MFE204TC
ARTIFICIAL INTELLIGENCE AND DATA ANALYSIS

LECTURE 3

LONG HUANG

Xi'an Jiaotong-Liverpool University
西交利物浦大学

DESCRIPTIVE STATISTICS

- Calculating Maximum, Mean, and Standard Deviation

```
% Load the sample data
load count.dat
% Find the maximum value in each
column
mx = max(count)
% Calculate the mean of each
column
mu = mean(count)
% Calculate the standard
deviation of each column
sigma = std(count)
```

```
mx =

   114   145   257


mu =

   32.0000   46.5417   65.5833


sigma =

   25.3703   41.4057   68.0281
```

DESCRIPTIVE STATISTICS

- Locate where the maximum data values occur in each data column.
- To find the minimum value in the entire count matrix, 24-by-3 matrix into a 72-by-1 column vector by using the syntax count(:). Then, to find the minimum value in the single column.

```
clear;
clc;

% Load the sample data
load count.dat
% Find the maximum value in each column
mx = max(count)
% Calculate the mean of each column
mu = mean(count)
% Calculate the standard deviation of
each column
sigma = std(count)

[mx,indx] = max(count)

min(count(:))
```

```
indx =

    20    20    20


ans =

     7
```

DESCRIPTIVE STATISTICS

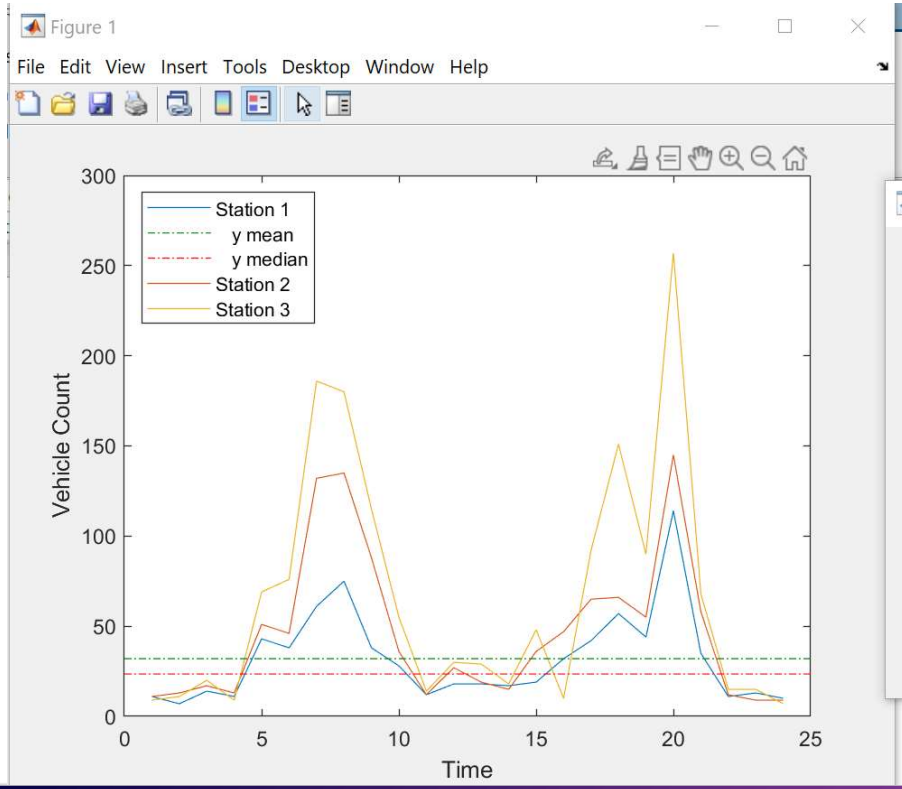
- Recall detrending data, alternatively we can subtract the mean from each column

