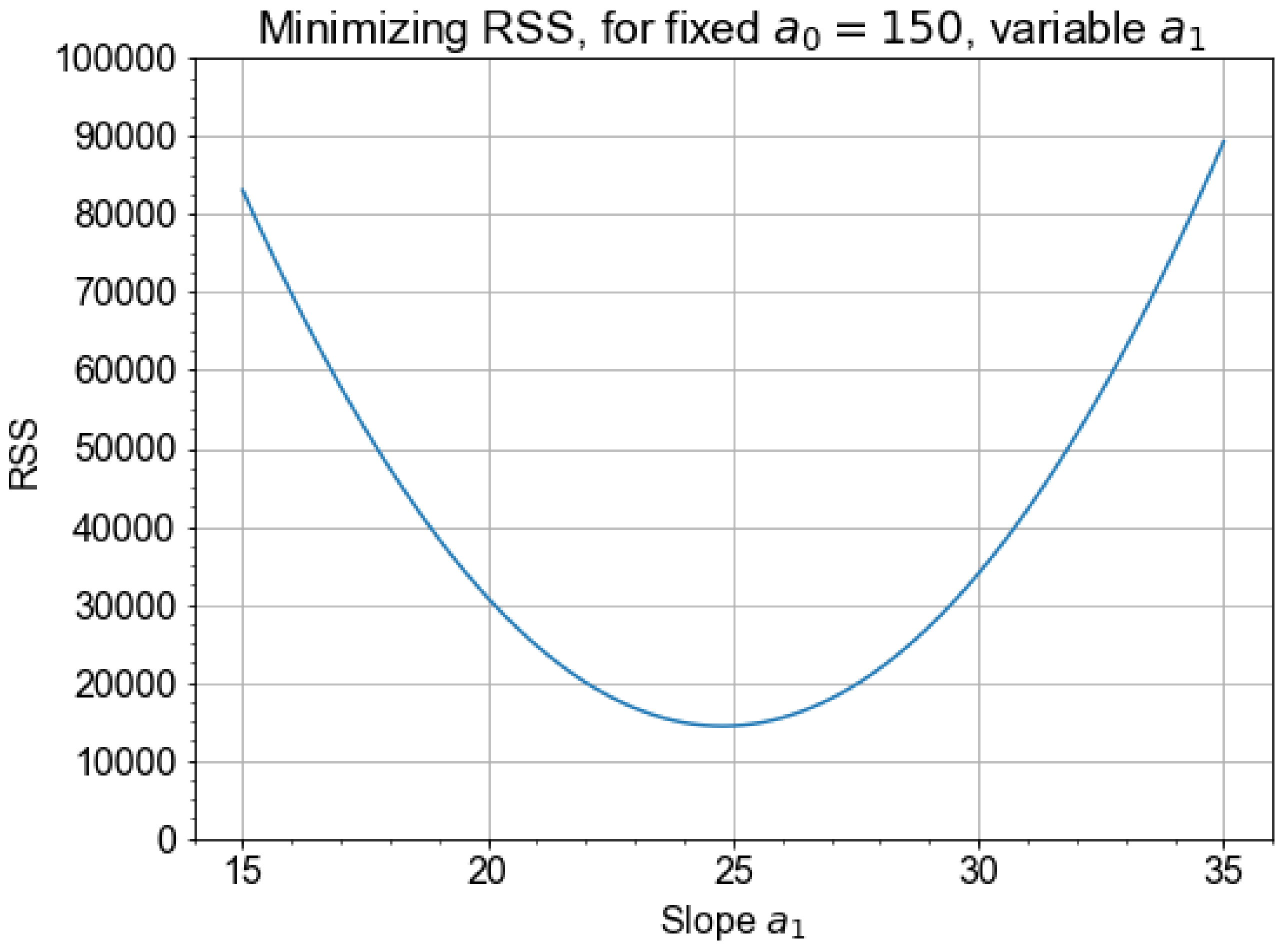
```
residuals_squared = np.square(y_model - y_data)
RSS = np.sum(residuals_squared)
```

