

# Regression 1

## Simple Linear Regression Model

### 1 What is regression?

The word was *originally* used by an English statistician Galton. In his 1885 paper to Royal Anthropological Institute, he derived regression and used it interchangeably with the word reversion.

regression  $\equiv$  regression to the mean (*e.g.*, Galton 1885).

**Example 1.1.** *Regression towards the mean.*

- Sports stars tend to have a poorer season following a really good one.
- Split the class into two groups – top 50% and lower 50%. We can expect the lower group to do better and the top group to do worse.

||

**Definition 1.1.** *Regression Analysis* is a statistical methodology that analyzes the relation between two or more quantitative variables so that one variable can be predicted from other(s). That is to say, it is equivalent to find a function relation of the form

$$Y = f(X).$$

---

## 2 Simple linear regression model

We will start with the simplest regression model (simple linear regression model),

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i.$$

Here *simple* means a single regressor variable  $X$ . The  $Y$  variables are called *dependent* or *response* variables, and the  $X$ 's are called *independent variables*, *explanatory variables*, *regressors*, *covariates* (usually continuous type), *control variables*, or *predictor variables*. The parameters  $\beta_0$  and  $\beta_1$  are called *regression coefficients*.

Our goal is to find  $\beta_0$  and  $\beta_1$ . It is natural to find  $\beta_0$  and  $\beta_1$  which minimize the errors  $\epsilon_i = Y_i - (\beta_0 + \beta_1 X_i)$ .

### Statistical assumptions and properties

Assumptions:  $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$

- $X_i$  is treated as a known constant.
- $\epsilon_i$  is a *random error* with:
  1.  $E(\epsilon_i) = 0$  (the errors tend to compensate each other).  
If  $E(\epsilon_i) = \alpha$ , then  $\beta_0^* = \alpha + \beta_0$  can absorb  $\alpha$ .
  2.  $\text{Var}(\epsilon_i) = \sigma^2$  (for each observation the error has the same variance).
  3.  $\text{Cov}(\epsilon_i, \epsilon_j) = 0$  for  $i \neq j$  (the errors corresponding to difference observations are uncorrelated so they do not influence each other).

Properties:

- The response variable  $Y_i$  is the sum of two components called constant term ( $f(X_i) = \beta_0 + \beta_1 X_i$ ) and random term ( $\epsilon_i$ ). The first term  $\beta_0 + \beta_1 X_i$  represents the basic relation between the variables and the latter  $\epsilon_i$  represents the disturbances (errors) which affect this relation.

- Even if  $X_i$  is assumed to be known and fixed,  $Y_i$  is a random variable for the presence of  $\epsilon_i$ .

- The expected value of  $Y_i$  is equal to the constant term

$$E(Y_i) = E(\beta_0 + \beta_1 X_i + \epsilon_i) = \beta_0 + \beta_1 X_i,$$

and so, in practice, there is a linear relation between  $E(Y)$  and  $X$  ;

- The variance of  $Y_i$  is equal to that of the error term

$$\text{Var}(Y_i) = \text{Var}(\beta_0 + \beta_1 X_i + \epsilon_i) = \text{Var}(\epsilon_i) = \sigma^2.$$

- $Y_i$  and  $Y_j$  ( $i \neq j$ ) are uncorrelated,

$$\text{Cov}(Y_i, Y_j) = \text{Cov}(\epsilon_i, \epsilon_j) = 0,$$

and so the measure of the response variable in correspondence of a subject does not affect the measure in correspondence of another subject.

## 3 Estimation of regression function

### 3.1 Parameter estimation by least squares methods

Let us denote

$$Q_1 = \sum_{i=1}^n |\epsilon_i| = \sum_{i=1}^n |Y_i - (\beta_0 + \beta_1 X_i)|$$

$$Q_2 = \sum_{i=1}^n \epsilon_i^2 = \sum_{i=1}^n \{Y_i - (\beta_0 + \beta_1 X_i)\}^2$$

$$Q_3 = \sum_{i=1}^n \epsilon_i = \sum_{i=1}^n \{Y_i - (\beta_0 + \beta_1 X_i)\}.$$

Minimizing  $Q_1$  is called  $L_1$  regression and minimizing  $Q_2$  is  $L_2$  or least-squares regression.

Then can we use  $Q_3$ ? *Answer*: no.

We will focus on a  $L_2$  regression. Differentiate  $Q_2$  with respect to  $\beta_0$  and  $\beta_1$  to get

$$\frac{\partial Q_2}{\partial \beta_0} = -2 \sum_{i=1}^n \{Y_i - (\beta_0 + \beta_1 X_i)\} = 0 \quad (1.1)$$

$$\frac{\partial Q_2}{\partial \beta_1} = -2 \sum_{i=1}^n X_i \{Y_i - (\beta_0 + \beta_1 X_i)\} = 0. \quad (1.2)$$

Let us denote  $\hat{\beta}_0$  (or  $b_0$ ) and  $\hat{\beta}_1$  (or  $b_1$ ) to be the solution of the above equations which are called normal equations. The solution  $\hat{\beta}_0$  and  $\hat{\beta}_1$  are called *point estimators* of  $\beta_0$  and  $\beta_1$ .

The normal equations can be solved simultaneously:

$$\hat{\beta}_1 = b_1 = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sum (X_i - \bar{X})^2} = \frac{\sum X_i Y_i - n \bar{X} \bar{Y}}{\sum X_i^2 - n \bar{X}^2} = \frac{S_{xy}}{S_{xx}} \quad (1.3)$$

$$\hat{\beta}_0 = b_0 = \frac{1}{n} \left\{ \sum_{i=1}^n Y_i - \hat{\beta}_1 \sum_{i=1}^n X_i \right\} = \bar{Y} - \hat{\beta}_1 \bar{X}, \quad (1.4)$$

where  $\bar{X} = (1/n) \sum_{i=1}^n X_i$  and  $\bar{Y} = (1/n) \sum_{i=1}^n Y_i$ .

The  $\hat{\beta}_0$  and  $\hat{\beta}_1$  are unbiased. That is

$$E(\hat{\beta}_0) = \beta_0 \quad \text{and} \quad E(\hat{\beta}_1) = \beta_1.$$

### 3.2 Normal error regression model and MLE

By minimizing  $Q_2$ , we found the parameters of the  $L_2$  regression model under the assumption that  $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$ , where  $\epsilon_i$  satisfy that

1.  $E(\epsilon_i) = 0$ ,
2.  $\text{Var}(\epsilon_i) = \sigma^2$ , and
3.  $\text{Cov}(\epsilon_i, \epsilon_j) = 0$  for  $i \neq j$ .

