ENCN305 Regression Modelling

Lecturer: Dr Raazesh Sainudiin (Raaz)

Notes by Dr Carl Scarrott

University of Canterbury

May 3 2013

Example: Evaporation at Different Air Velocities

Measurements from experiment considering the evaporation coefficient of a liquid at fixed air velocities

Velocity (cm/s)	20	60	100	140	180	220	260	300	340	380
Evaporation (mm^2/s)	0.18	0.37	0.35	0.78	0.56	0.75	1.18	1.36	1.17	1.65

What is Regression?

Regression analysis use various models and techniques for describing the relationships between two or more variables, which can potentially be used for prediction purposes.

What is Regression?

Regression analysis use various models and techniques for describing the relationships between two or more variables, which can potentially be used for prediction purposes.

We will consider two basic regression models:

What is Regression?

Regression analysis use various models and techniques for describing the relationships between two or more variables, which can potentially be used for prediction purposes.

We will consider two basic regression models:

- **Simple linear regression** has a *linear relationship* between a *single explanatory* variable and a *single response* variable.

What is Regression?

Regression analysis use various models and techniques for describing the relationships between two or more variables, which can potentially be used for prediction purposes.

We will consider two basic regression models:

- **Simple linear regression** has a *linear relationship* between a *single explanatory* variable and a *single response* variable.
- **Multiple linear regression** extends this basic model to allow for *multiple explanatory* variables.

What is Regression?

Regression analysis use various models and techniques for describing the relationships between two or more variables, which can potentially be used for prediction purposes.

We will consider two basic regression models:

- **Simple linear regression** has a *linear relationship* between a *single explanatory* variable and a *single response* variable.
- **Multiple linear regression** extends this basic model to allow for *multiple explanatory* variables.

Alternative terminologies:

What is Regression?

Regression analysis use various models and techniques for describing the relationships between two or more variables, which can potentially be used for prediction purposes.

We will consider two basic regression models:

- **Simple linear regression** has a *linear relationship* between a *single explanatory* variable and a *single response* variable.
- **Multiple linear regression** extends this basic model to allow for *multiple explanatory* variables.

Alternative terminologies:

- **explanatory variables** - predictor, input or independent variable

What is Regression?

Regression analysis use various models and techniques for describing the relationships between two or more variables, which can potentially be used for prediction purposes.

We will consider two basic regression models:

- **Simple linear regression** has a *linear relationship* between a *single explanatory* variable and a *single response* variable.
- **Multiple linear regression** extends this basic model to allow for *multiple explanatory* variables.

Alternative terminologies:

- **explanatory variables** - predictor, input or independent variable
- **response variable** - predictand, output or dependent variable.

Data for Regression Modelling

Observed "data vectors" of these variables are used to derive the form and quantify the strength of the linear relationship. The *fitted model* can then be used to *predict the response given a value of the explanatory variable*.

Data for Regression Modelling

Observed "data vectors" of these variables are used to derive the form and quantify the strength of the linear relationship. The *fitted model* can then be used to *predict the response given a value of the explanatory variable*.

For example, if you want to predict your exam mark on ENCN305 using your test and assignment marks, then the:

- **explanatory variables** -

Data for Regression Modelling

Observed "data vectors" of these variables are used to derive the form and quantify the strength of the linear relationship. The *fitted model* can then be used to *predict the response given a value of the explanatory variable*.

For example, if you want to predict your exam mark on ENCN305 using your test and assignment marks, then the:

- **explanatory variables** - test marks, assignment 1 - 3 marks, (possibly EMTH210 or MATH103 final grades?)

Data for Regression Modelling

Observed "data vectors" of these variables are used to derive the form and quantify the strength of the linear relationship. The *fitted model* can then be used to *predict the response given a value of the explanatory variable*.

For example, if you want to predict your exam mark on ENCN305 using your test and assignment marks, then the:

- **explanatory variables** - test marks, assignment 1 - 3 marks, (possibly EMTH210 or MATH103 final grades?)
- **response variable** -

Data for Regression Modelling

Observed "data vectors" of these variables are used to derive the form and quantify the strength of the linear relationship. The *fitted model* can then be used to *predict the response given a value of the explanatory variable*.

For example, if you want to predict your exam mark on ENCN305 using your test and assignment marks, then the:

- **explanatory variables** - test marks, assignment 1 - 3 marks, (possibly EMTH210 or MATH103 final grades?)
- **response variable** - ENCN305 exam mark.

Data for Regression Modelling

Observed "data vectors" of these variables are used to derive the form and quantify the strength of the linear relationship. The *fitted model* can then be used to *predict the response given a value of the explanatory variable*.

For example, if you want to predict your exam mark on ENCN305 using your test and assignment marks, then the:

- **explanatory variables** - test marks, assignment 1 - 3 marks, (possibly EMTH210 or MATH103 final grades?)
- **response variable** - ENCN305 exam mark.

We'd expect a **strong positive association** between the exam & test marks, though associations with assignments are often weaker

Data for Regression Modelling

Observed "data vectors" of these variables are used to derive the form and quantify the strength of the linear relationship. The *fitted model* can then be used to *predict the response given a value of the explanatory variable*.

For example, if you want to predict your exam mark on ENCN305 using your test and assignment marks, then the:

- **explanatory variables** - test marks, assignment 1 - 3 marks, (possibly EMTH210 or MATH103 final grades?)
- **response variable** - ENCN305 exam mark.

We'd expect a **strong positive association** between the exam & test marks, though associations with assignments are often weaker So you'd expect to be able to reliably **predict** your exam marks using the test marks, but often the assignment marks do not "**significantly improve the model fit**".

What Will We Cover In This Course?

On this course you will learn how to:

- **specify** appropriate simple/multiple linear regression models;

What Will We Cover In This Course?

On this course you will learn how to:

- **specify** appropriate simple/multiple linear regression models;
- **fit** the model to data;

What Will We Cover In This Course?

On this course you will learn how to:

- **specify** appropriate simple/multiple linear regression models;
- **fit** the model to data;
- **evaluate** the model performance;

What Will We Cover In This Course?

On this course you will learn how to:

- **specify** appropriate simple/multiple linear regression models;
- **fit** the model to data;
- **evaluate** the model performance;
- **choose** between alternative models;

What Will We Cover In This Course?

On this course you will learn how to:

- **specify** appropriate simple/multiple linear regression models;
- **fit** the model to data;
- **evaluate** the model performance;
- **choose** between alternative models;
- **predict** using the model; and

What Will We Cover In This Course?

On this course you will learn how to:

- **specify** appropriate simple/multiple linear regression models;
- **fit** the model to data;
- **evaluate** the model performance;
- **choose** between alternative models;
- **predict** using the model; and
- **quantify** the model uncertainties and evaluate it's statistical usefulness.

What Will We Cover In This Course?

On this course you will learn how to:

- **specify** appropriate simple/multiple linear regression models;
- **fit** the model to data;
- **evaluate** the model performance;
- **choose** between alternative models;
- **predict** using the model; and
- **quantify** the model uncertainties and evaluate it's statistical usefulness.

What Will We Cover In This Course?

On this course you will learn how to:

- **specify** appropriate simple/multiple linear regression models;
- **fit** the model to data;
- **evaluate** the model performance;
- **choose** between alternative models;
- **predict** using the model; and
- **quantify** the model uncertainties and evaluate it's statistical usefulness.

You will also see some of the **common pitfalls of regression models**.

Simple Linear Regression

[1] Google "generalised linear model" if you want to know more

Simple Linear Regression

A simple linear regression model with **explanatory variable** X and **response** Y can be written:

[1] Google "generalised linear model" if you want to know more

Simple Linear Regression

A simple linear regression model with **explanatory variable** X and **response** Y can be written:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

let's draw it →

[1] Google "generalised linear model" if you want to know more

Simple Linear Regression

A simple linear regression model with **explanatory variable** X and **response** Y can be written:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

let's draw it →

where ϵ is the error term. The properties of the errors are discussed further below.

[1] Google "generalised linear model" if you want to know more

Simple Linear Regression

A simple linear regression model with **explanatory variable** X and **response** Y can be written:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

let's draw it →

where ϵ is the error term. The properties of the errors are discussed further below.

In this course, the response and explanatory variables are assumed to be **continuous**, i.e. can take any value over a range.

[1] Google "generalised linear model" if you want to know more

Simple Linear Regression

A simple linear regression model with **explanatory variable** X and **response** Y can be written:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

let's draw it →

where ϵ is the error term. The properties of the errors are discussed further below.

In this course, the response and explanatory variables are assumed to be **continuous**, i.e. can take any value over a range.

Although extensions to discrete explanatory variables or alternative response distributions are available (e.g. discrete distributions or restriction to range [0,1] for predicting probabilities).[1]

[1] Google "generalised linear model" if you want to know more

Fixed Versus Conditional Modelling

Fixed Versus Conditional Modelling

Need to differentiate two situations for data collection of the explanatory variable(s):

Fixed Versus Conditional Modelling

Need to differentiate two situations for data collection of the explanatory variable(s):

- **fixed** X values, typically resulting from experiments where the values can be controlled in order to explore possible causal relationships; and

Fixed Versus Conditional Modelling

Need to differentiate two situations for data collection of the explanatory variable(s):

- **fixed** X values, typically resulting from experiments where the values can be controlled in order to explore possible causal relationships; and
- **random** X values, in which case the modelling is **conditional** on the observed values and care must be taken as formally the results cannot be used to explore causal relationships.

Fixed Versus Conditional Modelling

Need to differentiate two situations for data collection of the explanatory variable(s):

- **fixed** X values, typically resulting from experiments where the values can be controlled in order to explore possible causal relationships; and
- **random** X values, in which case the modelling is **conditional** on the observed values and care must be taken as formally the results cannot be used to explore causal relationships.

Note: in the latter case it is further assumed these measurement are not subject to error (e.g. measurement error), otherwise a more sophisticated "**errors-in-variables**" model is required.

A Bit of Terminology

A Bit of Terminology

The primary **parameters** of the model $Y = \beta_0 + \beta_1 X + \epsilon$ are the **intercept** β_0 and **gradient** β_1 (often called *regression coefficients*)

A Bit of Terminology

The primary **parameters** of the model $Y = \beta_0 + \beta_1 X + \epsilon$ are the **intercept** β_0 and **gradient** β_1 (often called *regression coefficients*)

Usually estimated using a sample dataset consisting of pairs of observations $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$.

A Bit of Terminology

The primary **parameters** of the model $Y = \beta_0 + \beta_1 X + \epsilon$ are the **intercept** β_0 and **gradient** β_1 (often called *regression coefficients*)

Usually estimated using a sample dataset consisting of pairs of observations $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$.

Notice that the **sample size** n is the number of observation pairs

A Bit of Terminology

The primary **parameters** of the model $Y = \beta_0 + \beta_1 X + \epsilon$ are the **intercept** β_0 and **gradient** β_1 (often called *regression coefficients*)

Usually estimated using a sample dataset consisting of pairs of observations $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$.

Notice that the **sample size** n is the number of observation pairs

Interpretation of coefficients:

A Bit of Terminology

The primary **parameters** of the model $Y = \beta_0 + \beta_1 X + \epsilon$ are the **intercept** β_0 and **gradient** β_1 (often called *regression coefficients*)

Usually estimated using a sample dataset consisting of pairs of observations $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$.

Notice that the **sample size** n is the number of observation pairs

Interpretation of coefficients:

- β_1 is effect of unit increase in the explanatory variable X on response Y

A Bit of Terminology

The primary **parameters** of the model $Y = \beta_0 + \beta_1 X + \epsilon$ are the **intercept** β_0 and **gradient** β_1 (often called *regression coefficients*)

Usually estimated using a sample dataset consisting of pairs of observations $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$.

Notice that the **sample size** n is the number of observation pairs

Interpretation of coefficients:

- β_1 is effect of unit increase in the explanatory variable X on response Y
- β_0 is value of response Y when explanatory variable $X = 0$

What Is Meant By “Linear” Regression Model?

What Is Meant By "Linear" Regression Model?

"**Linear**" regression models are **linear in the unknown parameters**, but there may be a non-linear relationship between the X and Y. This is a *common misconception*.

What Is Meant By "Linear" Regression Model?

"**Linear**" regression models are **linear in the unknown parameters**, but there may be a non-linear relationship between the X and Y. This is a *common misconception*.

For example, the following quadratic form is a linear model:

$$Y = \beta_0 + \beta_1 X^2 + \epsilon,$$

What Is Meant By "Linear" Regression Model?

"**Linear**" regression models are **linear in the unknown parameters**, but there may be a non-linear relationship between the X and Y. This is a *common misconception*.

For example, the following quadratic form is a linear model:

$$Y = \beta_0 + \beta_1 X^2 + \epsilon,$$

and this is a non-linear model:

$$Y = \beta_0 X_1^{\beta_1} X_2^{\beta_2} \epsilon.$$

What Is Meant By "Linear" Regression Model?

"**Linear**" regression models are **linear in the unknown parameters**, but there may be a non-linear relationship between the X and Y. This is a *common misconception*.

For example, the following quadratic form is a linear model:

$$Y = \beta_0 + \beta_1 X^2 + \epsilon,$$

and this is a non-linear model:

$$Y = \beta_0 X_1^{\beta_1} X_2^{\beta_2} \epsilon.$$

In this example, it is possible to turn this non-linear model into a linear model by taking the logarithm of both sides of the equation.

What Is Meant By "Linear" Regression Model?

"**Linear**" regression models are **linear in the unknown parameters**, but there may be a non-linear relationship between the X and Y. This is a *common misconception*.

For example, the following quadratic form is a linear model:

$$Y = \beta_0 + \beta_1 X^2 + \epsilon,$$

and this is a non-linear model:

$$Y = \beta_0 X_1^{\beta_1} X_2^{\beta_2} \epsilon.$$

In this example, it is possible to turn this non-linear model into a linear model by taking the logarithm of both sides of the equation. However, general techniques required for solving **non-linear models** are beyond the scope of this course.

Descriptive Terms

Descriptive Terms

The first step in regression is to determine whether or not there is an observed **association** between the variables of interest.

Descriptive Terms

The first step in regression is to determine whether or not there is an observed **association** between the variables of interest.

The term "association" is used to avoid complications of causal interpretations of the term relationship.

Descriptive Terms

The first step in regression is to determine whether or not there is an observed **association** between the variables of interest.

The term "association" is used to avoid complications of causal interpretations of the term relationship.

A **scatterplot** can be a useful tool to examine:

Descriptive Terms

The first step in regression is to determine whether or not there is an observed **association** between the variables of interest.

The term "association" is used to avoid complications of causal interpretations of the term relationship.

A **scatterplot** can be a useful tool to examine:

- **strength** - weak, moderate or strong association;

Descriptive Terms

The first step in regression is to determine whether or not there is an observed **association** between the variables of interest.

The term "association" is used to avoid complications of causal interpretations of the term relationship.

A **scatterplot** can be a useful tool to examine:

- **strength** - weak, moderate or strong association;
- **direction** - positive or negative;

Descriptive Terms

The first step in regression is to determine whether or not there is an observed **association** between the variables of interest.

The term "association" is used to avoid complications of causal interpretations of the term relationship.

A **scatterplot** can be a useful tool to examine:

- **strength** - weak, moderate or strong association;
- **direction** - positive or negative;
- **form** - e.g. linear or quadratic.

Example: Evaporation at Different Air Velocities

Example: Evaporation at Different Air Velocities

Let's return to the measurements from experiment considering the evaporation coefficient of a liquid at fixed air velocities

Example: Evaporation at Different Air Velocities

Let's return to the measurements from experiment considering the evaporation coefficient of a liquid at fixed air velocities

Velocity (cm/s)	20	60	100	140	180	220	260	300	340	380
Evaporation (mm^2/s)	0.18	0.37	0.35	0.78	0.56	0.75	1.18	1.36	1.17	1.65

Example: Evaporation at Different Air Velocities

Let's return to the measurements from experiment considering the evaporation coefficient of a liquid at fixed air velocities

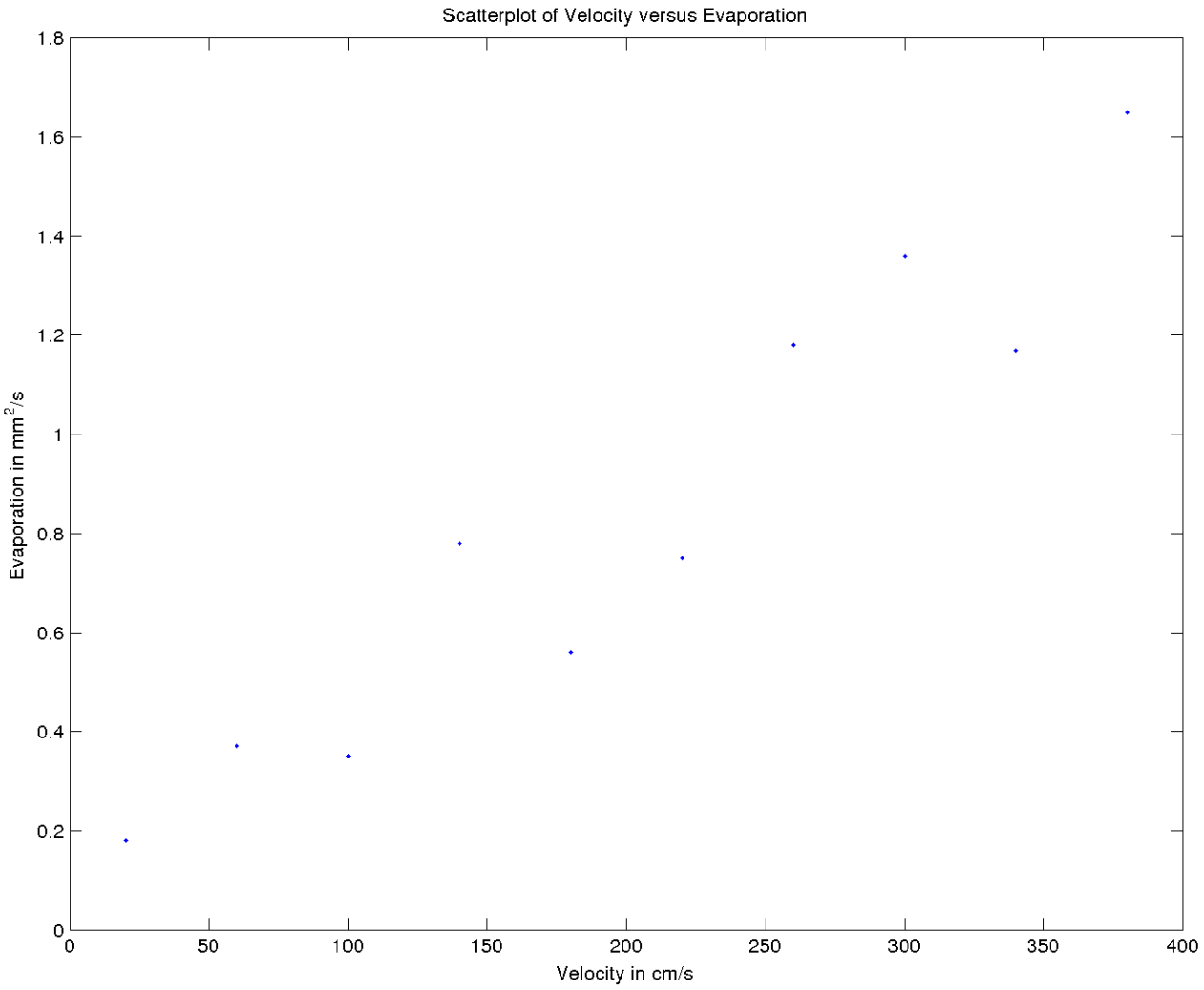
Velocity (cm/s)	20	60	100	140	180	220	260	300	340	380
Evaporation (mm^2/s)	0.18	0.37	0.35	0.78	0.56	0.75	1.18	1.36	1.17	1.65

Use `MATLAB` menu option `File > Import Data...` to import data (pretty intuitive interface). Then plot the data:

