Machine Learning

tanshuqiu

Email: tsq@cqut.edu.cn

Finding Relationships between Variables - Regression Techniques

1. Searching linear relationships
2. How to create a linear regression model
3. Polynomial regression

1. Searching linear relationships

Regression analysis is a statistical process of studying the relationship between a set of independent variables (explanatory variables) and the dependent variable (response variable). Through this technique, it is possible to understand how the value of the response variable changes when the explanatory variable is varied.

Consider a group of bikers about which some information has been collected: number of years of use, number of kilometers traveled in one year, and number of falls. Through these techniques, we can find that on an average, when the number of kilometers traveled increases, the number of falls also increases. By increasing the number of years of motorcycle usage and so increasing the experience, the number of falls tends to decrease.

1. Searching linear relationships

A regression analysis can be conducted for a dual purpose:

Explanatory, to understand and weigh the effects of the independent variable on the dependent variable according to a particular theoretical model.

Predictive, to locate a linear combination of the independent variable to predict the value assumed by the dependent variable-optimally.

1. Searching linear relationships

Calculating correlation and covariance is a useful way to investigate whether a linear relationship exists between variables, without having to assume or fit a specific model to our data. It can happen that two variables have a small or no linear correlation, meaning a strong nonlinear relationship. As it may happen that the two variables have a strong correlation, it will be necessary to find a model that approximates this trend. This indicates that calculating linear correlation before fitting a model is a great way to identify variables that have a simple relationship.

1. Searching linear relationships

To introduce the key concepts, we will get started with a simple linear regression example. In theory, there are an infinite number of lines that may interpolate the observations. In practice, there is only one mathematical model that optimizes the representation of the data. In the case of a linear mathematical relationship, the observations of the variable Y can be obtained by a linear function of observations of the variable X. For each observation, we will have:

$$y = \alpha * x + \beta$$

1. Searching linear relationships

$$y = \alpha * x + \beta$$

In this formula, x is the explanatory variable and y is the response variable. Parameters α and β, which represent respectively the intercept with the Y axis and the slope of the line, must be estimated based on the observations collected for the two variables included in the model.

1. Searching linear relationships

Least square regression

To introduce the key concepts, we will get started with a simple linear regression example; we just use a spreadsheet that contains the number of vehicles registered in Italy and the population of the different regions. We will try to find the line that best approximates a relationship that certainly exists between the number of registered vehicles and the population.

1. Searching linear relationships

Least square regression

Previously, we said that a linear relationship is represented by the following formula:

$$y = \alpha * x + \beta$$

If we have a set of observations in the form (x1, y1), (x2, y2), ... (xn, yn), for each of these pairs, we can write an equation of the type just seen. In this way, we get a system of linear equations. Represent this equation in matrix form as:

$$\begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \dots & \dots \\ x_n & 1 \end{bmatrix} \times \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

1. Searching linear relationships

Least square regression

Assuming:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix}; X = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \dots & \dots \\ x_n & 1 \end{bmatrix}; A = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

This can be re-expressed using a condensed formulation:

$$Y = X \times A$$

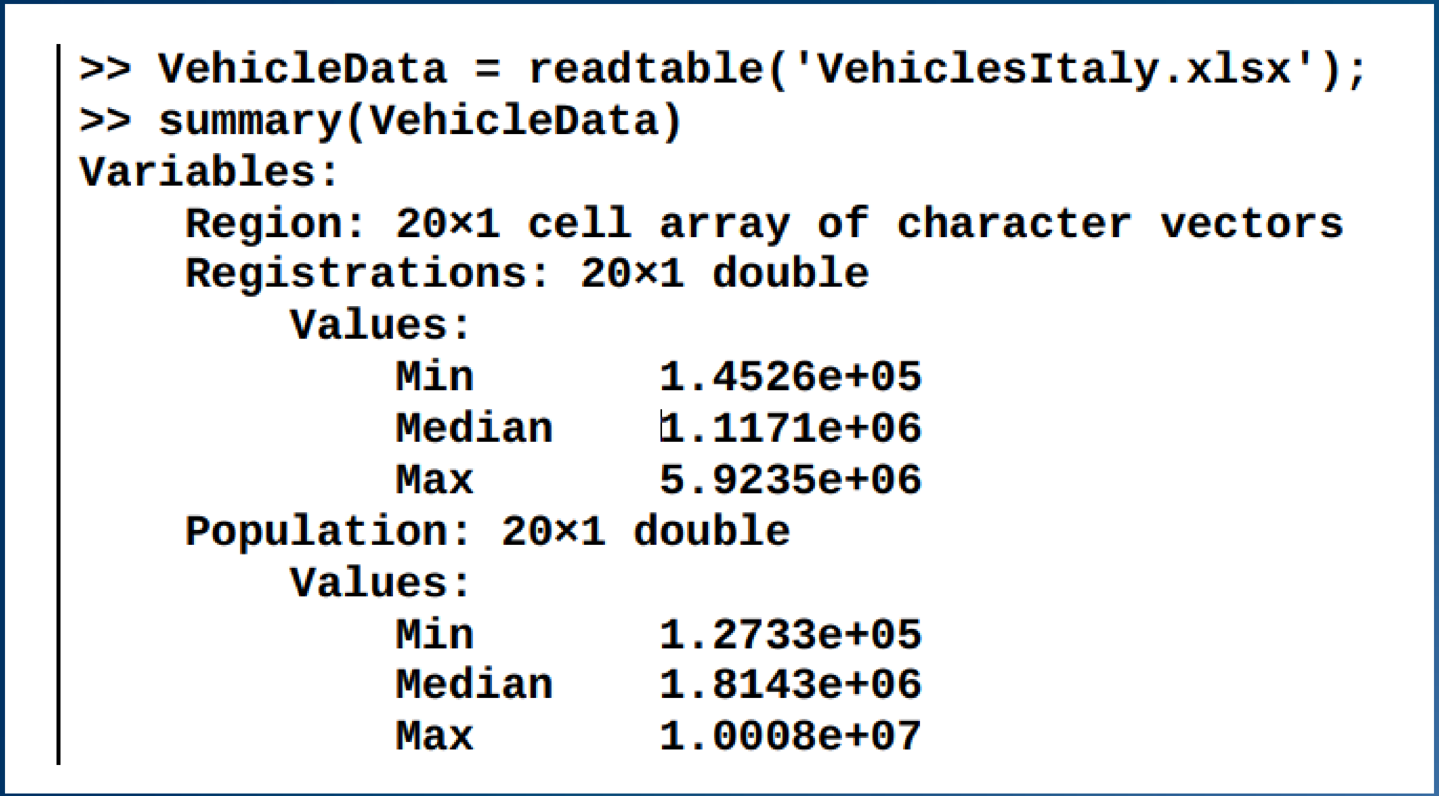
This represents a system of linear equations and MATLAB provides a specific function to locate the solution.

1. Searching linear relationships

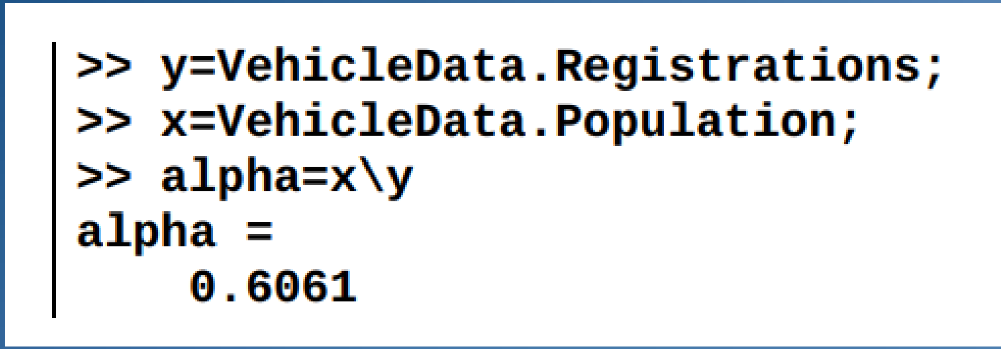
Least square regression

This represents a system of linear equations and MATLAB provides a specific function to locate the solution. It is the mldivide() function, and it is also performed by the \ operator. To determine the intercept and the slope, just do the following:

$$A = X\backslash Y$$



```
>> VehicleData = readtable('VehiclesItaly.xlsx');
>> summary(VehicleData)
Variables:
    Region: 20×1 cell array of character vectors
    Registrations: 20×1 double
        Values:
            Min          1.4526e+05
            Median       1.1171e+06
            Max          5.9235e+06
    Population: 20×1 double
        Values:
            Min          1.2733e+05
            Median       1.8143e+06
            Max          1.0008e+07
```



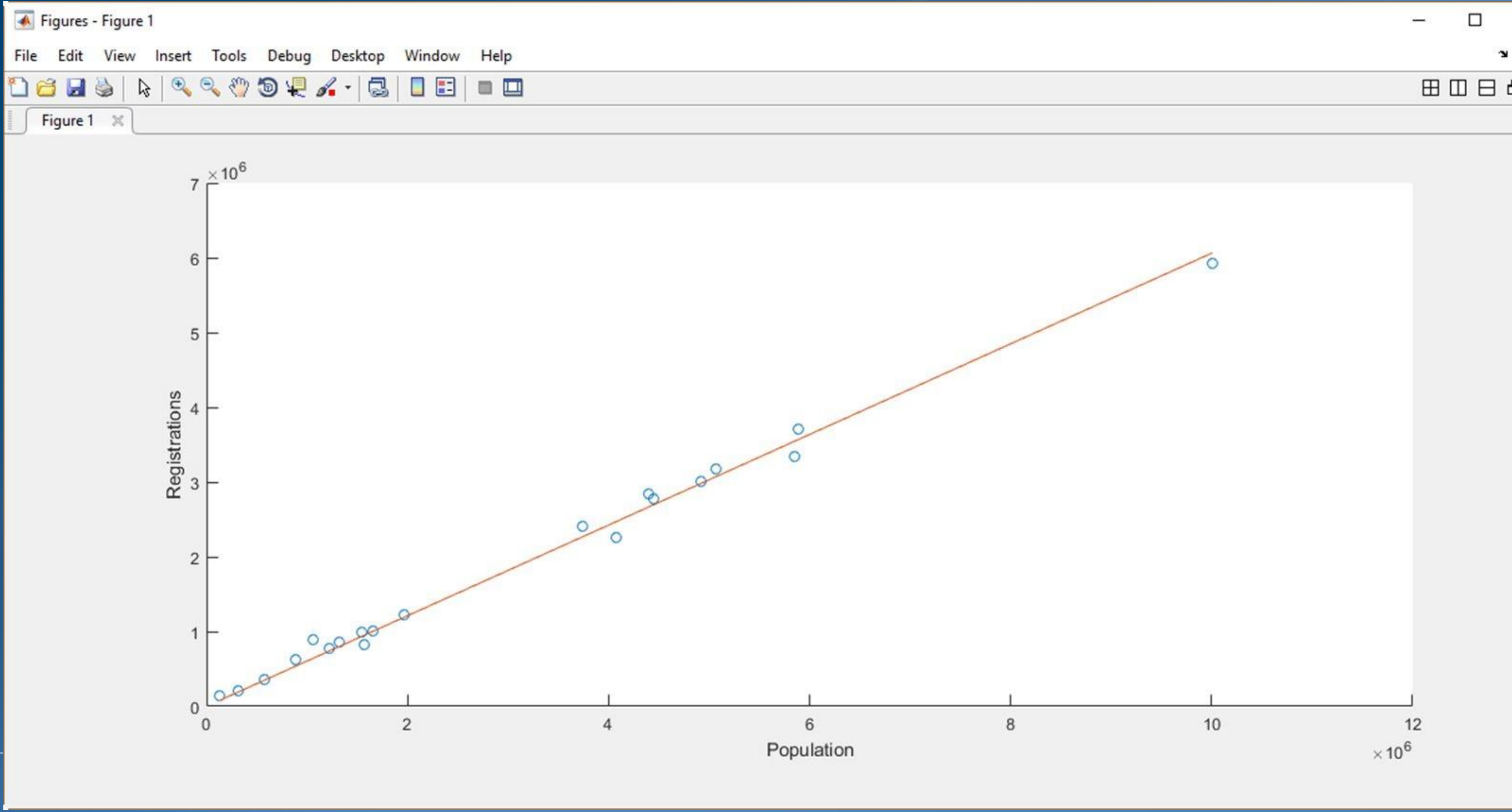
```
>> y=VehicleData.Registrations;
>> x=VehicleData.Population;
>> alpha=x\y
alpha =
    0.6061
```

