1. Linear Regression
The Least Squares Method

Bruce E. Shapiro

Getting Started in Machine Learning

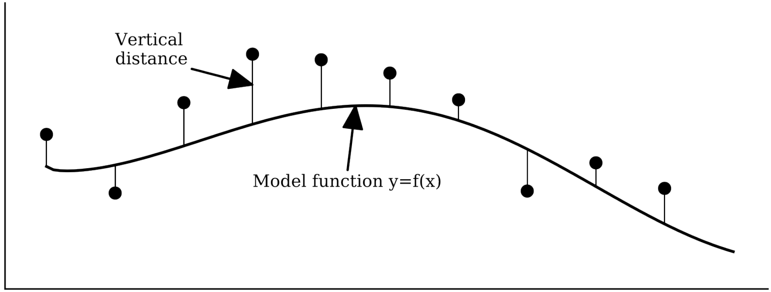
Last revised: February 3, 2019

Goal: Find curve $f(x)$ that gives the **best fit** to the points

$$(x_0, y_0), (x_1, y_1), (x_2, y_2), \ldots, (x_{n-1}, y_{n-1})$$

such that the **objective function** is minimized:

$$\mathcal{E} = \frac{1}{2} \sum_{i=0}^{n-1} (f(x_i) - y_i)^2$$



The form of $f(x)$ is specified by the user.
If $f(x) = a + bx$ then

$$\mathcal{E} = \frac{1}{2}\sum_{i=0}^{n-1}(a + bx_i - y_i)^2$$

The form of $f(x)$ is specified by the user.
If $f(x) = a + bx$ then

$$\mathcal{E} = \frac{1}{2}\sum_{i=0}^{n-1}(a + bx_i - y_i)^2$$

If $f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_k x^k$ then

$$\mathcal{E} = \frac{1}{2}\sum_{i=0}^{n-1}(a_0 + a_1 x_i + a_2 x_i^2 + \cdots + a_k x_i^k - y_i)^2$$

The form of $f(x)$ is specified by the user.
If $f(x) = a + bx$ then

$$\mathcal{E} = \frac{1}{2}\sum_{i=0}^{n-1}(a + bx_i - y_i)^2$$

If $f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_kx^k$ then

$$\mathcal{E} = \frac{1}{2}\sum_{i=0}^{n-1}(a_0 + a_1x_i + a_2x_i^2 + \cdots + a_kx_i^k - y_i)^2$$

If $f(x) = \dfrac{x^k}{a^k + x^k}$ then

$$\mathcal{E} = \frac{1}{2}\sum_{i=0}^{n-1}(\frac{x_i^k}{a^k + x_i^k} - y_i)^2$$

Each form for $f(x)$ depends on some unknown **constants** (also called **parameters**) that must be fit to the data.

If $f(x) = a + bx$ then the parameters are a, b.

Each form for $f(x)$ depends on some unknown **constants** (also called **parameters**) that must be fit to the data.

If $f(x) = a + bx$ then the parameters are a, b.

If $f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_k x^k$ then the unknown parameters are $a_0, a_1, \ldots, a_k$.

Each form for $f(x)$ depends on some unknown **constants** (also called **parameters**) that must be fit to the data.

If $f(x) = a + bx$ then the parameters are a, b.

If $f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_k x^k$ then the unknown parameters are $a_0, a_1, \ldots, a_k$.

If $f(x) = \dfrac{x^k}{a^k + x^k}$ then the unknowns are a and k.

We can determine the parameters by setting the partial derivatives $\mathcal{E}$ with respect to each partial equal to zero. For $f(x) = a + bx$,

$$\frac{\partial \mathcal{E}}{\partial a} = 0, \qquad \frac{\partial \mathcal{E}}{\partial b} = 0$$

For $f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_k x^k$, we solve

$$\frac{\partial \mathcal{E}}{\partial a_0} = 0, \qquad \frac{\partial \mathcal{E}}{\partial a_0} = 0, \qquad \ldots, \qquad \frac{\partial \mathcal{E}}{\partial a_k} = 0$$

For $f(x) = x^k/(a^k + x^k)$, we solve

$$\frac{\partial \mathcal{E}}{\partial a} = 0, \qquad \frac{\partial \mathcal{E}}{\partial k} = 0$$

- If the objective function is linear in the parameters then the system of partials will be a system of linear equations that can be solved exactly.
  - $y = a + bx$ - explicit solution commonly used
  - Polynomial - exact solution by linear methods

- If the objective function is linear in the parameters then the system of partials will be a system of linear equations that can be solved exactly.
  - $y = a + bx$ - explicit solution commonly used
  - Polynomial - exact solution by linear methods
- If the objective function is nonlinear
  - usually can't be solved analytically
  - numerical solvers like gradient descent
  - danger of falling into wrong minimum
  - Hill function, logistic regression, exponential

