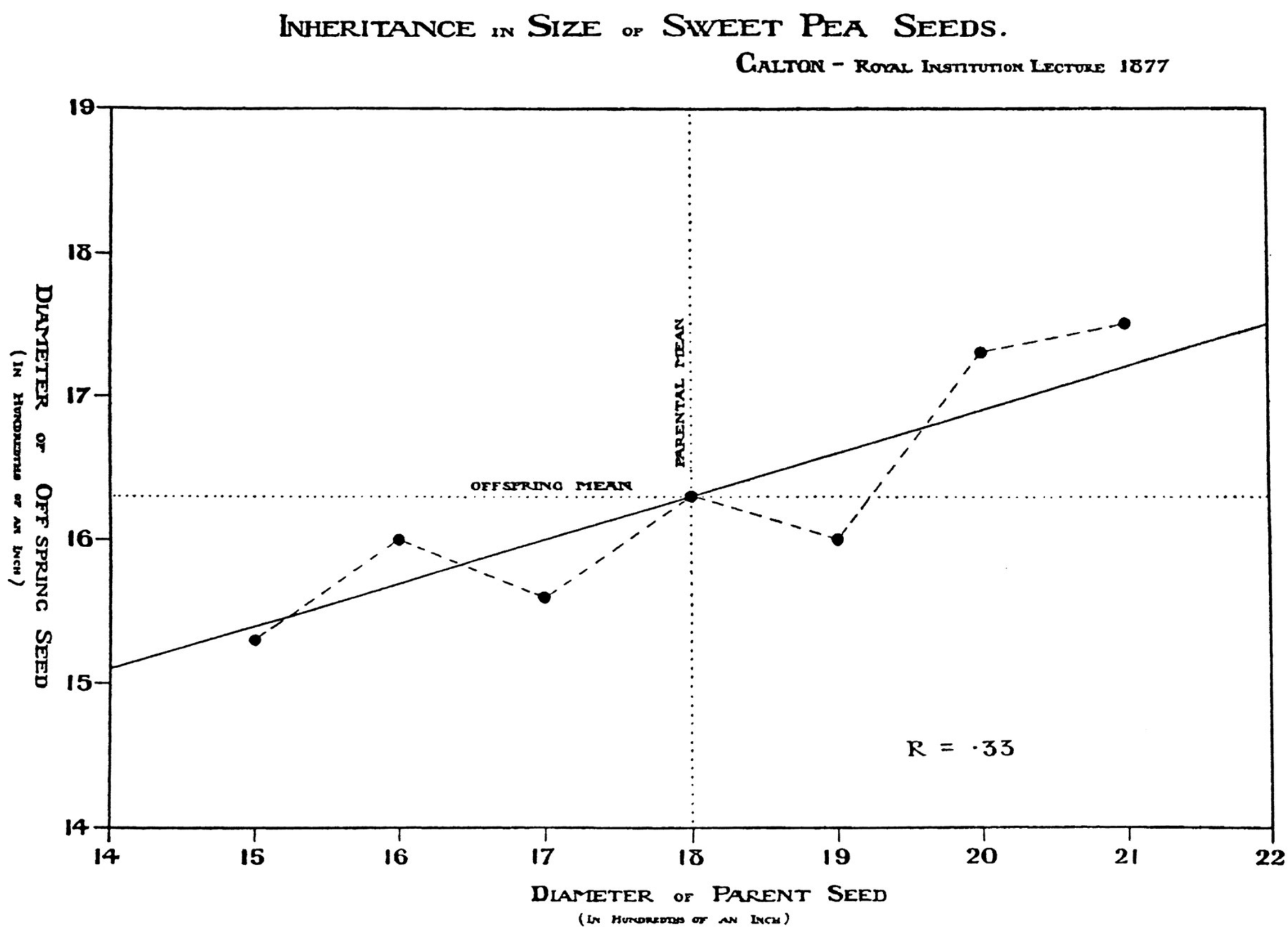
Linear Regression

Thomas Schwarz, SJ

Linear Regression

- Sir Francis Galton : 16 Feb 1822 — Jan 17 1911
  - Cousin of Charles Darwin
    - Discovered "Regression towards Mediocrity":
      - Individuals with exceptional measurable traits have more normal progreny
  - If parent's trait is at $x\sigma$ from μ, then progeny has traits at $\rho x\sigma$ from μ
    - ρ is the *coefficient of correlation* between trait of parent and of progeny

Linear Regression



Statistical Aside

- Regression towards mediocrity **does not mean**
  - Differences in future generations are smoothed out
- It reflects a selection biases
  - Trait of parent is mean + inherited trait + error
    - The parents we look at have both inherited trait and error >>0
  - Progeny also has mean + inherited trait + error
    - But the error is now random, and on average ~ 0.