The linear relationship between the variables is expressed by the following equation:

$$y = 0.6061 \times x$$

In this case, the intercept is equal to zero.

```
>> VehicleRegFit=alpha*x;
>> scatter(x,y)
>> hold on
>> plot(x,VehicleRegFit)
```



1. Searching linear relationships

Least square regression

Given n points (x1, y1), (x2, y2), ... (xn, yn), in the observed population, a least square regression line is defined as the equation line:

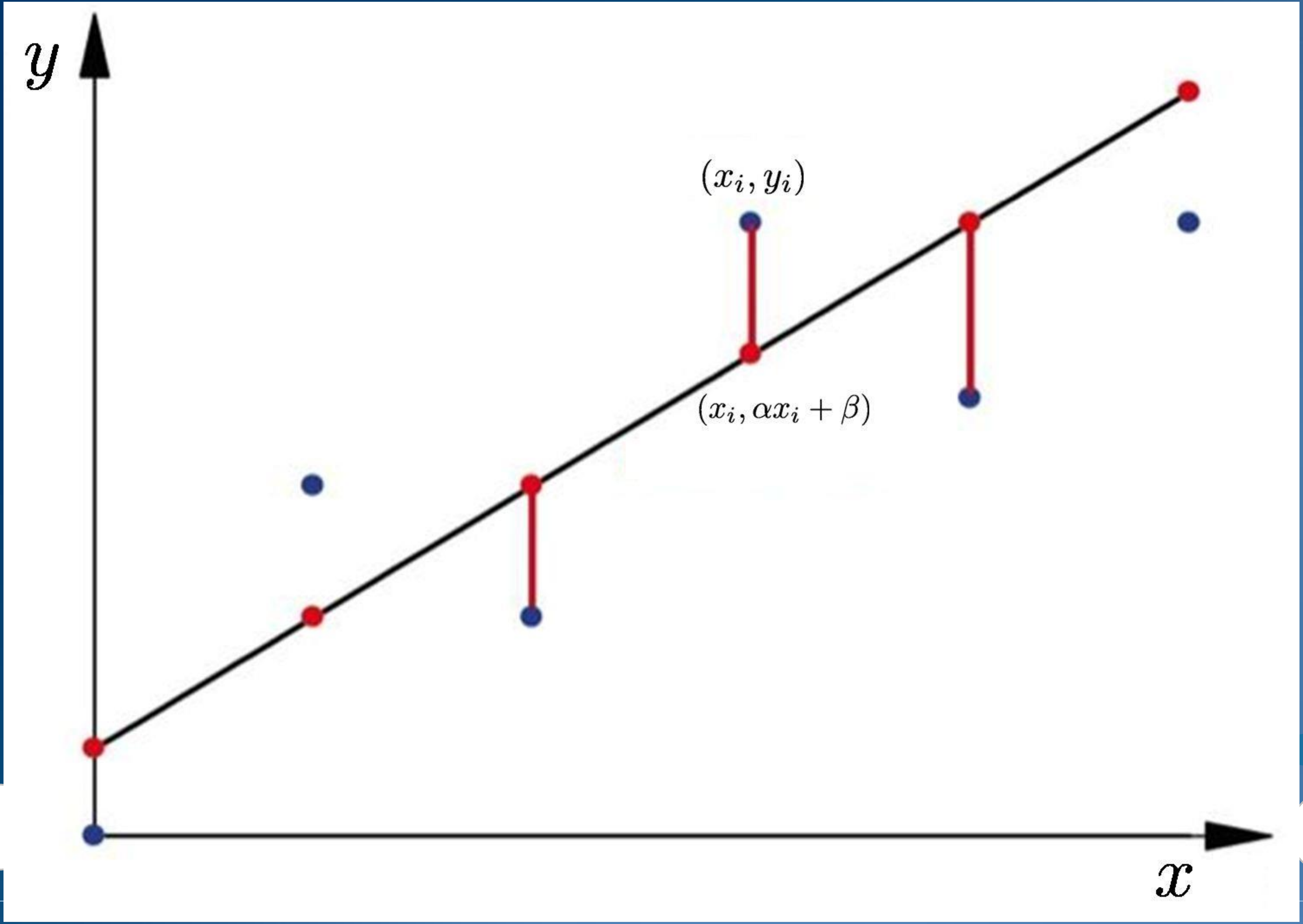
$$y = \alpha * x + \beta$$

For which the following quantity is minimal:

$$E = \sum_{i=1}^{n} (\alpha x_i + \beta - y_i)^2$$

This quantity represents the sum of squares of distances of each experimental datum (xi, yi) from the corresponding point on the straight line:

$$(x_i, \alpha x_i + \beta)$$


y
(x_i, y_i)
$(x_i, \alpha x_i + \beta)$
x

1. Searching linear relationships

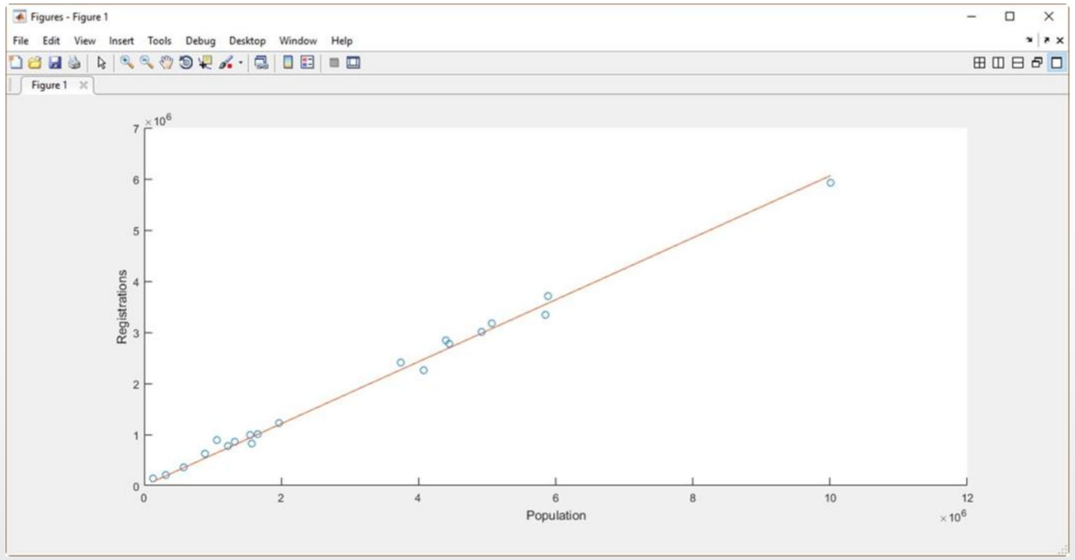
To understand this concept, it is easier to draw the distances between these points, formally called residuals, for a couple of data. Once the coefficients are obtained, calculating the residuals is really simple.

$$r_i = y_i - (\alpha x_i + \beta)$$

```
>> Residual=y-VehicleRegFit;
>> stem(Residual)
```

A residual is a measure of how well a regression line fits an individual data point. Therefore, if the residuals appear to behave randomly, it suggests that the model fits the data well. However, if the residuals display a systematic pattern, it is a clear sign that the model fits the data poorly.

1. Searching linear relationships



In the Figure, we have seen that the model fits the data very well. But how can we measure this feature? One method to find the better fit is to calculate the coefficient of determination (R-squared). To calculate this coefficient, it is necessary to evaluate two quantities: **residual sum of squares** and **total sum of square**.

1. Searching linear relationships

The sum of squares of residuals, also called the residual sum of squares (RSS):

$$SS_{res} = \sum_{i=1}^{n} (y_i - \alpha x_i + \beta)^2$$

The total sum of squares that is proportional to the variance of the data:

$$SS_{tot} = \sum_{i=1}^{n} (y_i - y_{mean})^2$$

The most general definition of the coefficient of determination is:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_{i=1}^{n} (y_i - \alpha x_i + \beta)^2}{\sum_{i=1}^{n} (y_i - y_{mean})^2}$$

The better the linear regression fits the data in comparison to the simple average, the closer the value of R^2 is to 1.

1. Searching linear relationships

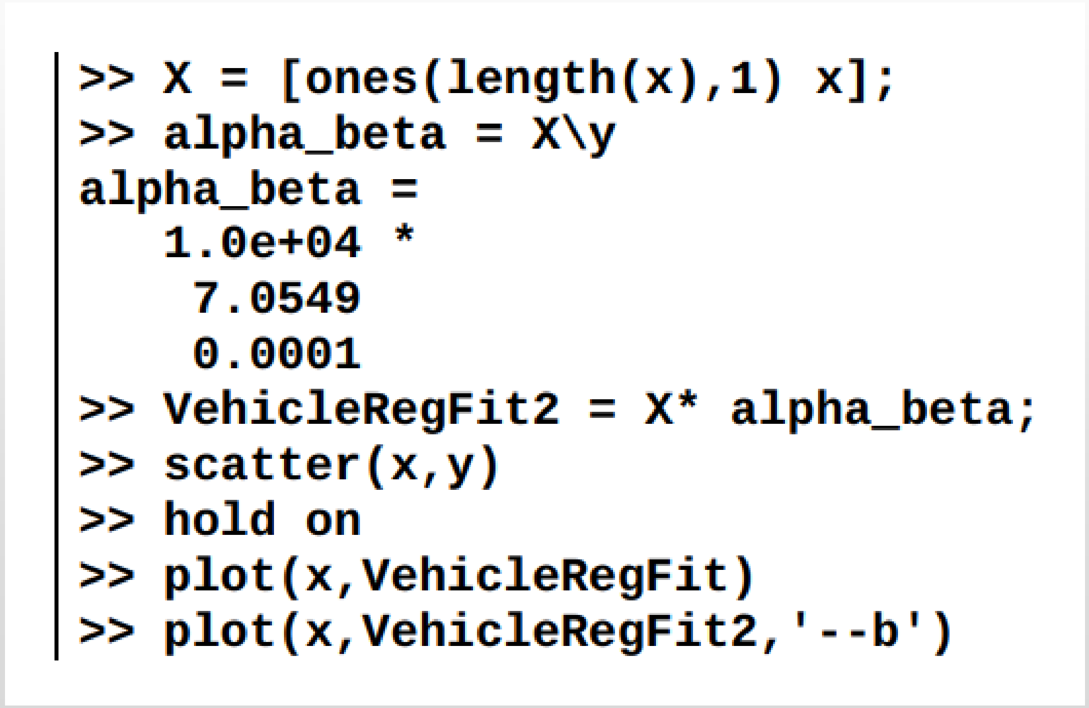
$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_{i=1}^{n}(y_i - \alpha x_i + \beta)^2}{\sum_{i=1}^{n}(y_i - y_{mean})^2}$$