Linear Regression: Solution From Statistics

- Let
$$\mu_x = \frac{1}{n}\sum_{i=0}^{n-1} x_i, \qquad \mu_y = \frac{1}{n}\sum_{i=0}^{n-1} y_i \qquad \text{means}$$

Linear Regression: Solution From Statistics

- Let

$$\mu_x = \frac{1}{n}\sum_{i=0}^{n-1} x_i, \qquad \mu_y = \frac{1}{n}\sum_{i=0}^{n-1} y_i \qquad \text{means}$$

$$\sigma_x^2 = \frac{1}{n-1}\sum_{i=0}^{n-1}(x_i - \overline{x})^2, \qquad \sigma_y^2 = \frac{1}{n-1}\sum_{i=0}^{n-1}(y_i - \overline{y})^2 \qquad \text{Std. Devs.}$$

Linear Regression: Solution From Statistics

- Let

$$\mu_x = \frac{1}{n}\sum_{i=0}^{n-1} x_i, \qquad \mu_y = \frac{1}{n}\sum_{i=0}^{n-1} y_i \qquad \text{means}$$

$$\sigma_x^2 = \frac{1}{n-1}\sum_{i=0}^{n-1}(x_i - \overline{x})^2, \qquad \sigma_y^2 = \frac{1}{n-1}\sum_{i=0}^{n-1}(y_i - \overline{y})^2 \qquad \text{Std. Devs.}$$

$$r = \frac{1}{(n-1)\sigma_x\sigma_y}\sum_{i=0}^{n-1}(x_i - \overline{x})(y_i - \overline{y}) \qquad \text{correlation}$$

Linear Regression: Solution From Statistics

- Let

$$\mu_x = \frac{1}{n}\sum_{i=0}^{n-1} x_i, \qquad \mu_y = \frac{1}{n}\sum_{i=0}^{n-1} y_i \qquad \text{means}$$

$$\sigma_x^2 = \frac{1}{n-1}\sum_{i=0}^{n-1}(x_i-\overline{x})^2, \qquad \sigma_y^2 = \frac{1}{n-1}\sum_{i=0}^{n-1}(y_i-\overline{y})^2 \qquad \text{Std. Devs.}$$

$$r = \frac{1}{(n-1)\sigma_x\sigma_y}\sum_{i=0}^{n-1}(x_i-\overline{x})(y_i-\overline{y}) \qquad \text{correlation}$$

- Then $y = a + bx$ where

$$b = r\sigma_y/\sigma_x, \qquad a = \mu_y - b\mu_x$$

Example: One way to do linear regression

```
import numpy as np
def linear_regression(x,y):
    mux=np.mean(x)
    muy=np.mean(y)
    sx=np.std(x,ddof=1)
    sy=np.std(y,ddof=1)
    r=np.corrcoef(x,y)[1,0]
    b=r*sy/sx
    a=muy-b*mux
    return(a,b,r)
a, b, r = linear_regression(xvals,yvals)
print("intercept a=",a)
print("slope b=    ",b)
print("correlation=",r)
```

Example

```
import pandas as pd
data = pd.read_csv("first-csv-file.csv")
data
```

	x	y
0	1.1	11.70
1	2.2	12.19
2	2.9	16.04
3	4.1	14.63
4	5.1	17.31
5	6.2	18.49
6	7.2	19.62
7	8.3	19.45

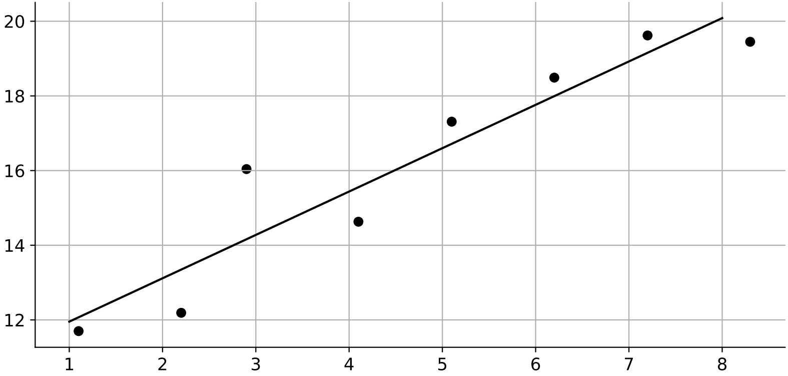
Example (continued)

```
a, b, r = linear_regression(data["x"],data["y"])
print("intercept a=",a)
print("slope b=     ",b)
print("correlation=",r)
```

```
intercept a= 10.791852248092571
slope b=     1.1615952025676397
correlation= 0.9417178321883759
```

```
plt.scatter(data["x"],data["y"], c="k")
plt.plot([1,8],[intercept + slope*1, intercept+slope*8], c="k"
plt.grid()
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.tick_params(axis = 'both', which = 'major', labelsize = 12)
```

Example (continued)