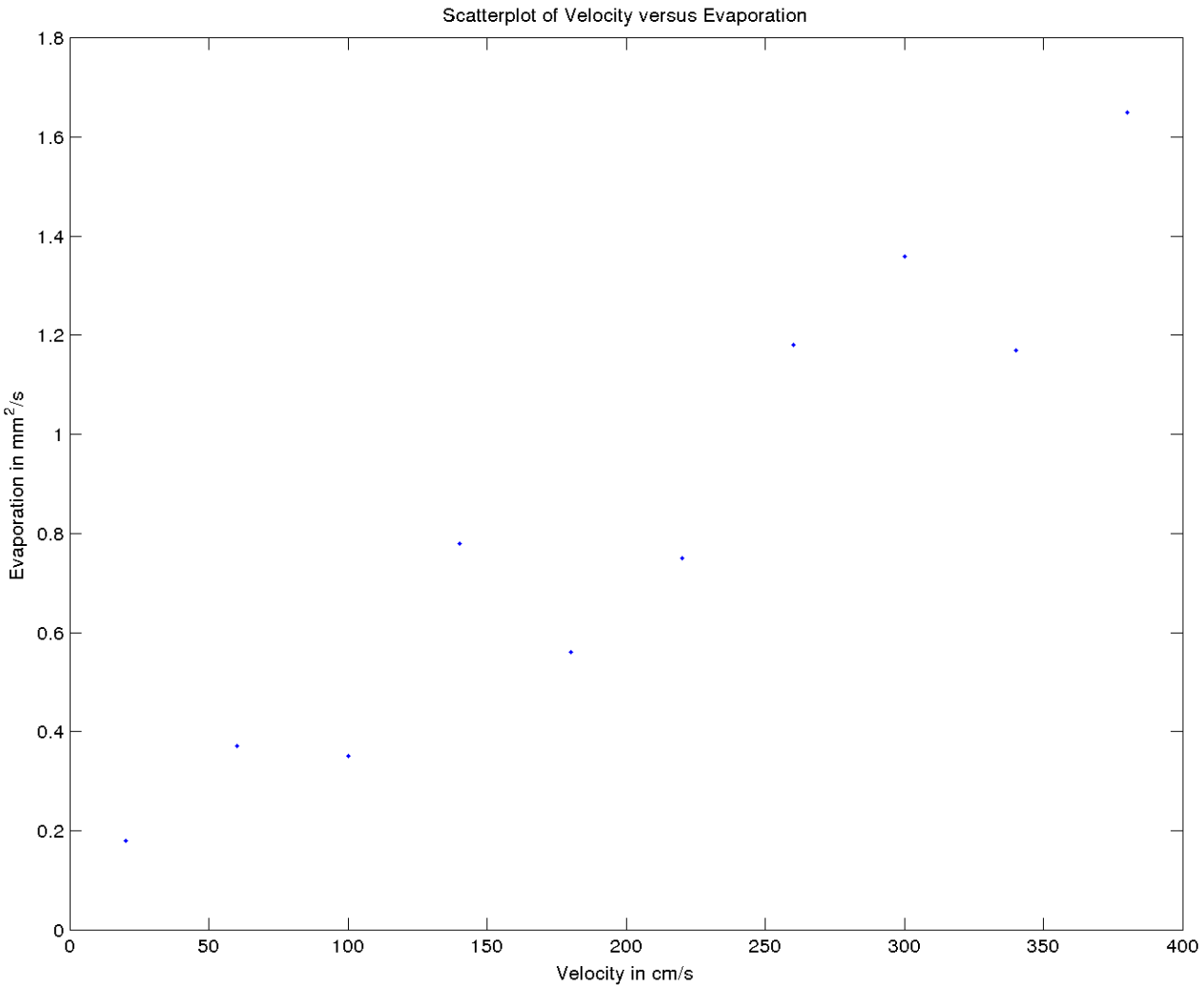
```
%
% Import evaporation data via File > Import Data... menu option
% Following produces scatterplot of data
%
plot(Velocity, Evaporation,'.')
xlabel('Velocity in cm/s')
ylabel('Evaporation in mm^2/s')
title('Scatterplot of Velocity versus Evaporation')
```

coded in `../Datasets/Evaporation.m`

Example: Evaporation Scatterplot

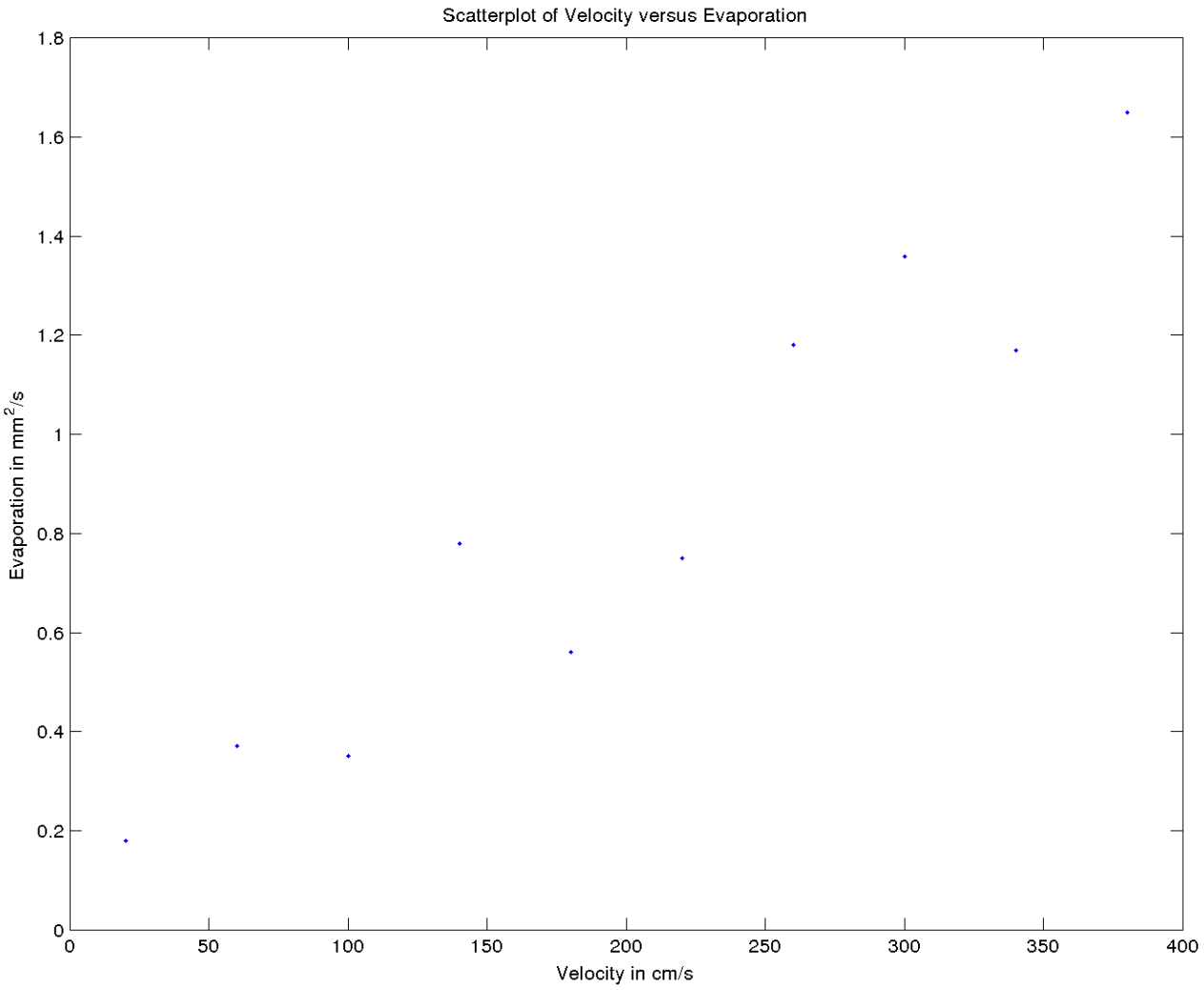


Example: Evaporation Scatterplot



- It looks as though there is a strong positive linear association between air velocity and evaporation

Example: Evaporation Scatterplot



- It looks as though there is a strong positive linear association between air velocity and evaporation
- But how strong is it? Could it be spurious (just due to natural sample variability with no underlying correlation)?

Correlation Coefficient

Correlation Coefficient

Strength of linear association can be measured using the **Pearson product moment "correlation coefficient"**:

Correlation Coefficient

Strength of linear association can be measured using the **Pearson product moment "correlation coefficient"**:

$$\rho_{XY} = \frac{Cov(X, Y)}{\sqrt{Var(X)Var(Y)}}$$

Correlation Coefficient

Strength of linear association can be measured using the **Pearson product moment "correlation coefficient"**:

$$\rho_{XY} = \frac{Cov(X, Y)}{\sqrt{Var(X)Var(Y)}}$$

where:

$$\begin{aligned} Var(X) &= E\{(X - \mu_X)^2\} \\ Var(Y) &= E\{(Y - \mu_Y)^2\}, \end{aligned}$$

Correlation Coefficient

Strength of linear association can be measured using the **Pearson product moment "correlation coefficient"**:

$$\rho_{XY} = \frac{Cov(X, Y)}{\sqrt{Var(X)Var(Y)}}$$

where:

$$\begin{aligned} Var(X) &= E\{(X - \mu_X)^2\} \\ Var(Y) &= E\{(Y - \mu_Y)^2\}, \end{aligned}$$

and the covariance measures how X and Y vary with each other:

$$Cov(X, Y) = E\{(X - \mu_X)(Y - \mu_Y)\}$$

.

Correlation Coefficient

Strength of linear association can be measured using the **Pearson product moment "correlation coefficient"**:

$$\rho_{XY} = \frac{Cov(X, Y)}{\sqrt{Var(X)Var(Y)}}$$

where:

$$\begin{aligned} Var(X) &= E\{(X - \mu_X)^2\} \\ Var(Y) &= E\{(Y - \mu_Y)^2\}, \end{aligned}$$

and the covariance measures how X and Y vary with each other:

$$Cov(X, Y) = E\{(X - \mu_X)(Y - \mu_Y)\}$$

- There are many different measures of correlation, this is the most commonly used

Problem With Covariance Interpretation

The covariance is a measure of how X and Y linearly vary with each other where:

- if Y tends to **increase** as X **increases** then the covariance will be **positive**;

Problem With Covariance Interpretation

The covariance is a measure of how X and Y linearly vary with each other where:

- if Y tends to **increase** as X **increases** then the covariance will be **positive**;
- if Y tends to **decrease** as X **increases** then the covariance will be **negative**; and

Problem With Covariance Interpretation

The covariance is a measure of how X and Y linearly vary with each other where:

- if Y tends to **increase** as X **increases** then the covariance will be **positive**;
- if Y tends to **decrease** as X **increases** then the covariance will be **negative**; and
- the unit of measurement is X times Y.

Problem With Covariance Interpretation

The covariance is a measure of how X and Y linearly vary with each other where:

- if Y tends to **increase** as X **increases** then the covariance will be **positive**;
- if Y tends to **decrease** as X **increases** then the covariance will be **negative**; and
- the unit of measurement is X times Y.

Problem With Covariance Interpretation

The covariance is a measure of how X and Y linearly vary with each other where:

- if Y tends to **increase** as X **increases** then the covariance will be **positive**;
- if Y tends to **decrease** as X **increases** then the covariance will be **negative**; and
- the unit of measurement is X times Y.

The lack of invariance to scale of measurement makes the covariance hard to interpret.

Problem With Covariance Interpretation

The covariance is a measure of how X and Y linearly vary with each other where:

- if Y tends to **increase** as X **increases** then the covariance will be **positive**;
- if Y tends to **decrease** as X **increases** then the covariance will be **negative**; and
- the unit of measurement is X times Y.

The lack of invariance to scale of measurement makes the covariance hard to interpret.

The correlation is on standardised scale so interpretation does not depend on the measurement scale of X and Y. The division by the standard deviations (square root of variances) ensure the correlations are on the range -1 to 1.

Calculating Correlation For Sample Data

Calculating Correlation For Sample Data

Sample correlation coefficient is estimated using:

$$\hat{\rho}_{XY} = \frac{\hat{\sigma}_{XY}}{\hat{\sigma}_X \hat{\sigma}_Y} \tag{1}$$

Calculating Correlation For Sample Data

Sample correlation coefficient is estimated using:

$$\hat{\rho}_{XY} = \frac{\hat{\sigma}_{XY}}{\hat{\sigma}_X \hat{\sigma}_Y} \tag{1}$$

where the sample covariance and variances estimators are:

$$\begin{aligned}
\hat{\sigma}_{XY} &= \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{n-1} && \left(\text{or } \frac{\sum x_i y_i - n\bar{x}\bar{y}}{n-1}\right) \\
\hat{\sigma}_X^2 &= \frac{\sum (x_i - \bar{x})^2}{n-1} && \left(\text{or } \frac{\sum x_i^2 - n\bar{x}^2}{n-1}\right) \\
\hat{\sigma}_Y^2 &= \frac{\sum (y_i - \bar{y})^2}{n-1} && \left(\text{or } \frac{\sum y_i^2 - n\bar{y}^2}{n-1}\right)
\end{aligned}$$

and the sample mean of the random variables are $\bar{x}$ and $\bar{y}$ respectively.

An Aside: Alternative Formulae for Sample Correlation

Denominators $(n-1)$ on the numerator and denominator of equation (1) cancel, so correlation is often simplified to:

$$\hat{\rho}_{XY} = \frac{SS_{XY}}{\sqrt{SS_{XX}SS_{YY}}}$$

An Aside: Alternative Formulae for Sample Correlation

Denominators $(n-1)$ on the numerator and denominator of equation (1) cancel, so correlation is often simplified to:

$$\hat{\rho}_{XY} = \frac{SS_{XY}}{\sqrt{SS_{XX} SS_{YY}}}$$

where the sum of square (SS) of mean corrected data are:

$$\begin{aligned} SS_{XX} &= \sum (x_i - \bar{x})^2 \\ SS_{YY} &= \sum (y_i - \bar{y})^2 \end{aligned}$$

An Aside: Alternative Formulae for Sample Correlation

Denominators $(n-1)$ on the numerator and denominator of equation (1) cancel, so correlation is often simplified to:

$$\hat{\rho}_{XY} = \frac{SS_{XY}}{\sqrt{SS_{XX} SS_{YY}}}$$

where the sum of square (SS) of mean corrected data are:

$$\begin{aligned} SS_{XX} &= \sum (x_i - \bar{x})^2 \\ SS_{YY} &= \sum (y_i - \bar{y})^2 \end{aligned}$$

and the sum of the mean corrected cross products:

$$SS_{XY} = \sum (x_i - \bar{x})(y_i - \bar{y}).$$

Correlation Terminology

Correlation Terminology

Correlation coefficient $-1 \leq \rho_{XY} \leq 1$ where:

Correlation Terminology

Correlation coefficient $-1 \leq \rho_{XY} \leq 1$ where:

- 0 indicates no association

Correlation Terminology

Correlation coefficient $-1 \leq \rho_{XY} \leq 1$ where:

- 0 indicates no association
- $+1$ and -1 represents **perfect positive and negative correlation**

Correlation Terminology

Correlation coefficient $-1 \leq \rho_{XY} \leq 1$ where:

- 0 indicates no association
- $+1$ and -1 represents **perfect positive and negative correlation**

Correlation Terminology

Correlation coefficient $-1 \leq \rho_{XY} \leq 1$ where:

- 0 indicates no association
- $+1$ and -1 represents **perfect positive and negative correlation**

Commonly used descriptors are:

- $|\rho_{XY}| > 0.7$ **strong correlation**

Correlation Terminology

Correlation coefficient $-1 \leq \rho_{XY} \leq 1$ where:

- 0 indicates no association
- $+1$ and -1 represents **perfect positive and negative correlation**

Commonly used descriptors are:

- $|\rho_{XY}| > 0.7$ **strong correlation**
- $0.3 < |\rho_{XY}| < 0.7$ **moderate correlation**

Correlation Terminology

Correlation coefficient $-1 \leq \rho_{XY} \leq 1$ where:

- 0 indicates no association
- $+1$ and -1 represents **perfect positive and negative correlation**

Commonly used descriptors are:

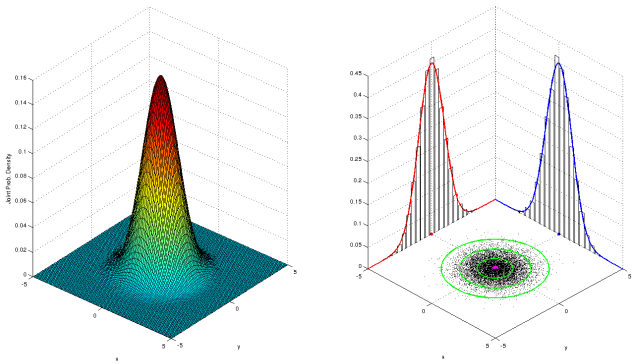
- $|\rho_{XY}| > 0.7$ **strong correlation**
- $0.3 < |\rho_{XY}| < 0.7$ **moderate correlation**
- $|\rho_{XY}| < 0.3$ **weak correlation**

Samples from bivariate Normal (Joint & Marginal) Densities

Simulated samples of bivariate normal ($\mu_1 = \mu_2 = 0$ and $\sigma_1 = \sigma_2 = 1$) of size 1000 with correlation $= 0, 0.3, 0.7, -0.7$)

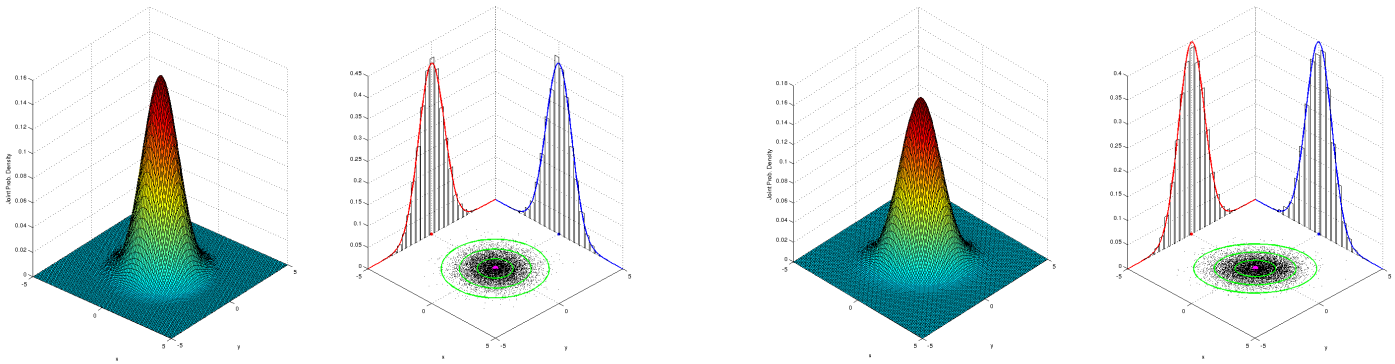
Samples from bivariate Normal (Joint & Marginal) Densities

Simulated samples of bivariate normal ($\mu_1 = \mu_2 = 0$ and $\sigma_1 = \sigma_2 = 1$) of size 1000 with correlation $= 0, 0.3, 0.7, -0.7$)



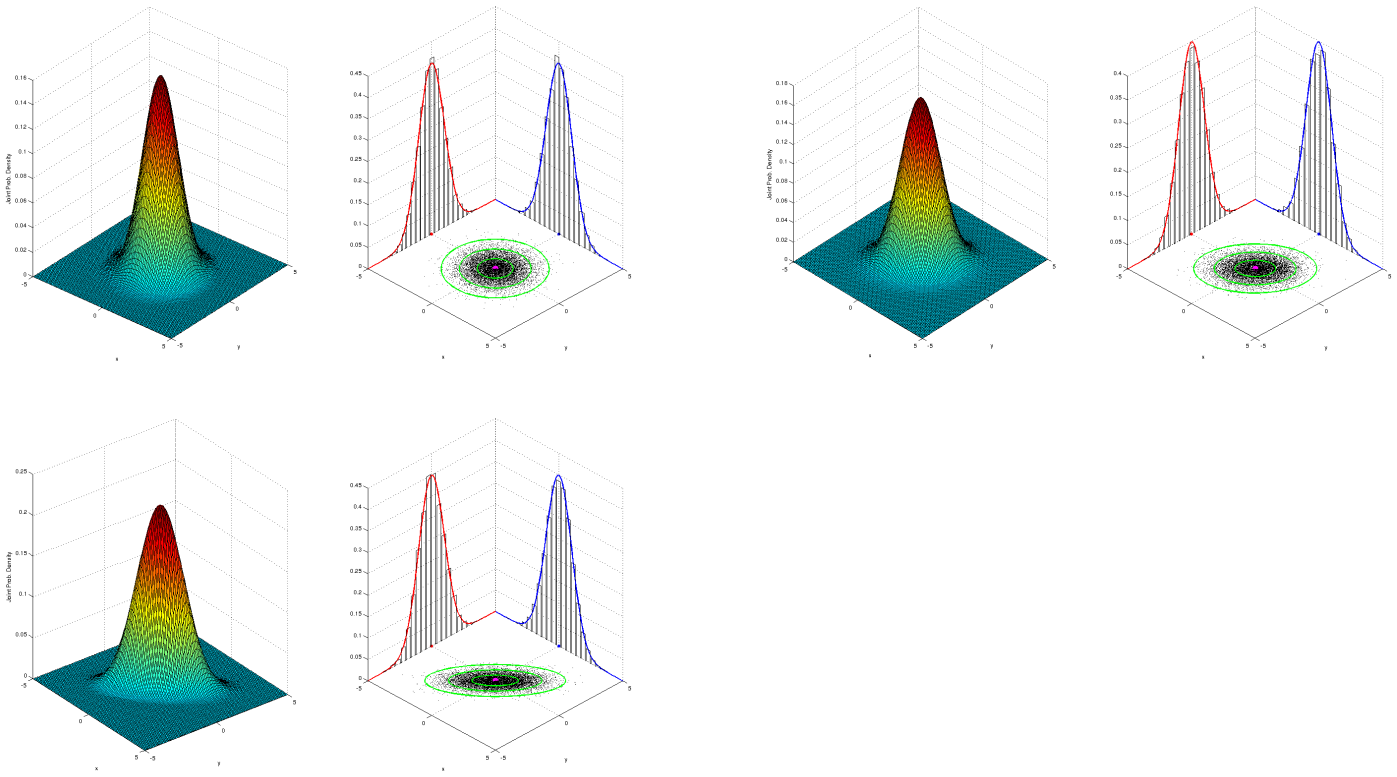
Samples from bivariate Normal (Joint & Marginal) Densities