The better the linear regression fits the data in comparison to the simple average, the closer the value of R^2 is to 1.

```
>> Rsq1 = 1 - sum((y - VehicleRegFit).^2)/sum((y - mean(y)).^2)
Rsq1 =
    0.9935
```

1. Searching linear relationships

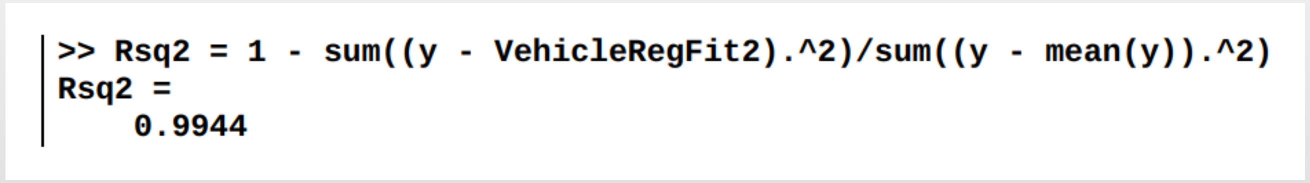
Let's improve the fit by including a y-intercept in our model. To do this, insert a column of one into x and use the \ operator, another time:



```
>> X = [ones(length(x),1) x];
>> alpha_beta = X\y
alpha_beta =
   1.0e+04 *
    7.0549
    0.0001
>> VehicleRegFit2 = X* alpha_beta;
>> scatter(x,y)
>> hold on
>> plot(x,VehicleRegFit)
>> plot(x,VehicleRegFit2,'--b')
```

1. Searching linear relationships

Calculate the determination coefficient again to compare the performance of the two models:



```
>> Rsq2 = 1 - sum((y - VehicleRegFit2).^2)/sum((y - mean(y)).^2)
Rsq2 =
    0.9944
```

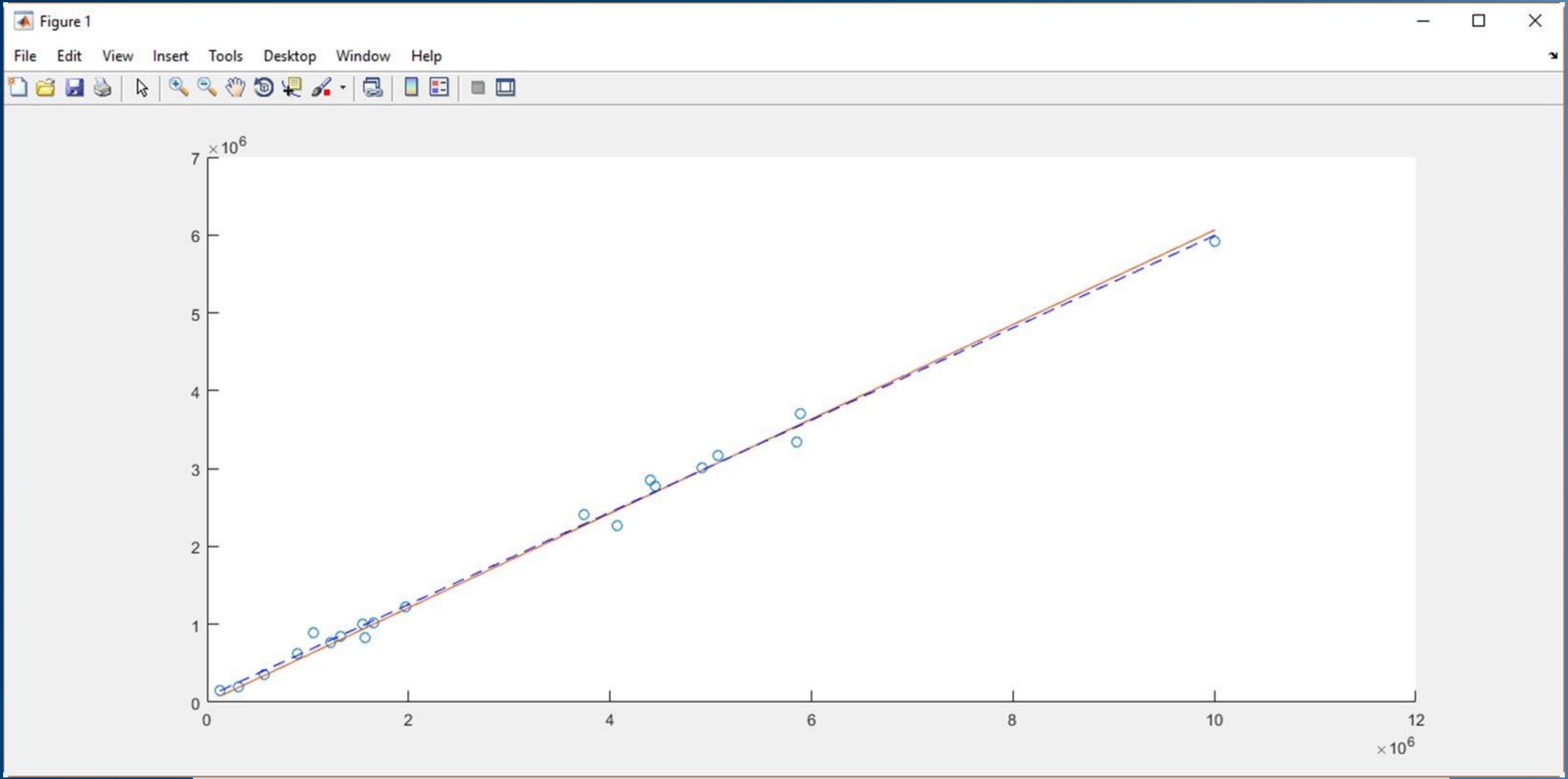
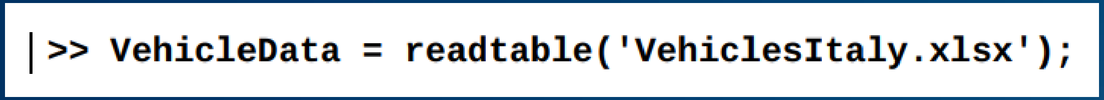


Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
$\times 10^6$
7
6
5
4
3
2
1
0
0
2
4
6
8
10
12
$\times 10^6$

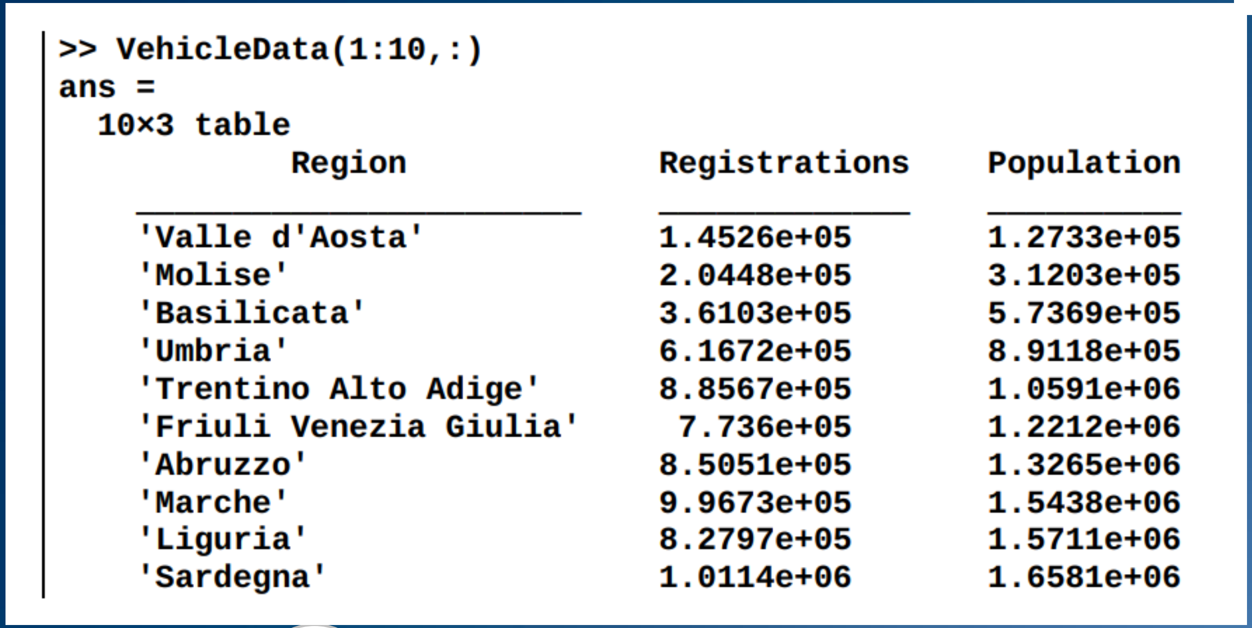
2. How to create a linear regression model

More generally, to create a linear regression model, use the fitlm() function. This function creates a LinearModel object. By default, fitlm() takes the last variable in the table or dataset array as the response.

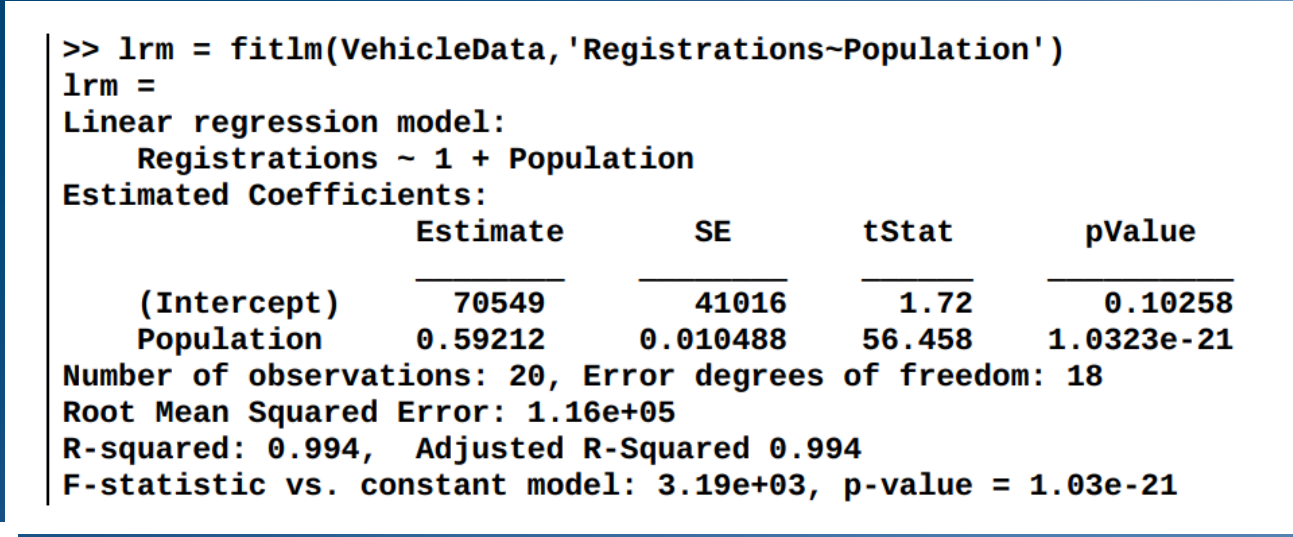
To understand how fitlm() works, we start from the same dataset used before; it contains the number of vehicles registered in Italy and the population of different regions.



```
>> VehicleData = readtable('VehiclesItaly.xlsx');
```



```
>> VehicleData(1:10,:)
ans =
  10×3 table
             Region               Registrations    Population
    ________________________    _____________    __________
    'Valle d'Aosta'              1.4526e+05       1.2733e+05
    'Molise'                     2.0448e+05       3.1203e+05
    'Basilicata'                 3.6103e+05       5.7369e+05
    'Umbria'                     6.1672e+05       8.9118e+05
    'Trentino Alto Adige'        8.8567e+05       1.0591e+06
    'Friuli Venezia Giulia'       7.736e+05       1.2212e+06
    'Abruzzo'                    8.5051e+05       1.3265e+06
    'Marche'                     9.9673e+05       1.5438e+06
    'Liguria'                    8.2797e+05       1.5711e+06
    'Sardegna'                   1.0114e+06       1.6581e+06
```