A Second Way to Do Linear Regression in Python

```
from scipy.stats import linregress as LR
LR(xvals,yvals)
```

```
LinregressResult(slope=1.1615952025676397,
   intercept=10.791852248092571,
   rvalue=0.941717832188376,
   pvalue=0.00047355160326094863,
   stderr=0.16940229837546086)
```

or

```
m,b,r,p,s=LR(xvals,yvals)
print((m,b,r,p,s))
```

```
(1.1615952025676397, 10.791852248092571, 0.941717832188
0.00047355160326094863, 0.16940229837546086)
```

Return values for linregress

returned	Description
slope	slope of line
intercept	y intercept
rvalue	correlation
pvalue	p-value of $H_0 : b = 0$ vs $H_A : b \neq 0$
stderr	standard error of slope

Third Way to do Linear Regression in Python

```
input numpy as np
np.polyfit(xvals,yvals,1)
```

```
array([ 1.1615952 , 10.79185225])
```

Linear Regression

Goal: Fit points $\{(x_i, y_i))\}_{0 \le i < n}$ to $y = a + bx$ where we minimize the **objective function**

$$\mathcal{E} = \frac{1}{2} \sum_{i=0}^{n-1} (a + bx_i - y_i)^2$$

Linear Regression

Goal: Fit points $\{(x_i, y_i))\}_{0 \le i < n}$ to $y = a + bx$ where we minimize the **objective function**

$$\mathcal{E} = \frac{1}{2} \sum_{i=0}^{n-1} (a + bx_i - y_i)^2$$

Process: set $\partial\mathcal{E}/\partial a = 0$ and $\partial\mathcal{E}/\partial b = 0$, solve for a, b

Equations for Partials

$$\mathcal{E} = \frac{1}{2}\sum_{i=0}^{n-1}(a + bx_i - y_i)^2$$

Equations for Partials

$$\mathcal{E} = \frac{1}{2}\sum_{i=0}^{n-1}(a + bx_i - y_i)^2$$

$$0 = \frac{\partial \mathcal{E}}{\partial a} = \frac{\partial}{\partial a}\frac{1}{2}\sum_{i=1}^{n}(a + bx_i - y_i)^2 = \sum_{i=1}^{n}(a + bx_i - y_i)$$

Equations for Partials

$$
\begin{aligned}
\mathcal{E} &= \frac{1}{2}\sum_{i=0}^{n-1}(a + bx_i - y_i)^2 \\
0 = \frac{\partial \mathcal{E}}{\partial a} &= \frac{\partial}{\partial a}\frac{1}{2}\sum_{i=1}^{n}(a + bx_i - y_i)^2 = \sum_{i=1}^{n}(a + bx_i - y_i) \\
&= \sum_{i=1}^{n} a + \sum_{i=1}^{n} bx_i - \sum_{i=1}^{n} y_i
\end{aligned}
$$

Equations for Partials

$$
\begin{aligned}
\mathcal{E} &= \frac{1}{2}\sum_{i=0}^{n-1}(a + bx_i - y_i)^2 \\
0 = \frac{\partial \mathcal{E}}{\partial a} &= \frac{\partial}{\partial a}\frac{1}{2}\sum_{i=1}^{n}(a + bx_i - y_i)^2 = \sum_{i=1}^{n}(a + bx_i - y_i) \\
&= \sum_{i=1}^{n} a + \sum_{i=1}^{n} bx_i - \sum_{i=1}^{n} y_i = na + b\sum_{i=1}^{n} x_i - \sum_{i=1}^{n} y_i
\end{aligned}
$$

Equations for Partials

$$
\begin{aligned}
\mathcal{E} &= \frac{1}{2}\sum_{i=0}^{n-1}(a + bx_i - y_i)^2 \\
0 = \frac{\partial \mathcal{E}}{\partial a} &= \frac{\partial}{\partial a}\frac{1}{2}\sum_{i=1}^{n}(a + bx_i - y_i)^2 = \sum_{i=1}^{n}(a + bx_i - y_i) \\
&= \sum_{i=1}^{n} a + \sum_{i=1}^{n} bx_i - \sum_{i=1}^{n} y_i = na + b\sum_{i=1}^{n} x_i - \sum_{i=1}^{n} y_i \\
0 = \frac{\partial \mathcal{E}}{\partial b} &= \frac{\partial}{\partial b}\frac{1}{2}\sum_{i=1}^{n}(a + bx_i - y_i)^2 = \sum_{i=1}^{n} x_i\,(a + bx_i - y_i)
\end{aligned}
$$

Equations for Partials

$$\mathcal{E} = \frac{1}{2}\sum_{i=0}^{n-1}(a + bx_i - y_i)^2$$

$$\begin{aligned}
0 = \frac{\partial \mathcal{E}}{\partial a} &= \frac{\partial}{\partial a}\frac{1}{2}\sum_{i=1}^{n}(a + bx_i - y_i)^2 = \sum_{i=1}^{n}(a + bx_i - y_i) \\
&= \sum_{i=1}^{n} a + \sum_{i=1}^{n} bx_i - \sum_{i=1}^{n} y_i = na + b\sum_{i=1}^{n} x_i - \sum_{i=1}^{n} y_i \\
0 = \frac{\partial \mathcal{E}}{\partial b} &= \frac{\partial}{\partial b}\frac{1}{2}\sum_{i=1}^{n}(a + bx_i - y_i)^2 = \sum_{i=1}^{n} x_i\,(a + bx_i - y_i) \\
&= a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i y_i
\end{aligned}$$