Statistical Aside

- Example:
  - You do exceptionally well in a chess tournament
    - Result is Skill + Luck
      - You probably will not do so well in the next one
        - Your skill might have increased, but you cannot expect your luck to stay the same
          - It might, and you might be even luckier, but the odds are against it

Review of Statistics

- We have a population with traits
  - We are interested in only one trait
    - We need to make predictions based on a sample, a (random) collection of population members
    - We estimate the population mean by the sample mean
      - $$m = \frac{1}{N}\sum_{i=1}^{N} x_i$$
    - We estimate the population standard deviations by the (unbiased) sample standard deviation
      - $$s^2 = \frac{1}{N-1}\sum_{i=1}^{N} (x_i - \mu)^2$$

Unbiased ?

- Normally distributed variable with mean μ and st. dev. σ
  - Take sample $\{x_1, \ldots, x_N\}$
  - Calculate $s^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$
- Turns out: expected value for s is less than σ
- Call $N - 1$ the *degree of freedom*

Forecasting

- Mean model
  - We have a sample
    - We predict the value of the next population member to be the sample mean
- What is the risk?
  - Measure the risk by the standard deviation

Forecasting

- Normally distributed variable with mean μ and st. dev. σ
    - Take sample $\{x_1, \ldots, x_N\}$
    - What is the expected squared difference of m and μ:

      $E((m-\mu)^2)$
- "*Standard error of the mean*"
    - $E((m-\mu)^2) = \frac{s}{\sqrt{N}}$

Forecasting

- Forecasting Error of : $x_{N+1} \leftarrow \frac{1}{N}\sum_{i=1}^{N} X_i$
- Two sources of error:
  - We estimate the standard deviation wrongly
    - x_{N+1} is on average one standard deviation away from the mean
- Expected error

$$= \sqrt{s^2 + (\frac{s}{\sqrt{N}})^2} = s\sqrt{1 + \frac{1}{N}}$$

model error parameter error

Forecasting

- There is still a *model risk*
  - We just might not have the right model
    - The underlying distribution is not normal

Confidence Intervals

- Assume that the model is correct
  - Simulate the model `run` times
  - The *x*-confidence interval then
    - contains *x%* of the runs contain the true value

Confidence Intervals

- Confidence intervals usually are
  $\pm t \times$ (standard error of forecast
    - Contained in t-tables and depend on sample size