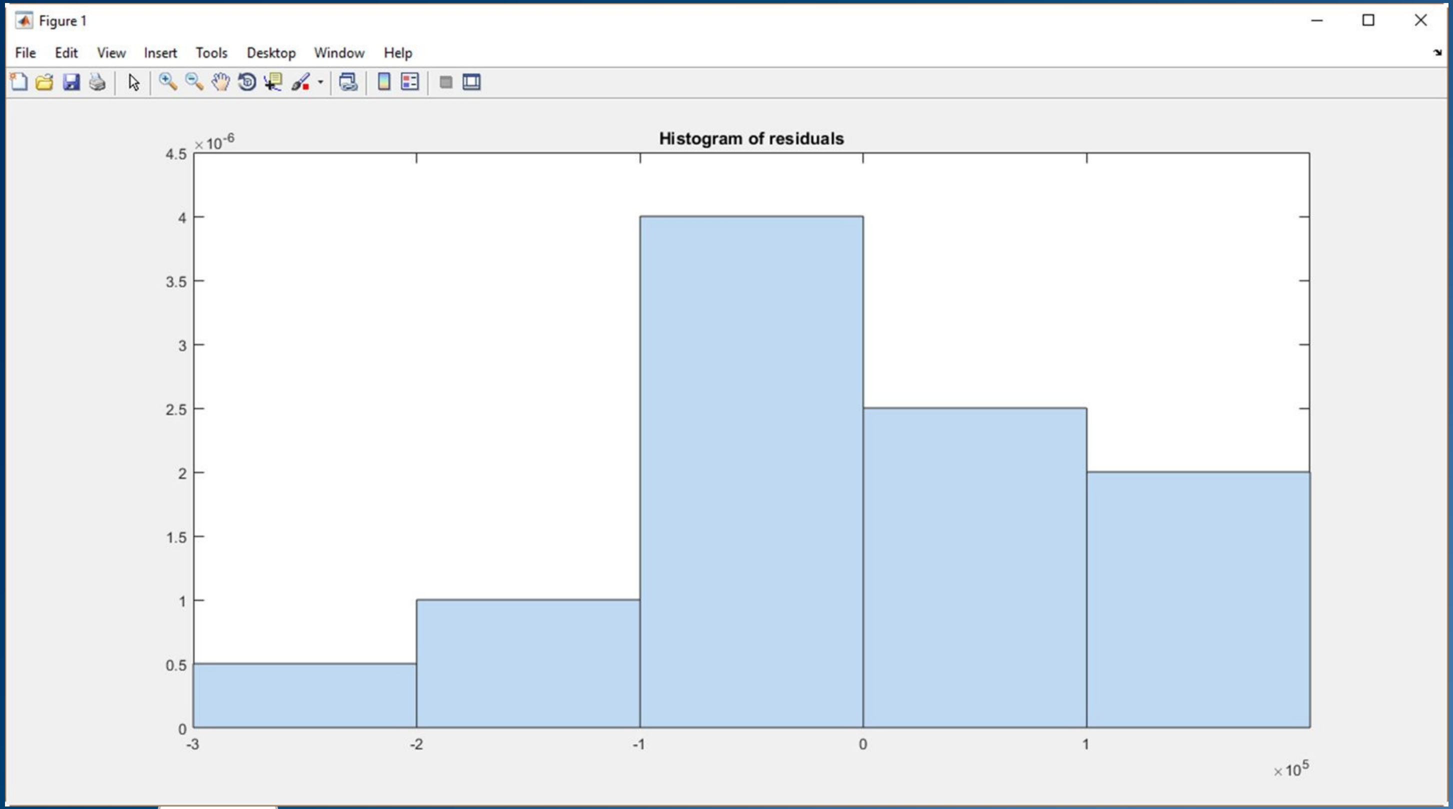
```
>> lrm = fitlm(VehicleData,'Registrations~Population')
lrm =
Linear regression model:
    Registrations ~ 1 + Population
Estimated Coefficients:
                   Estimate       SE        tStat       pValue
                   ________    ________    ______    __________

    (Intercept)      70549        41016      1.72        0.10258
    Population     0.59212     0.010488    56.458     1.0323e-21
Number of observations: 20, Error degrees of freedom: 18
Root Mean Squared Error: 1.16e+05
R-squared: 0.994,  Adjusted R-Squared 0.994
F-statistic vs. constant model: 3.19e+03, p-value = 1.03e-21
```

Fi ear regression model for Registrations (vehicle registrations for each region), using Population (resident population in each region) as explanatory variables (predictor):

We'll start plotting the residuals through a specific property of the LinearModel object:

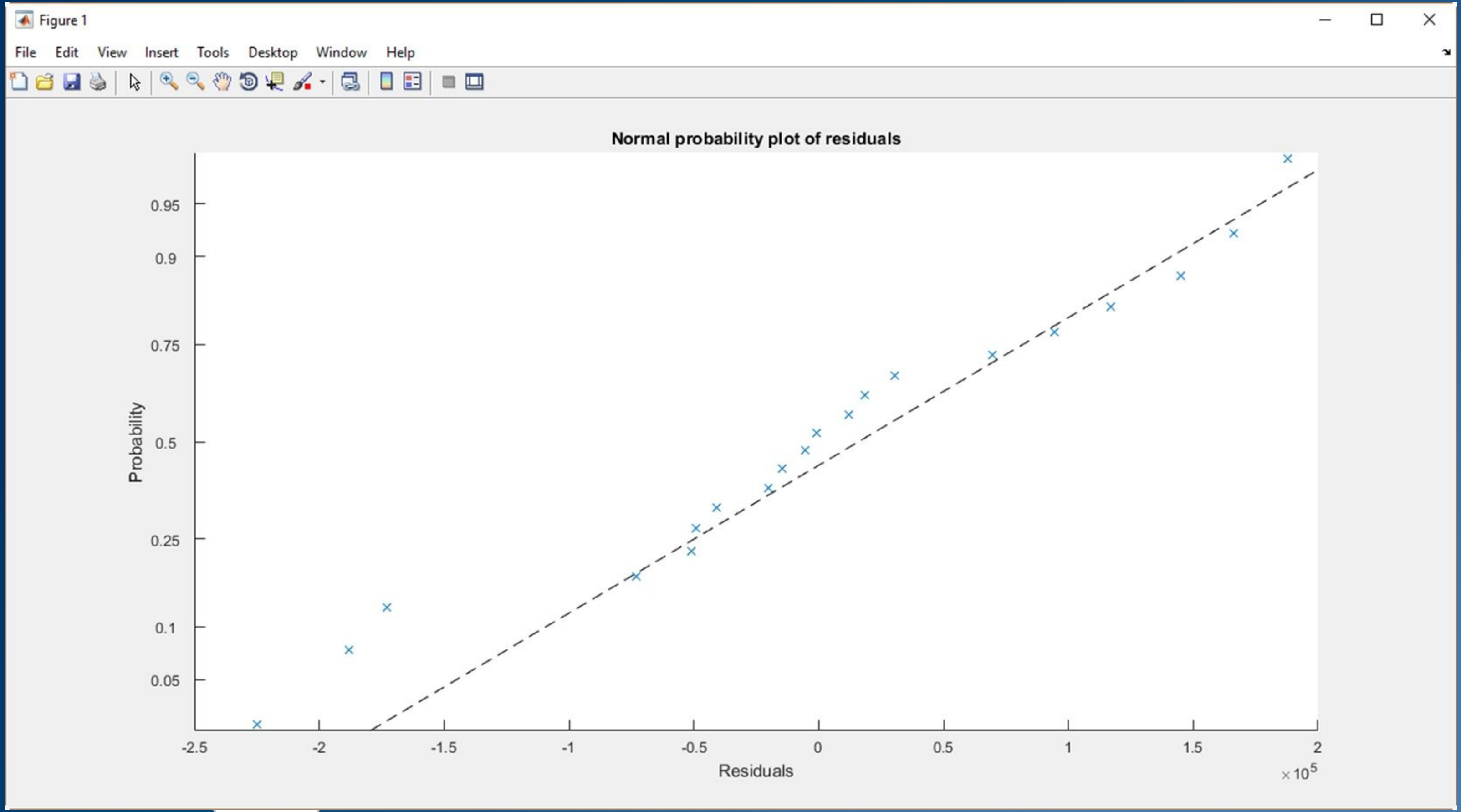
```
>> plotResiduals(lrm)
```



By analyzing the histogram obtained, we can notice that it has a particular asymmetry at the negative values. In particular, the observations under -2 *105 seem to be potential outliers.