Simulated samples of bivariate normal ($\mu_1 = \mu_2 = 0$ and $\sigma_1 = \sigma_2 = 1$) of size 1000 with correlation $= 0, 0.3, 0.7, -0.7$)



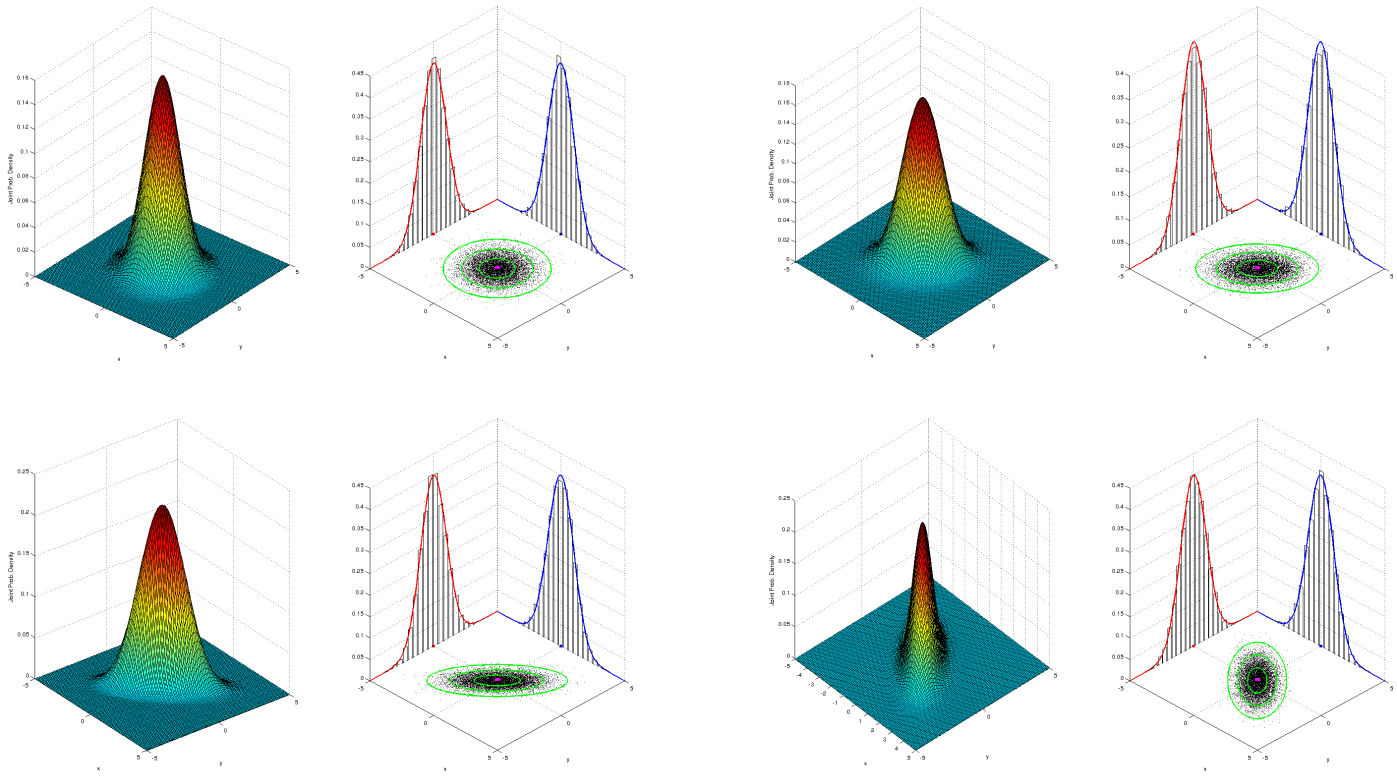
Samples from bivariate Normal (Joint & Marginal) Densities

Simulated samples of bivariate normal ($\mu_1 = \mu_2 = 0$ and $\sigma_1 = \sigma_2 = 1$) of size 1000 with correlation $= 0, 0.3, 0.7, -0.7$)



Samples from bivariate Normal (Joint & Marginal) Densities

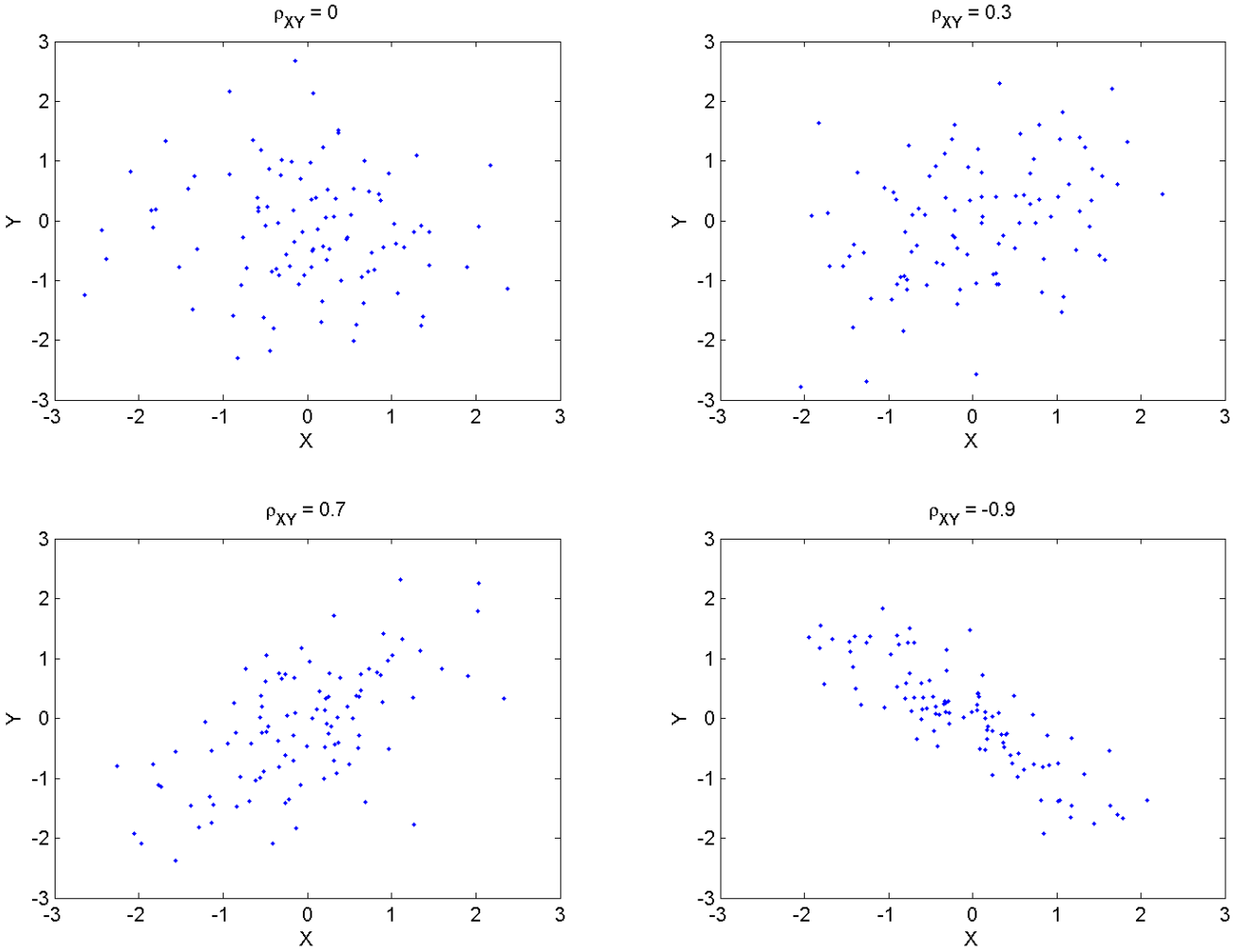
Simulated samples of bivariate normal ($\mu_1 = \mu_2 = 0$ and $\sigma_1 = \sigma_2 = 1$) of size 1000 with correlation $= 0, 0.3, 0.7, -0.7$)



Simulated Data Using `mvnrnd`

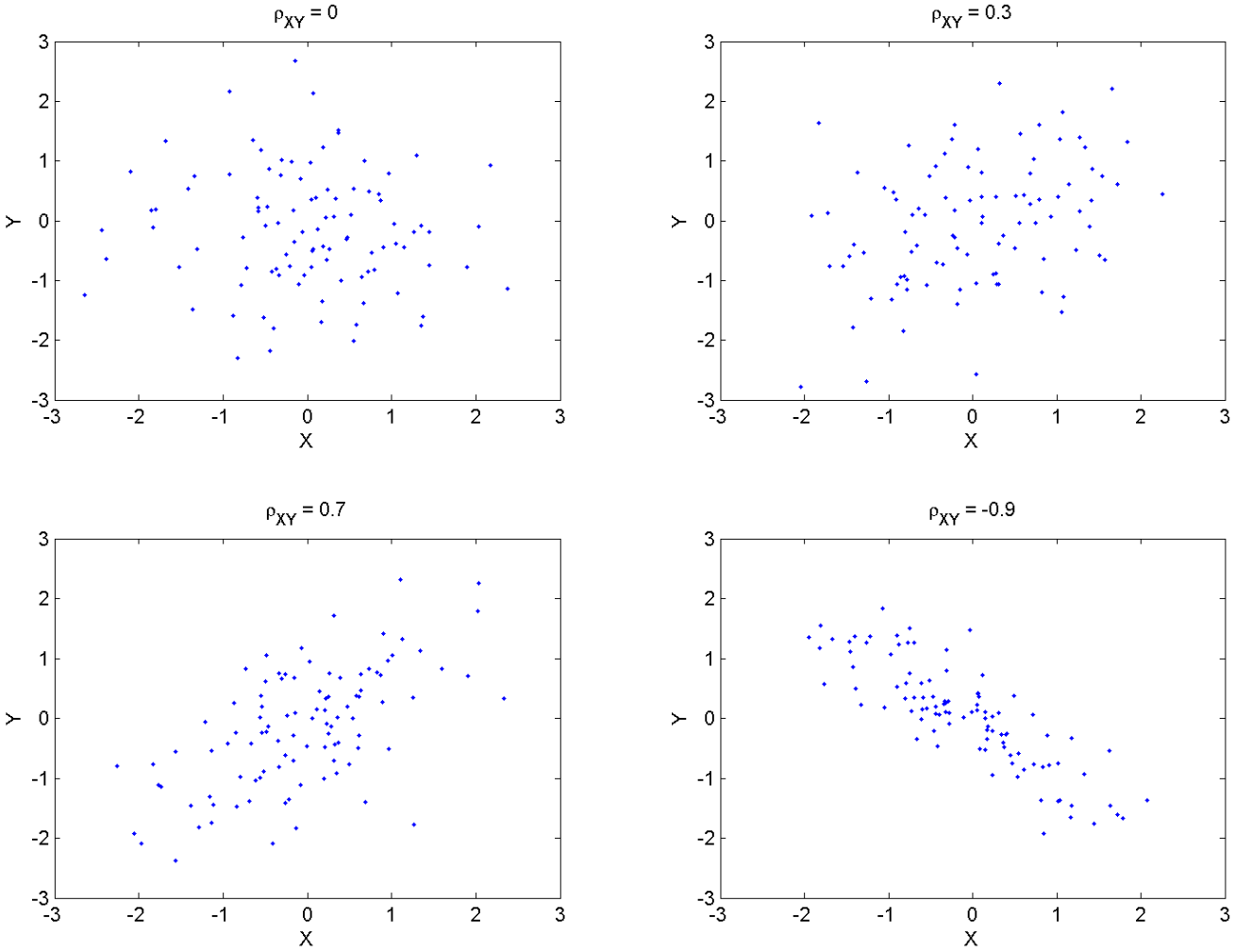
Simulated Data Using `mvnrnd`

Simulated samples of bivariate normal ($\mu_1 = \mu_2 = 0$ and $\sigma_1 = \sigma_2 = 1$) of size 100 (using `mvnrnd`) with different correlations:



Simulated Data Using `mvnrnd`

Simulated samples of bivariate normal ($\mu_1 = \mu_2 = 0$ and $\sigma_1 = \sigma_2 = 1$) of size 100 (using `mvnrnd`) with different correlations:



See `./ mfiles/Plot2DGaussianSamplesAndMarginals.m`, shall we?

Calculating Correlation in MATLAB

Calculating Correlation in MATLAB

Correlation coefficient calculated in MATLAB using:

Calculating Correlation in MATLAB

Correlation coefficient calculated in MATLAB using:

```
>> corr(Velocity,Evaporation) % if separate vectors
```

Calculating Correlation in MATLAB

Correlation coefficient calculated in MATLAB using:

```
>> corr(Velocity,Evaporation) % if separate vectors
>> corr([Velocity Evaporation]) % if matrix with column per variable
```

Calculating Correlation in MATLAB

Correlation coefficient calculated in MATLAB using:

```
>> corr(Velocity,Evaporation) % if separate vectors
>> corr([Velocity Evaporation]) % if matrix with column per variable
```

Calculating Correlation in MATLAB

Correlation coefficient calculated in MATLAB using:

```
>> corr(Velocity,Evaporation) % if separate vectors
>> corr([Velocity Evaporation]) % if matrix with column per variable
```

Sample correlation coefficient for Evaporation data is 0.95.
Bootstrap simulation are obtained from:

```
>> nsim=10000;
>> corrstats=bootstrp(nsim,@corr,Velocity,Evaporation)
```

Calculating Correlation in MATLAB

Correlation coefficient calculated in MATLAB using:

```
>> corr(Velocity,Evaporation) % if separate vectors
>> corr([Velocity Evaporation]) % if matrix with column per variable
```

Sample correlation coefficient for Evaporation data is 0.95. Bootstrap simulation are obtained from:

```
>> nsim=10000;
>> corrstats=bootstrp(nsim,@corr,Velocity,Evaporation)
```

A bootstrap based 95% confidence interval can be obtained from:

```
>> prctile(corrstats,[2.5 97.5]) % using bootstrap correlations
```

or more directly using `bootci` function:

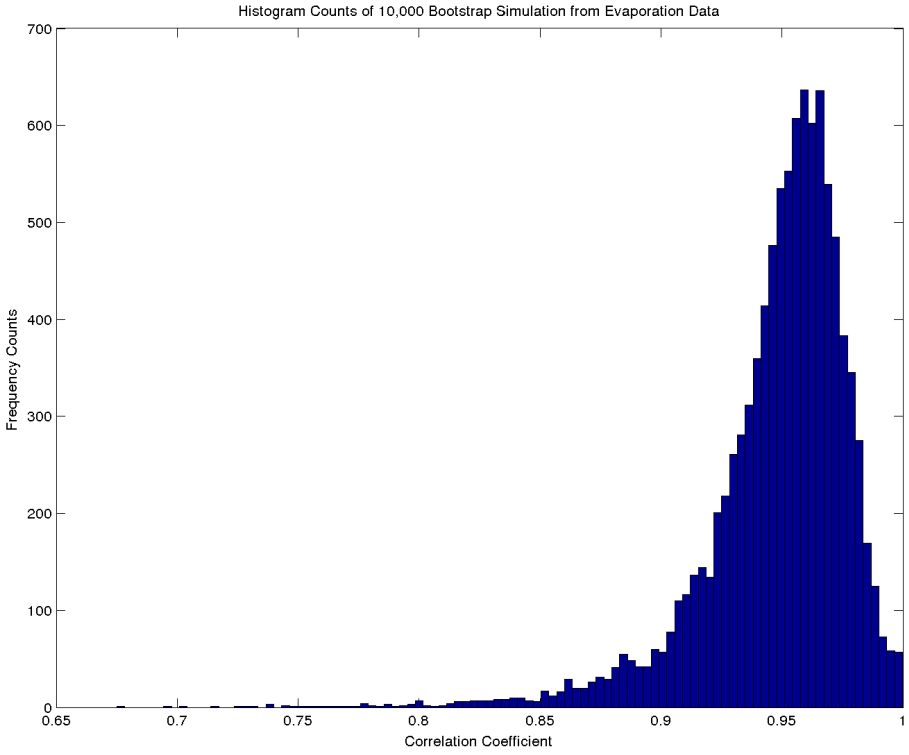
```
>> corrxy = @(x,y) corr(x,y)
>> bootci(nsim,{corrxy,Velocity,Evaporation},'alpha',0.05,'type','per')
```

Note: the '`type`' option specifies basic percentile method and '`alpha`' is the significance level.

Let's DIY `../../Datasets/Evaporation_PoorMansBoot.m`

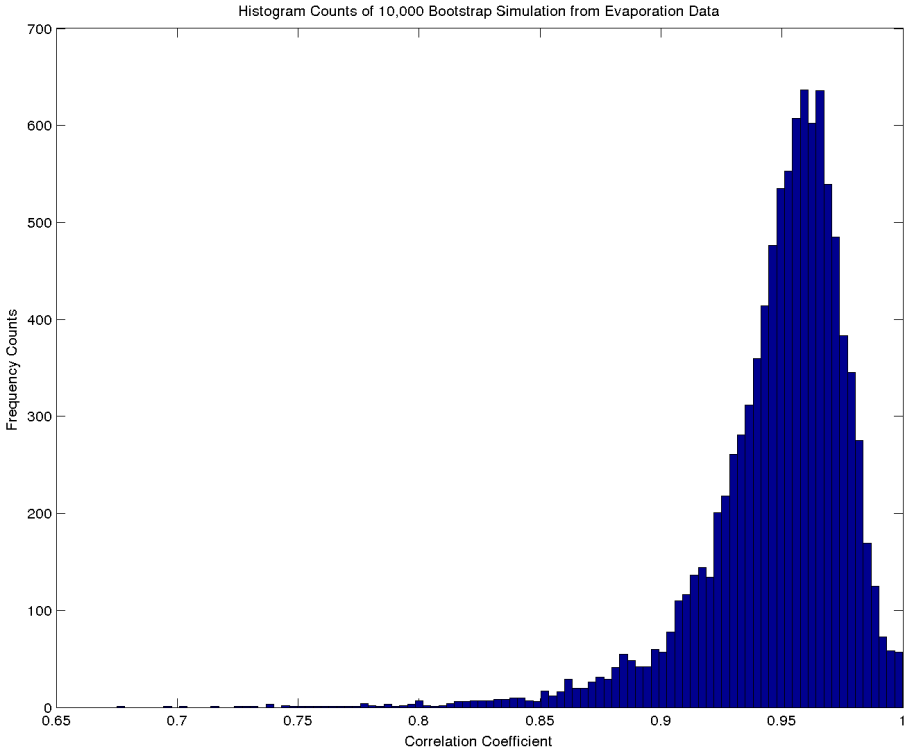
Bootstrapped Correlations

Bootstrap simulations below and estimated 95% confidence interval of $(0.88, 0.99)$ confirm the true correlation between evaporation and air velocity is statistically significantly different from zero



Bootstrapped Correlations

Bootstrap simulations below and estimated 95% confidence interval of $(0.88, 0.99)$ confirm the true correlation between evaporation and air velocity is statistically significantly different from zero