Solve Equations for Partials

Two Equations in two unknowns: a, b:

$$na + b\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i$$

$$a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i$$

Solve Equations for Partials

Two Equations in two unknowns: a, b:

$$na + b\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i$$

$$a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i$$

Make substitutions to simplify the algebra:

$$X = \sum_{i=1}^{n} x_i, \quad Y = \sum_{i=1}^{n} y_i, \quad A = \sum_{i=1}^{n} x_i^2, \quad C = \sum_{i=1}^{n} x_i y_i$$

Solve Equations for Partials

Two Equations in two unknowns: a, b:

$$na + b\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i$$

$$a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i$$

Make substitutions to simplify the algebra:

$$X = \sum_{i=1}^{n} x_i, \quad Y = \sum_{i=1}^{n} y_i, \quad A = \sum_{i=1}^{n} x_i^2, \quad C = \sum_{i=1}^{n} x_i y_i$$

Simplifies the equation to:

$$na + bX = Y$$

$$aX + bA = C$$

Solution by row reduction

$$\begin{bmatrix} n & X \\ X & A \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} Y \\ C \end{bmatrix}$$

Solution by row reduction

$$\begin{bmatrix} n & X \\ X & A \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} Y \\ C \end{bmatrix}$$

$$\begin{bmatrix} n & X & Y \\ X & A & C \end{bmatrix}$$ Write augmented matrix

Solution by row reduction

$$\begin{bmatrix} n & X \\ X & A \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} Y \\ C \end{bmatrix}$$

$$\begin{bmatrix} n & X & Y \\ X & A & C \end{bmatrix}$$ Write augmented matrix

$$\begin{bmatrix} nX & X^2 & XY \\ nX & nA & nC \end{bmatrix} \qquad \begin{matrix} R1 \to X \times R1 \\ R2 \to n \times R2 \end{matrix}$$

Solution by row reduction

$$\begin{bmatrix} n & X \\ X & A \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} Y \\ C \end{bmatrix}$$

$$\begin{bmatrix} n & X & Y \\ X & A & C \end{bmatrix}$$ Write augmented matrix

$$\begin{bmatrix} nX & X^2 & XY \\ nX & nA & nC \end{bmatrix} \qquad \begin{matrix} R1 \to X \times R1 \\ R2 \to n \times R2 \end{matrix}$$

$$\begin{bmatrix} nX & X^2 & XY \\ 0 & nA - X^2 & nC - XY \end{bmatrix} \qquad R2 \to R2 - R1$$

Solution by row reduction

$$\begin{bmatrix} n & X \\ X & A \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} Y \\ C \end{bmatrix}$$

$$\begin{bmatrix} n & X & Y \\ X & A & C \end{bmatrix}$$ Write augmented matrix

$$\begin{bmatrix} nX & X^2 & XY \\ nX & nA & nC \end{bmatrix} \qquad \begin{matrix} R1 \to X \times R1 \\ R2 \to n \times R2 \end{matrix}$$

$$\begin{bmatrix} nX & X^2 & XY \\ 0 & nA - X^2 & nC - XY \end{bmatrix} \qquad R2 \to R2 - R1$$

$$\begin{bmatrix} nX(nA - X^2) & X^2(nA - X^2) & XY(nA - X^2) \\ 0 & X^2(nA - X^2) & X^2(nC - XY) \end{bmatrix} \qquad \begin{matrix} R1 \to (nA - X^2) \times R1 \\ R2 \to X^2 \times R2 \end{matrix}$$

Solution by row reduction

$$\begin{bmatrix} n & X \\ X & A \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} Y \\ C \end{bmatrix}$$

$$\begin{bmatrix} n & X & Y \\ X & A & C \end{bmatrix} \quad \text{Write augmented matrix}$$

$$\begin{bmatrix} nX & X^2 & XY \\ nX & nA & nC \end{bmatrix} \quad \begin{array}{c} R1 \to X \times R1 \\ R2 \to n \times R2 \end{array}$$

$$\begin{bmatrix} nX & X^2 & XY \\ 0 & nA - X^2 & nC - XY \end{bmatrix} \quad R2 \to R2 - R1$$

$$\begin{bmatrix} nX(nA - X^2) & X^2(nA - X^2) & XY(nA - X^2) \\ 0 & X^2(nA - X^2) & X^2(nC - XY) \end{bmatrix} \quad \begin{array}{c} R1 \to (nA - X^2) \times R1 \\ R2 \to X^2 \times R2 \end{array}$$

$$\begin{bmatrix} nX(nA - X^2) & 0 & nX(YA - XC) \\ 0 & X^2(nA - X^2) & X^2(nC - XY) \end{bmatrix} \quad R1 \to R1 - R2$$