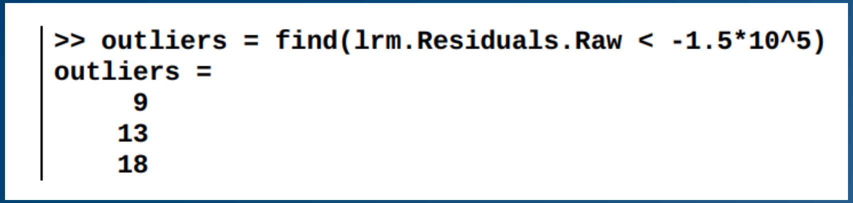
Histogram of a residual for a linear regression model

```
>> plotResiduals(lrm,'probability')
```



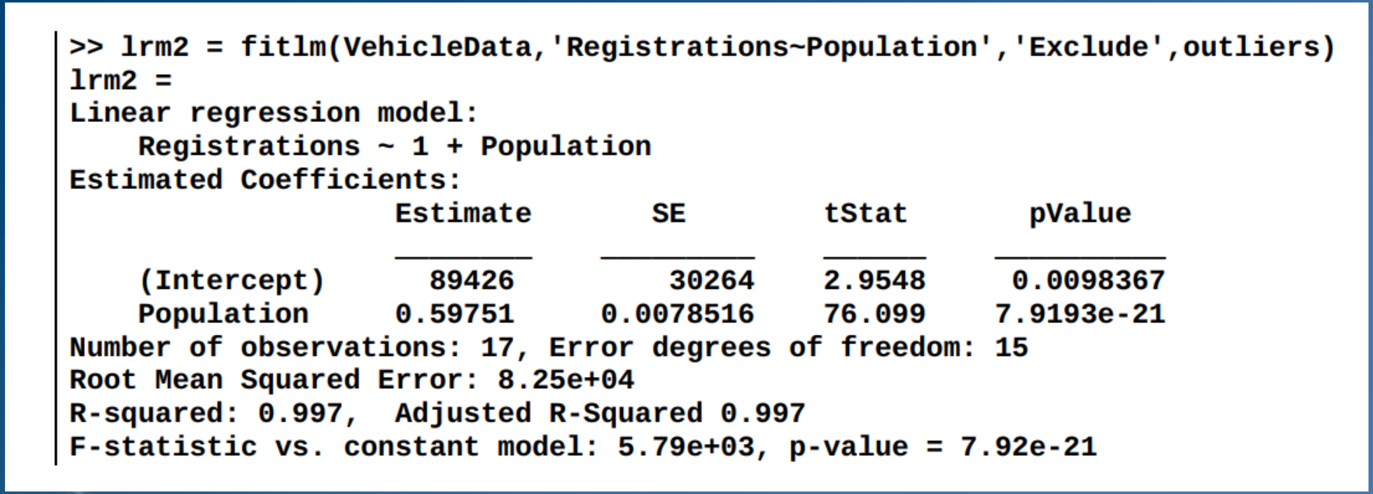
Probability plot of residual for a linear regression model

Potential outliers appear on this plot as well, notably in the bottom left of the plot shown in Figure, we can notice three values that deviate significantly from the dotted line. Otherwise, for other residual values, the probability plot seems reasonably straight, meaning a reasonable fit for normally distributed residuals. We can identify the outliers and remove them from the data; let's see how. In the probability plot, three possible outliers appear to the left of the abscissa equal to -1.5*105. To find them, we will use the find() function:



```
>> outliers = find(lrm.Residuals.Raw < -1.5*10^5)
outliers =
     9
    13
    18
```

At this point, we can create the model, once again excluding those values.

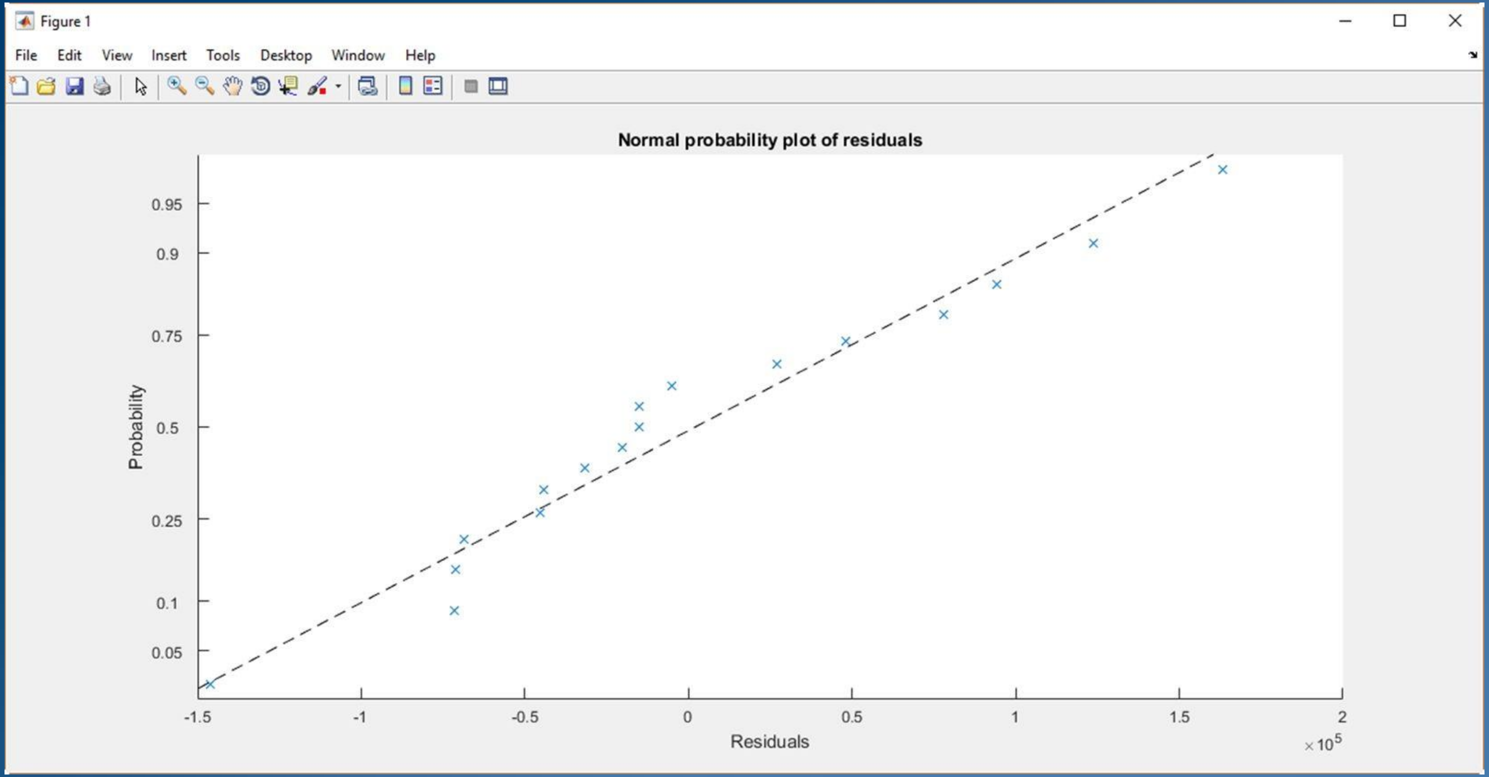


```
>> lrm2 = fitlm(VehicleData,'Registrations~Population','Exclude',outliers)
lrm2 =
Linear regression model:
    Registrations ~ 1 + Population
Estimated Coefficients:
                   Estimate       SE         tStat       pValue
                   ________    _________    ______    __________
    (Intercept)     89426          30264    2.9548     0.0098367
    Population     0.59751     0.0078516    76.099    7.9193e-21
Number of observations: 17, Error degrees of freedom: 15
Root Mean Squared Error: 8.25e+04
R-squared: 0.997,  Adjusted R-Squared 0.997
F-statistic vs. constant model: 5.79e+03, p-value = 7.92e-21
```

we immediately notice that R-squared is clearly improved by recording a value of 0.997, ompared to the 0.994 value of the previous model.

```
>> plotResiduals(lrm2,'probability')
```

In the following figure is shown the probability plot of residual for linear regression model obtained from removing the outliners.



we immediately notice that R-squared is clearly improved by recording a value of 0.997,

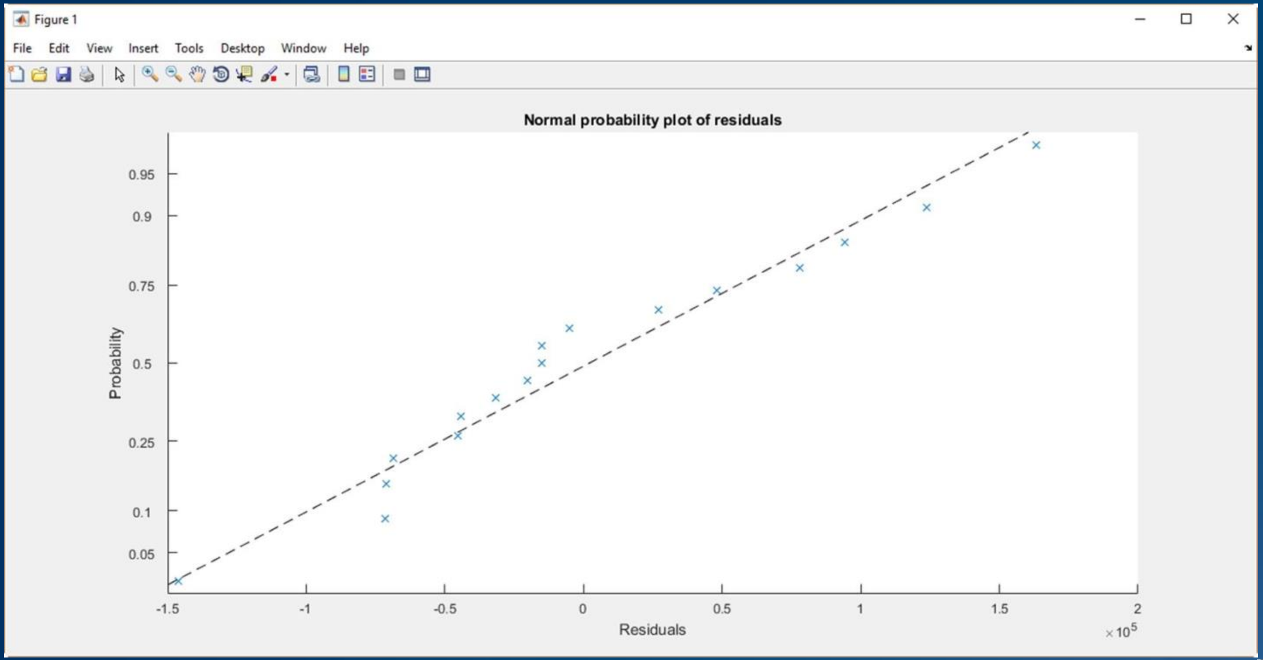


Figure 1
File Edit View Insert Tools Desktop Window Help
Normal probability plot of residuals
Probability
0.95
0.9
0.75
0.5
0.25
0.1
0.05
-1.5
-1
-0.5
0
0.5
1
1.5
2
Residuals
$\times 10^5$

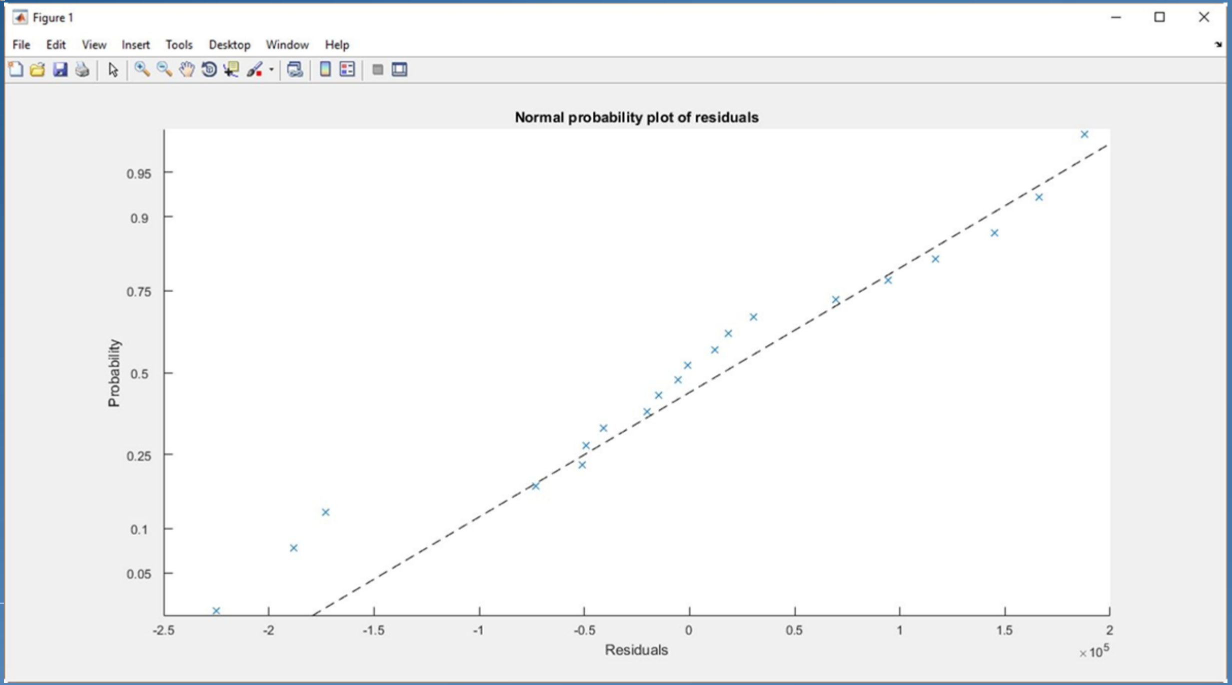


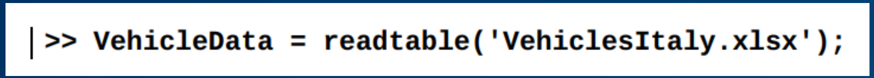
Figure 1
File Edit View Insert Tools Desktop Window Help
Normal probability plot of residuals
Probability
0.95
0.9
0.75
0.5
0.25
0.1
0.05
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
Residuals
$\times 10^5$