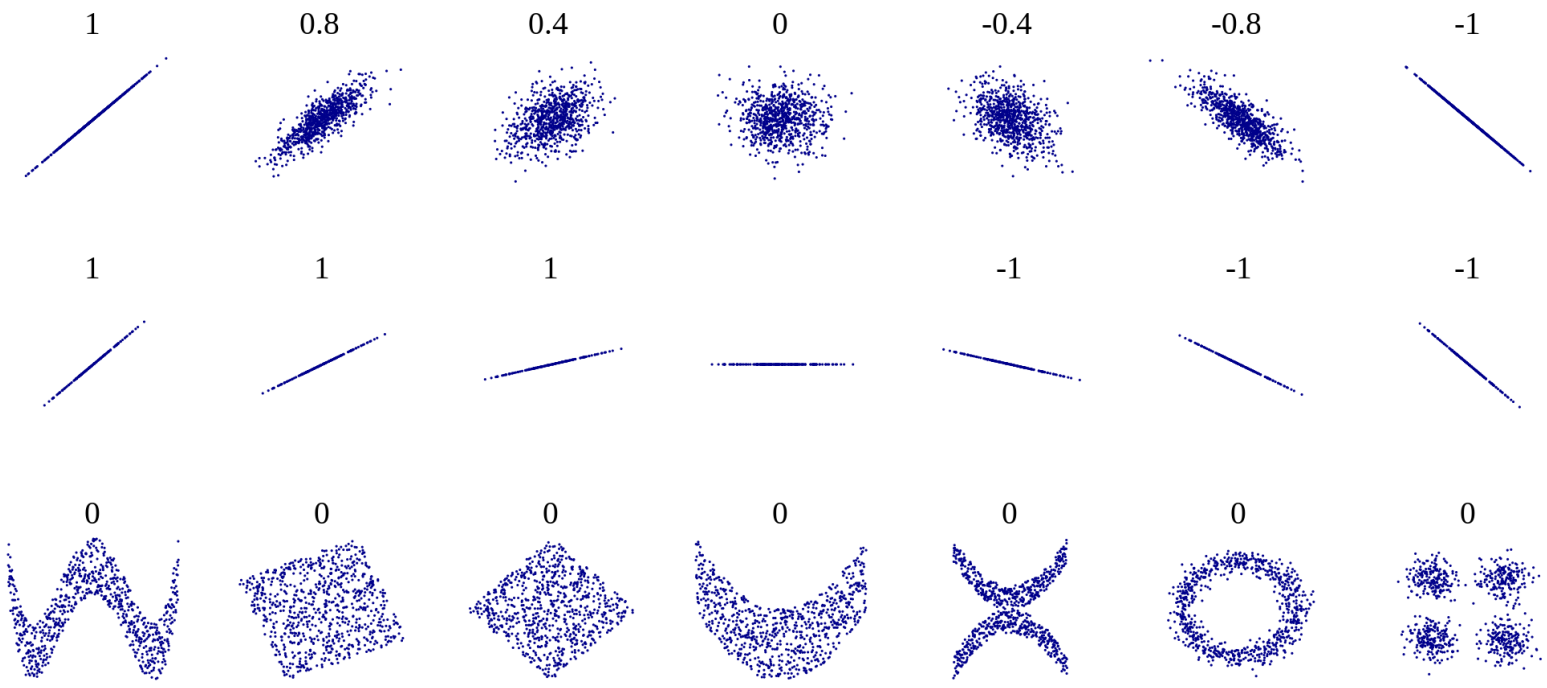


Note: highly skewed sampling distribution (would be symmetric for normal populations with zero correlation). Traditional confidence interval estimation approaches typically use transformations to adjust for this (e.g. see Fisher's Z transform on Wikipedia)

Pitfalls of Correlation

Some pitfalls of correlation:

- We all know "correlation DOES NOT imply causation"
- Independence implies zero correlation, but zero correlation does not imply independence
- In particular, Pearson product moment correlation coefficient only measures linear association



Source: http://en.wikipedia.org/wiki/File:Correlation_examples2.svg

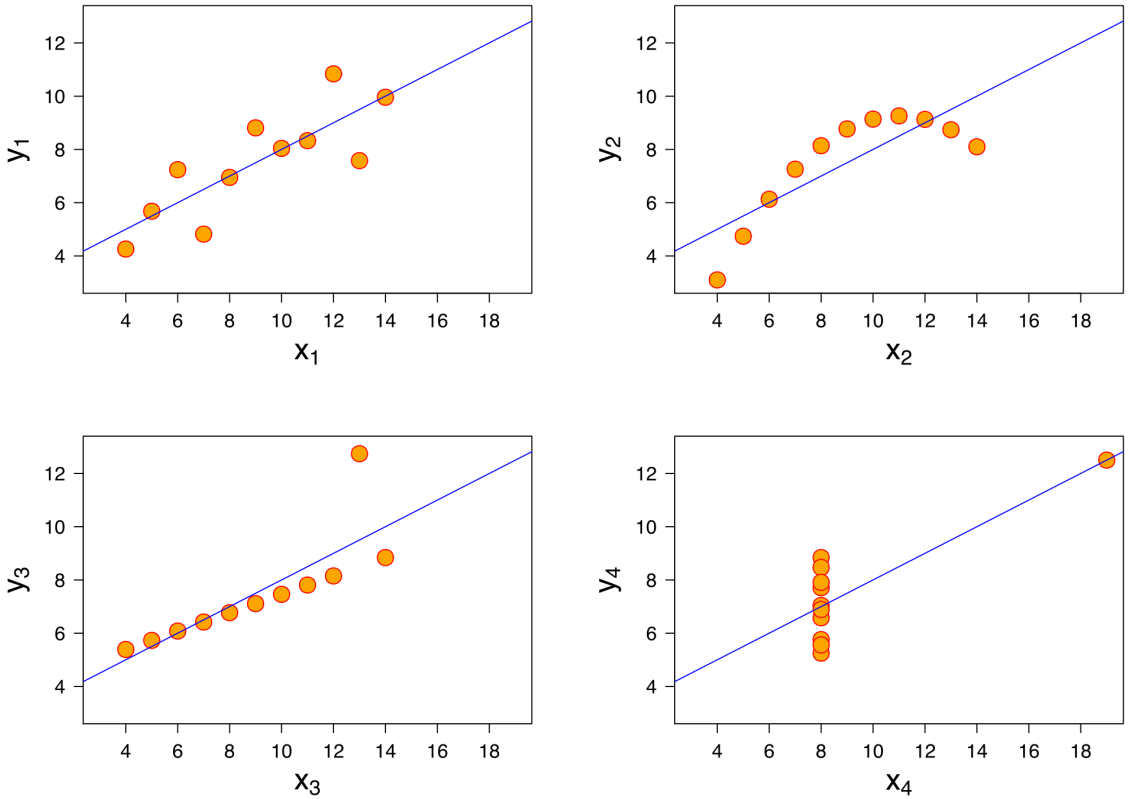
Correlation Is Not Full Story...

- Correlation is only a statistical summary, so **always plot the data**

Correlation Is Not Full Story...

- Correlation is only a statistical summary, so **always plot the data**

Famous example of datasets with same correlation (and many other statistics) is Anscombe's Quartet (see Wikipedia for more details):



Source: `http://en.wikipedia.org/wiki/File:Anscombe's_quartet_3.svg`

Simple Linear Regression Matrix Representation

The simple linear regression equations for the observation pairs can be concisely written using matrix notation:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{2}$$

where:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \quad \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}.$$

- **Design matrix X** has column of ones for intercept term and second column of explanatory variable observations corresponding to elements of response vector **y**

System Of Equations

Each row represents one of an overdetermined system of linear equations:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \beta_0 + \beta_1 x_1 + \epsilon_1 \\ \beta_0 + \beta_1 x_2 + \epsilon_2 \\ \vdots \\ \beta_0 + \beta_1 x_n + \epsilon_n \end{bmatrix}$$

to which an approximately solution is calculated to find the regression coefficients β_0 and β_1.

System Of Equations

Each row represents one of an overdetermined system of linear equations:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \beta_0 + \beta_1 x_1 + \epsilon_1 \\ \beta_0 + \beta_1 x_2 + \epsilon_2 \\ \vdots \\ \beta_0 + \beta_1 x_n + \epsilon_n \end{bmatrix}$$

to which an approximately solution is calculated to find the regression coefficients β_0 and β_1.

- The estimated regression coefficient vector is $\hat{\boldsymbol{\beta}}$

System Of Equations

Each row represents one of an overdetermined system of linear equations:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \beta_0 + \beta_1 x_1 + \epsilon_1 \\ \beta_0 + \beta_1 x_2 + \epsilon_2 \\ \vdots \\ \beta_0 + \beta_1 x_n + \epsilon_n \end{bmatrix}$$

to which an approximately solution is calculated to find the regression coefficients β_0 and β_1.

- The estimated regression coefficient vector is $\hat{\boldsymbol{\beta}}$
- Error vector $\boldsymbol{\epsilon}$ will also be estimated, giving the **residuals** vector $\hat{\boldsymbol{\epsilon}}$

Ordinary Least Squares Estimation

- Ordinary least squares (OLS) is a commonly used criteria for fitting regression models

Ordinary Least Squares Estimation

- Ordinary least squares (OLS) is a commonly used criteria for fitting regression models
- Find $\boldsymbol{\beta}$ to minimise the **sum of the square of the errors**:

$$\begin{aligned} S &= \sum_{i=1}^{n} \epsilon_i^2 \\ &= \boldsymbol{\epsilon}'\boldsymbol{\epsilon} \\ &= (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \end{aligned} \tag{3}$$

Ordinary Least Squares Estimation

- Ordinary least squares (OLS) is a commonly used criteria for fitting regression models
- Find $\boldsymbol{\beta}$ to minimise the **sum of the square of the errors**:

$$\begin{aligned} S &= \sum_{i=1}^{n} \epsilon_i^2 \\ &= \boldsymbol{\epsilon}'\boldsymbol{\epsilon} \\ &= (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \end{aligned} \tag{3}$$

- which is a quadratic in the unknown vector $\boldsymbol{\beta}$

Ordinary Least Squares Estimation

- Ordinary least squares (OLS) is a commonly used criteria for fitting regression models
- Find $\boldsymbol{\beta}$ to minimise the **sum of the square of the errors**:

$$\begin{aligned} S &= \sum_{i=1}^{n} \epsilon_i^2 \\ &= \boldsymbol{\epsilon}'\boldsymbol{\epsilon} \\ &= (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \end{aligned} \tag{3}$$

- which is a quadratic in the unknown vector $\boldsymbol{\beta}$
- Therefore, minimum is found by setting partial derivative of S (w.r.t $\boldsymbol{\beta}$) equal to zero

OLS Solution

The partial derivative of S with respect to $\boldsymbol{\beta}$ is:

$$\frac{\partial S}{\partial \boldsymbol{\beta}} = -2\mathbf{X}'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}),$$

OLS Solution

The partial derivative of S with respect to $\boldsymbol{\beta}$ is:

$$\frac{\partial S}{\partial \boldsymbol{\beta}} = -2\mathbf{X}'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}),$$

which when set equal to zero gives:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}. \tag{4}$$

OLS Solution

The partial derivative of S with respect to $\boldsymbol{\beta}$ is:

$$\frac{\partial S}{\partial \boldsymbol{\beta}} = -2\mathbf{X}'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}),$$

which when set equal to zero gives:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}. \tag{4}$$

Note that:

$$\begin{aligned}
\mathbf{X}'\mathbf{X} &= \begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \\
(\mathbf{X}'\mathbf{X})^{-1} &= \frac{1}{n\sum(x_i - \bar{x})^2}\begin{bmatrix} \sum x_i^2 & -\sum x_i \\ -\sum x_i & n \end{bmatrix} \\
\mathbf{X}'\mathbf{y} &= \begin{bmatrix} \sum y_i \\ \sum x_i y_i \end{bmatrix}
\end{aligned}$$

.

Simple Linear Regression Assumptions

The OLS criterion is intuitively sensible and the same solution (5) can be derived in many different ways under various assumptions. The assumptions underlying OLS are:

Simple Linear Regression Assumptions

The OLS criterion is intuitively sensible and the same solution (5) can be derived in many different ways under various assumptions. The assumptions underlying OLS are:

- weak exogeneity - **X** values are fixed, or can be treated as fixed (i.e. conditioned upon), otherwise an "errors-in-variables" model is needed. This also implies that errors are uncorrelated with explanatory variable;

Simple Linear Regression Assumptions

The OLS criterion is intuitively sensible and the same solution (5) can be derived in many different ways under various assumptions. The assumptions underlying OLS are:

- weak exogeneity - $\mathbf{X}$ values are fixed, or can be treated as fixed (i.e. conditioned upon), otherwise an "errors-in-variables" model is needed. This also implies that errors are uncorrelated with explanatory variable;
- linearity - response is linear function of explanatory variables;

Simple Linear Regression Assumptions

The OLS criterion is intuitively sensible and the same solution (5) can be derived in many different ways under various assumptions. The assumptions underlying OLS are:

- weak exogeneity - **X** values are fixed, or can be treated as fixed (i.e. conditioned upon), otherwise an "errors-in-variables" model is needed. This also implies that errors are uncorrelated with explanatory variable;
- linearity - response is linear function of explanatory variables;
- homoscedastic - errors have a constant variance (essentially all errors are equally important) so $Var(\epsilon_i) = \sigma^2$; and

Simple Linear Regression Assumptions

The OLS criterion is intuitively sensible and the same solution (5) can be derived in many different ways under various assumptions. The assumptions underlying OLS are:

- weak exogeneity - **X** values are fixed, or can be treated as fixed (i.e. conditioned upon), otherwise an "errors-in-variables" model is needed. This also implies that errors are uncorrelated with explanatory variable;
- linearity - response is linear function of explanatory variables;
- homoscedastic - errors have a constant variance (essentially all errors are equally important) so $Var(\epsilon_i) = \sigma^2$; and
- uncorrelated - errors are uncorrelated with each other, so $Corr(\epsilon_i, \epsilon_j) = 0$ for $\{i, j = 1, \ldots, n : i \neq j\}$.

Simple Linear Regression Assumptions

The OLS criterion is intuitively sensible and the same solution (5) can be derived in many different ways under various assumptions. The assumptions underlying OLS are:

- weak exogeneity - **X** values are fixed, or can be treated as fixed (i.e. conditioned upon), otherwise an "errors-in-variables" model is needed. This also implies that errors are uncorrelated with explanatory variable;
- linearity - response is linear function of explanatory variables;
- homoscedastic - errors have a constant variance (essentially all errors are equally important) so $Var(\epsilon_i) = \sigma^2$; and
- uncorrelated - errors are uncorrelated with each other, so $Corr(\epsilon_i, \epsilon_j) = 0$ for $\{i, j = 1, \ldots, n : i \neq j\}$.

Simple Linear Regression Assumptions

The OLS criterion is intuitively sensible and the same solution (5) can be derived in many different ways under various assumptions. The assumptions underlying OLS are:

- weak exogeneity - $\mathbf{X}$ values are fixed, or can be treated as fixed (i.e. conditioned upon), otherwise an "errors-in-variables" model is needed. This also implies that errors are uncorrelated with explanatory variable;
- linearity - response is linear function of explanatory variables;
- homoscedastic - errors have a constant variance (essentially all errors are equally important) so $Var(\epsilon_i) = \sigma^2$; and
- uncorrelated - errors are uncorrelated with each other, so $Corr(\epsilon_i, \epsilon_j) = 0$ for $\{i, j = 1, \ldots, n : i \neq j\}$.

Or in matrix terms: $E(\boldsymbol{\epsilon}) = \mathbf{0}$, $Var(\boldsymbol{\epsilon}) = \sigma^2\mathbf{I}$, and $Corr(\mathbf{X}, \boldsymbol{\epsilon}) = \mathbf{0}$.

Simple Linear Regression Assumptions

The OLS criterion is intuitively sensible and the same solution (5) can be derived in many different ways under various assumptions. The assumptions underlying OLS are:

- weak exogeneity - $\mathbf{X}$ values are fixed, or can be treated as fixed (i.e. conditioned upon), otherwise an "errors-in-variables" model is needed. This also implies that errors are uncorrelated with explanatory variable;
- linearity - response is linear function of explanatory variables;
- homoscedastic - errors have a constant variance (essentially all errors are equally important) so $Var(\epsilon_i) = \sigma^2$; and
- uncorrelated - errors are uncorrelated with each other, so $Corr(\epsilon_i, \epsilon_j) = 0$ for $\{i, j = 1, \ldots, n : i \neq j\}$.

Or in matrix terms: $E(\boldsymbol{\epsilon}) = \mathbf{0}$, $Var(\boldsymbol{\epsilon}) = \sigma^2\mathbf{I}$, and $Corr(\mathbf{X}, \boldsymbol{\epsilon}) = \mathbf{0}$.

It is also commonly assumed that the errors are normally distributed, but we'll get to this later.

Fitted Regression Model

The fitted regression model is then:

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} \tag{5}$$

The estimated prediction error (**residuals**) can be written:

$$\begin{aligned} \hat{\epsilon} &= \mathbf{y} - \hat{\mathbf{y}} \\ &= \mathbf{y} - \mathbf{X}\hat{\beta} \end{aligned} \tag{6}$$

An unbiased estimate of the error variance is given by:

$$\begin{aligned} \hat{\sigma}^2 &= \frac{\sum(y_i - \hat{y}_i)^2}{n-2} \\ &= \frac{\hat{\epsilon}'\hat{\epsilon}}{n-2}. \end{aligned} \tag{7}$$

Vector Geometry of Least Squares

There is a nice geometric interpretation of the OLS solution:

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \mathbf{H}\mathbf{y}, \tag{8}$$

where $\mathbf{H}$ is called the hat or **projection matrix** which has the following properties:

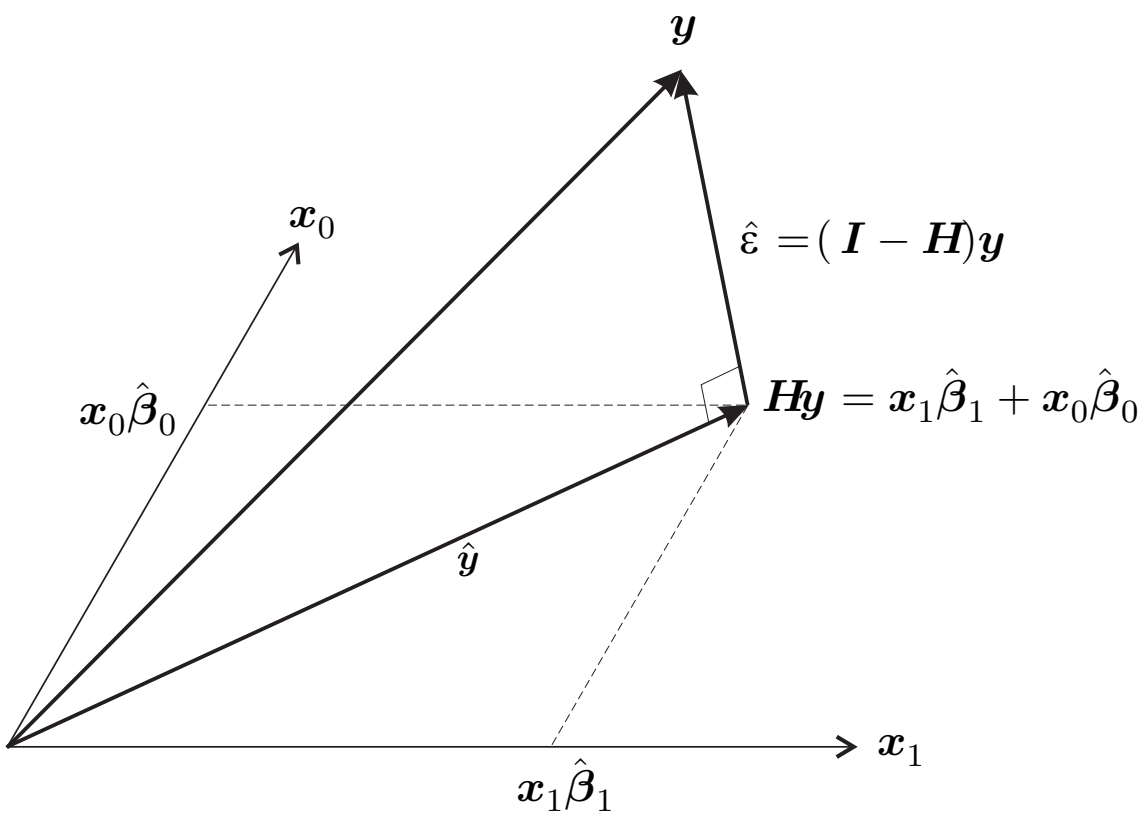
- Symmetric: $\mathbf{H}' = \mathbf{H}$
- Idempotent: $\mathbf{H}^2 = \mathbf{H}$.

The prediction error can also be written:

$$\begin{aligned} \hat{\epsilon} &= \mathbf{y} - \hat{\mathbf{y}} \\ &= \mathbf{y} - \mathbf{X}\hat{\beta} \\ &= \mathbf{y} - \mathbf{H}\mathbf{y} \\ &= (\mathbf{I} - \mathbf{H})\mathbf{y}. \end{aligned} \tag{9}$$

Vector Geometry of Least Squares Diagram

The prediction vector is then simply an orthogonal projection of the original response vector onto the subspace spanned by the explanatory variables, *span*(**X**), and the residuals are the leftovers which are perpendicular to this subspace:



Example: Evaporation Model Fitting in MATLAB

The design matrix **X** must be created first:

```
>> X=[ones(length(Velocity),1) Velocity];
```

There are various ways to fit regression model in MATLAB, with an extensive set of options available using the regress function:

```
>> [B,BINT,R,RINT,STATS] = regress(Evaporation,X,0.05)
```

But we will explain all the results from this function later.