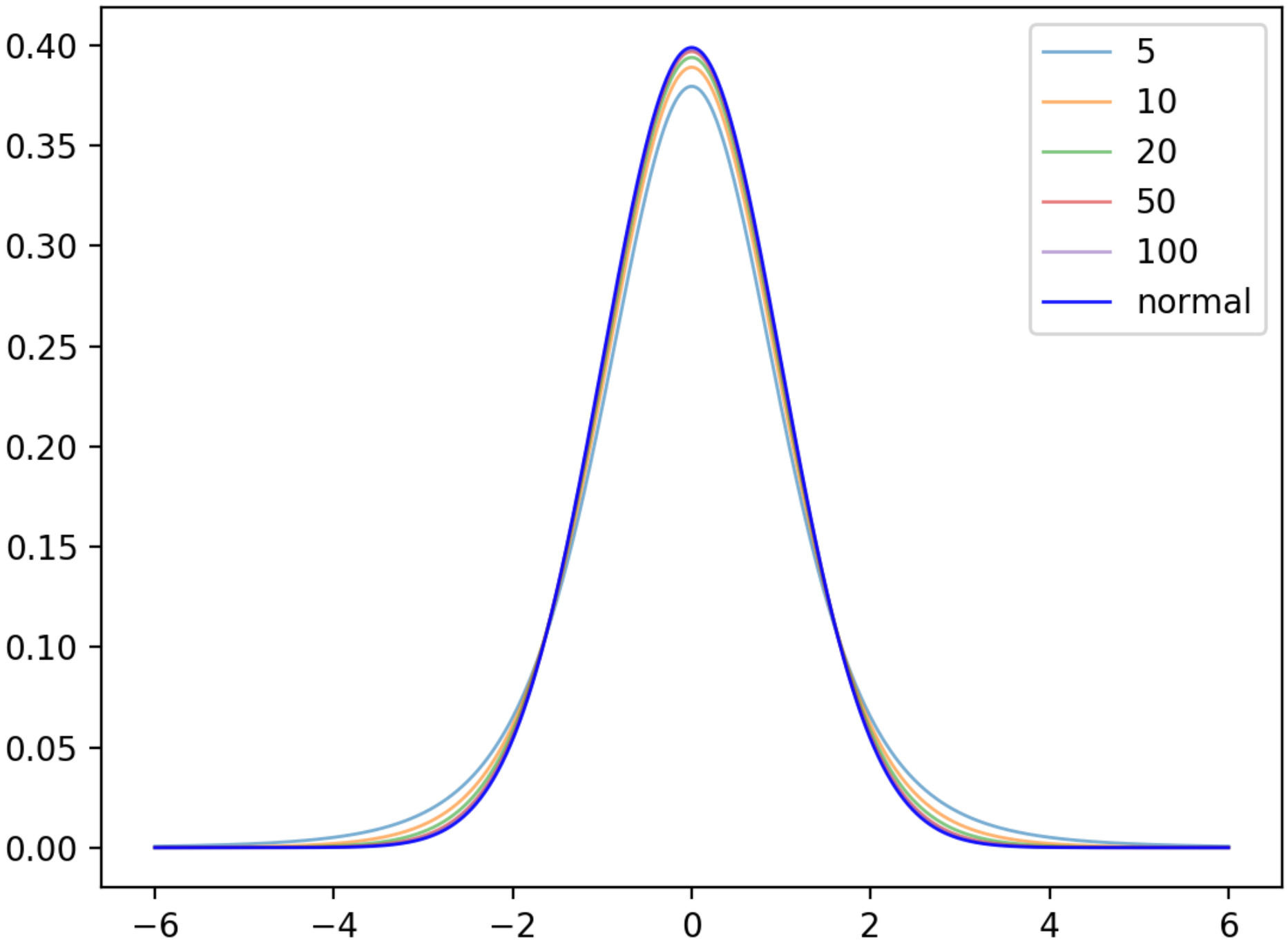
Student *t*-distribution

- Gossett (writing as "Student")
    - Distribution of

$$\frac{(m-\mu)}{s/\sqrt{N}}$$

    - With increasing N comes close to normal distribution

Student-*t* distribution



Simple Linear Regression

- Linear regression uses straight lines for prediction
  - Model:
    - "Causal variable" x, "observed variable" y
    - Connection is linear (with or without a constant)
    - There is an additive "error" component
      - Subsuming "unknown" causes
      - With expected value of 0
      - Usually assumed to be normally distributed

Simple Linear Regression

- Model:

$$y = b_0 + b_1 x + \epsilon$$

Simple Linear Regression

- Assume $y = b_0 + b_1 x$

  Minimize $S = \sum_{i=1}^{n} (y_i - (b_0 + b_1 x_i))^2$

  Take the derivative with respect to b_0 and set it to zero:

$$\frac{\delta S}{\delta b_0} = \sum_{i=1}^{n} -2(y_i - b_0 - b_1 x_i) = 0$$

$$\Rightarrow \sum_{i=1}^{n} y_i = b_0 n + b_1 \sum_{i=1}^{n} x_i \Rightarrow \; b_0 = \frac{1}{n}\sum_{i=1}^{n} y_i - b_1 \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\Rightarrow b_0 = \bar{y} - b_1 \bar{x}$$

Simple Linear Regression

- Assume $y = b_0 + b_1 x$

$$\text{Minimize } S = \sum_{i=1}^{n} (y_i - \left(b_0 + b_1 x_i\right))^2$$

Take the derivative with respect to b_1 and set it to zero:

$$\frac{\delta S}{\delta b_1} = \sum_{i=1}^{n} - 2x_i(y_i - b_0 - b_1 x_i) = 0$$

$$\Rightarrow \sum_{i=1}^{n} x_i(y_i - b_0 - b_1 x_i) = \sum_{i=1}^{n} (x_i y_i - b_0 x_i - b_1 x_i^2) = 0$$

Simple Linear Regression

From previous, we know $b_0 = \bar{y} - b_1\bar{x}$

Our formula $\sum_{i=1}^{n}(x_i y_i - b_0 x_i - b_1 x_i^2) = 0$ becomes

$$\sum_{i=1}^{n}\left(x_i y_i - (\bar{y} - b_1\bar{x})x_i - b_1 x_i^2\right) = 0$$

$$\sum_{i=1}^{n}\left(x_i y_i - \bar{y}x_i\right) + b_1\sum_{i=1}^{n}\left(\bar{x}x_i - x_i^2\right) = 0$$

Simple Linear Regression

- This finally gives us a solution:

$$b_1 = \frac{\sum_{i=1}^{n}(x_i y_i - \bar{y} x_i)}{\sum_{i=1}^{n}(x_i^2 - \bar{x} x_i)}$$

$$b_0 = \bar{y} - b_1 \bar{x}$$

Simple Linear Regression

- Measuring fit:

Calculate the sum of squares $SS_{\text{tot}} = \sum_{i=1}^{n} (y_i - \bar{y})^2$

Residual sum of squares $SS_{res} = \sum_{i=1}^{n} (b_0 + b_1 x_i - y_i)^2$

Coefficient of determination $R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$

Simple Linear Regression

- R^2 can be used as a goodness of fit
  - Value of 1: perfect fit
  - Value of 0: no fit
  - Negative values: wrong model was chosen

Simple Linear Regression

- Look at residuals:
  - Determine statistics on the residuals
  - Question: do they look normally distributed?