2. How to create a linear regression model

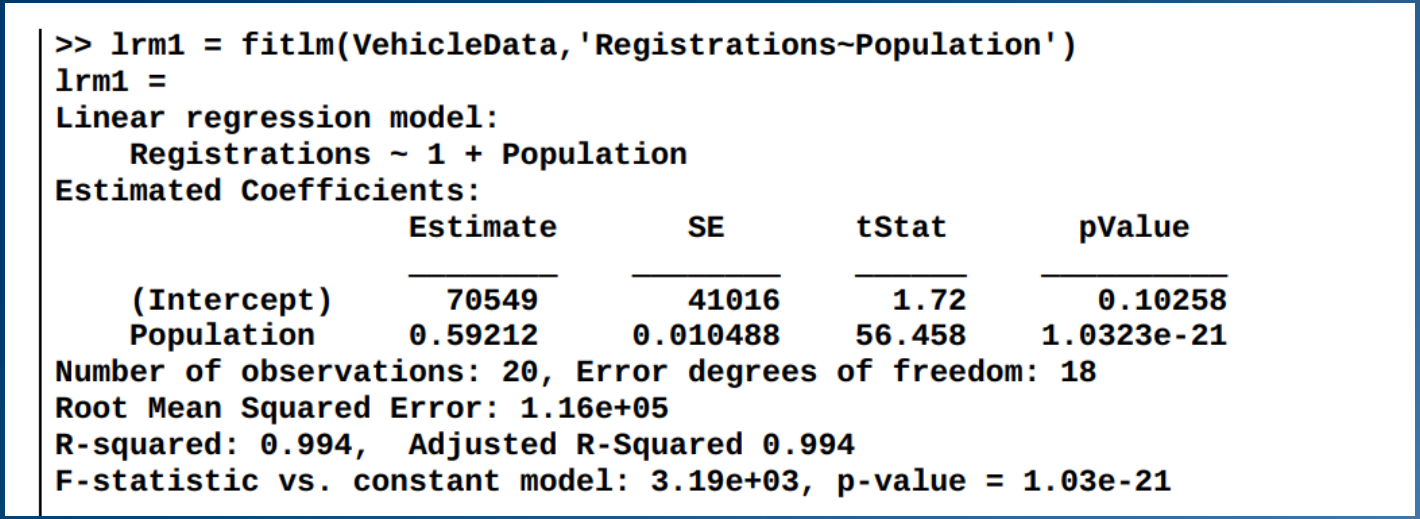
Reducing outlier effects with robust regression

Outliers tend to change the direction of the regression line by getting much more weight than they are worth. Thus, the estimate of the regression coefficients is clearly distorted. These effects are difficult to identify since their residuals are much smaller than they would be if the distortion wasn't present.

To reduce outlier effects, we can fit our data using robust least squares regression. Robust regression methods provide an alternative to least squares regression; they attempts to dampen the influence of outlying cases in order to provide a better fit to the majority of the data.

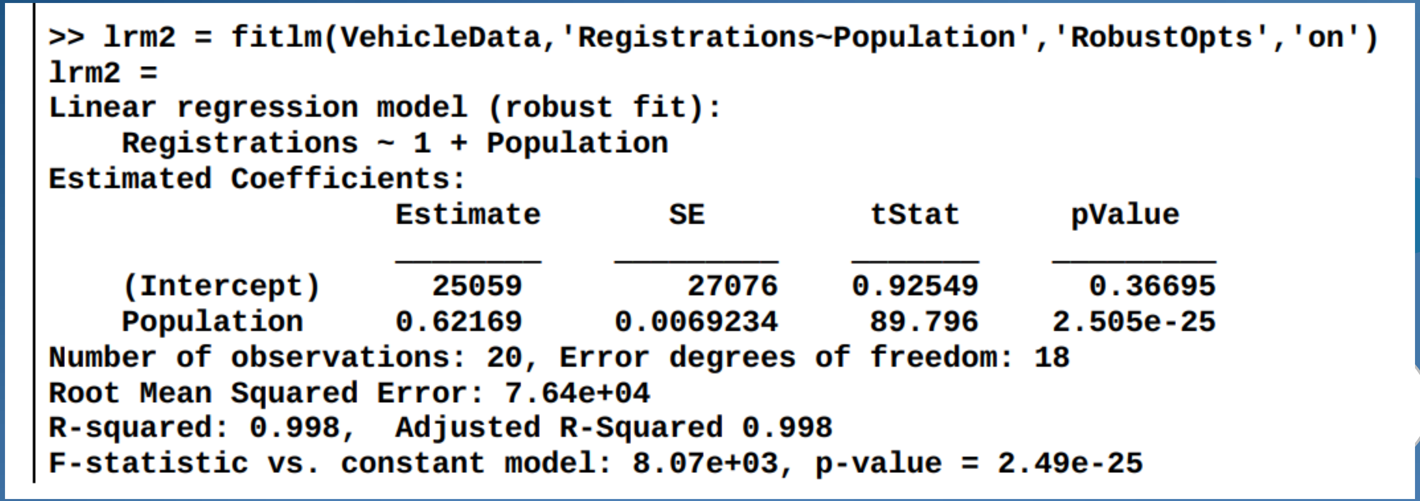


```
>> VehicleData = readtable('VehiclesItaly.xlsx');
```



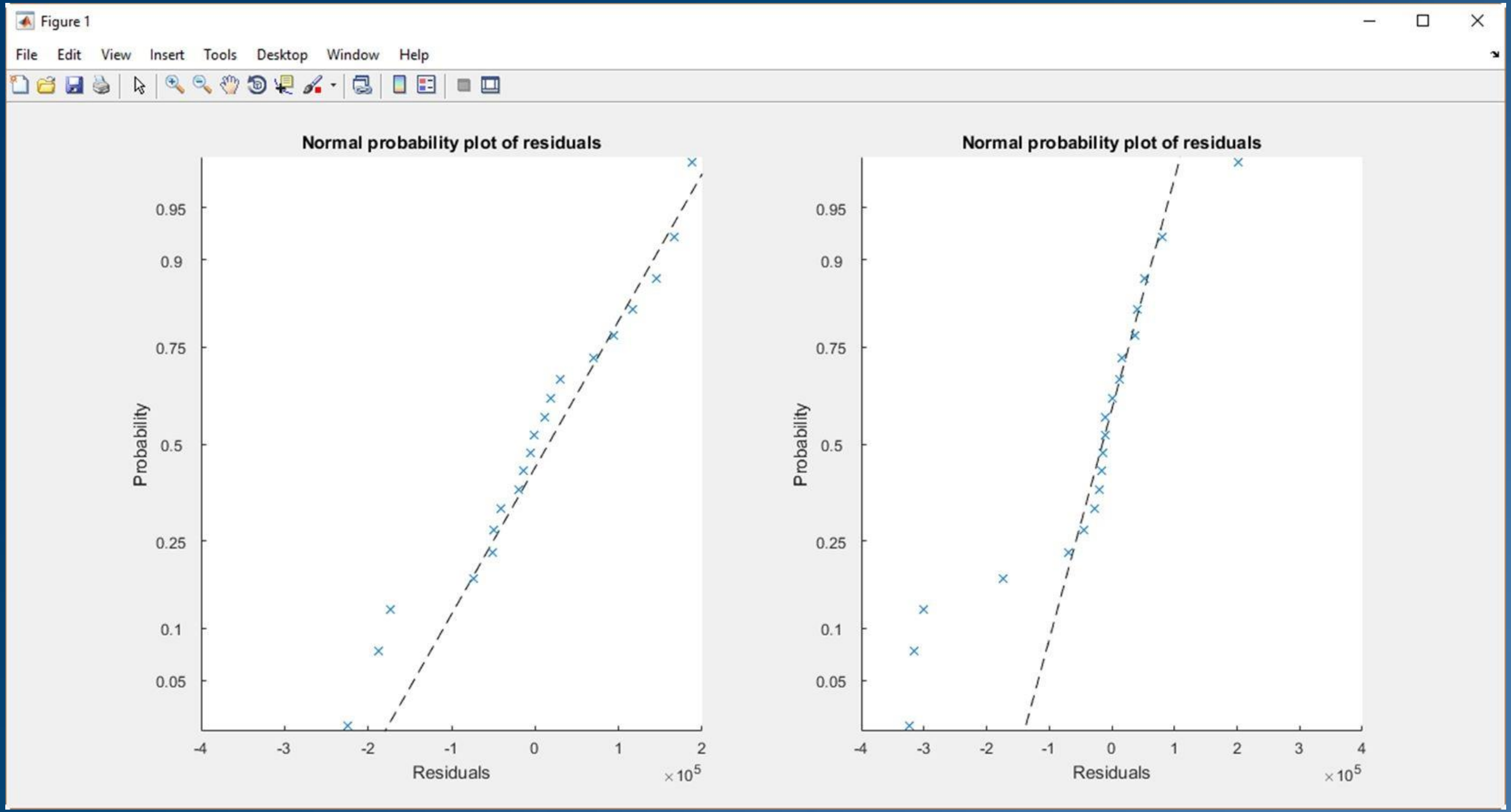
```
>> lrm1 = fitlm(VehicleData,'Registrations~Population')
lrm1 =
Linear regression model:
    Registrations ~ 1 + Population
Estimated Coefficients:
                   Estimate       SE        tStat       pValue
                   ________    ________    ______    __________

    (Intercept)      70549        41016      1.72       0.10258
    Population     0.59212     0.010488    56.458    1.0323e-21
Number of observations: 20, Error degrees of freedom: 18
Root Mean Squared Error: 1.16e+05
R-squared: 0.994,  Adjusted R-Squared 0.994
F-statistic vs. constant model: 3.19e+03, p-value = 1.03e-21
```