There are so many possible distributions of  $\epsilon_i$  satisfying  $E(\epsilon_i) = 0$ ,  $\text{Var}(\epsilon_i) = \sigma^2$ , and  $\text{Cov}(\epsilon_i, \epsilon_j) = 0$  for  $i \neq j$ . One appealing choice is  $\epsilon_i \sim N(0, \sigma^2)$ . If the distribution of the error terms is specified, estimators of the parameters  $\beta_0$ ,  $\beta_1$  and  $\sigma^2$  can be obtained by the method of *maximum likelihood*.

**Fact.** *It is shown that  $\text{Cov}(U, V) = 0$  if the random variables  $U$  and  $V$  are independent. The converse, in general, is not true (see Example 1.2). However, if the random variables  $U$  and  $V$  have a bivariate normal distribution, then  $\text{Cov}(U, V) = 0$  guarantees that  $U$  and  $V$  are independent.*

---

*Proof.* For more details, see Theorem 4.5-1 of Hogg et al. (2015). Also, refer to § 7.8.2 of Ross (2014) and Theorem 4.6.12 and Lemma 5.3.3 of Casella and Berger (2002).  $\square$

**Example 1.2.** A probability mass function

$$f(u, v) = 1/4$$

for  $(u, v) = (0, 1), (1, 0), (0, -1), (-1, 0)$ . The marginal  $f_1(1) = f_1(-1) = 1/4$ ,  $f_1(0) = 1/2$  and  $f_2(1) = f_2(-1) = 1/4$ ,  $f_2(0) = 1/2$ . Then we have  $\text{Cov}(U, V) = 0$  but  $U$  and  $V$  are not independent (i.e.,  $f(u, v) \neq f_1(u)f_2(v)$ ).  $\parallel$

Since  $\epsilon_i \sim N(0, \sigma^2)$  and  $\epsilon_i$  and  $\epsilon_j$  ( $i \neq j$ ) are uncorrelated,  $\epsilon_i$  are independent and identically distributed (iid) with  $N(0, \sigma^2)$ . The joint pdf of  $(\epsilon_1, \epsilon_2, \dots, \epsilon_n)$  is

$$f(\epsilon_1, \epsilon_2, \dots, \epsilon_n) = \prod_{i=1}^n f(\epsilon_i).$$

Hence the likelihood function  $L(\beta_0, \beta_1, \sigma^2)$  is given by

$$\begin{aligned} L(\beta_0, \beta_1, \sigma^2) &= \prod_{i=1}^n f(\epsilon_i) \\ &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} \exp \left[ -\frac{1}{2\sigma^2} (Y_i - \beta_0 - \beta_1 X_i)^2 \right] \\ &= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp \left[ -\frac{1}{2\sigma^2} \sum_{i=1}^n (Y_i - \beta_0 - \beta_1 X_i)^2 \right]. \end{aligned}$$

In general, it is easy to deal with the log likelihood  $l(\cdot) = \ln L(\cdot)$  rather than  $L(\cdot)$ . The log likelihood is given by

$$l(\beta_0, \beta_1, \sigma^2) = \text{Constant} - \frac{n}{2} \ln \sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^n (Y_i - \beta_0 - \beta_1 X_i)^2.$$

Hence, if we assume  $\epsilon_i \sim N(0, \sigma^2)$ , then the MLE of the regression coefficients  $\beta_0$  and  $\beta_1$  are the same as the least-squares estimators. It should be made clear that if we can not make the normal assumption, the least squares estimator is not the MLE. Note that if we assume that  $\epsilon_i$  comes from the double exponential distribution, the MLE of the  $\beta_0$  and  $\beta_1$  are the same as the  $L_1$  estimators (i.e., minimizing  $Q_1$ ).

---

### 3.3 Estimation of mean response

Given parameter estimators  $\hat{\beta}_0$  and  $\hat{\beta}_1$  in the regression function:

$$\mu_Y = E(Y) = \beta_0 + \beta_1 X,$$

we can estimate  $\mu_Y$  as  $\hat{\mu}_Y = \hat{\beta}_0 + \hat{\beta}_1 X$ . For convenience, we denote

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X.$$

Note that for convenience  $\hat{Y}$  is widely used instead of  $\hat{\mu}_Y$  or  $\widehat{E(Y)}$ . We call a value of the response variable ( $Y$ ) a response value and  $E(Y)$  the mean response.  $\hat{Y}$  is a point estimator of the mean response when the level of the predictor variable is  $X$ . For the  $i$ th observed  $X_i$ , we call  $\widehat{E(Y_i)}$  the fitted value for the  $i$ th case, where

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i, \quad i = 1, \dots, n.$$

Note that for convenience  $\hat{Y}_i$  is widely used instead of  $\widehat{E(Y_i)}$ .

### 3.4 Residuals

The  $i$ th residual is the difference between the observed value  $Y_i$  and the corresponding fitted value  $\hat{Y}_i$ . This residual is denoted by  $\hat{\epsilon}_i$  (or  $e_i$ ) and is defined as

$$\hat{\epsilon}_i = e_i = Y_i - \hat{Y}_i.$$

Note that  $\epsilon_i = Y_i - E(Y_i)$ .

The estimated simple linear regression by the least-squares method has the following properties:

1.  $\sum_{i=1}^n \hat{\epsilon}_i = 0$ .
2.  $\sum_{i=1}^n \hat{\epsilon}_i^2$  is a minimum.
3.  $\sum_{i=1}^n Y_i = \sum_{i=1}^n \hat{Y}_i$ .

---


$$4. \sum_{i=1}^n X_i \hat{\epsilon}_i = 0.$$

$$5. \sum_{i=1}^n \hat{Y}_i \hat{\epsilon}_i = 0.$$

$$6. \sum_{i=1}^n Y_i \hat{\epsilon}_i = \sum_{i=1}^n \hat{\epsilon}_i^2.$$

7. The regression line always passes through the point  $(\bar{X}, \bar{Y})$ .

## 4 Estimating $\sigma^2$

The variance  $\sigma^2$  of the error term  $\epsilon_i$  in regression needs to be estimated to obtain an indication of the variability of the probability distributions of  $Y$ .

Let us look at the simplest case, where we just have repeated observations  $Y_1, \dots, Y_n$ . The variability in the  $Y$ 's is given by the usual sample variance:

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (Y_i - \bar{Y})^2.$$

This is made unbiased by dividing by  $n-1$  not  $n$ . In regression, we use

$$\text{MSE} = \frac{1}{n-2} \sum_{i=1}^n (\hat{\epsilon}_i - \bar{\hat{\epsilon}})^2 = \frac{1}{n-2} \sum_{i=1}^n (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i)^2.$$

Notice that  $\bar{\hat{\epsilon}} = \frac{1}{n} \sum \hat{\epsilon}_i = 0$ . Why  $(n-2)$  is used instead of  $(n-1)$  or  $n$ ? To make MSE unbiased *i.e.*,  $E(\text{MSE}) = \sigma^2$ . The denominator term  $(n-2)$  is degrees of freedom (the number of quantities that are free to vary).