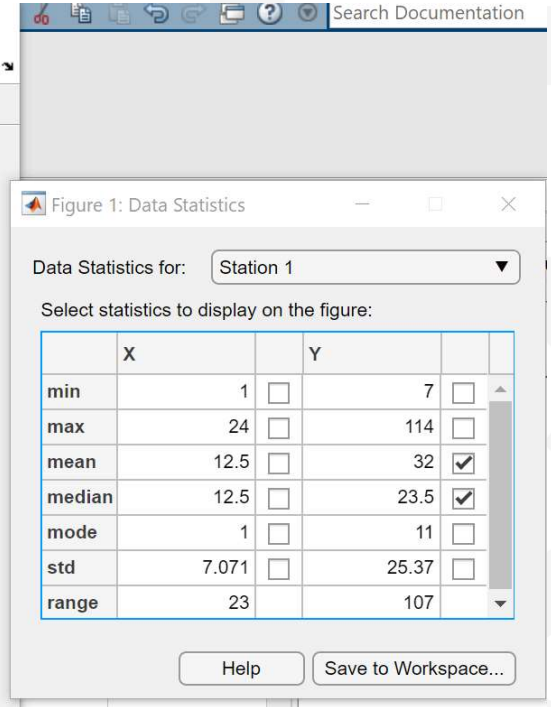
```
% Load the sample data
load count.dat
% Get the size of the count matrix
[n,p] = size(count)
% Compute the mean of each column
mu = mean(count)
% Create a matrix of mean values by
% replicating the mu vector for n rows
MeanMat = repmat(mu,n,1)
% Subtract the column mean from each
element
% in that column
x = count - MeanMat
```

DESCRIPTIVE STATISTICS

- E.g. Calculating and Plotting Descriptive Statistics

```
load count.dat
[n,p] = size(count);
% Define the x-values
t = 1:n;
% Plot the data and annotate the graph
plot(t,count)
legend('Station 1','Station 2','Station
3','Location','northwest')
xlabel('Time')
ylabel('Vehicle Count')
```





LINEAR CORRELATION

- Correlation quantifies the strength of a linear relationship between two variables.
  - When there is no correlation between two variables, then there is no tendency for the values of the variables to increase or decrease in tandem.
  - Two variables that are uncorrelated are not necessarily independent, however, because they might have a nonlinear relationship.
- You can use linear correlation to investigate whether a linear relationship exists between variables without having to assume or fit a specific model to your data.

COVARIANCE

- Covariance quantifies the strength of a linear relationship between two variables in units relative to their variances.

$$\begin{bmatrix} s^2_{11} & s^2_{12} & s^2_{13} \\ s^2_{21} & s^2_{22} & s^2_{23} \\ s^2_{31} & s^2_{32} & s^2_{33} \end{bmatrix}$$

$$s^2_{ij} = s^2_{ji}$$

s^2_{ij} is the sample covariance between column *i* and column *j* of the data. Because the count matrix contains three columns, the covariance matrix is 3-by-3.

For two random variable vectors A and B, the covariance is defined as

$$\mathrm{cov}(A, B) = \frac{1}{N-1} \sum_{i=1}^{N} (A_i - \mu_A)^* (B_i - \mu_B)$$

where μ_A is the mean of A, μ_B is the mean of B, and * denotes the complex conjugate.

https://ww2.mathworks.cn/help/matlab/ref/cov.html#buo3tdl-1

COVARIANCE

- Use the MATLAB *cov* function to calculate the sample covariance matrix for a data matrix (where each column represents a separate quantity).

```
load count.dat
%Calculate the covariance matrix
for this data:
cov = cov(count);
```

```
cov =

   1.0e+03 *

    0.6437    0.9802    1.6567
    0.9802    1.7144    2.6908
    1.6567    2.6908    4.6278
```

CORRELATION COEFFICIENTS

- The function *corrcoef* produces a matrix of sample correlation coefficients for a data matrix (where each column represents a separate quantity). The correlation coefficients range from -1 to 1, where
  - Values close to 1 indicate that there is a positive linear relationship between the data columns.
  - Values close to -1 indicate that one column of data has a negative linear relationship to another column of data (anticorrelation).
  - Values close to or equal to 0 suggest there is no linear relationship between the data columns.

CORRELATION COEFFICIENTS

- Use *corrcoef* to calculate the correlation coefficients:

```
load count.dat
%Calculate the correlation
coefficients for this data:
corr=corrcoef(count);
corr
```

```
corr =

    1.0000    0.9331    0.9599
    0.9331    1.0000    0.9553
    0.9599    0.9553    1.0000
```

- Because all correlation coefficients are close to 1, there is a strong positive correlation between each pair of data columns in the count matrix.