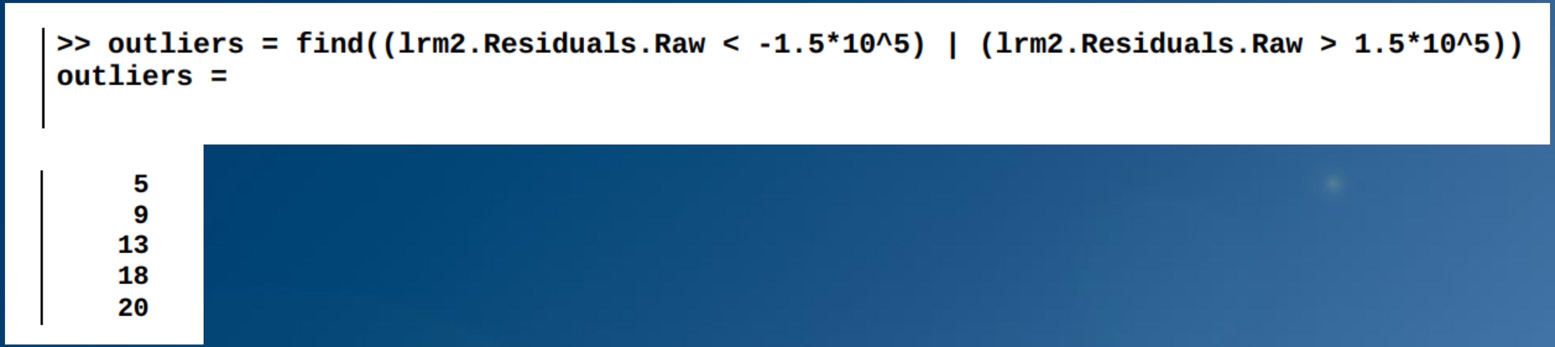
```
>> lrm2 = fitlm(VehicleData,'Registrations~Population','RobustOpts','on')
lrm2 =
Linear regression model (robust fit):
    Registrations ~ 1 + Population
Estimated Coefficients:
                   Estimate        SE         tStat       pValue
                   ________    _________    _______    _________

    (Intercept)      25059         27076    0.92549      0.36695
    Population     0.62169     0.0069234     89.796    2.505e-25
Number of observations: 20, Error degrees of freedom: 18
Root Mean Squared Error: 7.64e+04
R-squared: 0.998,  Adjusted R-Squared 0.998
F-statistic vs. constant model: 8.07e+03, p-value = 2.49e-25
```



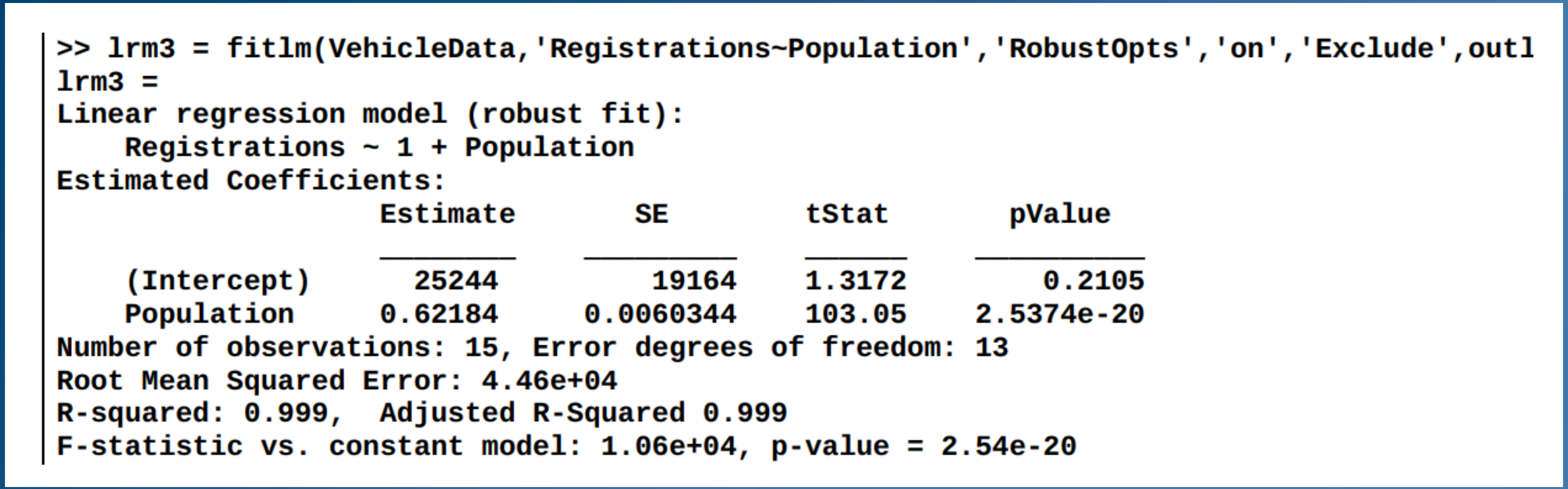
```
>> subplot(1,2,1)
>> plotResiduals(lrm1,'probability')
>> subplot(1,2,2)
>> plotResiduals(lrm2,'probability')
```

By comparing the two plots, it is obvious that the Robust model is closer to the straight line in this plot, the outliers appear more distant from the straight line and therefore are more highlighted, which helps us in the detection of them.



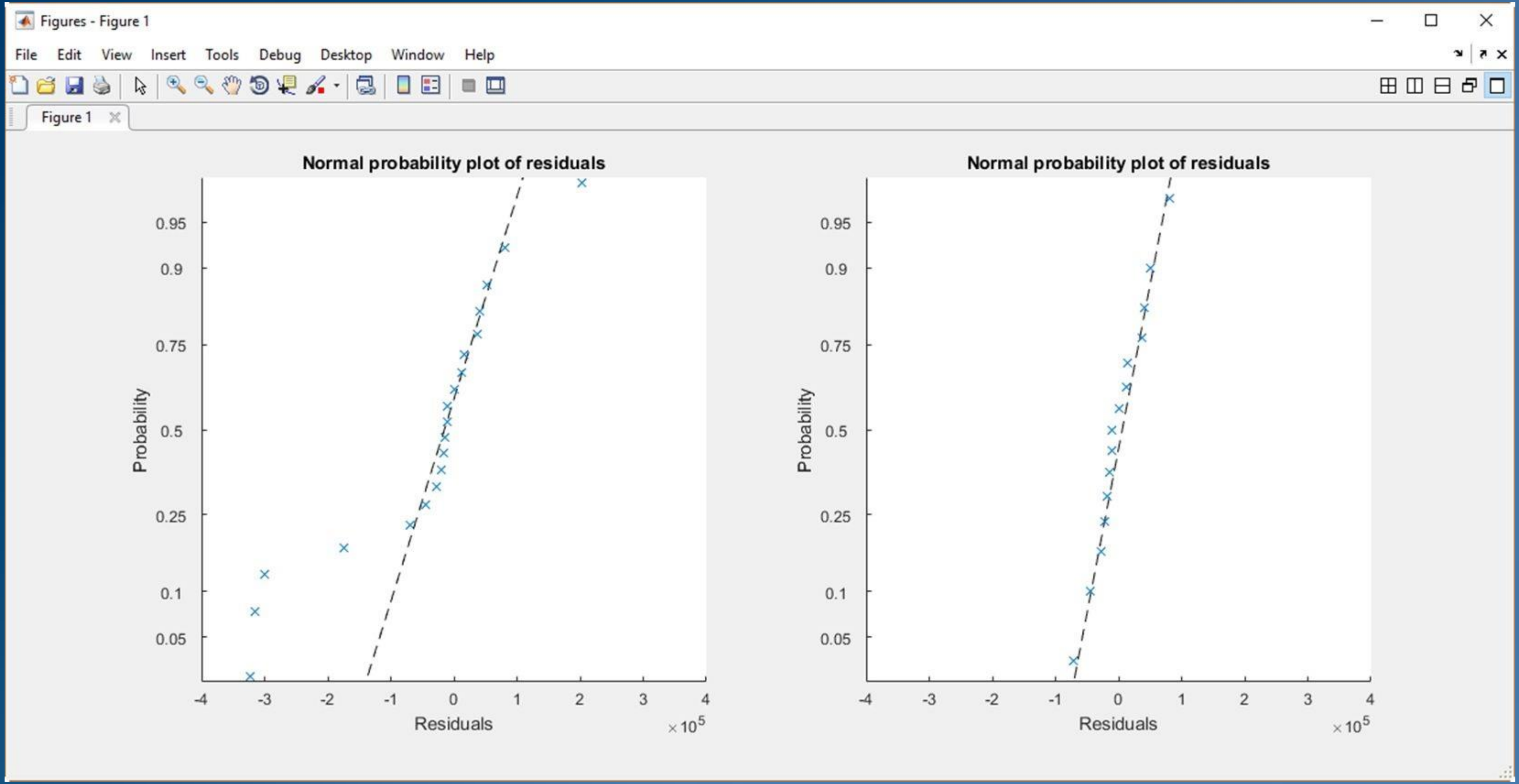
```
>> outliers = find((lrm2.Residuals.Raw < -1.5*10^5) | (lrm2.Residuals.Raw > 1.5*10^5))
outliers =


     5
     9
    13
    18
    20
```



```
>> lrm3 = fitlm(VehicleData,'Registrations~Population','RobustOpts','on','Exclude',outl
lrm3 =
Linear regression model (robust fit):
    Registrations ~ 1 + Population
Estimated Coefficients:
                   Estimate       SE         tStat       pValue
                   ________    _________    _______    __________
    (Intercept)      25244        19164     1.3172        0.2105
    Population     0.62184    0.0060344     103.05    2.5374e-20
Number of observations: 15, Error degrees of freedom: 13
Root Mean Squared Error: 4.46e+04
R-squared: 0.999,  Adjusted R-Squared 0.999
F-statistic vs. constant model: 1.06e+04, p-value = 2.54e-20
```

```
>> subplot(1,2,1)
>> plotResiduals(lrm2,'probability')
>> subplot(1,2,2)
>> plotResiduals(lrm3,'probability')
```



Normal probability plot of residuals (robust fit to the left, robust fit without outliers to the right)

2. How to create a linear regression model

Multiple linear regression

So far, we have solved simple linear regression problems, which study the relation between the dependent variable y and the independent variable x based on the regression equation:

$$y = \alpha * x + \beta$$

In this equation, x is the explanatory variable and y is the response variable. To solve this problem, we used the least squares method. In this method, the best fitting line can be found by minimizing the sum of the squares of the vertical distance from each data point on the line.

2. How to create a linear regression model

More often, the response variable depends on at least two predictors. In practice, we will have to create models with response variables that depend on more than one predictor. These models are named multiple linear regressions. According to multiple linear regression models, the dependent variable is related to two or more independent variables. The general model for n variables is of the form:

$$y = \beta_0 + \beta_1 \times x_1 + \beta_2 \times x_2 + ... + \beta_n x_n$$

Here, x_1, x_2,.. xn are the n predictors and y is the only response variable. The coefficients β measure the change in the y value associated with a change in x_i, keeping all the other variables constant.