We can also estimate  $\sigma^2$  from the log likelihood  $l(\beta_0, \beta_1, \sigma^2)$ ,

$$\frac{\partial l(\beta_0, \beta_1, \sigma^2)}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} \sum (Y_i - \beta_0 - \beta_1 X_i)^2 = 0.$$

Solving for  $\sigma^2$ , we have

$$\hat{\sigma}_M^2 = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i)^2.$$

Thus,  $\hat{\sigma}_M^2$  is biased.

---

## 5 Example I

A company produces refrigeration equipment and its replacement parts. In the past, one of the replacement parts has been produced periodically in different size lots. The company is interested in the optimum lot size. The data in the first column are different lot sizes and those in the second column are their corresponding work hours required to produce the lot(Kutner et al., 2005).

Minitab

### (1) Write and Read Data

---

```
1 MTB > # -----
2 MTB > # Write the data into C1 and C2 variables directly.
3 MTB > # -----
4 MTB > READ C1 C2 .
5 DATA>      80  399
6 DATA>      30  121
7 .....
8 .....
9 .....
10 DATA>      80  342
11 DATA>      70  323
12 DATA> END .
13 25 rows read.
14
15 MTB > # Write C1 only
16 MTB > SET C1 .
17 DATA>      80  30  50 90 70 60 120 80 100 50 40 70 90
18 DATA>      20 110 100 30 50 90 110 30  90 40 80 70
19 DATA> END .
20
21 MTB > # Write C2 only
22 MTB > SET C2 .
23 DATA>  399 121 221 376 361 224 546 352 353 157 160 252 389
24 DATA>  113 435 420 212 268 377 421 273 468 244 342 323
25 DATA> END .
26 MTB >
27 MTB > # Double check if they are read well
28 MTB > PRINT C1 C2.
29
30 Data Display
31 Row    C1    C2
32   1    80   399
33   2    30   121
34 .....
35 .....
36  25    70   323
37
38 MTB > READ C1 C2;
39 SUBC> file "S:\LM\CH01TA01.txt" .
40 Entering data from file: S:\LM\CH01TA01.TXT
41 25 rows read.
42
43 MTB > # Double check if they are read well
44 MTB > PRINT C1-C2 .
45
46 Data Display
47 Row    C1    C2
48   1    80   399
49   2    30   121
50 .....
```



```

51 .....
52 25 70 323

```

## (2) Scatter plot

```

1 MTB > GSTD .
2 * NOTE * The character graph commands are obsolete.
3 * NOTE * Standard Graphics are now enabled, and Professional Graphics are
4   * disabled. Use the GPRO command when you want to re-enable
5   * Professional Graphics.
6
7 MTB > PLOT C2 C1 .
8 Scatterplot
9
10
11 C2
12
13
14 450+
15
16
17
18
19 300+
20
21
22
23
24 150+
25
26
27 -----+-----+-----+-----+-----+-----+-----C1
28          20          40          60          80          100          120
29
30 MTB > GPRO .
31 * NOTE * Professional Graphics are now enabled, and Standard Graphics are
32   * disabled. Use the GSTD command when you want to re-enable Standard
33   * Graphics.
34
35 MTB > PLOT C2*C1 .    # NB: not PLOT C2 C1. (* is needed).

```

## (3) Parameter estimation

```

1 MTB > # Minitab provides estimation and ANOVA as well.
2 MTB > REGR C2 1 C1;
3 SUBC> FITS C3 .
4
5 Regression Analysis: C2 versus C1
6 The regression equation is
7 C2 = 62.4 + 3.57 C1
8
9 Predictor      Coef    SE Coef      T      P
10 Constant      62.37     26.18     2.38  0.026
11 C1             3.5702    0.3470    10.29  0.000
12
13 S = 48.8233    R-Sq = 82.2%    R-Sq(adj) = 81.4%
14
15
16 Analysis of Variance
17 Source         DF      SS      MS      F      P
18 Regression      1  252378  252378  105.88  0.000
19 Residual Error  23   54825   2384
20 Total          24  307203
21
22 Unusual Observations
23 Obs   C1      C2      Fit   SE Fit   Residual   St Resid
24  21   30   273.00  169.47   16.97    103.53     2.26R
25
26 R denotes an observation with a large standardized residual.
27
28 MTB > #-----

```

---

```

29 MTB > # Scatter plot with the fitted line
30 MTB > #-----
31 MTB > GPRO .
32 * NOTE * Professional Graphics are now enabled, and Standard Graphics are
33     * disabled. Use the GSTD command when you want to re-enable Standard
34     * Graphics.
35
36 MTB > PLOT C2*C1 ;
37 SUBC> Symbol ;
38 SUBC> Regress .

```

---

#### (4) Some Calculations

---

```

1  MTB > LET C4 = C2 - C3 # No period(.) after LET command.
2  MTB > LET C5 = C4**2
3  MTB > PRINT C1-C5 .
4
5  Data Display
6  Row   C1   C2   C3   C4   C5
7    1   80  399 347.982  51.018 2602.8
8    2   30  121 169.472 -48.472 2349.5
9  .....
10 .....
11
12   25   70  323 312.280  10.720  114.9
13
14 MTB > LET C11 = SUM(C4)
15 MTB > LET C12 = SUM(C2)
16 MTB > LET C13 = SUM(C3)
17 MTB > LET C14 = C1*C4
18 MTB > LET C15 = C3*C4
19 MTB > LET C16 = C2*C4
20 MTB > LET C14 = SUM(C14)
21 MTB > LET C15 = SUM(C15)
22 MTB > LET C16 = SUM(C16)
23 MTB > PRINT C11-C16 .
24
25 Data Display
26 Row   C11   C12   C13   C14   C15   C16
27    1 -0.0000000  7807  7807 -0.0000000 -0.0000000 54825.5
28
29 MTB > ##- MSE
30 MTB > LET C21 = SUM(C5) / (25-2)
31 MTB > PRINT C21 .
32
33 Data Display
34 C21
35    2383.72