LINEAR REGRESSION

- A data model explicitly describes a relationship between predictor and response variables.
  - Linear regression fits a data model that is linear in the model coefficients.
  - We are going to:
    - Perform simple linear regression using the \ operator.
    - Use correlation analysis to determine whether two quantities are related to justify fitting the data.
    - Fit a linear model to the data.
    - Evaluate the goodness of fit by plotting residuals and looking for patterns.
    - Calculate measures of goodness of fit R^2 and adjusted R^2

SIMPLE LINEAR REGRESSION

- Linear regression models the relation between a dependent, or response, variable y and one or more independent, or predictor, variables $x_1, \ldots, x_n$. Simple linear regression considers only one independent variable using the relation.

$$y = \beta_0 + \beta_1 x + \epsilon$$

- where β_0 is the y-intercept, β_1 is the slope (or regression coefficient), and ϵ is the error term.
- For a data set: $Y = XB$

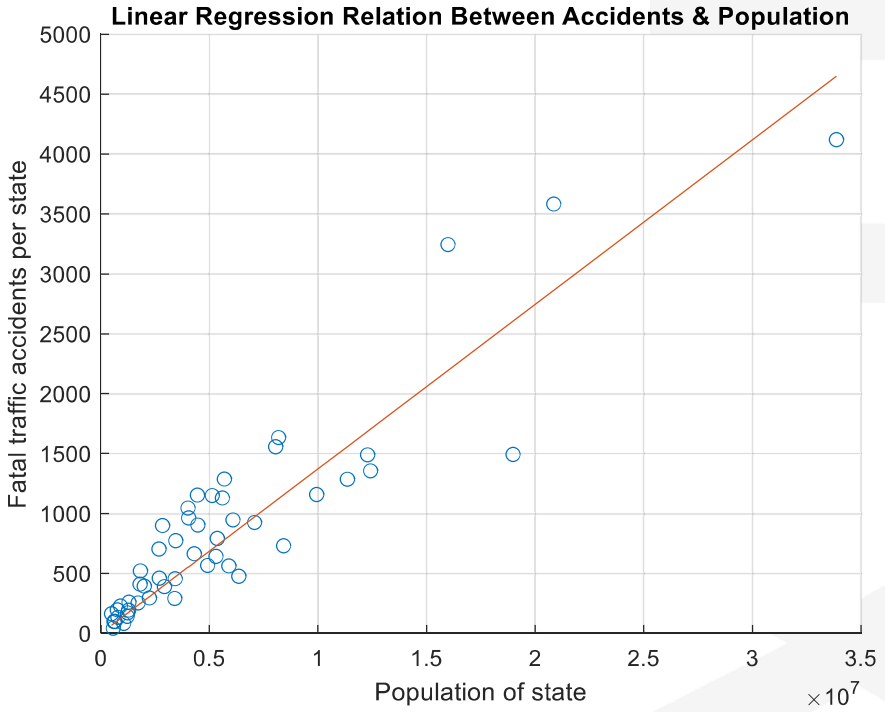
$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}, B = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}.$$

SIMPLE LINEAR REGRESSION

- From the dataset accidents, load accident data in y and state population data in x. Find the linear regression relation $y = \beta_1 x$ between the accidents in a state and the population of a state using the \ operator.
- The \ operator performs a least-squares regression.

```
load accidents
x = hwydata(:,14); %Population of states
y = hwydata(:,4); %Accidents per state
format long
b1 = x\y

yCalc1 = b1*x;
scatter(x,y)
hold on
plot(x,yCalc1)
xlabel('Population of state')
ylabel('Fatal traffic accidents per state')
title('Linear Regression Relation Between
Accidents & Population')
grid on
```