First lets see the how to fit the model and obtain the previous results by doing the explicit calculations manually in MATLAB.

Example: Evaporation Model Fitting (Manually) in MATLAB

The simplest inbuilt option to solve the regression equations for $\hat{\boldsymbol{\beta}}$ is to use the backslash operator (left matrix divide):

```
>> B = X\Evaporation
```

which gives estimated regression coefficient vector $\hat{\boldsymbol{\beta}}$:

```
>> X\Evaporation

ans =

    0.0692
    0.0038
```

Calculating the estimated coefficient vector $\hat{\boldsymbol{\beta}}$ explicitly gives the same result:

```
>> B = inv(X'*X) * X'*Evaporation
```

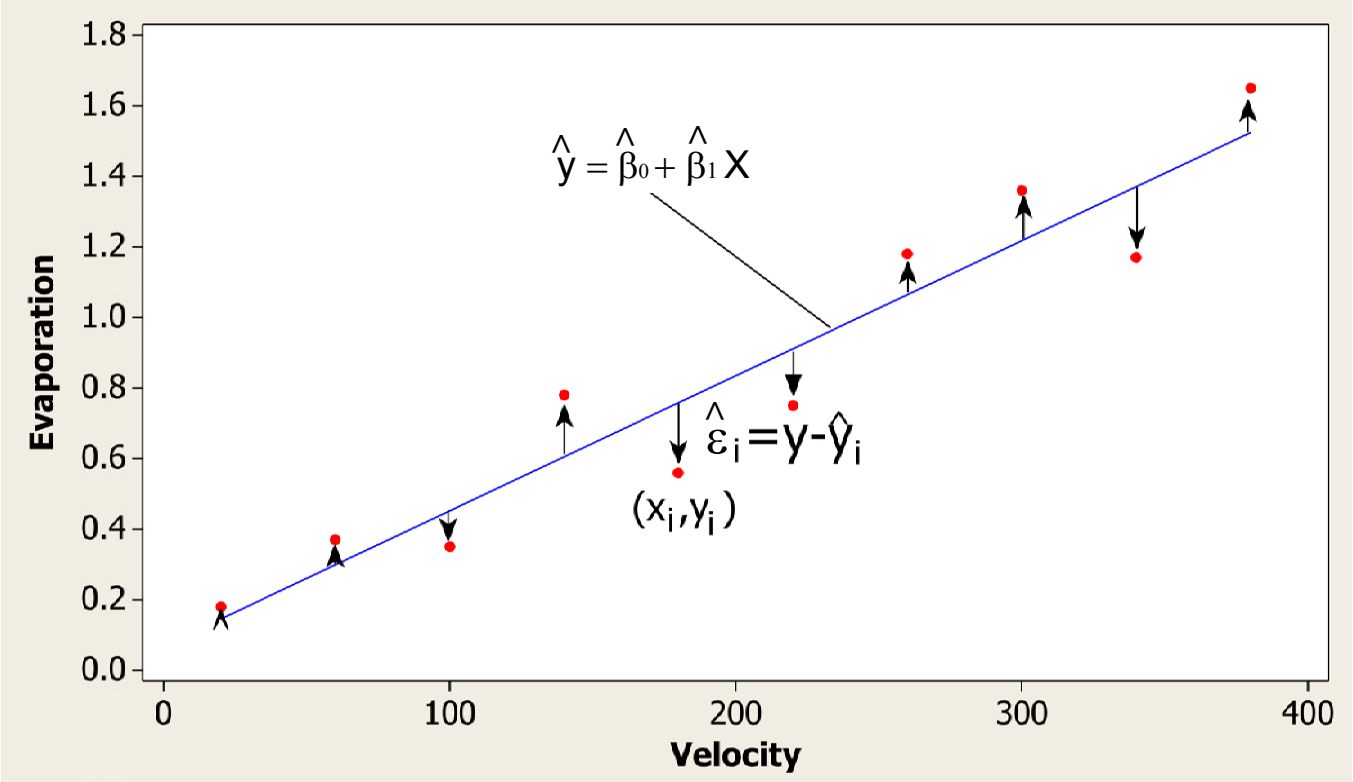
Hint: the backslash operator is very numerically stable, so is generally more reliable than using the latter explicit solution.

Example: Evaporation Model Fit

The OLS estimate of the regression is then:

- `Evaporation = 0.0692 + 0.0038 Velocity`

which is plotted below including marking on the residuals:



- A subjective examination indicates the least square line seems to provide good fit to the observed response.

Example: Evaporation Model Predictions (Manually) in MATLAB

The estimated coefficient can then be used to calculate predictions and residuals:

```
>> yhat = X*B
>> resids = Evaporation - yhat
```

these are the same as vector `R` from the `regress` function above.

The sample standard deviation of the errors is then:

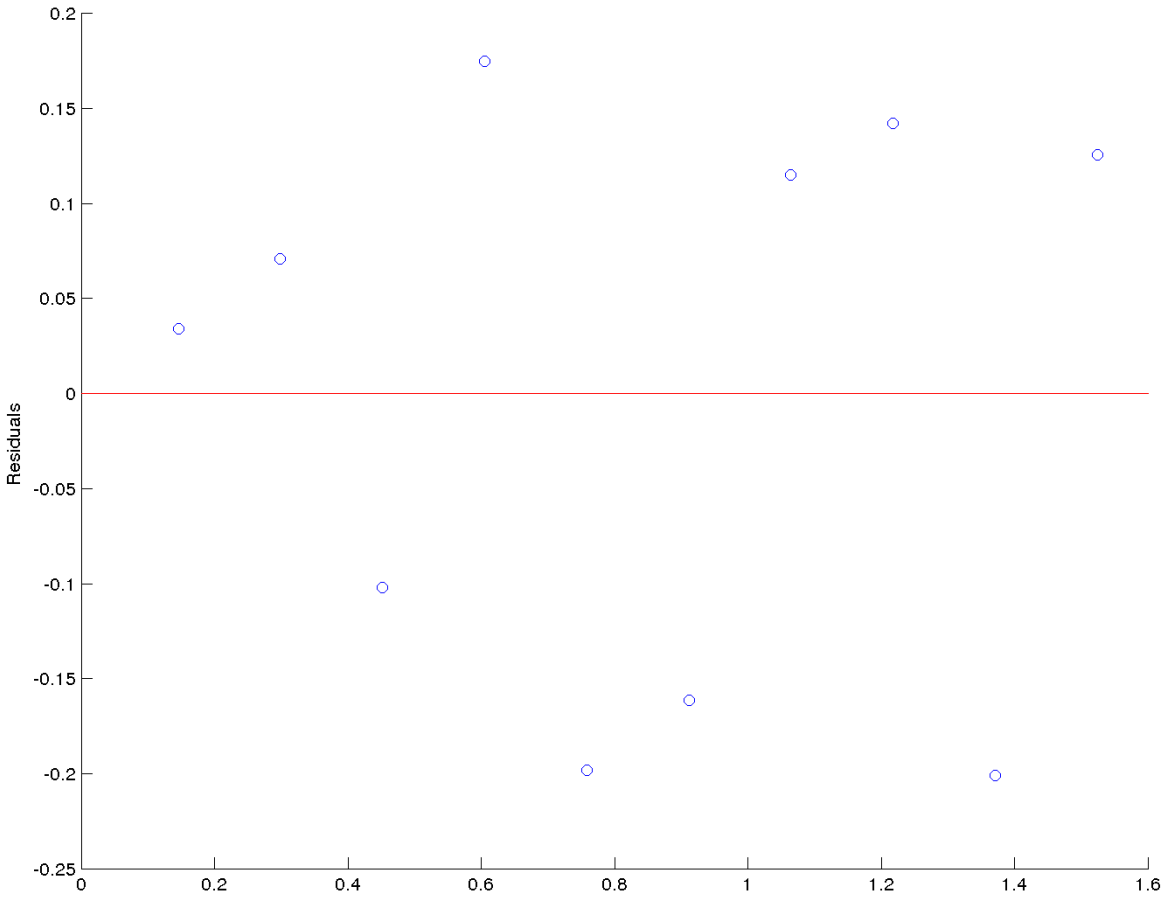
```
>> sigma2 = resids'*resids / (length(resids)-2)
```

which is automatically provided as the fourth element in the `STATS(4)` component of `regress` function above.

Example: Evaporation Model Fit Diagnostic in MATLAB

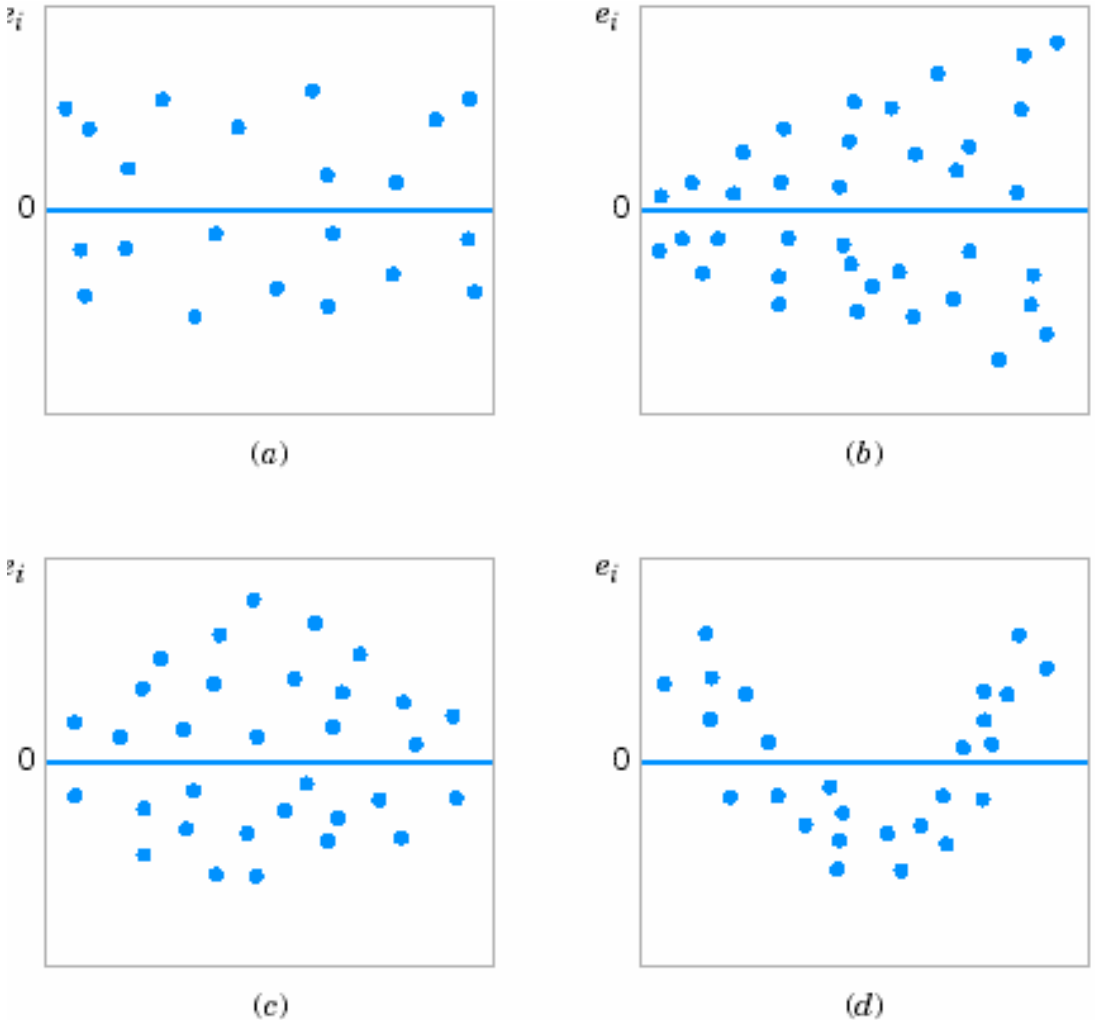
A common diagnostic to assess the model fit (and in particular the assumptions) is to plot residuals against the predicted values:

```
>> evapstats=regstats(Evaporation,Velocity,'linear',{'yhat','r'});
>> clf;
>> scatter(evapstats.yhat,evapstats.r)
>> hold on;
>> plot([0 1.6],[0 0],'r-') % add horizontal line at zero
>> xlabel('Fitted Values');ylabel('Residuals');
```



Regression Diagnostic - What to Look For?

random with constant variance **increasing variance**



changing variance **non-zero mean over X**

Assessing Model Performance

- Variability (in terms of sum of squares around mean) in the response can be decomposed into 2 components:

$$\begin{aligned} \sum(y_i - \bar{y})^2 &= \sum(\hat{y}_i - \bar{y})^2 + \sum(y_i - \hat{y}_i)^2 \\ \text{Total} &= \text{Regression} + \text{Residual.} \end{aligned}$$

- A commonly used measure of performance is the **coefficient of determination** R^2
- Percentage of the total response sum of squares explained by the model:

$$\begin{aligned} R^2 &= \frac{\text{Explained (Regression) variation}}{\text{Total Variation}} \qquad (10) \\ &= \frac{\sum(\hat{y}_i - \bar{y})^2}{S_{yy}} \qquad (\text{or } \hat{\rho}^2_{xy}) \\ &= 1 - \frac{\sum(y_i - \hat{y}_i)^2}{S_{yy}} \end{aligned}$$

Example: Evaporation Model Fit Diagnostic in MATLAB

It is easy to calculate R^2 directly:

```
>> R2 = 1 - (resids'*resids /
   sum((Evaporation-mean(Evaporation)).^2))
```

But it also provided as the first part of the `STATS(1)` output from the `regress` function:

```
>> [B,BINT,R,RINT,STATS] = regress(Evaporation,X,0.05)
>> STATS(1)
```

Or you could use the `regstats` function above.

Resampling Methods for Simple Linear Regression

The two types of explanatory variables (fixed or random) impacts on the resampling scheme to explore the properties of the model:

- random X (called **observation resampling**) - resample observation pairs as with correlation; or
- fixed X (called **residual resampling**) - resample residuals and reapply fitted model:
  - fit model and compute residuals
  - generate bootstrap sample residual vector $\tilde{\epsilon}$
  - obtain bootstrap simulations of response by applying fitted model: $\tilde{\mathbf{y}} = \hat{y} + \tilde{\epsilon}$

Comments on Bootstrap for Regression

Obtain the bootstrap simulated response vectors $\tilde{\mathbf{y}}_1, \ldots, \tilde{\mathbf{y}}_{N_{BOOT}}$ and corresponding explanatory variable matrices $\tilde{\mathbf{X}}_1, \ldots, \tilde{\mathbf{X}}_{N_{BOOT}}$ (either resampled under "observation resampling" or same X for "residual resampling"). For each one:

- refit the linear regression model;
- calculate the statistic of interest (e.g. coefficients, predictions, leverages).

Thus the bootstrap simulations provide a "sampling distribution" for any statistic of interest.

Generally, "observation resampling" approach leads to wider confidence intervals (i.e. indicating more uncertainty) than residual resampling, as the latter ignores uncertainty due to explanatory variable levels and relies on the original model fit.

Example: Evaporation Model Bootstrapping in MATLAB

The evaporation example has fixed X values for the explanatory variables as it was a controlled experiment. So therefore, "residual resampling" is required.

The bootstrap regression coefficients (and corresponding estimated 95% confidence interval) can be calculated in MATLAB using

```
>> bootbetas = bootstrp(nsim, @(bootr)regress(yhat+bootr,X), resids);
>> prctile(bootbetas,[2.5 97.5],1)
```

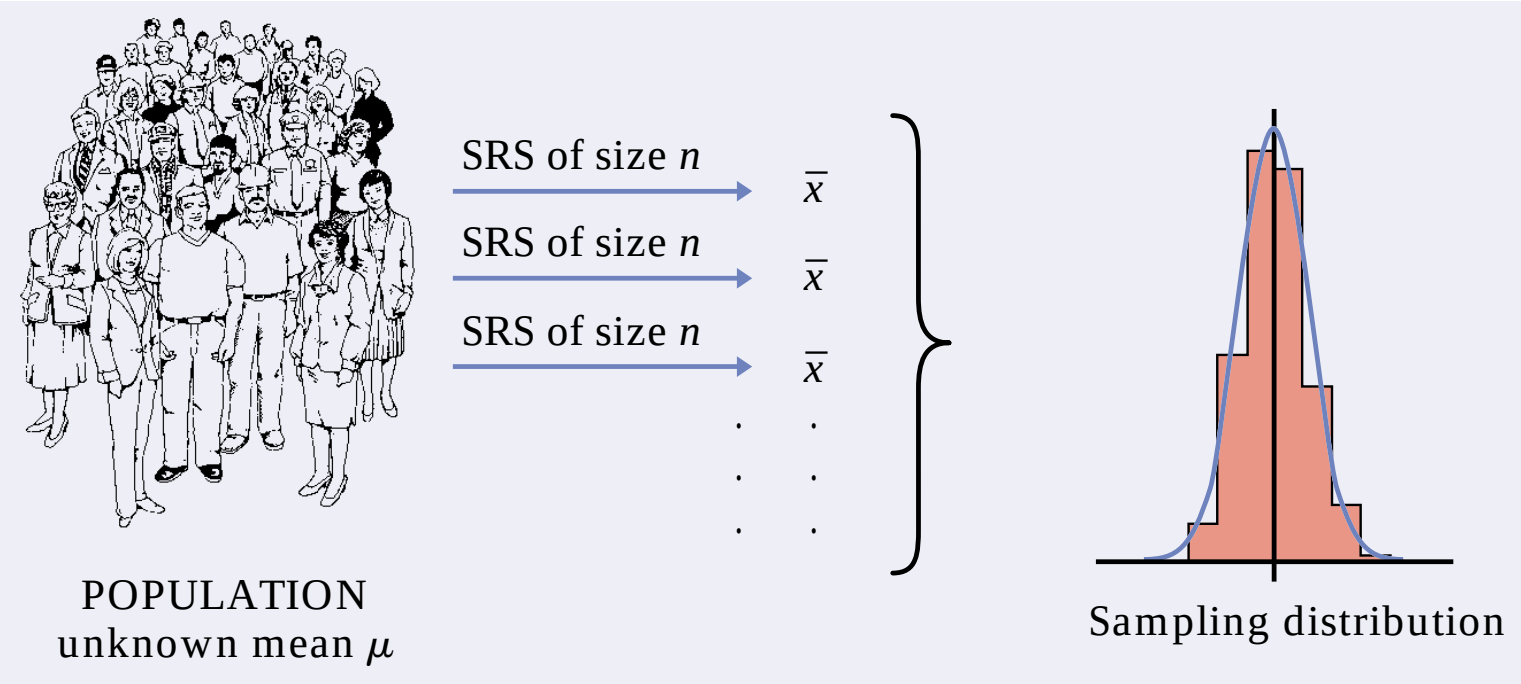
which gives a CI for β_1 of (0.0031, 0.0046) which is a little thinner than the (0.0028, 0.0048) given by BINT result from the regress function above.