Simple Linear Regression

- Example 1: brain sizes versus IQ
  - A number of female students were given an IQ test
  - They were also given an MRI to measure the size of their brain
- Is there a relationship between brains size and IQ?

VerbalIQ	Brain Size
132	816.932
132	951.545
90	928.799
136	991.305
90	854.258
129	833.868
120	856.472
100	878.897
71	865.363
132	852.244
112	808.02
129	790.619
86	831.772
90	798.612
83	793.549
126	866.662
126	857.782
90	834.344
129	948.066
86	893.983

Simple Linear Regression

- Can use statsmodels

```
import statsmodels.api as sm

df = pd.read_csv('brain-size.txt', sep = '\t')
Y = df['VerbalIQ']
X = df['Brain Size']
X = sm.add_constant(X)

model = sm.OLS(Y,X).fit()
predictions = model.predict(X)
print(model.summary())
```

Simple Linear Regression

- Gives a very detailed feed-back

OLS Regression Results

Dep. Variable:	VerbalIQ	R-squared:	0.065
Model:	OLS	Adj. R-squared:	0.013
Method:	Least Squares	F-statistic:	1.251
Date:	Thu, 02 Jul 2020	Prob (F-statistic):	0.278
Time:	16:22:00	Log-Likelihood:	-88.713
No. Observations:	20	AIC:	181.4
Df Residuals:	18	BIC:	183.4
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	24.1835	76.382	0.317	0.755	-136.288	184.655
Brain Size	0.0988	0.088	1.119	0.278	-0.087	0.284

Omnibus:	5.812	Durbin-Watson:	2.260
Prob(Omnibus):	0.055	Jarque-Bera (JB):	1.819
Skew:	-0.259	Prob(JB):	0.403
Kurtosis:	1.616	Cond. No.	1.37e+04

Simple Linear Regression

- Interpreting the outcome:

Omnibus:	5.812	Durbin-Watson:	2.260
Prob(Omnibus):	0.055	Jarque-Bera (JB):	1.819
Skew:	-0.259	Prob(JB):	0.403
Kurtosis:	1.616	Cond. No.	1.37e+04

- Are the residuals normally distributed?
- Omnibus: test for skew and kurtosis
  - Should be zero
- In this case: Probability of this or worse is 0.055