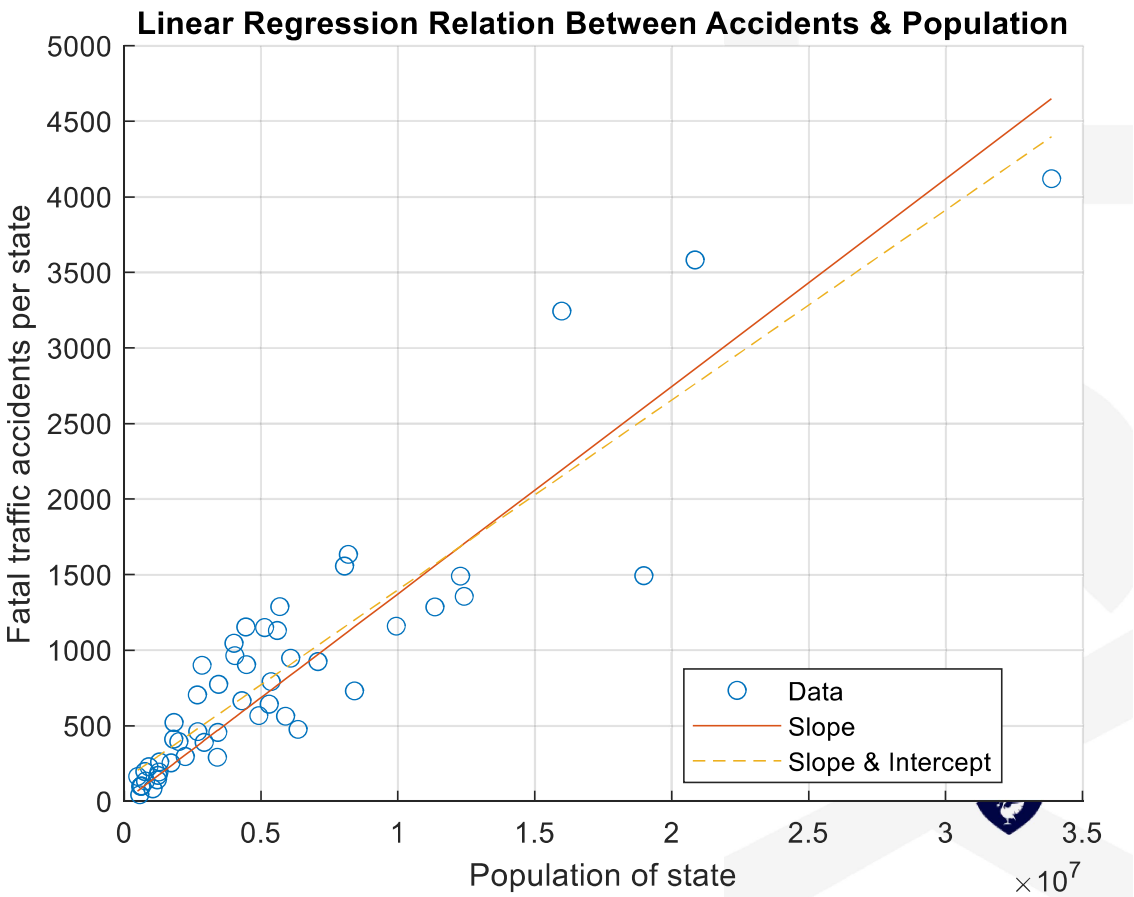
SIMPLE LINEAR REGRESSION

- Improve the fit by including a y-intercept β_0 in your model as $y = \beta_0 + \beta_1 x$. Calculate β_0 by padding x with a column of ones and using the \ operator.

```
load accidents
x = hwydata(:,14); %Population of states
y = hwydata(:,4); %Accidents per state
format long
b1 = x\y

yCalc1 = b1*x;
scatter(x,y)
hold on
plot(x,yCalc1)
xlabel('Population of state')
ylabel('Fatal traffic accidents per state')
title('Linear Regression Relation Between
Accidents & Population')
grid on

X = [ones(length(x),1) x];
b = X\y
yCalc2 = X*b;
plot(x,yCalc2,'--')
legend('Data','Slope','Slope &
Intercept','Location','best');
```



RESIDUALS AND GOODNESS OF FIT

- Residuals are the difference between the observed values of the response (dependent) variable and the values that a model predicts.
- Producing a fit using a linear model requires minimizing the sum of the squares of the residuals. This minimization yields what is called a least-squares fit.
- One measure of goodness of fit is the coefficient of determination, or R^2

$$R^2 = 1 - SS_{resid} / SS_{total}$$

- SS_{resid} is the sum of the squared residuals from the regression.
- SS_{total} is the sum of the squareddifferences from the mean of the dependent variable (total sum of squares).
- Q: calculate R-square for the examples in last 2 slides

POLYNOMIAL REGRESSIONS

- *polyfit* can be used to compute a linear regression that predicts y from x
- p = polyfit(x,y,n) returns the coefficients for a polynomial p(x) of degree n that is a best fit (in a least-squares sense) for the data in y.

$$p(x)=p_1x^n+p_2x^{n-1}+...+p_nx+p_{n+1}.$$

- A linear fit has a degree of 1, a quadratic fit 2, a cubic fit 3, and so on.

ADJUSTED R^2 FOR POLYNOMIAL REGRESSIONS

- For higher degree polynomial. When you add more terms, you increase the coefficient of determination, R^2. You get a closer fit to the data, but at the expense of a more complex model, for which R2 cannot account.
- Adjusted R2, does include a penalty for the number of terms in a model.

$$R^2_{adjusted} = 1 - (SS_{resid} / SS_{total})*((n-1)/(n-d-1))$$

- *d* is the degree of the polynomial.
- n is the number of data points
- In-class practice: Compare linear fit, a quadratic fit, cubic fit. Compute R^2 and adjusted R^2

```
load count.dat
x = count(:,1);
y = count(:,2);
```