Row Reduction (continued)

$$\begin{bmatrix} nX(nA - X^2) & 0 & nX(YA - XC) \\ 0 & X^2(nA - X^2) & X^2(nC - XY) \end{bmatrix}$$ (copy from prev. page)

Row Reduction (continued)

$$\begin{bmatrix} nX(nA-X^2) & 0 & nX(YA-XC) \\ 0 & X^2(nA-X^2) & X^2(nC-XY) \end{bmatrix} \quad \text{(copy from prev. page)}$$

$$\begin{bmatrix} 1 & 0 & \dfrac{YA-XC}{nA-X^2} \\ 0 & 1 & \dfrac{nC-XY}{nA-X^2} \end{bmatrix} \quad \begin{array}{l} R1 \to \dfrac{R1}{nX(nA-X^2)} \\ R2 \to \dfrac{R2}{X^2(nA-X^2)} \end{array}$$

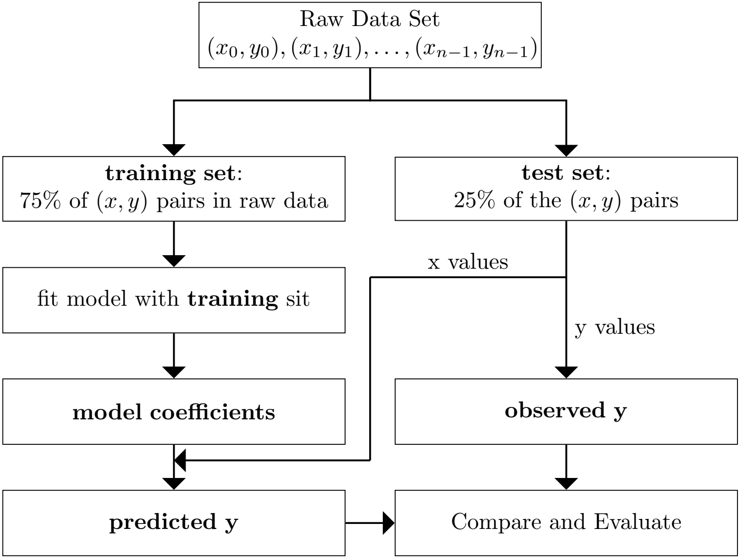
Row Reduction (continued)

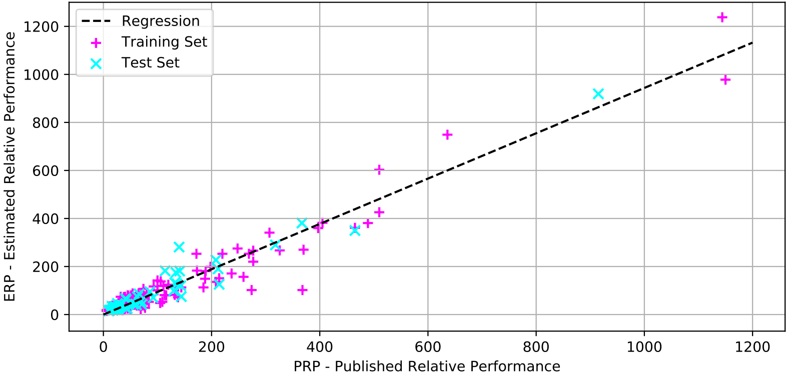
$$\begin{bmatrix} nX(nA-X^2) & 0 & nX(YA-XC) \\ 0 & X^2(nA-X^2) & X^2(nC-XY) \end{bmatrix} \quad \text{(copy from prev. page)}$$

$$\begin{bmatrix} 1 & 0 & \dfrac{YA-XC}{nA-X^2} \\ 0 & 1 & \dfrac{nC-XY}{nA-X^2} \end{bmatrix} \quad \begin{array}{l} R1 \to \dfrac{R1}{nX(nA-X^2)} \\ R2 \to \dfrac{R2}{X^2(nA-X^2)} \end{array}$$

Therefore:

$$a = \frac{YA-XC}{nA-X^2}, \qquad b = \frac{nC-XY}{nA-X^2}$$


Raw Data Set
$(x_0, y_0), (x_1, y_1), \ldots, (x_{n-1}, y_{n-1})$
training set:
75% of (x, y) pairs in raw data
test set:
25% of the (x, y) pairs
x values
fit model with **training** sit
y values
model coefficients
observed y
predicted y
Compare and Evaluate


Regression
Training Set
Test Set
ERP - Estimated Relative Performance
PRP - Published Relative Performance
0
200
400
600
800
1000
1200