```

---

R

#### (1) Write and Read Data

---

```

1  > # -----
2  > # Write the data into C1 and C2 variables directly.
3  > # -----
4  > x1 = c(80, 30, 50, 90, 70, 60,120, 80,100, 50,
5  +       40, 70, 90, 20,110,100, 30, 50, 90,110,
6  +       30, 90, 40, 80, 70)
7  > y1 = c(399,121,221,376,361,224,546,352,353,157,
8  +       160,252,389,113,435,420,212,268,377,421,
9  +       273,468,244,342,323)
10 >
11 > # -----
12 > # From Hard disc
13 > # -----
14 > # Note: use ANSI ascii text file. UTF-8 text is not supported.
15 > mydata = read.table("S:/LM/CH01TA01.txt")
16 >

```

---

```

17 > # The above is the same as:
18 > # setwd("S:/LM")
19 > # mydata = read.table("CH01TA01.txt")
20 >
21 > # Double-check if they are read well
22 > x2 = mydata[,1]
23 > y2 = mydata[,2]
24 > cbind(x2,y2)
25       x2 y2
26 [1,]  80 399
27 [2,]  30 121
28 [3,]  50 221
29 .....
30 .....
31 .....
32 [22,]  90 468
33 [23,]  40 244
34 [24,]  80 342
35 [25,]  70 323
36 >
37 > # -----
38 > # From URL
39 > # -----
40 > # See the URL: https://github.com/AppliedStat/LM
41 > # If your computer is connected to Internet, then the following should work:
42 > url = "https://raw.githubusercontent.com/AppliedStat/LM/master/CH01TA01.txt"
43 > mydata = read.table(url)
44 >
45 > # Check below
46 > # is.matrix(mydata)
47 > # is.list(mydata)
48 > # as.matrix(mydata)
49 >
50 > # Double-check if they are read well
51 > x3 = mydata[,1]
52 > y3 = mydata[,2]
53 > cbind(x3,y3)
54       x3 y3
55 [1,]  80 399
56 [2,]  30 121
57 [3,]  50 221
58 .....
59 .....
60 .....
61 [22,]  90 468
62 [23,]  40 244
63 [24,]  80 342
64 [25,]  70 323
65 >
66 > # -----
67 > # For convenience,
68 > # -----
69 > x = x1
70 > y = y1

```

---

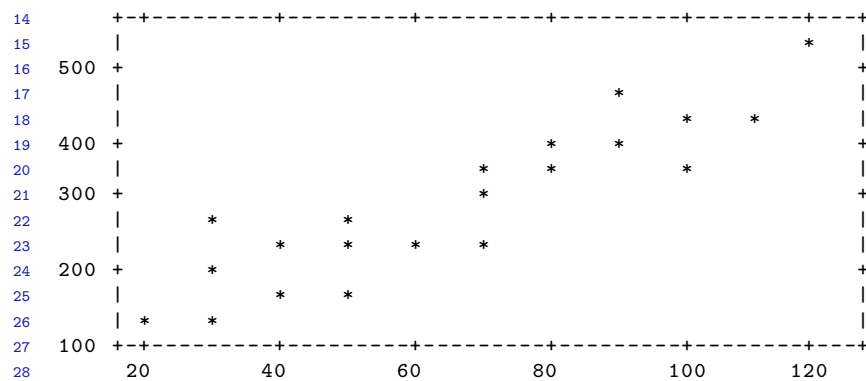
## (2) Scatter plot

---

```

1 > # ?Devices
2 > # postscript( "ex1.ps", width=4, height=4 )
3 > # pdf(file="ex1.pdf", width=4, height=4 )
4 > plot(x,y)
5 > dev.off()
6 null device
7      1
8 >
9 > # plot with text is optional in R
10 > # It may need to install txtplot package
11 > # install.packages("txtplot")
12 > library("txtplot")
13 > txtplot(x,y)

```



### (3) Parameter estimation

---

```

1 > lm( y ~ x)
2 Call:
3 lm(formula = y ~ x)
4
5 Coefficients:
6 (Intercept)          x
7    62.37         3.57
8
9 > output = lm (y ~ x)
10 > summary(output)
11 Call:
12 lm(formula = y ~ x)
13
14 Residuals:
15     Min       1Q   Median       3Q      Max
16 -83.876 -34.088  -5.982   38.826  103.528
17
18 Coefficients:
19             Estimate Std. Error t value Pr(>|t|)
20 (Intercept)   62.366     26.177    2.382  0.0259 *
21 x             3.570      0.347   10.290 4.45e-10 ***
22 ---
23 Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
24
25 Residual standard error: 48.82 on 23 degrees of freedom
26 Multiple R-squared:  0.8215, Adjusted R-squared:  0.8138
27 F-statistic: 105.9 on 1 and 23 DF, p-value: 4.449e-10
28
29 > anova(output)

```

---

```

30 Analysis of Variance Table
31
32 Response: y
33      Df Sum Sq Mean Sq F value    Pr(>F)
34 x      1 252378  252378   105.88 4.449e-10 ***
35 Residuals 23   54825    2384
36 ---
37 Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

---

#### (4) Some Calculations

---

```

1  > # To compare with Minitab codes
2  > # NB: Cx (x=1,2,...) are variables in Minitab
3  > C2 = y # copy of y to C2
4  > C1 = x
5  >
6  > C3 = fitted(output)
7  >
8  > C4 = C2 - C3
9  > C5 = C4^2
10 >
11 > cbind(C1, C2, C3, C4, C5)
12      C1  C2      C3      C4      C5
13 1    80 399 347.9820  51.0179798 2.602834e+03
14 2    30 121 169.4719 -48.4719192 2.349527e+03
15 3    50 221 240.8760 -19.8759596 3.950538e+02
16 4    90 376 383.6840 -7.6840404 5.904448e+01
17 5    70 361 312.2800  48.7200000 2.373638e+03
18 .....
19 .....
20 .....
21 21   30 273 169.4719 103.5280808 1.071806e+04
22 22   90 468 383.6840  84.3159596 7.109181e+03
23 23   40 244 205.1739  38.8260606 1.507463e+03
24 24   80 342 347.9820 -5.9820202 3.578457e+01
25 25   70 323 312.2800  10.7200000 1.149184e+02
26 >
27 > C11 = sum(C4)
28 > C12 = sum(C2)
29 > C13 = sum(C3)
30 > C14 = C1*C4
31 > C15 = C3*C4
32 > C16 = C2*C4
33 > C14 = sum(C14)
34 > C15 = sum(C15)
35 > C16 = sum(C16)
36 > C17 = sum(C5)
37 > cbind(C11, C12, C13, C14, C15, C16, C17)
38      C11  C12  C13      C14      C15      C16      C17
39 [1,] 2.273737e-13 7807 7807 1.921308e-11 7.958079e-11 54825.46 54825.46
40 >
41 > # MSE
42 > sum(C5) / (25-2)
43 [1] 2383.716
44 > sum ( (y-fitted(output))^2 ) / (25-2)
45 [1] 2383.716

```

---

### Python

#### (1) Write and Read Data

---