Simple Linear Regression

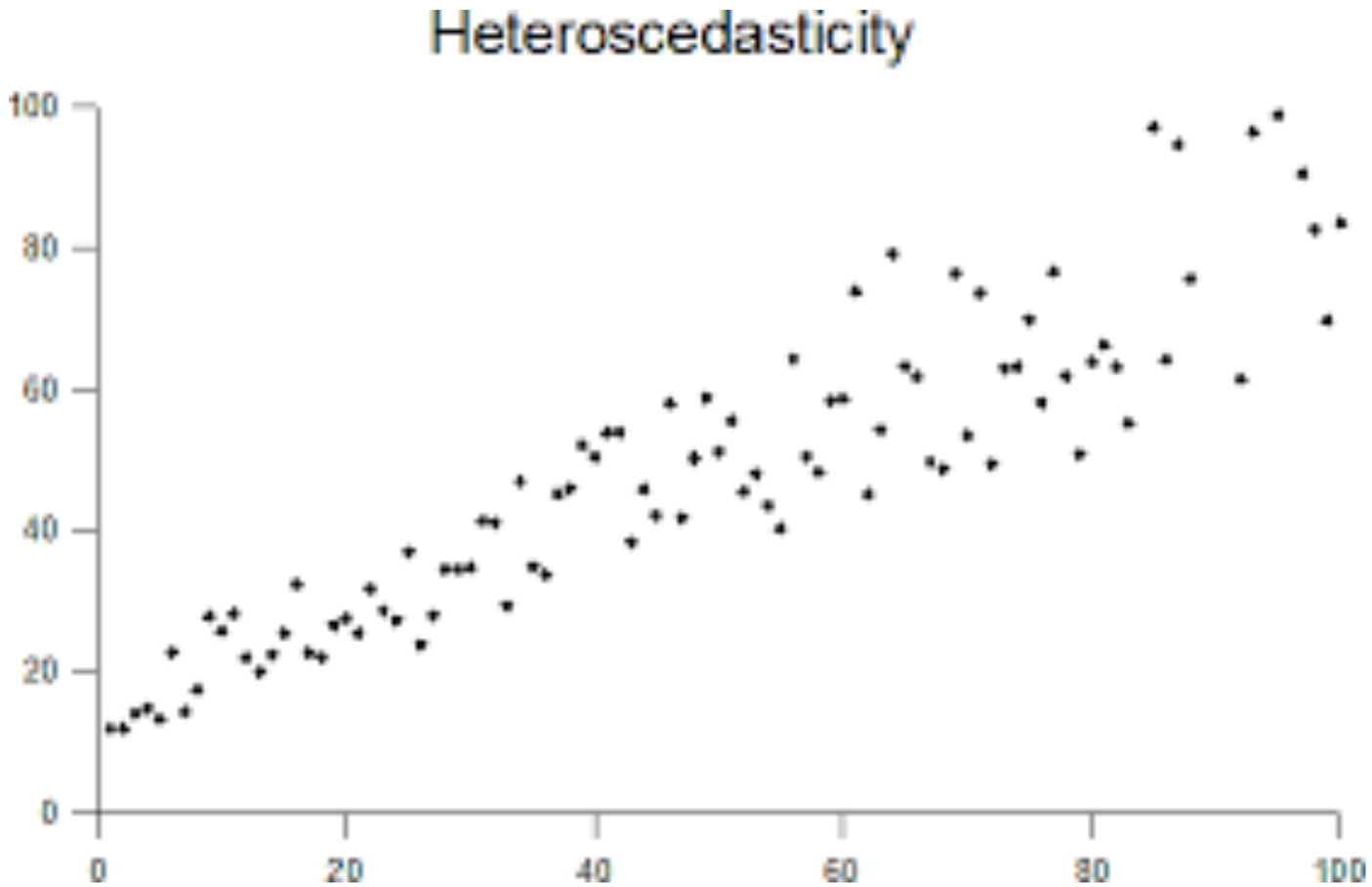
- Interpreting the outcome:

Omnibus:	5.812	Durbin-Watson:	2.260
Prob(Omnibus):	0.055	Jarque-Bera (JB):	1.819
Skew:	-0.259	Prob(JB):	0.403
Kurtosis:	1.616	Cond. No.	1.37e+04

- Are the residuals normally distributed?
- Durbin-Watson: tests homoscedasticity
  - Is the Variance of the errors consistent

Simple Linear Regression

- Homoscedasticity
  - Observe that variance increases



Simple Linear Regression

- Interpreting the outcome:

Omnibus:	5.812	Durbin-Watson:	2.260
Prob(Omnibus):	0.055	Jarque-Bera (JB):	1.819
Skew:	-0.259	Prob(JB):	0.403
Kurtosis:	1.616	Cond. No.	1.37e+04

- Jarque-Bera:
  - Tests skew and kurtosis of residuals
  - Here acceptable probability

Simple Linear Regression

- Interpreting the outcome:

Omnibus:	5.812	Durbin-Watson:	2.260
Prob(Omnibus):	0.055	Jarque-Bera (JB):	1.819
Skew:	-0.259	Prob(JB):	0.403
Kurtosis:	1.616	Cond. No.	1.37e+04