If "observation resampling" was appropriate then the following MATLAB code will do this:

```
>> bootbetas = bootstrp(nsim,@(y,x) regress(y,x),Evaporation,X);
>> prctile(bootbetas,[2.5 97.5],1)
```

Revision: Idea Underlying Bootstrapping

- Unknown population parameter (some feature or characteristic of interest)
- Estimate using sample statistic
- SRS - **S**imple **R**andom **S**ampling



Source: Moore & McCabe. Introduction to Practice of Statistics. Chapter 14 on "Bootstrap Methods and Permutation Tests" freely available from `http://bcs.whfreeman.com/ips5e/content/cat_080/pdf/moore14.pdf`

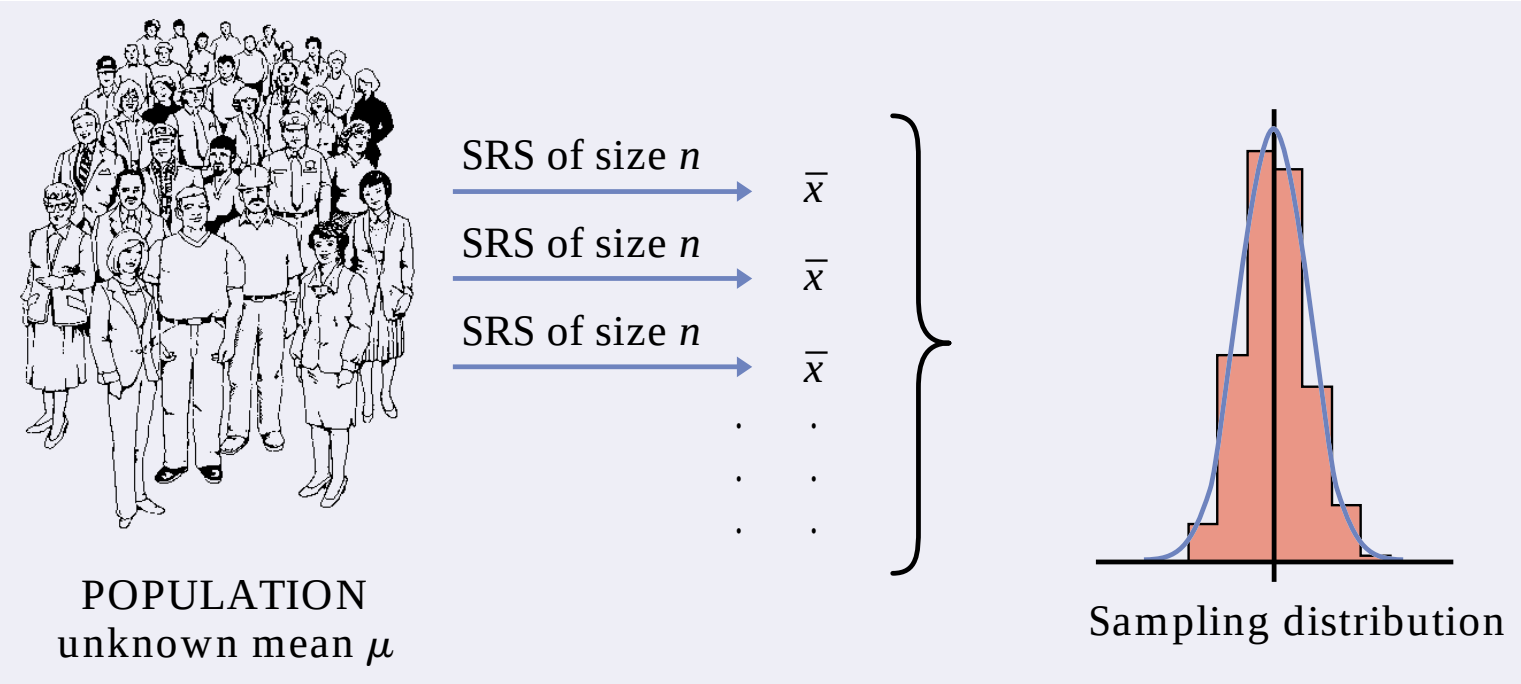
Idea Underlying Sample Estimators

- How reliable is that estimator?
- Some key properties of estimators[1]:
  - **bias** (*accuracy*) - how far is "average" of estimator from the true parameter;
  - **variance** (*precision*) - variability of estimator across samples;
  - **consistency** - does estimator converge (in probability) to true parameter as sample size increases;
  - **efficiency** - is this the best estimator, is there one that makes better use of data
- **Bootstrap simulations** can be used in exploring these properties of estimators
- On this course, *bootstrapping is used to explore the variance (precision) of the estimator*, i.e. how much variability do we get in samples of that sample size?

[1] See Wikipedia page on "Estimator" for formal definitions

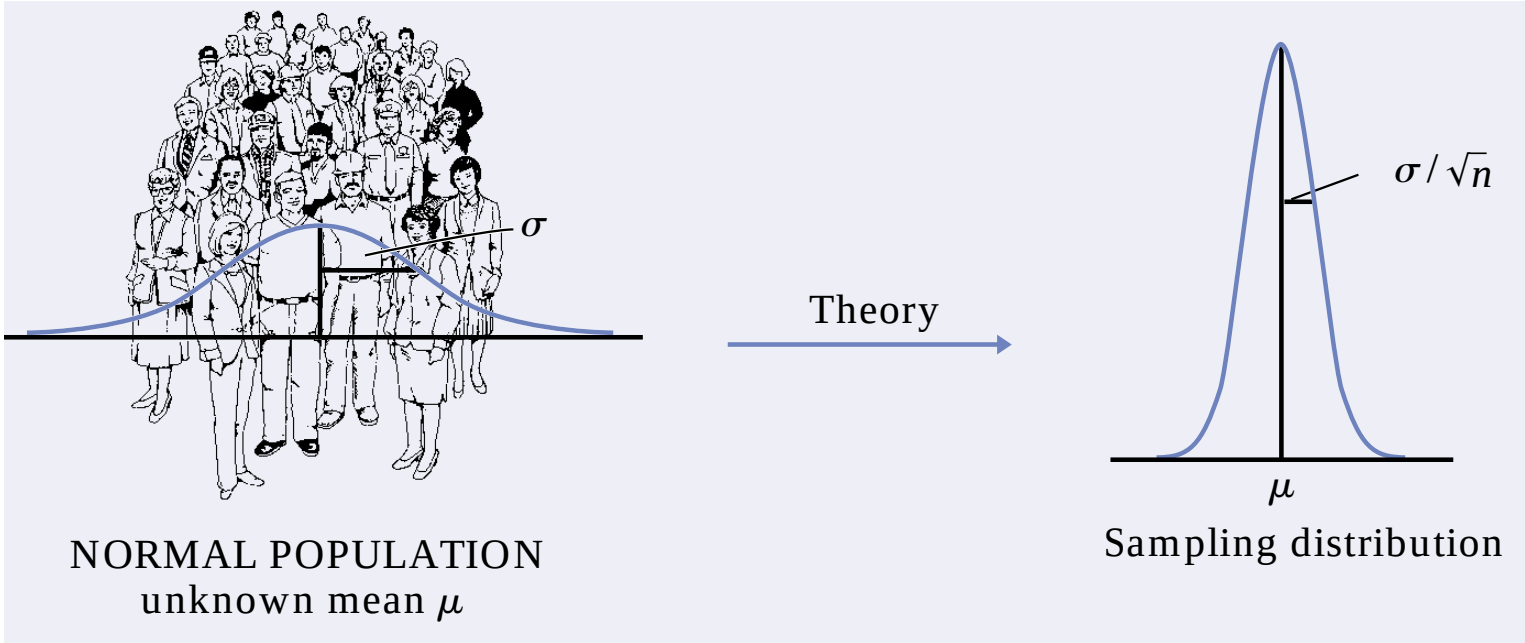
Revision: Idea Underlying Bootstrapping

- In general, the **sampling distribution** for the statistic can be used to explore the variability (uncertainty) due to the sampling process itself
- i.e. to summarise the information content in samples of that sample size



What sampling distribution?

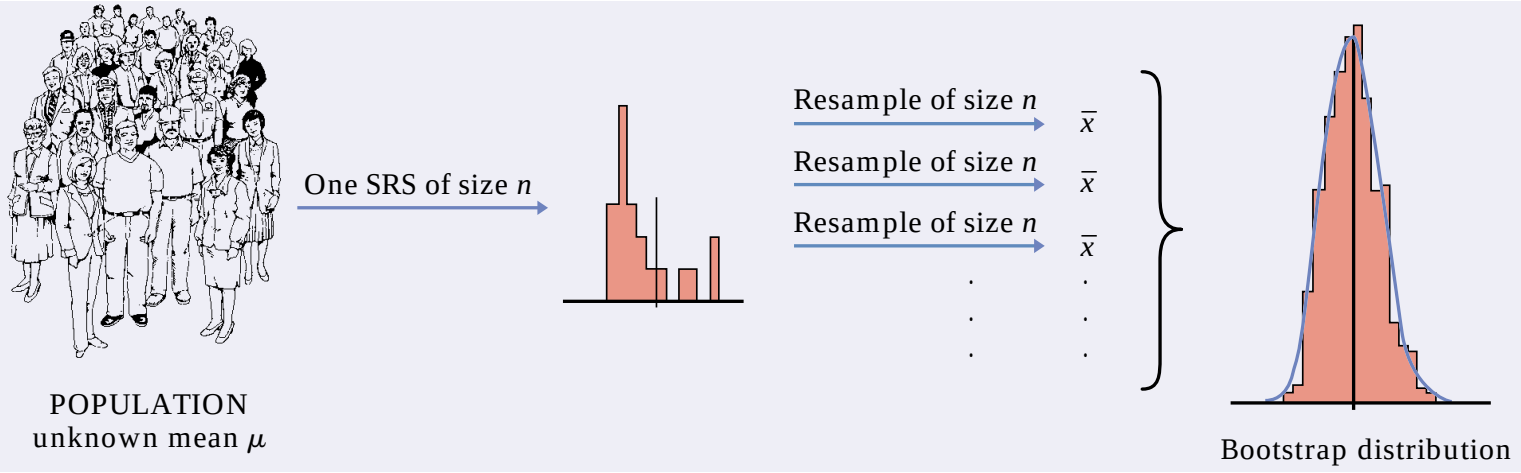
- If we know the true population distribution then, generally, we can determine the theoretical sampling distribution



- But in most real world applications we will not know the population distribution. So what can we do?
  - Traditionally, statisticians used asymptotic models for sampling distributions under various assumptions about population
- Key problem: results may be invalid if assumptions are incorrect (Remember: **all models are wrong!**)

So Just Bootstrap It!

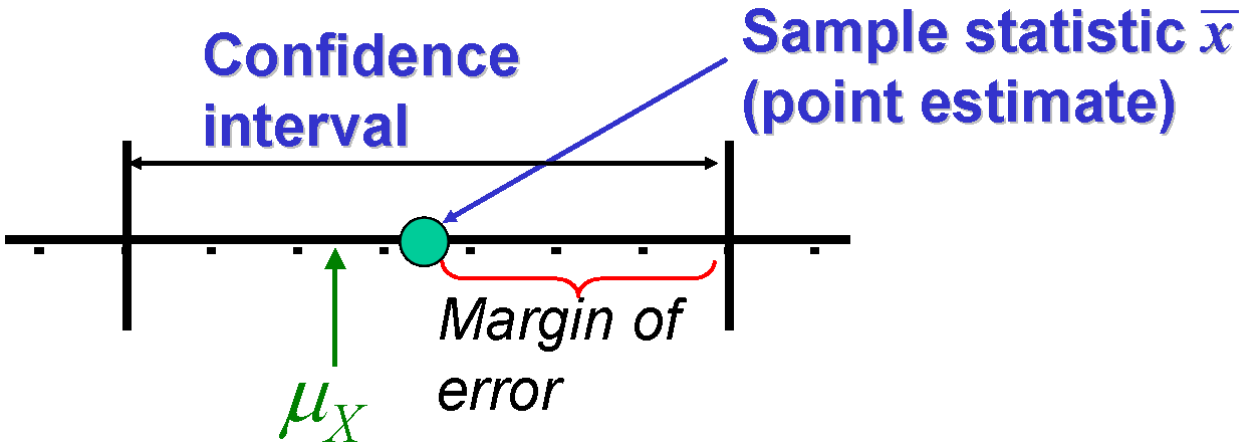
- Bootstrapping essentially only requires assumption that:
  - original sample is assumed to be representative of the population distribution
- Which is basic assumption in most statistical analyses anyway
- So why not resample (with replacement) from original sample?
- Resample to get new samples with same sample size and calculate sample statistic



- Use resample simulations to approximate sampling distribution

Revision: Confidence Intervals

- Use sample data to construct an interval with high probability of containing actual (**unknown**) population parameter
- Interval includes possible values for the population parameter which are in most agreement with the sample data
- Measure of uncertainty (level of confidence) in point estimate



- Let α be a number between 0 and 1.
- An interval (a, b) is a $1 - \alpha$ confidence interval for a population mean, μ_X, if $P(a < \mu_X < b) = 1 - \alpha$
- The probability, $1 - \alpha$, is called the **coverage probability** or **confidence coefficient** or **confidence level** of the interval.
- Common choice of 'significance level' α is 0.1, 0.05 or 0.01.

Confidence Interval Interpretation

- Confidence intervals are **commonly mis-interpreted**[2]
- A confidence interval calculated from a sample either contains the population parameter or it doesn't (i.e. probability of 1 or 0), so what is meant by 95% confidence?
- The bootstrap 95% confidence interval comes from a repeated sampling of the population
- Correct interpretation of 95% confidence interval:
  - If we repeatedly sample from population, with same sample size, and use boostrapping to calculate confidence interval, then 95% of intervals will include population parameter
- Confidence is statement about estimation procedure, rather than any particular interval calculated from a single sample.

[2] See Wikipedia page on "Confidence Intervals" for fuller discussion

Choice of Confidence Coefficient $1 - \alpha$

- We used 95% confidence in evaporation data example
- This is a commonly used but completely arbitrary choice
- The choice of confidence level should be determined by the application
  - If we wanted to be hyper-conservative that the interval will include the truth then perhaps we could use 99.9% but this would be **a lot wider** and may not really be acceptable in other applications.
  - If the application is less sensitive to the truth being within the interval then an 80% confidence level may suffice
- Common (somewhat arbitrary) values are 90%, 95% and 99%

Expanding on the Bootstrapping Code

Take another look at bootstrapping correlation code:

```
>> nsim=10000;
>> corrstats=bootstrp(nsim,@corr,Velocity,Evaporation)
```

`@` operator before the `corr` function name is "function handle"

Check out the help for more information:

```
>> help function_handle
>> doc function_handle
```

They allow you to:

- indirectly call a function (i.e. `corr`) within another function (i.e. `bootstrp`);
- include new input variables defined inside the function (i.e. Velocity and Evaporation bootstrap simulations)
- include existing variables from current workspace (WARNING!!!: these are locked in when handle is created).

Function Handles Are Powerful But Use With Care

Take a look at the following code:

```
>> temp=100 % define existing workspace variable
>> newhandle = @(x) sum([temp x]) % use new input and existing
   workspace variable
```

When the function handle `newhandle` was created on the second line the `temp` was locked in

The following gives $100 + 1 = 101$ as expected:

```
>> newhandle(1)
```

If you subsequently change the `temp` variable in the workspace **IT WILL NOT BE UPDATED IN THE FUNCTION HANDLE**

```
>> temp=200 % reassign different value
>> newhandle(1) % not updated in function handle
```

Hence, final call still gives 101 and not 201 as you might expect

Discussion of Function Handle Examples

In the example for bootstrapping the correlation:

```
>> nsim=10000;
>> corrstats=bootstrp(nsim,@corr,Velocity,Evaporation)
```

The function handle points to the existing `corr` function with no specified inputs

Inside the `bootstrp` program the `corr` function will be called with the new bootstrap simulated datasets resampled from subsequent `Velocity` and `Evaporation` vector arguments

Discussion of Function Handle Examples

The bootstrap correlation confidence interval example code was:

```
>> corrxy = @(x,y) corr(x,y)
>> bootci(nsim,{corrxy,Velocity,Evaporation},'alpha',0.05,'type','per')
```

The first line creates a new "anonymous" function with two inputs `(x,y)`, which is given function handle called `corrxy`

The second line includes this function handle (don't need extra `@` as `corrxy` is already a function handle)

Note the extra { } brackets in `bootci` call are needed to specify bootstrapping of `corrxy` statistic is over `Velocity` and `Evaporation` variables, and that the inputs following the { } brackets are optional arguments `'alpha'` and `'type'` which are not to be boostrapped

Further Discussion of Function Handle Examples

The first line of the "observation resampling" code:

```
>> bootbetas = bootstrp(nsim,@(y,x) regress(y,x),Evaporation,X);
>> prctile(bootbetas,[2.5 97.5],1)
```

Creates a new function handle with two input variables `(y,x)` to the existing function `regress`

The boostrapping is applied to the rows of the response **y** vector (`Evaporation`) and design matrix `X`

The `regress` function outputs the regression coefficient vector $\boldsymbol{\beta}$, the pair of which is collated in each row of the resultant `bootbetas` matrix

The `prctile` function calculates the 95% confidence interval for each column of `bootbetas` (specified by last option `1` in `prctile` call above), i.e for each coefficient (intercept and gradient)

Further Discussion of Function Handle Examples

The first line of the "residual resampling" code

```
>> bootbetas = bootstrp(nsim, @(bootr)regress(yhat+bootr,X), resids);
>> prctile(bootbetas,[2.5 97.5],1)
```

specifies a new function handle with input variable `bootr` and **existing variables from the workspace**: `yhat` and `X`

Therefore, `yhat` and `X` are now fixed in the function handle, so are not changed in each bootstrap

The `bootr` input is specified in the `bootstrp` function as the resamples from the `resids` vector

The residual resampling takes the existing predictions `yhat` and adds the bootstrapped residuals `bootr`, to get the new bootstrap simulation of **y** response vector.

The existing design matrix `X` is used as this is **fixed** for each boostrap simulation.

Further Properties of the Residuals

The true errors are assumed to have the following properties:

$$\begin{aligned} E(\boldsymbol{\epsilon}|\mathbf{X}) &= \mathbf{0} \text{ (i.e. mean zero)} \\ Var(\boldsymbol{\epsilon}|\mathbf{X}) &= \sigma^2\mathbf{I} \text{ (i.e. constant variance and uncorrelated)} \end{aligned} \tag{11}$$

It is easy to show (and intuitive) the mean of the residuals is zero. But what about the variance (and covariance) of the estimated errors (i.e. residuals)?

Using equation (9) above we have

$$Var(\hat{\boldsymbol{\epsilon}}|\mathbf{X}) = \sigma^2(\mathbf{I} - \mathbf{H}). \tag{12}$$

Notice this is not the same as (11) unless the hat matrix $\mathbf{H}$ is the null matrix. **So the estimated errors (residuals) have different properties than the 'true' errors**.

Leverages

In particular, the variance of the residual for observation i is given by the ith diagonal element of equation (12):

$$Var(\hat{\epsilon}_i) = \sigma^2(1 - h_{ii}). \tag{13}$$

This has important implications when considering the impact of outliers on the linear regression estimates. These diagonal elements h_{ii} are called the **leverages** of the observations. They indicate the **influence** of the observed explanatory variables on the estimated errors (and more generally on the fit of the whole model). **The larger the leverage the smaller the variance of the residual error, i.e. greater influence on the regression line.**
For simple linear regression the leverage h_{ii} is given by:

$$h_{ii} = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum(x_i - \bar{x})^2}. \tag{14}$$

So key factor impacting the leverage is how far the observation is from the mean of the explanatory variables $\bar{x}$.