```

1  ~/MyFiles/teaching/IE-68722-regr/pgm> python3
2  Python 3.5.2 (default, Nov 12 2018, 13:43:14)
3  [GCC 5.4.0 20160609] on linux
4  Type "help", "copyright", "credits" or "license" for more information.
5  >>> # -----
6  ... # Write the data into x1 and y1 variables directly.
7  ... # -----

```

---

```

8 ... x1 = [ 80, 30, 50, 90, 70, 60,120, 80,100, 50,
9 ...      40, 70, 90, 20,110,100, 30, 50, 90,110,
10 ...      30, 90, 40, 80, 70 ]
11 >>> y1 = [399,121,221,376,361,224,546,352,353,157,
12 ...      160,252,389,113,435,420,212,268,377,421,
13 ...      273,468,244,342,323 ]
14 >>>
15 >>> # -----
16 ... # Read From Hard disc and write into x2 and y2
17 ... # -----
18 ... f = open("S:/LM/CH01TA01.txt", "r")
19 >>> file2 = f.read().splitlines()
20 >>> print(file2)
21 [ ' 80 399', ' 30 121', ' 50 221', ' 90 376', ' 70 361', ' 60 224',
   ' 120 546', ' 80 352', ' 100 353', ' 50 157', ' 40 160', ' 70
   252', ' 90 389', ' 20 113', ' 110 435', ' 100 420', ' 30 212', '
   50 268', ' 90 377', ' 110 421', ' 30 273', ' 90 468', ' 40 244',
   ' 80 342', ' 70 323' ]
22 >>> f.close()
23
24 >>> # Write the data in the file into x2 and y2 variables
25 ... x2 = []; y2 = [] # Make space for the data
26 >>>
27 >>> for line in file2[0:]:
28 ...     p = line.split()
29 ...     x2 = x2 + [float(p[0])]
30 ...     y2 = y2 + [float(p[1])]
31 ...
32 >>> # Double-check if they are read well
33 ... for i in range(len(x2)):
34 ...     print(x2[i],y2[i])
35 80.0 399.0
36 .....
37 .....
38 70.0 323.0
39 >>>
40 >>> # -----
41 ... # From URL
42 ... # -----
43 ... # See the URL: https://github.com/AppliedStat/LM
44 ... # If your computer is connected to Internet, then the following should work:
45 ... from urllib.request import urlopen
46 >>> link = "https://raw.githubusercontent.com/AppliedStat/LM/master/CH01TA01.txt"
47 >>> url = urlopen(link)
48 >>> file3= url.readlines()
49 >>> print(file3)
50 [b' 80 399\n', b' 30 121\n', b' 50 221\n', b' 90 376\n', b' 70
   361\n', b' 60 224\n', b' 120 546\n', b' 80 352\n', b' 100 353\n', b'
   50 157\n', b' 40 160\n', b' 70 252\n', b' 90 389\n', b' 20 113\n',
   b' 110 435\n', b' 100 420\n', b' 30 212\n', b' 50 268\n', b' 90
   377\n', b' 110 421\n', b' 30 273\n', b' 90 468\n', b' 40 244\n', b'
   80 342\n', b' 70 323\n']
51 >>> url.close()
52 >>>
53 >>> # Write the data in the file into x3 and y3 variables
54 ... x3 = [] # Make space for the data
55 ... y3 = [] # Make space for the data
56 >>> for line in file3[0:]:
57 ...     p = line.split()
58 ...     x3 = x3 + [float(p[0])]
59 ...     y3 = y3 + [float(p[1])]
60 ...     # y3 += [float(p[1])] # Same as the above
61 ...     # y3.append(float(p[1]))
62 ...
63 >>> # Double-check if they are read well
64 ... print(x3)
65 [80.0, 30.0, 50.0, 90.0, 70.0, 60.0, 120.0, 80.0, 100.0, 50.0, 40.0, 70.0, 90.0,
   20.0, 110.0, 100.0, 30.0, 50.0, 90.0, 110.0, 30.0, 90.0, 40.0, 80.0, 70.0]
66 >>> print(y3)
67 [399.0, 121.0, 221.0, 376.0, 361.0, 224.0, 546.0, 352.0, 353.0, 157.0, 160.0, 252.0,
   389.0, 113.0, 435.0, 420.0, 212.0, 268.0, 377.0, 421.0, 273.0, 468.0, 244.0,
   342.0, 323.0]

```

---

```

68 >>> for i in range(len(x3)):
69     ...     print(x3[i],y3[i])
70
71 80.0 399.0
72 .....
73 .....
74 70.0 323.0
75
76 >>> # -----
77 ... # For convenience,
78 ... # -----
79 >>> x = x1
80 >>> y = y1

```

---

## (2) Scatter plot

---

```

1 ... import matplotlib.pyplot as plot # https://matplotlib.org/
2 >>> # Windows10: C:\> pip install matplotlib
3 ... # The below may be needed for upgrading pip.
4 ... # Windows10: C:\> python -m pip install --upgrade pip --user
5 ...
6 >>> plot.figure( figsize=(10,10) )
7 <matplotlib.figure.Figure object at 0x7f0de8411e48>
8 >>> plot.scatter(x, y, c="black" )
9 <matplotlib.collections.PathCollection object at 0x7f0dc958a6d8>
10 >>> # plot.savefig("ch01-example1.eps")
11 ... plot.show()

```

---

## (3) Parameter estimation

---

```

1 ... # https://www.statsmodels.org
2 ... # Windows10: C:\> pip install -U statsmodels --user
3 ... from statsmodels.formula.api import ols
4 >>> from statsmodels.stats.anova import anova_lm
5 >>> import pandas # https://pandas.pydata.org
6 >>>
7 >>> # This data structure is needed for ols and anova_lm
8 ... data = pandas.DataFrame({"x": x, "y": y})

```

---