Example Using sklearn

```
import pandas as pd
data = pd.read_csv("https://archive.ics.uci.edu/ml/
   machine-learning-databases/cpu-performance/
   machine.data",header=None)
data.columns=["vendor","Model","MYCT","MMIN",
   "MMAX","CACH","CHMIN","CHMAX","PRP","ERP"]
print(data[:5])
```

```
vendor  Model    MYCT     MMIN  MMAX  CACH   CHMIN  CHMAX  PRP
0       adviser  32/60    125   256   6000   256    16     128
1       amdahl   470v/7   29    8000  32000  32     8      32
2       amdahl   470v/7a  29    8000  32000  32     8      32
3       amdahl   470v/7b  29    8000  32000  32     8      32
4       amdahl   470v/7c  29    8000  16000  32     8      16
```

Extract X and Y data and reshape for compatibility with sklearn

```
import numpy as np
X=np.array(data["PRP"]).reshape(-1,1)
Y=np.array(data["ERP"])
```

Extract X and Y data and reshape for compatibility with sklearn

```
import numpy as np
X=np.array(data["PRP"]).reshape(-1,1)
Y=np.array(data["ERP"])
```

Split into test and training set

```
from sklearn.model_selection import train_test_split
XTRAIN, XTEST, YTRAIN, YTEST=train_test_split(X,Y)
```

Extract X and Y data and reshape for compatibility with sklearn

```
import numpy as np
X=np.array(data["PRP"]).reshape(-1,1)
Y=np.array(data["ERP"])
```

Split into test and training set

```
from sklearn.model_selection import train_test_split
XTRAIN, XTEST, YTRAIN, YTEST=train_test_split(X,Y)
```

Perform linear regression

```
from sklearn.linear_model import LinearRegression
LR=LinearRegression()
fit=LR.fit(XTRAIN,YTRAIN)
fit.intercept_, fit.coef_
```

```
(1.4909600579608195, array([0.95410822]))
```

```
from sklearn.metrics import mean_squared_error, r2_score
YP=LR.predict(XTEST)
r2=r2_score(YTEST, YP)
mse=mean_squared_error(YTEST,YP)
r2,mse
```

```
(0.8725711655873432, 2592.260236271053)
```

```
from sklearn.metrics import mean_squared_error, r2_score
YP=LR.predict(XTEST)
r2=r2_score(YTEST, YP)
mse=mean_squared_error(YTEST,YP)
r2,mse
```

```
(0.8725711655873432, 2592.260236271053)
```

Generate plot:

```
xplot=[[0],[1200]]
yplot=LR.predict(xplot)
plt.scatter(XTRAIN, YTRAIN, color="magenta",
   marker="+",s=50,label="Training Set")
plt.scatter(XTEST, YTEST,color="cyan",marker="x",
   s=50, label="Test Set")
plt.plot(xplot,yplot,label="Regression",ls="--",c="k")
plt.grid()
plt.legend()
```

Additional Material: Conversion to Statistical Form

$$(n-1)\sigma_x^2 = \sum (x - \mu_x)^2$$

Additional Material: Conversion to Statistical Form

$$(n-1)\sigma_x^2 = \sum(x-\mu_x)^2 = \sum(x^2) - \sum(2x\mu_x) + \sum(\mu_x)^2$$

Additional Material: Conversion to Statistical Form

$$
\begin{aligned}
(n-1)\sigma_x^2 = \sum(x-\mu_x)^2 &= \sum(x^2) - \sum(2x\mu_x) + \sum(\mu_x)^2 \\
&= \sum(x^2) - 2n\mu_x^2 + n\mu_x^2
\end{aligned}
$$

Additional Material: Conversion to Statistical Form

$$
\begin{aligned}
(n-1)\sigma_x^2 &= \sum(x-\mu_x)^2 = \sum(x^2) - \sum(2x\mu_x) + \sum(\mu_x)^2 \\
&= \sum(x^2) - 2n\mu_x^2 + n\mu_x^2 = \sum(x^2) - n\mu_x^2
\end{aligned}
$$

Additional Material: Conversion to Statistical Form

$$
\begin{aligned}
(n-1)\sigma_x^2 &= \sum(x-\mu_x)^2 = \sum(x^2) - \sum(2x\mu_x) + \sum(\mu_x)^2 \\
&= \sum(x^2) - 2n\mu_x^2 + n\mu_x^2 = \sum(x^2) - n\mu_x^2 \\
n(n-1)\sigma_x^2 &= n\sum(x^2) - n^2\mu_x^2
\end{aligned}
$$

Additional Material: Conversion to Statistical Form

$$
\begin{aligned}
(n-1)\sigma_x^2 &= \sum(x-\mu_x)^2 = \sum(x^2) - \sum(2x\mu_x) + \sum(\mu_x)^2 \\
&= \sum(x^2) - 2n\mu_x^2 + n\mu_x^2 = \sum(x^2) - n\mu_x^2 \\
n(n-1)\sigma_x^2 &= n\sum(x^2) - n^2\mu_x^2 = n\sum(x^2) - \left(\sum x\right)^2
\end{aligned}
$$