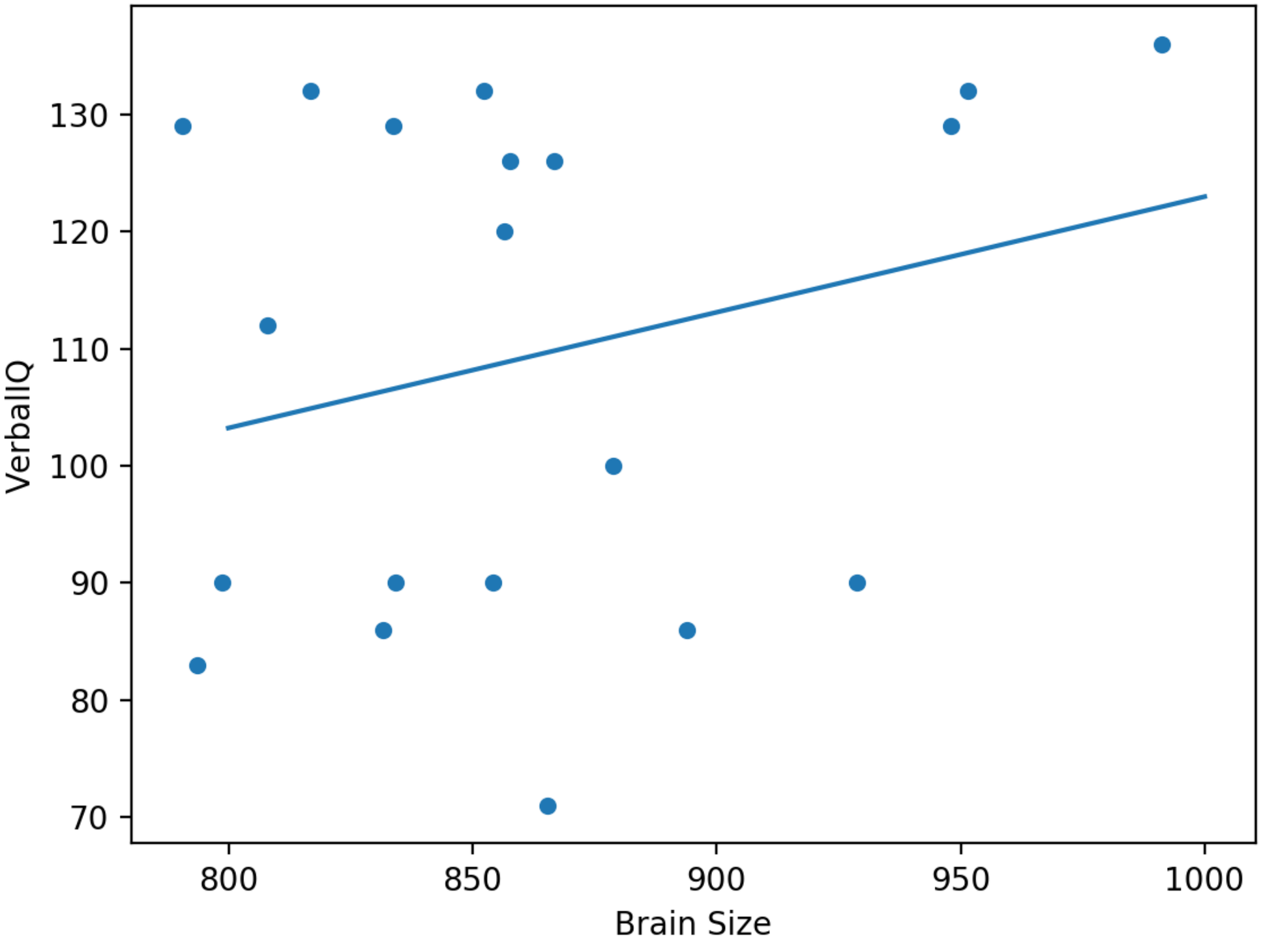
  - Condition number
    - Indicates either multicollinearity or numerical problems

Simple Linear Regression

- Plotting

```
my_ax = df.plot.scatter(x='Brain Size', y='VerbalIQ')

x=np.linspace(start=800,stop=1000)
my_ax.plot(x,24.1835+0.0988*x)
```