```

9 >>>
10 >>> model = ols("y~x", data) # NB: ols("y~0+x", data)
11 >>> OUT = model.fit()
12 >>> OUT.summary()
13 <class 'statsmodels.iolib.summary.Summary'>
14 """
15
16                      OLS Regression Results
17 =====
18 Dep. Variable:          y      R-squared:          0.822
19 Model:                OLS      Adj. R-squared:      0.814
20 Method:              Least Squares    F-statistic:      105.9
21 Date:                Fri, 28 Jun 2019    Prob (F-statistic):  4.45e-10
22 Time:                13:28:05    Log-Likelihood:    -131.64
23 No. Observations:      25      AIC:              267.3
24 Df Residuals:          23      BIC:              269.7
25 Df Model:              1
26 Covariance Type:      nonrobust
27 =====
28                      coef      std err          t      P>|t|      [0.025      0.975]
29 -----
29 Intercept            62.3659      26.177       2.382     0.026     8.214    116.518
30 x                   3.5702       0.347     10.290     0.000     2.852     4.288
31 =====
32 Omnibus:              0.608    Durbin-Watson:      1.432
33 Prob(Omnibus):        0.738    Jarque-Bera (JB):    0.684
34 Skew:                 0.298    Prob(JB):            0.710
35 Kurtosis:             2.450    Cond. No.:           202.
36 =====
37
38 Warnings:
39 [1] Standard Errors assume that the covariance matrix of the errors is correctly
    specified.
40 """
41
42 >>> # NB: Compare the following
43 ... # ols("y~ x", data).fit().summary()
44 ... # ols("y~0+x", data).fit().summary()
45 ... anova_lm(OUT)
46
47      df      sum_sq      mean_sq      F      PR(>F)
48 x          1.0    252377.580808    252377.580808    105.875709    4.448828e-10
49 Residual    23.0     54825.459192     2383.715617         NaN         NaN

```

---

#### (4) Some Calculations

---

```

1 ... # To compare with Minitab codes
2 ... # NB: Cx (x=1,2,...) are variables in Minitab
3 ... C2 = y
4 >>> C1 = x
5 >>>
6 >>> C3 = OUT.fittedvalues
7 >>>
8 >>> C4 = C2 - C3
9 >>> C5 = C4**2
10 >>>
11 >>> for i in range(len(C1)): # ugly
12 ...     print(C1[i],C2[i],C3[i],C4[i],C5[i])
13 ...
14 80.0 399.0 347.98202020202024 51.01797979797976 2602.834262667071
15 .....
16 .....
17 70.0 323.0 312.28000000000003 10.71999999999997 114.91839999999937
18 .....
19 >>> for i in range(len(C1)): # better
20 ...     print("%5d %5d %10.5f %10.5f %10.5f" %(C1[i],C2[i],C3[i],C4[i],C5[i]))
21 ...
22      80      399      347.98202      51.01798      2602.83426
23      30      121      169.47192     -48.47192      2349.52695
24 .....
25 .....
26      70      323      312.28000      10.72000      114.91840
27 >>> C11 = sum(C4)

```



---

```

28 >>> C12 = sum(C2)
29 >>> C13 = sum(C3)
30 >>>
31 >>> import numpy as np # We need np.multiply from numpy (https://www.numpy.org/)
32 >>> # Windows10: C:\> pip install -U numpy
33 ... C14 = np.multiply(C1,C4)
34 >>> C15 = np.multiply(C3,C4)
35 >>> C16 = np.multiply(C2,C4)
36 >>>
37 >>> C14 = sum(C14)
38 >>> C15 = sum(C15)
39 >>> C16 = sum(C16)
40 >>> C17 = sum(C5)
41 >>>
42 >>> print(C11, C12, C13, C14, C15, C16, C17) # ugly
43 -1.1937117960769683e-12 7807.0 7807.000000000002 -8.151346264639869e-11
    -3.6288838600739837e-10 54825.459191918846 54825.4591919192
44
45 >>> print((" %11.7G"*7) %(C11, C12, C13, C14, C15, C16, C17)) # better
46 -1.193712E-12 7807 7807 -8.151346E-11 -3.628884E-10 54825.46 54825.46

```

---

## 6 Example II

Over the past decades, the data useful to assess the impact of climate change have been collected from satellite. Especially since 1979, satellites have measured the area or extended area of sea ice in the Arctic Ocean on a daily basis. It should be noted that the extended areas (the fifth column in the data set) include the areas near the pole not imaged by the sensor. It is assumed to be entirely ice covered with at least 15% concentration. On the other hand, the areas (the sixth column) *exclude* the area not imaged by the sensor.

The averages of areas (or extended areas) of sea ice on a daily basis are calculated for each month. We consider the average *extended* area of sea ice in September. It should be noted that September is the month when the ice stops melting each summer and reaches its minimum extent. Witt (2013) analyzed a time series of September Arctic sea ice extent from 1979 until 2015. The original data (Fetterer et al., 2016) can be obtained at:

[www.amstat.org/publications/jse/v21n1/witt.pdf](http://www.amstat.org/publications/jse/v21n1/witt.pdf)

<http://dx.doi.org/10.7265/N5736NV7>

Minitab

### (1) Write and Read Data

---

```

1 MTB > SET C1 .
2 DATA> 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988

```



---

```

14 Analysis of Variance
15 Source          DF      SS      MS      F      P
16 Regression        1  32.162  32.162  100.48  0.000
17 Residual Error    35  11.203   0.320
18 Total             36  43.366
19
20 Unusual Observations
21 Obs      C1      C2      Fit  SE Fit  Residual  St Resid
22  18  1996  7.9100  6.4362  0.0934   1.4738    2.64R
23  29  2007  4.3200  5.4757  0.1274  -1.1557   -2.10R
24  34  2012  3.6300  5.0391  0.1604  -1.4091   -2.60R
25
26 R denotes an observation with a large standardized residual.
27
28 MTB > #-----
29 MTB > # Scatter plot with the fitted line
30 MTB > #-----
31 MTB > GPRO .
32 * NOTE * Professional Graphics are now enabled, and Standard Graphics are
33         * disabled. Use the GSTD command when you want to re-enable Standard
34         * Graphics.
35
36 MTB > PLOT C2*C1 ;
37 SUBC> Symbol ;
38 SUBC> Regress .

```

---

#### (4) To compare with Figure 2 of Witt (2013).

---

```

1 MTB > DELETE 35:37 C1.
2 MTB > DELETE 35:37 C2.
3 MTB > REGR C2 1 C1;
4 SUBC> FITS C3 .
5
6 Regression Analysis: C2 versus C1
7 The regression equation is
8 C2 = 189 - 0.0913 C1
9
10 Predictor      Coef  SE Coef      T      P
11 Constant      188.60   20.25    9.31  0.000
12 C1            -0.09128  0.01015  -8.99  0.000
13
14 S = 0.580615    R-Sq = 71.7%    R-Sq(adj) = 70.8%
15
16 Analysis of Variance
17 Source          DF      SS      MS      F      P
18 Regression        1  27.265  27.265  80.88  0.000
19 Residual Error    32  10.788   0.337
20 Total            33  38.053
21
22 Unusual Observations
23 Obs      C1      C2      Fit  SE Fit  Residual  St Resid
24  18  1996  7.9100  6.4129  0.0997   1.4971    2.62R
25  34  2012  3.6300  4.9525  0.1948  -1.3225   -2.42R
26
27 R denotes an observation with a large standardized residual.
28
29 MTB > #-----
30 MTB > # Scatter plot with the fitted line
31 MTB > #-----
32 MTB > GPRO .
33 * NOTE * Professional Graphics are now enabled, and Standard Graphics are
34         * disabled. Use the GSTD command when you want to re-enable Standard
35         * Graphics.
36
37 MTB > PLOT C2*C1 ;
38 SUBC> Symbol ;
39 SUBC> Regress .

```

---

R

---

### (1) Write and Read Data

---