Additional Material: Conversion to Statistical Form

$$
\begin{aligned}
(n-1)\sigma_x^2 &= \sum(x-\mu_x)^2 = \sum(x^2) - \sum(2x\mu_x) + \sum(\mu_x)^2 \\
&= \sum(x^2) - 2n\mu_x^2 + n\mu_x^2 = \sum(x^2) - n\mu_x^2 \\
n(n-1)\sigma_x^2 &= n\sum(x^2) - n^2\mu_x^2 = n\sum(x^2) - \left(\sum x\right)^2 = nA - X^2
\end{aligned}
$$

Additional Material: Conversion to Statistical Form

$$
\begin{aligned}
(n-1)\sigma_x^2 &= \sum(x-\mu_x)^2 = \sum(x^2) - \sum(2x\mu_x) + \sum(\mu_x)^2 \\
&= \sum(x^2) - 2n\mu_x^2 + n\mu_x^2 = \sum(x^2) - n\mu_x^2 \\
n(n-1)\sigma_x^2 &= n\sum(x^2) - n^2\mu_x^2 = n\sum(x^2) - \left(\sum x\right)^2 = nA - X^2 \\
(n-1)\sigma_x\sigma_y r &= \sum(x-\mu_x)(x-\mu_y)
\end{aligned}
$$

Additional Material: Conversion to Statistical Form

$$
\begin{aligned}
(n-1)\sigma_x^2 &= \sum (x-\mu_x)^2 = \sum (x^2) - \sum (2x\mu_x) + \sum (\mu_x)^2 \\
&= \sum (x^2) - 2n\mu_x^2 + n\mu_x^2 = \sum (x^2) - n\mu_x^2 \\
n(n-1)\sigma_x^2 &= n\sum (x^2) - n^2\mu_x^2 = n\sum (x^2) - \Big(\sum x\Big)^2 = nA - X^2 \\
(n-1)\sigma_x\sigma_y r &= \sum (x-\mu_x)(x-\mu_y) = \sum (xy) - \mu_y \sum x - \mu_x \sum y + n\mu_x\mu
\end{aligned}
$$

Additional Material: Conversion to Statistical Form

$$
\begin{aligned}
(n-1)\sigma_x^2 &= \sum(x-\mu_x)^2 = \sum(x^2) - \sum(2x\mu_x) + \sum(\mu_x)^2 \\
&= \sum(x^2) - 2n\mu_x^2 + n\mu_x^2 = \sum(x^2) - n\mu_x^2 \\
n(n-1)\sigma_x^2 &= n\sum(x^2) - n^2\mu_x^2 = n\sum(x^2) - \Big(\sum x\Big)^2 = nA - X^2 \\
(n-1)\sigma_x\sigma_y r &= \sum(x-\mu_x)(x-\mu_y) = \sum(xy) - \mu_y\sum x - \mu_x\sum y + n\mu_x\mu \\
&= C - n\mu_x\mu_y - n\mu_x\mu_y + n\mu_x\mu_y = C - n\mu_x\mu_y
\end{aligned}
$$

Additional Material: Conversion to Statistical Form

$$
\begin{aligned}
(n-1)\sigma_x^2 &= \sum(x-\mu_x)^2 = \sum(x^2) - \sum(2x\mu_x) + \sum(\mu_x)^2 \\
&= \sum(x^2) - 2n\mu_x^2 + n\mu_x^2 = \sum(x^2) - n\mu_x^2 \\
n(n-1)\sigma_x^2 &= n\sum(x^2) - n^2\mu_x^2 = n\sum(x^2) - \Big(\sum x\Big)^2 = nA - X^2 \\
(n-1)\sigma_x\sigma_y r &= \sum(x-\mu_x)(x-\mu_y) = \sum(xy) - \mu_y\sum x - \mu_x\sum y + n\mu_x\mu \\
&= C - n\mu_x\mu_y - n\mu_x\mu_y + n\mu_x\mu_y = C - n\mu_x\mu_y \\
&= C - n(X/n)(Y/n) = C - XY/n
\end{aligned}
$$

Additional Material: Conversion to Statistical Form

$$
\begin{aligned}
(n-1)\sigma_x^2 &= \sum(x-\mu_x)^2 = \sum(x^2) - \sum(2x\mu_x) + \sum(\mu_x)^2 \\
&= \sum(x^2) - 2n\mu_x^2 + n\mu_x^2 = \sum(x^2) - n\mu_x^2 \\
n(n-1)\sigma_x^2 &= n\sum(x^2) - n^2\mu_x^2 = n\sum(x^2) - \left(\sum x\right)^2 = nA - X^2 \\
(n-1)\sigma_x\sigma_y r &= \sum(x-\mu_x)(x-\mu_y) = \sum(xy) - \mu_y\sum x - \mu_x\sum y + n\mu_x\mu \\
&= C - n\mu_x\mu_y - n\mu_x\mu_y + n\mu_x\mu_y = C - n\mu_x\mu_y \\
&= C - n(X/n)(Y/n) = C - XY/n \\
n(n-1)\sigma_x\sigma_y r &= nC - XY
\end{aligned}
$$