Simple Linear Regression



Simple Linear Regression

- scipy has a stats package

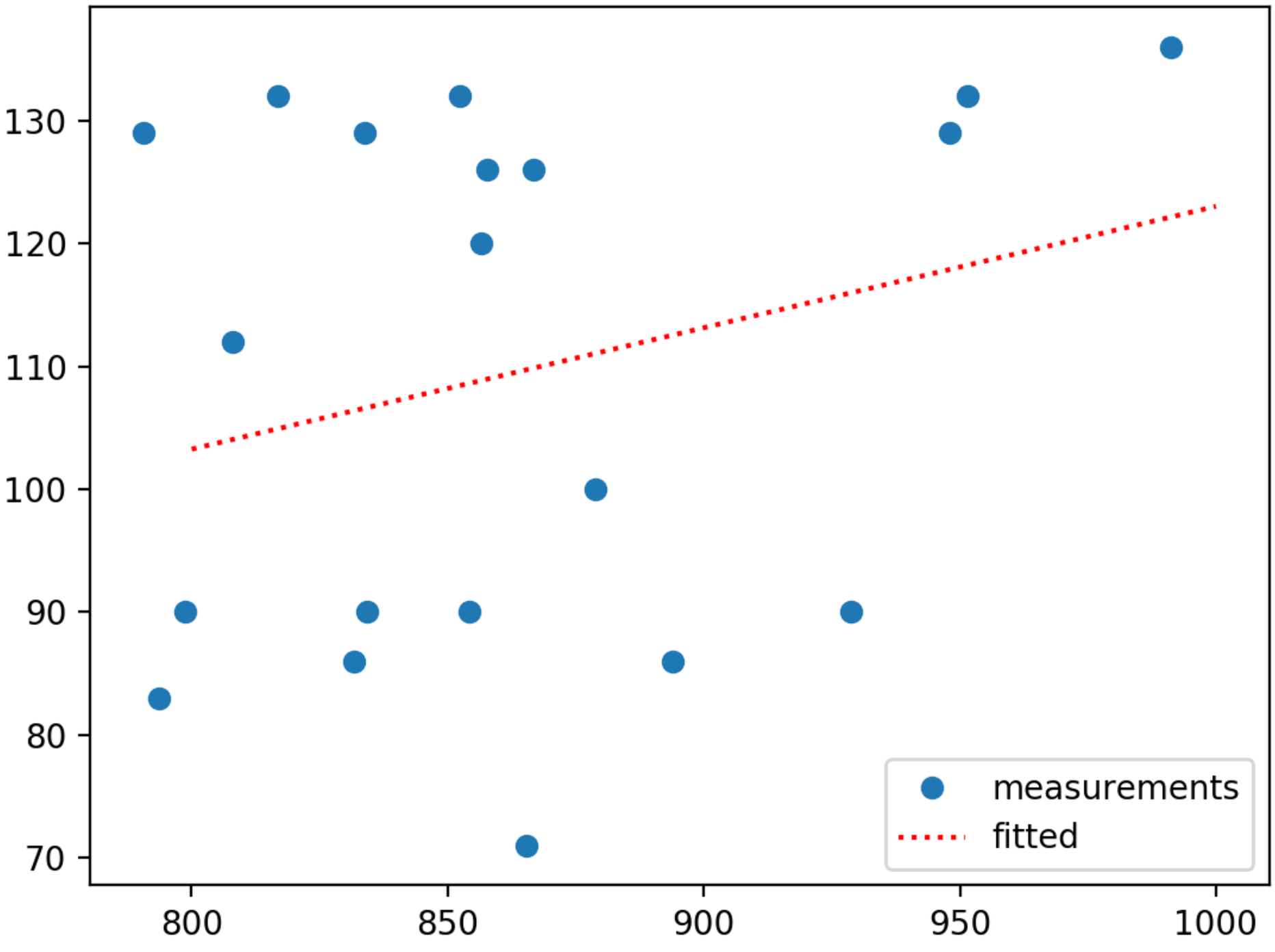
```
from scipy import stats

df = pd.read_csv('brain-size.txt', sep = '\t')
Y = df['VerbalIQ']
X = df['Brain Size']
x = np.linspace(800,1000)

slope, intercept, r_value, p_value, std_err =
stats.linregress(X, Y)
```

Simple Linear Regression

- plotting using plt

```
plt.plot(X, Y, 'o', label='measurements')
plt.plot(x, intercept+slope*x, 'r:', label='fitted')
plt.legend(loc='lower right')
print(slope, intercept, r_value, p_value)
plt.show()
```

Simple Linear Regression



Multiple Regression

- Assume now more explanatory variables
- $y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_r x_r$