```
1 > year = c(1979, 1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988,
2 +         1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998,
3 +         1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008,
4 +         2009, 2010, 2011, 2012, 2013, 2014, 2015 )
5 > area = c(7.22, 7.86, 7.25, 7.45, 7.54, 7.11, 6.93, 7.55, 7.51, 7.53,
6 +         7.08, 6.27, 6.59, 7.59, 6.54, 7.24, 6.18, 7.91, 6.78, 6.62,
7 +         6.29, 6.36, 6.78, 5.98, 6.18, 6.08, 5.59, 5.95, 4.32, 4.73,
8 +         5.39, 4.93, 4.63, 3.63, 5.35, 5.29, 4.68)
```

---

### (2) Scatter plot

---

```
1 > plot( year, area )
```

---

### (3) Parameter estimation

---

```
1 > OUT = lm (area ~ year)
2 > summary(OUT)
3 Call:
4 lm(formula = area ~ year)
5
6 Residuals:
7     Min       1Q   Median       3Q      Max
8 -1.40910 -0.34356  0.03251  0.35840  1.47376
9
10 Coefficients:
11             Estimate Std. Error t value Pr(>|t|)
12 (Intercept) 180.728966   17.396966   10.39 3.10e-12 ***
13 year        -0.087321    0.008711  -10.02 7.97e-12 ***
14 ---
15 Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
16
17 Residual standard error: 0.5658 on 35 degrees of freedom
18 Multiple R-squared:  0.7417, Adjusted R-squared:  0.7343
19 F-statistic: 100.5 on 1 and 35 DF, p-value: 7.968e-12
20
21 > # NB: Compare the following
22 > # lm (area ~ year)
23 > # lm (area ~ 0 + year)
24 >
25 > anova(OUT)
26 Analysis of Variance Table
27 Response: area
28      Df Sum Sq Mean Sq F value    Pr(>F)
29 year    1 32.162   32.162   100.48 7.968e-12 ***
30 Residuals 35 11.203    0.320
```

---

```

31 ---
32 Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
33 >
34 > #-----
35 > # Scatter plot with the fitted line
36 > #-----
37 > plot( year, area )
38 > abline(OUT)

```

---

#### (4) To compare with Figure 2 of Witt (2013)

---

```

1  > yr = year[1:34]
2  > ar = area[1:34]
3  >
4  > OUT2 = lm (ar ~ yr)
5  > summary(OUT2)
6  Call:
7  lm(formula = ar ~ yr)
8
9  Residuals:
10      Min       1Q   Median       3Q      Max
11 -1.32245 -0.37971  0.01339  0.38897  1.49711
12
13 Coefficients:
14             Estimate Std. Error t value Pr(>|t|)
15 (Intercept) 188.60240    20.25374   9.312 1.26e-10 ***
16 yr          -0.09128     0.01015  -8.993 2.85e-10 ***
17 ---
18 Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
19
20 Residual standard error: 0.5806 on 32 degrees of freedom
21 Multiple R-squared:  0.7165, Adjusted R-squared:  0.7076
22 F-statistic: 80.88 on 1 and 32 DF, p-value: 2.845e-10
23
24 > anova(OUT2)
25 Analysis of Variance Table
26
27 Response: ar
28      Df Sum Sq Mean Sq F value    Pr(>F)
29 yr      1 27.265  27.2650   80.878 2.845e-10 ***
30 Residuals 32 10.788   0.3371
31 ---
32 Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
33 >
34 > #-----
35 > # Scatter plot with the fitted line
36 > #-----
37 > plot(yr, ar, ylim=c(0,9) )
38 > abline( OUT2 )

```

---

### Python

#### (1) Write and Read Data

---

```

1 >>> x = [1979, 1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988,
2 ...     1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998,
3 ...     1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008,
4 ...     2009, 2010, 2011, 2012, 2013, 2014, 2015]
5 >>> y = [7.22, 7.86, 7.25, 7.45, 7.54, 7.11, 6.93, 7.55, 7.51, 7.53,
6 ...     7.08, 6.27, 6.59, 7.59, 6.54, 7.24, 6.18, 7.91, 6.78, 6.62,
7 ...     6.29, 6.36, 6.78, 5.98, 6.18, 6.08, 5.59, 5.95, 4.32, 4.73,
8 ...     5.39, 4.93, 4.63, 3.63, 5.35, 5.29, 4.68]

```

---

#### (2) Scatter plot

---

```

1 >>> #=====
2 ... # (2) Scatter plot

```

---

```

3 ... #=====
4 >>> import matplotlib.pyplot as plot # https://matplotlib.org/
5 >>> import numpy as np # https://www.numpy.org
6 >>> year = np.array(x).reshape((-1,1))
7 >>> area = np.array(y).reshape((-1,1))
8 >>>
9 >>> plot.figure( figsize=(10,10) )
10 <matplotlib.figure.Figure object at 0x7fec71fa7860>
11 >>> plot.scatter(year, area, c="black" )
12 <matplotlib.collections.PathCollection object at 0x7fec51acf518>
13 >>> plot.show()

```

---

### (3) Parameter estimation

---

```

1 >>> # https://www.statsmodels.org
2 ... # Windows10: C:\> pip install -U statsmodels --user
3 ... from statsmodels.formula.api import ols
4 >>> from statsmodels.stats.anova import anova_lm
5 >>> import pandas # https://pandas.pydata.org
6 >>>
7 >>> # This data structure is needed for ols and anova_lm
8 ... data = pandas.DataFrame({"year": x, "area": y})
9 >>> model = ols("area~year", data=data) # NB: ols("y~0+x", data)
10 >>> OUT = model.fit()
11 >>> print(OUT.summary())

```

OLS Regression Results						
=====						
Dep. Variable:	area	R-squared:	0.742			
Model:	OLS	Adj. R-squared:	0.734			
Method:	Least Squares	F-statistic:	100.5			
Date:	Wed, 26 Jun 2019	Prob (F-statistic):	7.97e-12			
Time:	19:27:27	Log-Likelihood:	-30.399			
No. Observations:	37	AIC:	64.80			
Df Residuals:	35	BIC:	68.02			
Df Model:	1					
Covariance Type:	nonrobust					
=====						
	coef	std err	t	P> t	[0.025	0.975]
-----						
Intercept	180.7290	17.397	10.389	0.000	145.411	216.047
year	-0.0873	0.009	-10.024	0.000	-0.105	-0.070

---