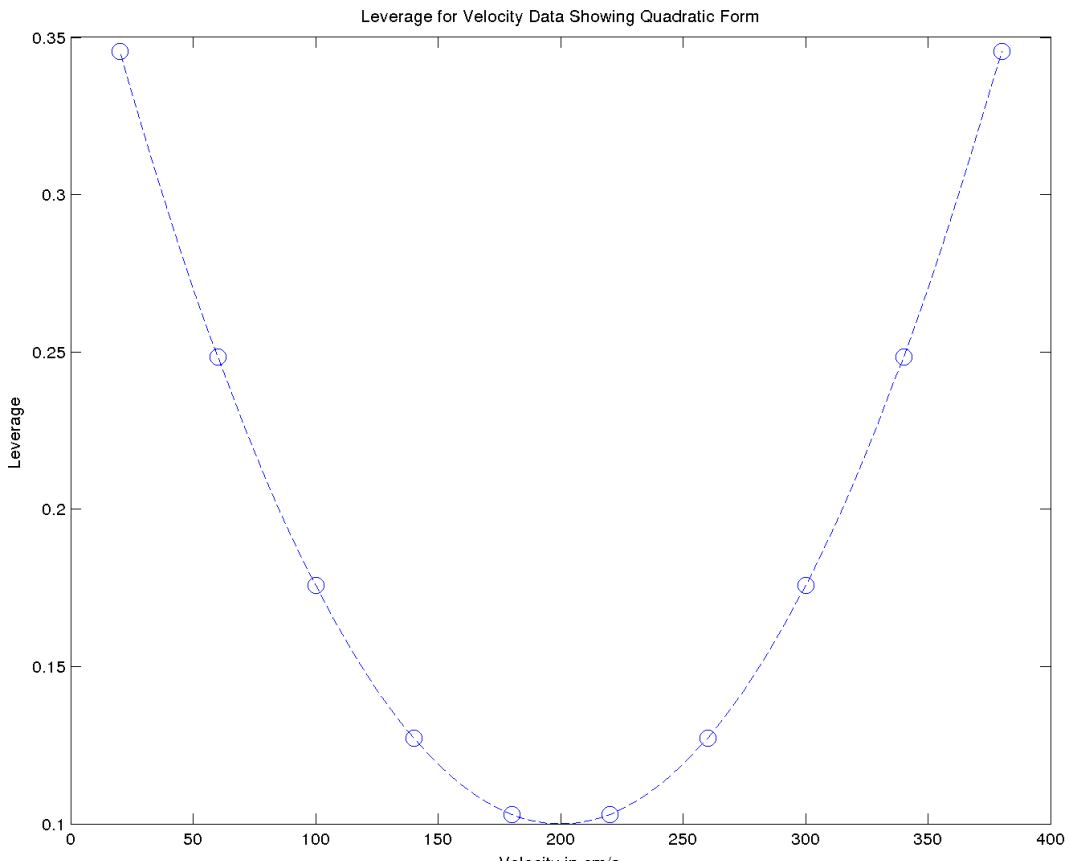
Example: Leverages for Evaporation Model

To calculate the leverage manually in MATLAB:

```
>> hatmatrix = X*inv(X'*X)*X' % whole H matrix
>> hii = diag(hatmatrix) % leverages are on diagonal
```

For simple linear regression there is only one explanatory variable so you can plot leverages against it:

```
>> plot(Velocity, hii, '.')
>> xlabel('Velocity in cm/s'); ylabel('Leverage')
>> title('Leverage for Velocity Data Showing Quadratic Form')
```



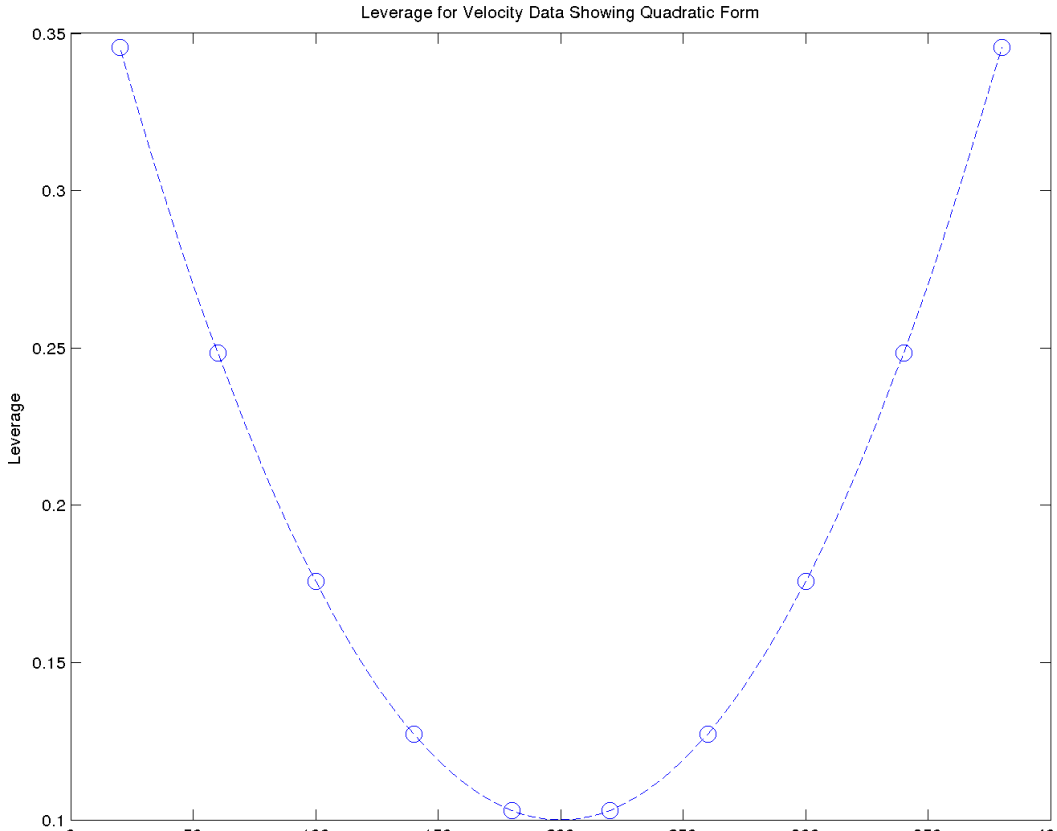
Example: Leverages for Evaporation Model

Leverages are also optional output of `regstats` function:

```
>> evapstats=regstats(Evaporation,Velocity,'linear',{'yhat','r','leverage'});
```

The quadratic nature of leverages about mean in the x-direction ($\bar{x}$) is evident below:

```
>> plot(Velocity, evapstats.leverage, '.')
>> xlabel('Velocity in cm/s'); ylabel('Leverage')
>> title('Leverage for Velocity Data Showing Quadratic Form')
```



High Leverage Good, Outliers With High Leverage Bad!

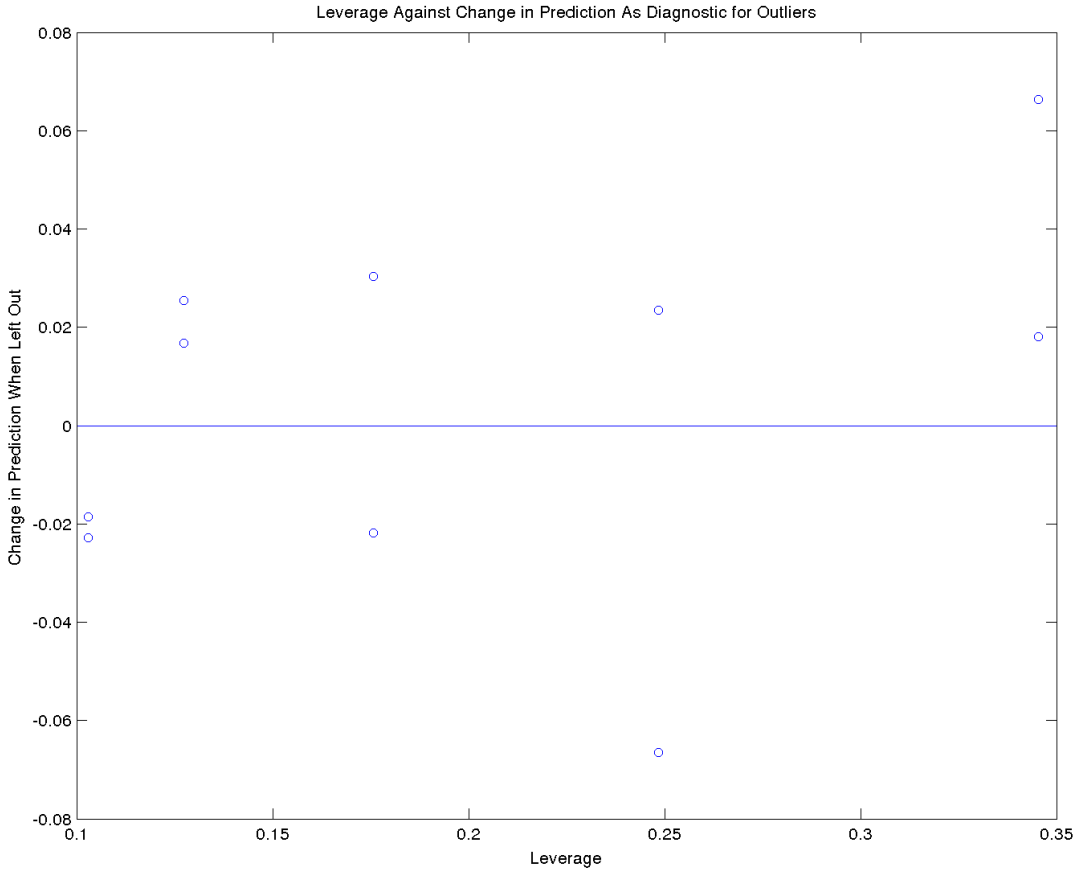
- Datapoints with high leverage provide a good source of information for determining a regression line;
- Note: high leverage points are those far away from the mean in the x-direction (i.e. outlying points in the x direction), **the leverage does not depend on the y variable at all**
- Outliers (in the y variable) can be problematic in regression situations, particularly for small datasets
- However, when outliers are also combined with high leverage then they are particularly troublesome
- There are lots of diagnostics used to highlight outliers with high leverage
- You saw in the lab last week (Problem B) that the boostrap coefficients and correlations can give an indication of influential outliers

Leave-one-out Diagnostics

- A common set of diagnostics are based on refitting the model when leaving each datapoint out in turn and using the new fitted model
- Suppose $\boldsymbol{\beta}_{(-i)}$ is the estimated regression coefficient when the ith datapoint is left out (i.e. ith element of $\mathbf{y}$ and ith row of design matrix $\mathbf{X}$ are ignored)
- These estimated coefficients can be used to get predictions of the response for the ith response $\hat{y}_{(-i)}$
- The difference between the original prediction and the leave-one-out prediction $\hat{y}_i - \hat{y}_{(-i)}$ are plotted against the leverages as a diagnostic
- Both the leverage and leave-one-out change in predictions are calculated using `regstats` function:

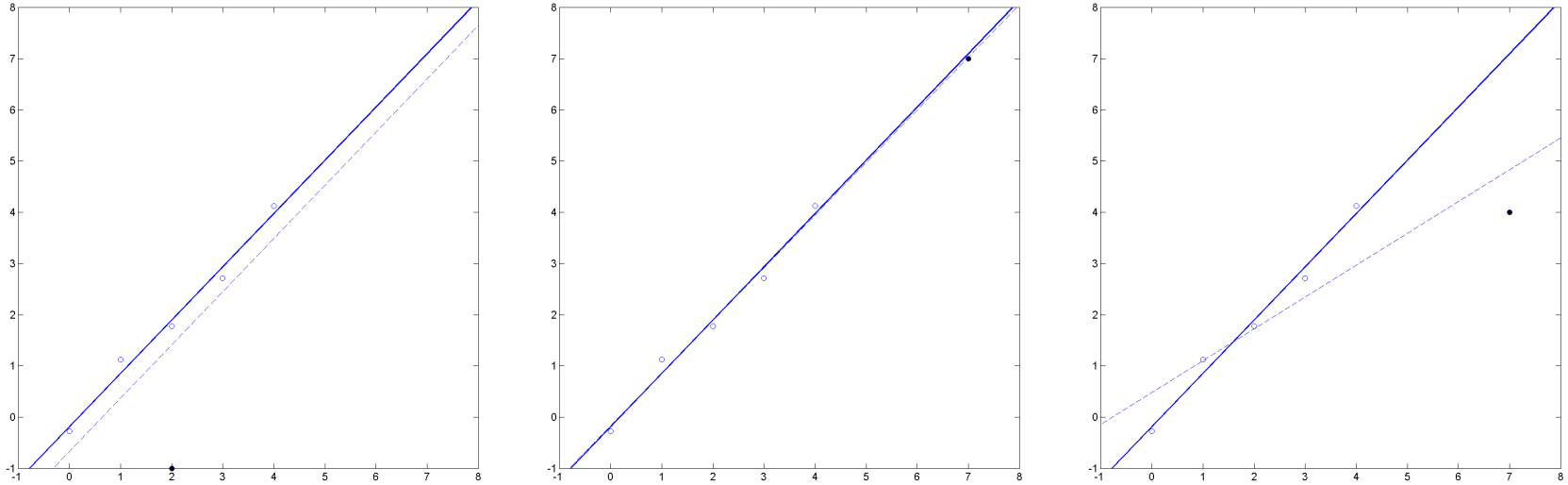
Example: Leave-one-out Diagnostics for Evaporation Data

```
>> evapstats=regstats(Evaporation,Velocity,'linear',{'leverage','dffit'});
>> plot(evapstats.leverage,evapstats.dffit,'o')
>> refline(0,0); % (slope=0,intercept=0)
>> xlabel('Leverage');ylabel('Change in Prediction When Left Out')
>> title('Leverage Against Change in Prediction As Diagnostic for Outliers')
```



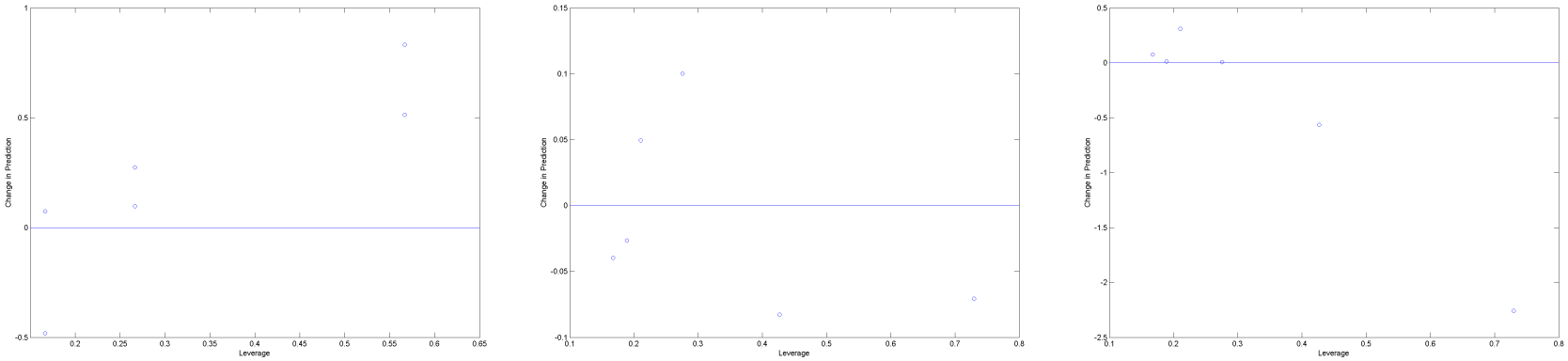
Conceptual Example of High Leverage Outlier

- Solid circle:
  - left - outlier with low leverage
  - middle - not outlier with high leverage (i.e informative datapoint)
  - right - outlier with high leverage
- Solid line regression fit excluding solid circle
- Dashed line regression fit including solid circle



Conceptual Example of High Leverage Outlier

Corresponding diagnostic plots:



- Outlying observation does not have a big effect on the fitted regression line (and therefore predictions) when it has low leverage, so does not show up in the left plot
- Whereas when outlier has high leverage it shows up very clearly (right plot) as a problem
- A high leverage point which is not outlying also does not stand out in the middle plot

Properties of Coefficient Estimates
(proofs non-examinable)

The parameter estimates $\hat{\beta}$ are unbiased:

$$\begin{aligned} E(\hat{\beta}|\mathbf{X}) &= E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}|\mathbf{X}] \\ &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'E(\mathbf{y}|\mathbf{X}) \\ &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}\beta \\ &= \beta \end{aligned} \tag{15}$$

and they have variance-covariance matrix:

$$\begin{aligned} Var(\hat{\beta}|\mathbf{X}) &= Var[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}|\mathbf{X}] \\ &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'Var(\mathbf{y}|\mathbf{X})\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} \\ &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'(\sigma^2\mathbf{I})\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} \\ &= \sigma^2(\mathbf{X}'\mathbf{X})^{-1} \end{aligned} \tag{16}$$

Properties of Predictions

Consider an observation row vector $\mathbf{x}$ (i.e. $(1, x)$ for simple linear regression) then the variance of the predicted "**mean response**" $\hat{y}^* = \mathbf{x}\hat{\boldsymbol{\beta}}$ is given by:

$$\begin{aligned} Var(\hat{y}^*|\mathbf{x}, \mathbf{X}) &= Var(\mathbf{x}\hat{\boldsymbol{\beta}}|\mathbf{x}, \mathbf{X}) \\ &= \mathbf{x} Var(\hat{\boldsymbol{\beta}}|\mathbf{X})\mathbf{x}' \\ &= \sigma^2 \mathbf{x}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}', \end{aligned} \tag{17}$$

with the corresponding standard error the square root of this.

However, when predicting a "**new response**" then $\hat{y}^*_{new} = \mathbf{x}\hat{\boldsymbol{\beta}} + \epsilon$ then the variance is larger:

$$Var(\hat{y}^*_{new}|\mathbf{x}, \mathbf{X}) = \sigma^2[1 + \mathbf{x}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}']. \tag{18}$$

Note: only difference is that the prediction of a new observation includes an extra σ^2 to account for the extra error associated with a new measurement of the response (e.g. accounts for extra uncertainty associated with measurement error).

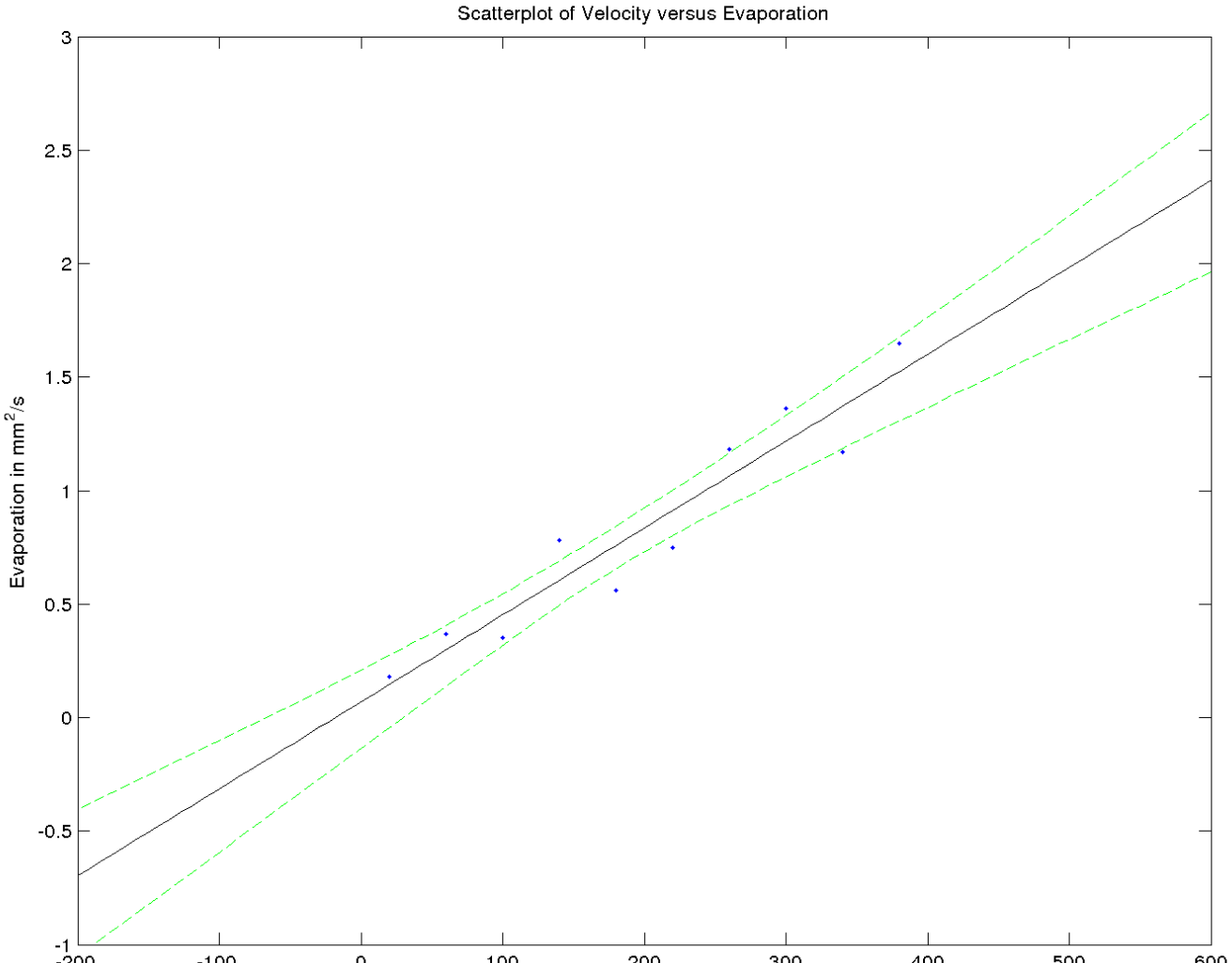
Example: Evaporation Mean Response Confidence Intervals

- A confidence interval for the fitted regression line is called a "confidence interval for the mean response"
- Bootstrap can be used to estimate this confidence interval:

```
>> % Set sequence of x values to plot over
>> npoints=100; minx=-200; maxx=600;
>> xtoplot=linspace(minx, maxx, npoints);
>> % Put together in design matrix form
>> Xtoplot=[ones(length(xtoplot),1) xtoplot']
>> yhats=Xtoplot*B; % predicted line for all points
>> % Calculate predictions for each point across all bootstrap beta vectors
>> bootyhat=Xtoplot*bootbetas'; % matrix of size npoints*nsim
>> ciyhat=prctile(bootyhat,[2.5 97.5],2) % 95% confidence interval
>> figure; plot(Velocity,Evaporation,'.'); hold on; % plot data
>> plot(xtoplot,yhats,'k-') % regression line
>> plot(xtoplot,ciyhat,'g--') % confidence interval
>> xlabel('Velocity in cm/s'); ylabel('Evaporation in mm2̂/s')
>> title('Scatterplot of Velocity versus Evaporation')
```