2. How to create a linear regression model

The simple linear regression model is used to find the straight line that best fits the data. On the other hand, multiple linear regression models, for example with two independent variables, are used to find the plane that best fits the data, more generally a multidimensional plane. The goal is to find the surface that best fits our predictors in terms of minimizing the overall squared distance between itself and the response.

$$\sum_i \left[y_i - (\beta_0 + \beta_1 \times x_{1,i} + \beta_2 \times x_{2,i} + \ldots + \beta_n x_{n,i})\right]^2$$

$$\sum_i \left[y_i - (\beta_0 + \beta_1 \times x_{1,i} + \beta_2 \times x_{2,i} + ... + \beta_n x_{n,i})\right]^2$$

$$\begin{bmatrix} y_1 \\ y_2 \\ ... \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & ... & x_{1,n} \\ 1 & x_{2,1} & x_{2,2} & ... & x_{2,n} \\ ... & & & & \\ 1 & x_{n,1} & x_{n,2} & ... & x_{n,n} \end{bmatrix} \times \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ ... \\ \beta_n \end{bmatrix}$$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ ... \\ y_n \end{bmatrix}; X = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & ... & x_{1,n} \\ 1 & x_{2,1} & x_{2,2} & ... & x_{2,n} \\ ... & & & & \\ 1 & x_{n,1} & x_{n,2} & ... & x_{n,n} \end{bmatrix}; A = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ ... \\ \beta_n \end{bmatrix}$$

$Y = X \times A$

$A = X \backslash Y$

```
>> lm=fitlm(ingredients,Y)
```

```
>> b=regress(Y,X)
```

3. Polynomial regression

The linear model also includes polynomial regression, in which some predictors appear in degrees equal to or greater than 2. The model continues to be linear in the parameters. For example, a second-degree parabolic regression model looks like this:

$$y = \beta_0 + \beta_1 \times x + \beta_2 \times x^2$$

Polynomial regression is usually used when the relationship between the variables looks curved. A simple curve can sometimes be straightened out by transforming one or both of the variables. A more complicated curve, however, is best handled by polynomial regression.

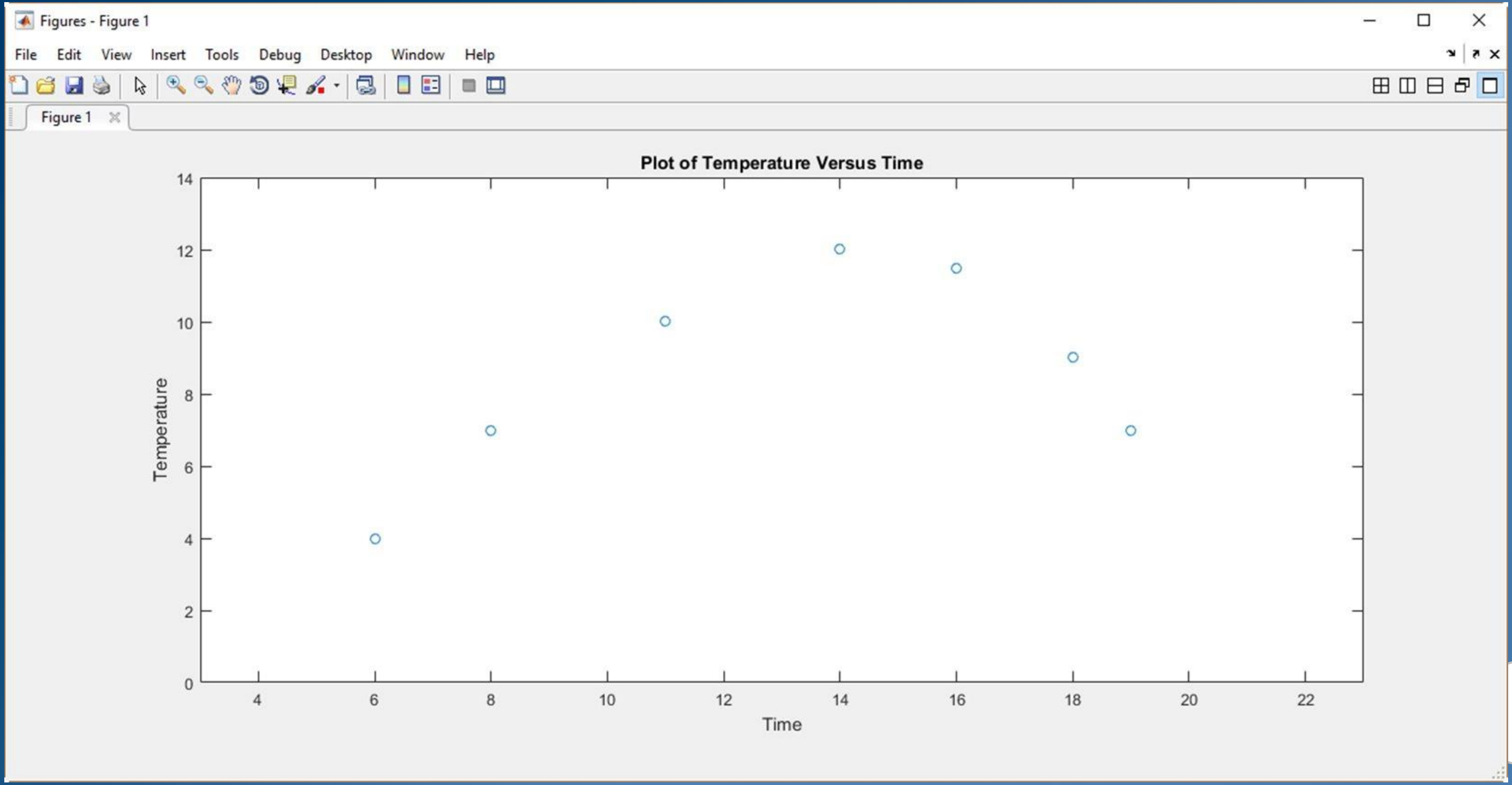
3. Polynomial regression

More generally, a polynomial regression equation assumes the following form:

$$y = \beta_0 + \beta_1 \times x + \beta_2 \times x^2 + \beta_3 \times x^3 + ... + \beta_n \times x^n$$

Now, we'll show how to model data with a polynomial. We measured the temperature in a few hours of the day. We want to know the temperature trend even in moments of the day when we did not notice it.

```
>> Time = [6 8 11 14 16 18 19];
>> Temp = [4 7 10 12 11.5 9 7];
>> plot(Time,Temp,'o')
>> title('Plot of Temperature Versus Time')
```

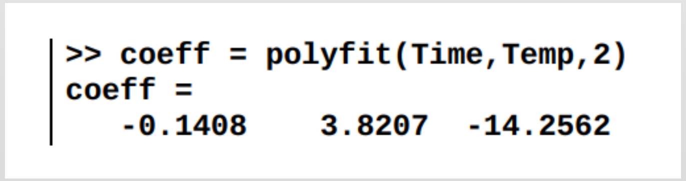

Figures - Figure 1
File Edit View Insert Tools Debug Desktop Window Help
Figure 1
Plot of Temperature Versus Time
Temperature
Time
0
2
4
6
8
10
12
14
4
6
8
10
12
14
16
18
20
22

3. Polynomial regression

From the analysis of the figure, it is possible to note a curvilinear pattern of the data that can be modeled through a second-degree polynomial as following equation:

$$temp = \beta_0 + \beta_1 \times time + \beta_2 \times time^2$$

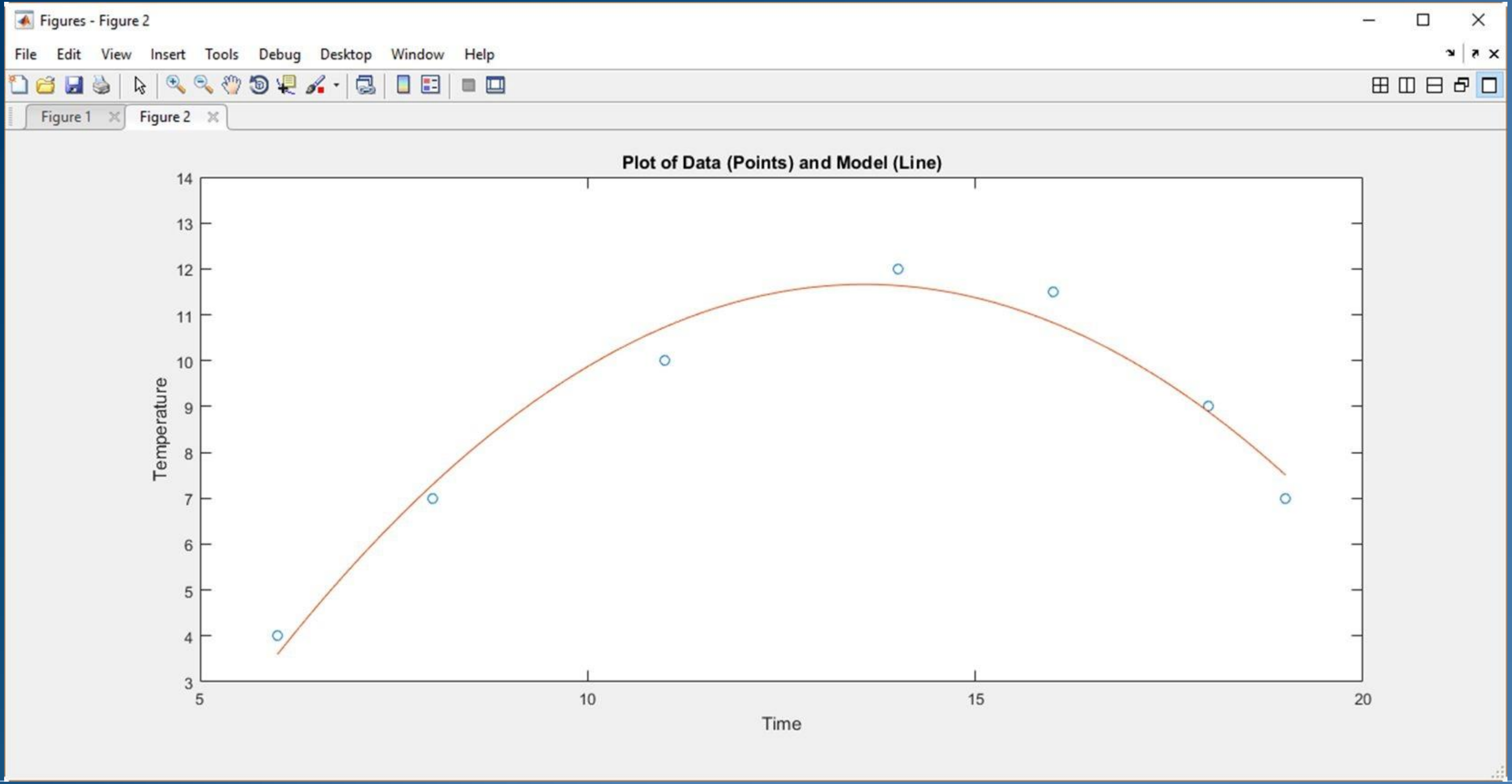
The unknown coefficients β0, β1, and β2 are computed by minimizing the sum of the squares of the deviations of the data from the model (least squares fit).



```
>> coeff = polyfit(Time,Temp,2)
coeff =
   -0.1408    3.8207  -14.2562
```

$$Temp = -0.1408 \times time^2 + 3.8207 \times time - 14.2562$$

```
>> TimeNew = 6:0.1:19;
>> TempNew = polyval(coeff,TimeNew);
>> figure
>> plot(Time,Temp,'o', TimeNew, TempNew)
>> title('Plot of Data (Points) and Model (Line)')
```


Figures - Figure 2
File
Edit
View
Insert
Tools
Debug
Desktop
Window
Help
Figure 1
Figure 2
Plot of Data (Points) and Model (Line)
Temperature
Time