Additional Material: Conversion to Statistical Form

$$
\begin{aligned}
(n-1)\sigma_x^2 &= \sum(x-\mu_x)^2 = \sum(x^2) - \sum(2x\mu_x) + \sum(\mu_x)^2 \\
&= \sum(x^2) - 2n\mu_x^2 + n\mu_x^2 = \sum(x^2) - n\mu_x^2 \\
n(n-1)\sigma_x^2 &= n\sum(x^2) - n^2\mu_x^2 = n\sum(x^2) - \Big(\sum x\Big)^2 = nA - X^2 \\
(n-1)\sigma_x\sigma_y r &= \sum(x-\mu_x)(x-\mu_y) = \sum(xy) - \mu_y\sum x - \mu_x\sum y + n\mu_x\mu \\
&= C - n\mu_x\mu_y - n\mu_x\mu_y + n\mu_x\mu_y = C - n\mu_x\mu_y \\
&= C - n(X/n)(Y/n) = C - XY/n \\
n(n-1)\sigma_x\sigma_y r &= nC - XY \\
b &= \frac{XY - nC}{X^2 - nA} = \frac{-n(n-1)\sigma_x\sigma_y r}{-n(n-1)\sigma_x^2} = -\frac{\sigma_y}{\sigma_x} r
\end{aligned}
$$

From the previous calculations,

$$a = \frac{AY - XC}{nA - X^2}$$

From the previous calculations,

$$a = \frac{AY - XC}{nA - X^2}$$
$$A = \sum x^2 = \sum (x - \mu_x)^2 + \sum 2\mu_x x - \sum \mu_x^2 = (n-1)\sigma_x^2 + n\mu_x^2$$

From the previous calculations,

$$
\begin{aligned}
a &= \frac{AY - XC}{nA - X^2} \\
A &= \sum x^2 = \sum (x - \mu_x)^2 + \sum 2\mu_x x - \sum \mu_x^2 = (n-1)\sigma_x^2 + n\mu_x^2 \\
C &= \sum xy = \sum (x - \mu_x)(y - \mu_y) - \sum x\mu_y - \sum y\mu_x + \sum \mu_x\mu_y \\
&= (n-1)\sigma_x\sigma_y r + n\mu_x\mu_y
\end{aligned}
$$

From the previous calculations,

$$
\begin{aligned}
a &= \frac{AY - XC}{nA - X^2} \\
A &= \sum x^2 = \sum (x-\mu_x)^2 + \sum 2\mu_x x - \sum \mu_x^2 = (n-1)\sigma_x^2 + n\mu_x^2 \\
C &= \sum xy = \sum (x-\mu_x)(y-\mu_y) - \sum x\mu_y - \sum y\mu_x + \sum \mu_x\mu_y \\
&= (n-1)\sigma_x\sigma_y r + n\mu_x\mu_y \\
AY - XC &= ((n-1)\sigma_x^2 + n\mu_x^2)n\mu_y - ((n-1)\sigma_x\sigma_y r + n\mu_x\mu_y)(n\mu_x) \\
&= n(n-1)\sigma_x^2\mu_y - n(n-1)\sigma_x\sigma_y\mu_x \\
&= n(n-1)\sigma_x^2\Big[\mu_y - \frac{\sigma_y}{\sigma_x}\mu_x\Big] = (nA - X^2)(\mu_y - b\mu_x)
\end{aligned}
$$

From the previous calculations,

$$
\begin{aligned}
a &= \frac{AY - XC}{nA - X^2} \\
A &= \sum x^2 = \sum (x - \mu_x)^2 + \sum 2\mu_x x - \sum \mu_x^2 = (n-1)\sigma_x^2 + n\mu_x^2 \\
C &= \sum xy = \sum (x - \mu_x)(y - \mu_y) - \sum x\mu_y - \sum y\mu_x + \sum \mu_x\mu_y \\
&= (n-1)\sigma_x\sigma_y r + n\mu_x\mu_y \\
AY - XC &= ((n-1)\sigma_x^2 + n\mu_x^2)n\mu_y - ((n-1)\sigma_x\sigma_y r + n\mu_x\mu_y)(n\mu_x) \\
&= n(n-1)\sigma_x^2\mu_y - n(n-1)\sigma_x\sigma_y\mu_x \\
&= n(n-1)\sigma_x^2\left[\mu_y - \frac{\sigma_y}{\sigma_x}\mu_x\right] = (nA - X^2)(\mu_y - b\mu_x) \\
a &= \mu_y - b\mu_x
\end{aligned}
$$

Citations

1. Dua, D. and Karra Taniskidou, E. (2017). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
2. Kibler,D. & Aha,D. (1988). Instance-Based Prediction of Real-Valued Attributes. In Proceedings of the CSCSI (Canadian AI) Conference. (data set: [https://archive.ics.uci.edu/ml/datasets/Computer+Hardware])