RSS



```
RSS = np.sum(np.square(y_model - y_data))
print( RSS )
    5.9
```

Variation of RSS



INTRODUCTION TO LINEAR MODELING IN PYTHON

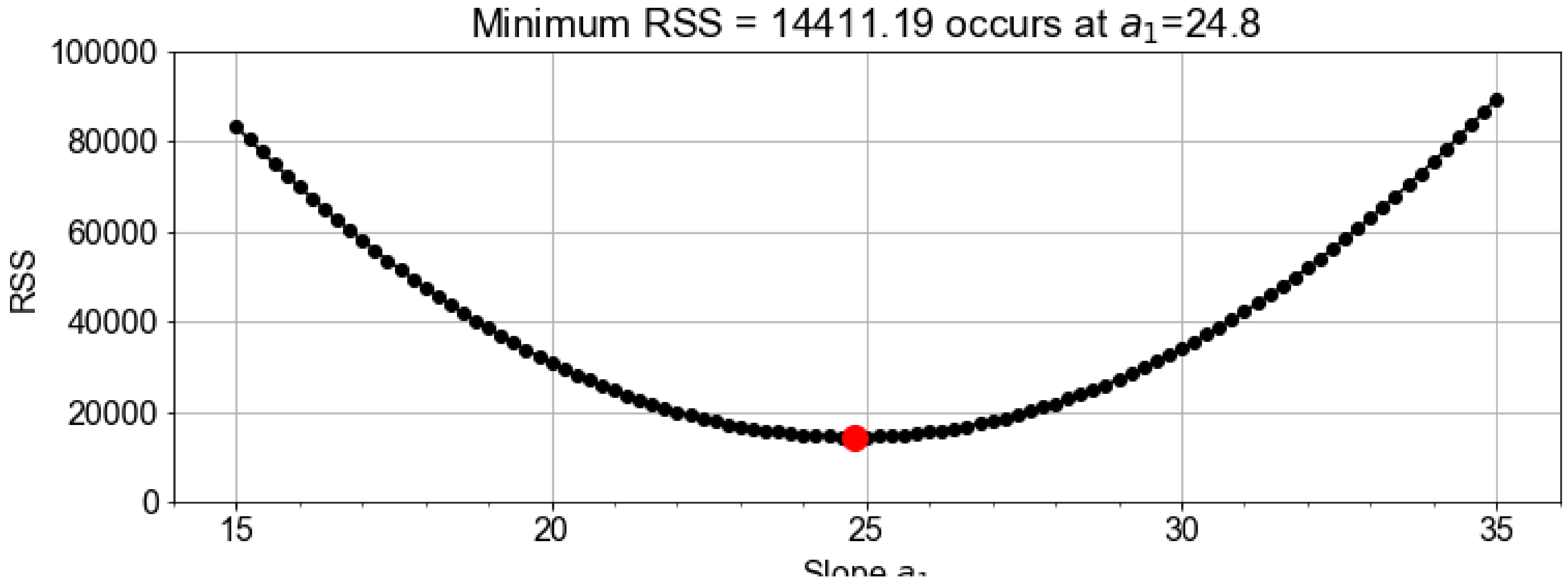
Let's practice!

INTRODUCTION TO LINEAR MODELING IN PYTHON

Least-Squares Optimization

Jason Vestuto
Data Scientist

Minima of RSS



Setting RSS slope = zero, and some calculus, yields:

- $a_1 = covariance(x, y) / variance(x)$
- $a_0 = mean(y) - a_1 \times mean(x)$

Optimized by Numpy

Numpy expressions of optimal slope and intercept

```
x_mean = np.sum(x)/len(x)
y_mean = np.sum(y)/len(y)
```

```
x_dev = x - x_mean
y_dev = y - y_mean
```

```
a1 = np.sum( x_dev * y_dev ) / np.sum( x_dev**2 )
```

```
a0 = y_mean - (a1*x_mean)
```

Optimized by Scipy

```
from scipy import optimize
```

```
x_data, y_data  = load_data()
def model_func(x, a0, a1):
    return a0 + (a1*x)
```

```
param_opt, param_cov = optimize.curve_fit(model_func, x_data, y_data)
```

```
a0 = param_opt[0]  # a0 is the intercept in y = a0 + a1*x
a1 = param_opt[1]  # a1 is the slope     in y = a0 + a1*x
```

Optimized by Statsmodels

```
from statsmodels.formula.api import ols
```

```
x_data, y_data = load_data()
df = pd.DataFrame(dict(x_name=x_data, y_name=y_data))
```

```
model_fit = ols(formula="y_name ~ x_name", data=df).fit()
```

```
y_model = model_fit.predict(df)
x_model = x_data
```

```
a0 = model_fit.params['Intercept']
a1 = model_fit.params['x_name']
```

INTRODUCTION TO LINEAR MODELING IN PYTHON

Let's practice!