```

28 =====
29 Omnibus:                1.794    Durbin-Watson:                1.739
30 Prob(Omnibus):          0.408    Jarque-Bera (JB):        0.811
31 Skew:                   -0.133    Prob(JB):                0.667
32 Kurtosis:              3.675    Cond. No.                3.74e+05
33 =====
34
35 Warnings:
36 [1] Standard Errors assume that the covariance matrix of the errors is correctly
    specified.
37 [2] The condition number is large, 3.74e+05. This might indicate that there are
    strong multicollinearity or other numerical problems.
38
39 >>>
40 >>> # NB: Compare the following
41 ... # ols("area~year", data).fit().summary()
42 ... # ols("area~0+year", data).fit().summary()
43 ...
44 >>> print(anova_lm(OUT))
45
46      df      sum_sq      mean_sq      F      PR(>F)
47 year      1.0    32.162073    32.162073    100.475221    7.967937e-12
48 Residual  35.0    11.203484     0.320100         NaN         NaN
49
50 >>> #-----
51 ... # Scatter plot with the fitted line
52 ... #-----
53 ... plot.figure( figsize=(10,10) )
54 <matplotlib.figure.Figure object at 0x7fec2e7a1be0>
55 >>> plot.scatter(year, area, c="black" )
56 <matplotlib.collections.PathCollection object at 0x7fec2e744eb8>
57 >>> plot.plot(year, OUT.fittedvalues, c="red", linewidth=1 )
58 <matplotlib.lines.Line2D object at 0x7fec2e798cf8>
59 >>> plot.xlabel("Year")
60 <matplotlib.text.Text object at 0x7fec2e73ff60>
61 >>> plot.ylabel("Extended area")
62 <matplotlib.text.Text object at 0x7fec2e74c668>
63 >>> plot.show()

```

---

#### (4) To compare with Figure 2 of Witt (2013)

---

```

1 >>> # NB: Shorten the data by slicing.
2 ... # (it starts with zero and ends with 34).
3 >>> data2 = pandas.DataFrame({"yr": x[0:34], "ar": y[0:34]})
4 >>> model2 = ols("ar~yr", data=data2) # NB: ols("y~0+x", data)
5 >>> OUT2 = model2.fit()
6 >>> print(OUT2.summary())
7
8 OLS Regression Results
9 =====
10 Dep. Variable:          ar      R-squared:                0.717
11 Model:                OLS      Adj. R-squared:           0.708
12 Method:                Least Squares      F-statistic:         80.88
13 Date:                  Wed, 26 Jun 2019    Prob (F-statistic):    2.85e-10
14 Time:                  19:27:36      Log-Likelihood:       -28.729
15 No. Observations:      34      AIC:                  61.46
16 Df Residuals:          32      BIC:                  64.51
17 Df Model:              1
18 Covariance Type:       nonrobust
19 =====
20
21      coef      std err      t      P>|t|      [0.025      0.975]
22 -----
23 Intercept    188.6024     20.254     9.312     0.000    147.347    229.858
24 yr          -0.0913      0.010    -8.993     0.000    -0.112    -0.071
25
26 Omnibus:                0.986    Durbin-Watson:                1.477
27 Prob(Omnibus):          0.611    Jarque-Bera (JB):        0.228
28 Skew:                   0.032    Prob(JB):                0.892
29 Kurtosis:              3.396    Cond. No.                4.06e+05
30 =====
31
32 Warnings:
33 [1] Standard Errors assume that the covariance matrix of the errors is correctly
    specified.

```

---

---

```

32 [2] The condition number is large, 4.06e+05. This might indicate that there are
33 strong multicollinearity or other numerical problems.
34 >>> print(anova_lm(OUT2))
35          df      sum_sq   mean_sq         F         PR(>F)
36 yr          1.0    27.264989   27.264989   80.877733   2.845158e-10
37 Residual    32.0    10.787637    0.337114         NaN         NaN
38 >>>
39 >>> #-----
40 ... # Scatter plot with the fitted line
41 ... #-----
42 ... yr = year[0:34] # NB: slicing (takes 34 observations)
43 >>> ar = area[0:34] # NB: it starts with zero and ends with 34.
44 >>> plot.figure( figsize=(10,10) )
45 <matplotlib.figure.Figure object at 0x7fec2e2c5dd8>
46 >>> plot.scatter(yr, ar, c="black" )
47 <matplotlib.collections.PathCollection object at 0x7fec2de5b780>
48 >>> plot.plot(yr, OUT2.fittedvalues, c="red", linewidth=1 )
49 <matplotlib.lines.Line2D object at 0x7fec2de61358>
50 >>> plot.xlabel("Year")
51 <matplotlib.text.Text object at 0x7fec2e2e0198>
52 >>> plot.ylabel("Extended area")
53 <matplotlib.text.Text object at 0x7fec2e2e7860>
54 >>> plot.show()

```

---

## 7 Spurious Correlation

There are examples that attempt to demonstrate how no cause-and-effect (causation) is necessarily implied by the regression model.

For example, an observational study of elementary school children aged 6 – 11 finds a high positive correlation between shoe size  $X$  and score  $Y$  on a test of reading comprehension. Note that such data are observational data since the explanatory variable, shoe size, is not controlled.

This relation does not imply, however, that an increase in shoe size causes reading ability. There are hidden factors (adequate explanatory variables) such as age of child and amount of education which affect both the shoe size ( $X$ ) and reading ability ( $Y$ ).

Even when a strong statistical relationship reflects causal conditions, the causal conditions may act in the *opposite* direction, from  $Y$  to  $X$ .



---

## 8 Miscellaneous

---

## References

- Casella, G. and Berger, R. L. (2002). *Statistical Inference*. Duxbury, Pacific Grove, CA, second edition.
- Fetterer, F., Knowles, K., Meier, W., and Savoie, M. (2016). *Sea Ice Index (updated daily), Version 2*. NSIDC: National Snow and Ice Data Center, Boulder, Colorado USA.
- Hogg, R. V., Tanis, E. A., and Zimmerman, D. L. (2015). *Probability and Statistical Inference*. Pearson, 9th edition.
- Kutner, M. H., Nachtsheim, C. J., Neter, J., and Li, W. (2005). *Applied Linear Statistical Models*. McGraw-Hill, New York, 5th edition.
- Ross, S. M. (2014). *A First Course in Probability*. Prentice Hall, 9th edition.
- Witt, G. (2013). Using data from climate science to teach introductory statistics. *Journal of Statistics Education*, 21:1–24.