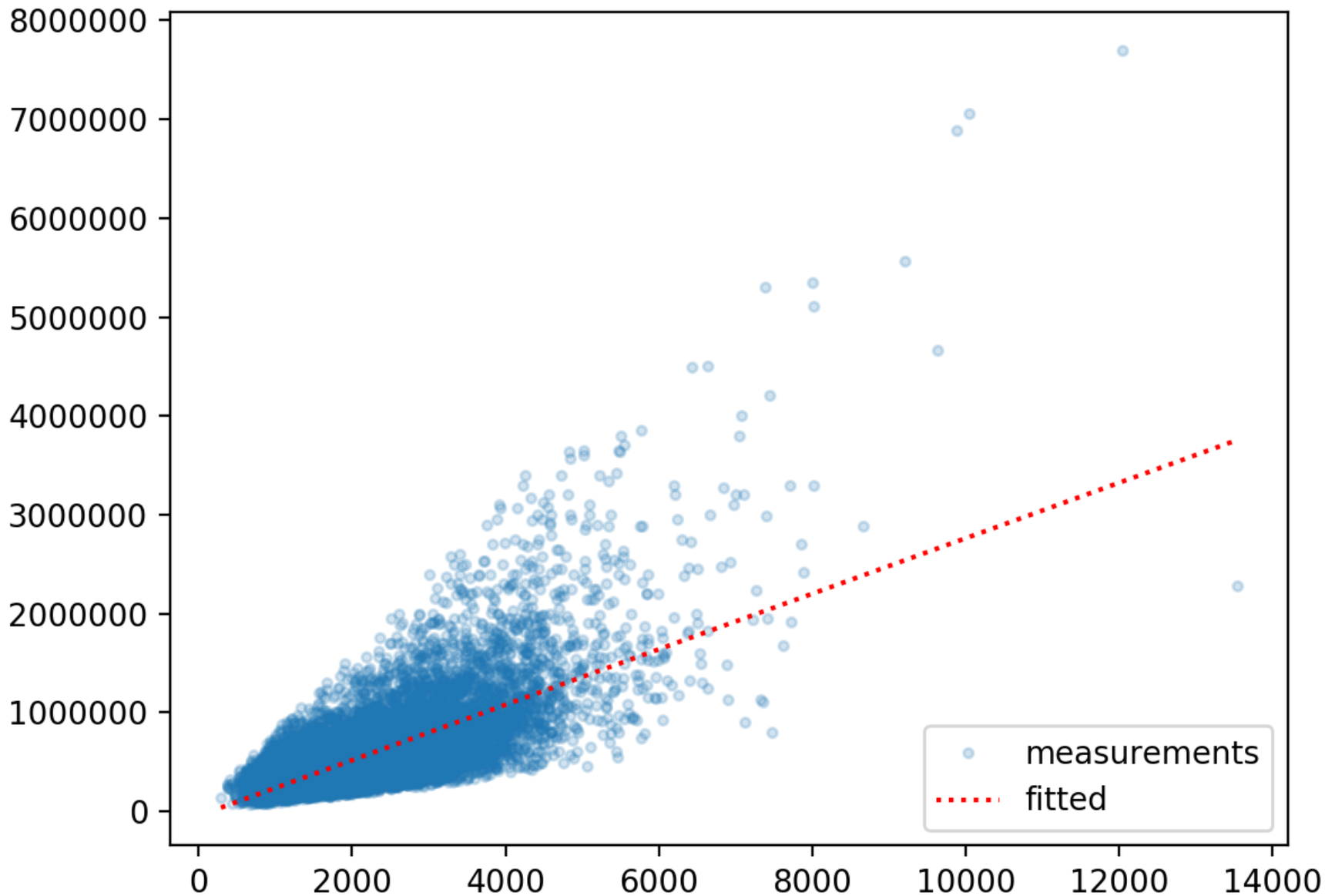
Multiple Regression

- Seattle Housing Market
  - Data from Kaggle

```
df = pd.read_csv('kc_house_data.csv')
df.dropna( inplace=True)
```

Multiple Regression

- Linear regression: price — grade



Multiple Regression

- Can use the same Pandas recipes

```
df = pd.read_csv('kc_house_data.csv')
df.dropna( inplace=True)
Y = df['price']
X = df[ ['sqft_living', 'bedrooms', 'condition',
'waterfront'] ]


model = sm.OLS(Y,X).fit()
predictions = model.predict(X)
print(model.summary())
```

Multiple Regression

OLS Regression Results

Dep. Variable:	price	R-squared (uncentered):	0.857
Model:	OLS	Adj. R-squared (uncentered):	0.857
Method:	Least Squares	F-statistic:	3.231e+04
Date:	Thu, 02 Jul 2020	Prob (F-statistic):	0.00
Time:	20:47:11	Log-Likelihood:	-2.9905e+05
No. Observations:	21613	AIC:	5.981e+05
Df Residuals:	21609	BIC:	5.981e+05
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
sqft_living	303.8804	2.258	134.598	0.000	299.455	308.306
bedrooms	-5.919e+04	2062.324	-28.703	0.000	-6.32e+04	-5.52e+04
condition	3.04e+04	1527.531	19.901	0.000	2.74e+04	3.34e+04
waterfront	7.854e+05	1.96e+04	40.043	0.000	7.47e+05	8.24e+05

Omnibus:	13438.261	Durbin-Watson:	1.985
Prob(Omnibus):	0.000	Jarque-Bera (JB):	437567.612
Skew:	2.471	Prob(JB):	0.00
Kurtosis:	24.482	Cond. No.	2.65e+04

Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly

Multiple Regression

- sklearn

```
from sklearn import linear_model

df = pd.read_csv('kc_house_data.csv')
df.dropna( inplace=True)
Y = df['price']
X = df[ ['sqft_living', 'bedrooms', 'condition',
'waterfront'] ]

regr = linear_model.LinearRegression()
regr.fit(X, Y)

print('Intercept: \n', regr.intercept_)
print('Coefficients: \n', regr.coef_)
```

Polynomial Regression

- What if the explanatory variables enter as powers?
- Can still apply multi-linear regression

$$y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_1^2 + b_4 x_1 x_2 + b_5 x_2^2$$