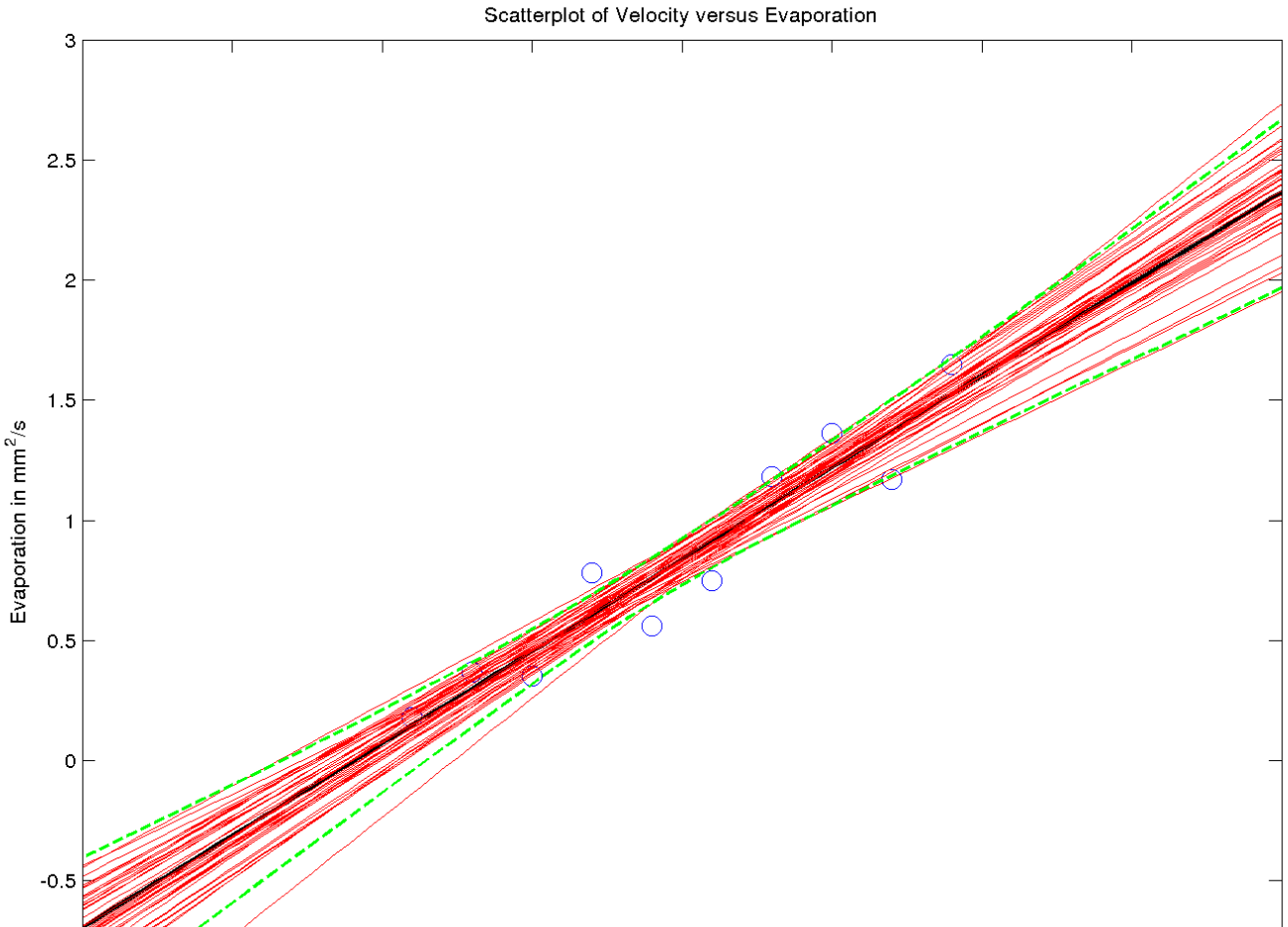
Example: Evaporation Mean Response Confidence Intervals

- Clearly, extrapolation of this model is not going to be reliable for negative velocity!
- Intervals get wider the further away from centroid in x you go
- In particular, wider for extrapolation than interpolation



Example: Evaporation Mean Response Confidence Intervals

- Conceptually way to think of boostrap is that there are many sample regressions lines (40 shown below in red) being fit in the bootstrap simulations
- For each x value the 95% confidence interval includes 95% of lines
- Described as "**pointwise**" confidence intervals



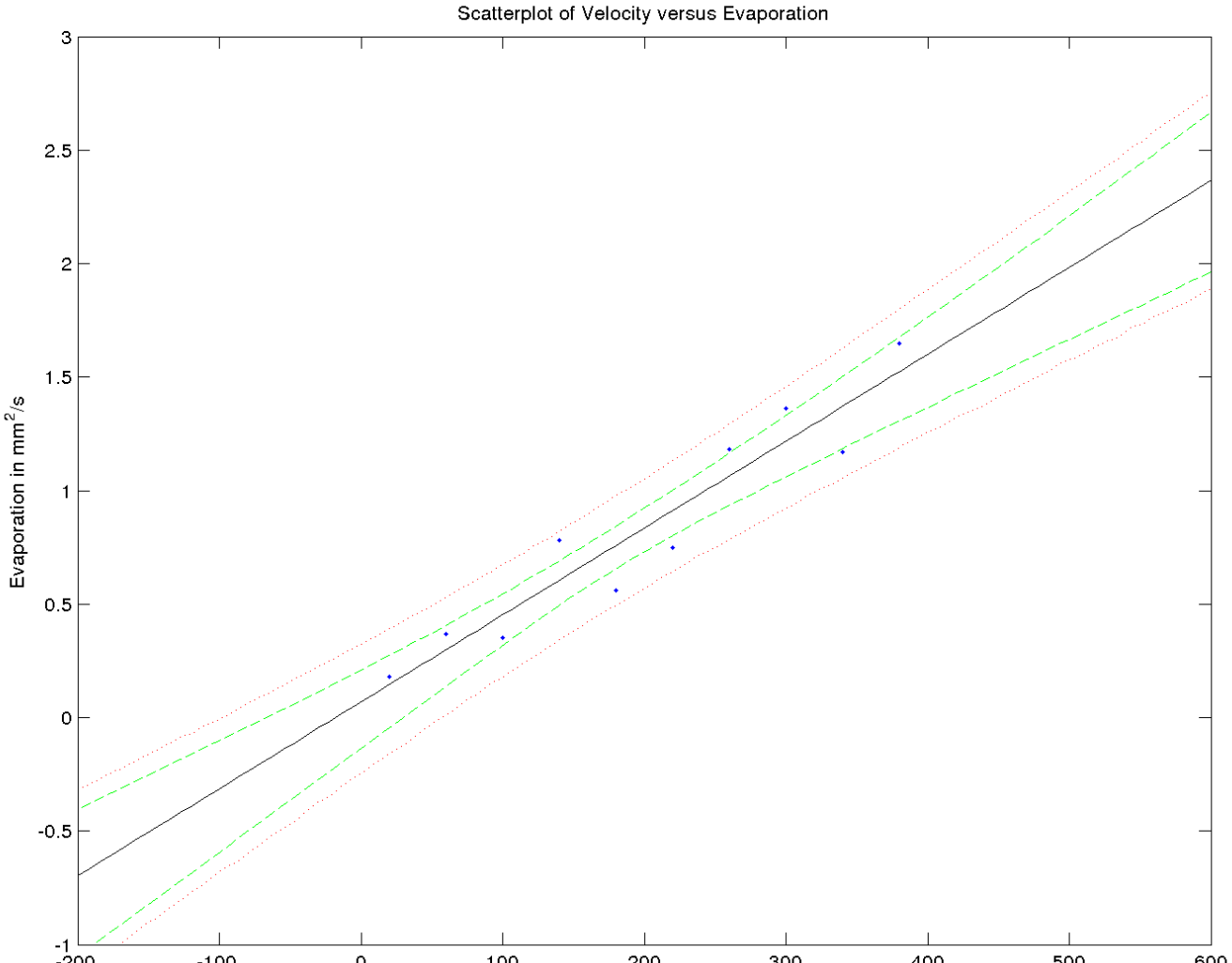
Example: Evaporation Prediction Intervals

- A confidence interval for prediction of a new observation (i.e. including error term) is called a "prediction interval"
- Bootstrap predictions for "new observations" have to account for both prediction of response and a boostrapped residual

```
>> % First simulate a residual for each prediction
>> % One for each of nsim boostraps and npoints to be plotted
>> bootresids=reshape(randsample(resids,npoints*nsim,true),npoints,nsim);
>> bootyhatnew=bootyhat+bootresids;
>> ciyhatnew=prctile(bootyhatnew,[2.5 97.5],2)
```

Example: Evaporation Prediction Intervals

- Notice that "pointwise" prediction intervals are similarly behaved to confidence intervals for the mean response
- Wider due to extra error variance (more uncertainty from errors in predictions of new observation)



Transformation of Variables

- The relationships between the response and predictors are frequently non-linear in nature
- Non-linear function can sometimes be expressed in linear form by a suitable transformation of the predictor or response variables or both
- The most common and useful transformations are taking reciprocals, powers and logarithms.
- For example:

  $$y = e^{\alpha+\beta x}$$

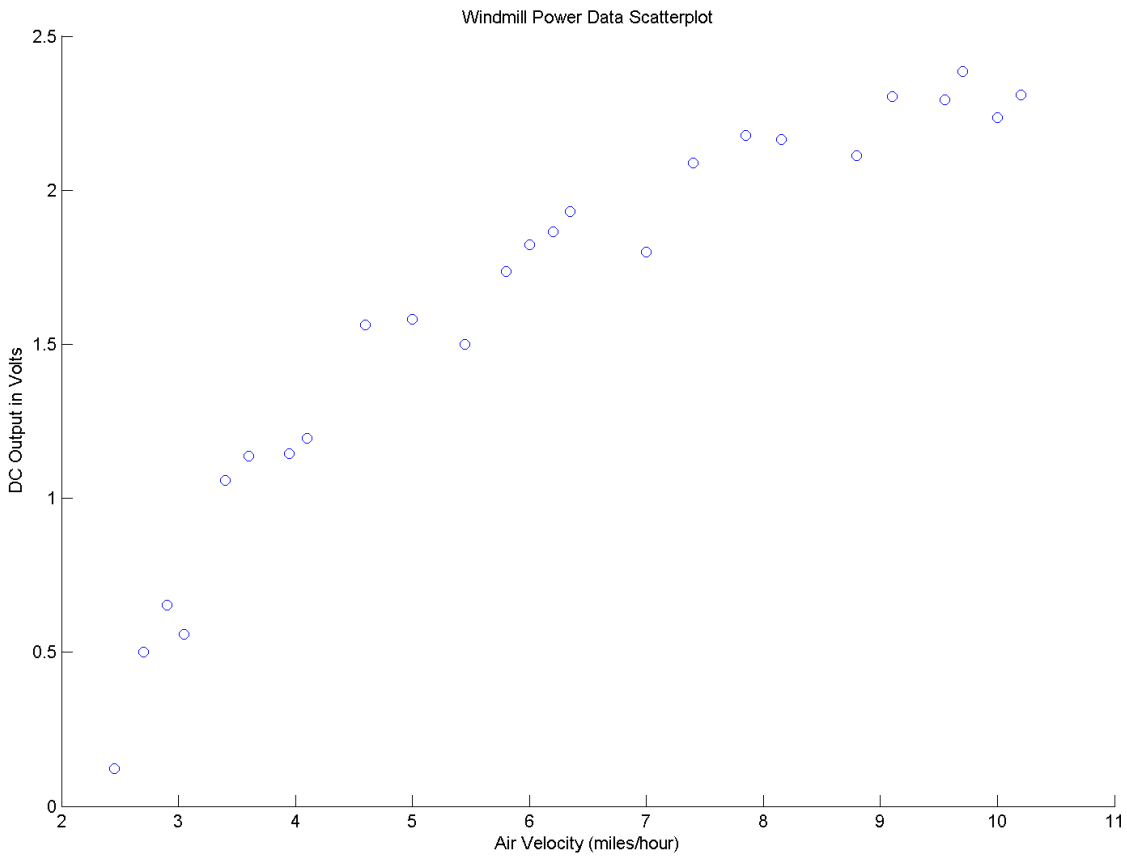
  can be transformed by taking natural logarithm:

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

  to give a linear form.
- Beware: non-linear transformation of either the response or explanatory variables will also impact the properties of errors
- Transformations can sometimes be used to resolve issues of a non-constant variance

Example: Windmill Power Production

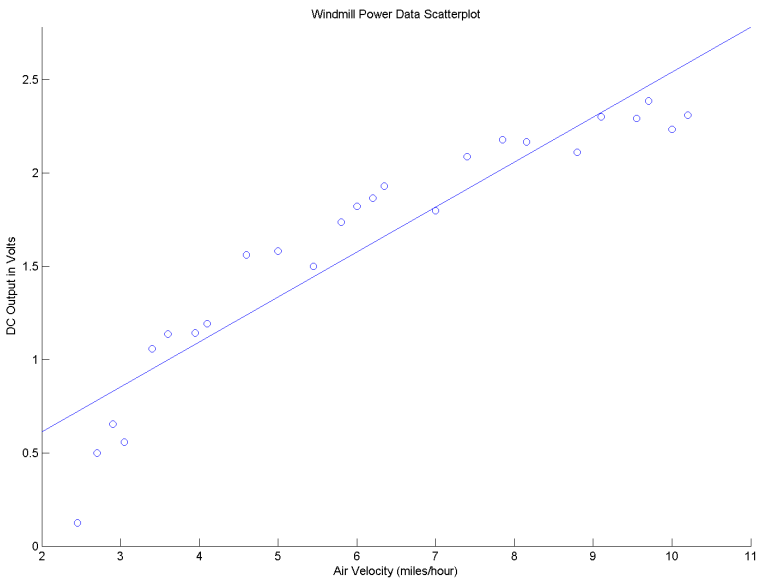
- The DC output from a windmill generator at different wind velocities are shown in scatterplot below.
- MATLAB code and datafile provided on Learn

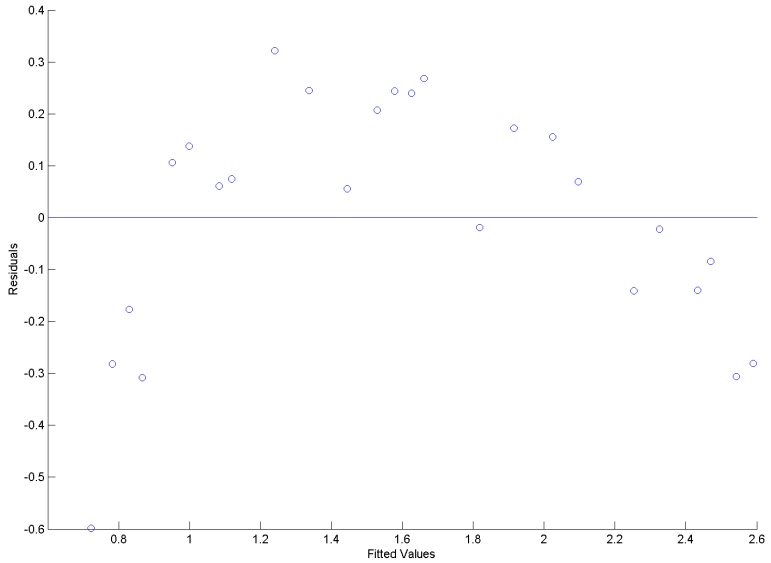


- There is a clear non-linear association between DC output and velocity.

Example: Windmill Power Production

- A simple linear regression with straight line provides a very poor fit.

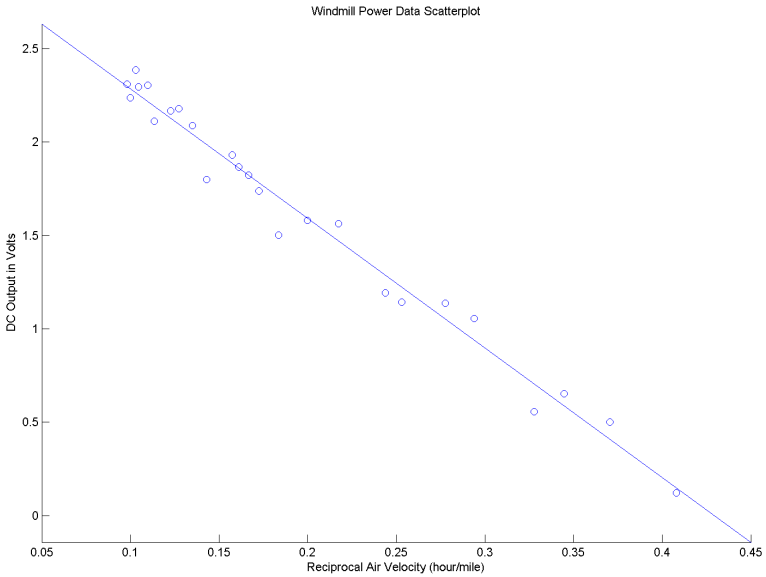


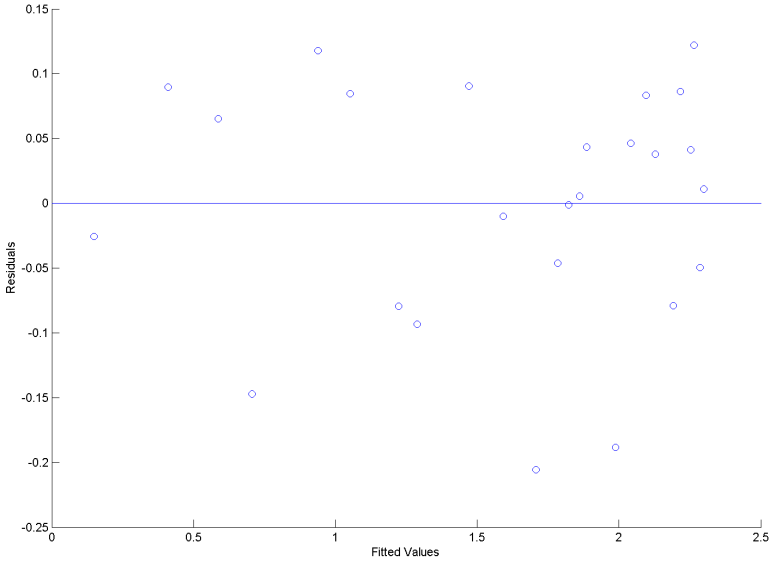


- There is a very strong non-random pattern (i.e. errors do not satisfy the zero mean assumption) in the residuals
- Proportion of variance explained: $R^2 = 87.5\%$ which is rather high (suggesting a good model)
- Another example to not rely on simple statistics and always plot the data!

Example: Windmill Power Production

- Linear regression of DC output against (1/wind velocity) gives a better straight line fit.





- Residuals now appear random centred around zero, with constant variance.

Multiple Linear Regression

- Multiple linear regression aims to predict a single response variable using multiple (usually denoted p) explanatory variables:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon \tag{19}$$

- where the random errors ϵ have the same properties as for simple linear regression
- The only addition is that the errors are also uncorrelated with all p explanatory variables $\{X_i : i = 1, \ldots, p\}$.

Linear In Coefficients

- Remember that the term "linear" indicates that the model is linear in the coefficients.
- So that the following polynomial (quadratic) function of the explanatory variable is still a linear model:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \epsilon$$

- as it is linear in the unknown coefficients

Additional Complications with Multiple Regression

- Most of the concepts from simple linear regression carry over to multiple linear regression
- Though there are a few extra complications, with the most obvious being:
  - graphical summaries/model diagnostics;
  - interpretation of coefficients; and
  - choice between competing models is challenging due to dimensionality of 2^p possible models to choose from

Graphics For Multiple Linear Regression

- In simple linear regression it is possible to produce a two dimensional scatterplot to explore the observed relationship between the explanatory variable and the response.
- When there are two explanatory variables the relationship is defined by a two dimensional plane in 3D space, a plot of which is often difficult to interpret.
- For more than two explanatory variables it gets more challenging to visualise the relationship between the explanatory variables and the response.
- On this course we will demonstrate some of the most commonly used graphical tools

Interpretation of coefficients

- The coefficient β_i determines the contribution of 'just' the variable X_i for predicting the response Y.
- More precisely, β_1 is the effect of unit increase in X_1 on the response Y, **when all the other components are held fixed**.
- Of course, in many situations the explanatory variables are related to each other and therefore it is often not physically possible to change one variable whilst keeping all the others fixed
- Thus complicating interpretation, as multiple explanatory variables have to be adjusted at once in a physically appropriate fashion

Examples: Interpretation of coefficients

- For example, if you are trying to build a basic model to describe compressive strength of concrete slabs the explanatory variables could be water/cement ratio, quantity of cement, coarse aggregates and fine aggregates.
  - Clearly it is not possible to vary just one of these variables at a time, as if the quantity of each ingredient is reduced the relative quantities of the others must increase.
- Another example, consider a regression model to predict the winter time PM_{10} air pollution concentration here in Christchurch, which requires multiple meteorological explanatory variables (temperature, wind speed, rainfall, cloud cover, etc.)
  - Again, the meteorological variables are highly interrelated, so it is not physically sensible to vary a single variable (e.g. temperature) on its own without considering the effect on the other variables (e.g. cloud cover and rainfall)

Fitting of Multiple Regression Model

- Thankfully, estimation of the multiple linear regression model is straightforward using the matrix notation for simple linear regression detailed above
- In fact, many of the corresponding results for multiple regression are unchanged
- The matrix representation of the multiple regression equation (19) is exactly the same as equation (2) for simple linear regression above, i.e.

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

but now the design matrix $\mathbf{X}$ is:

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & & & & \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix},$$

where each column represents an explanatory variable.

Fitting of Multiple Regression Model

- The n equations can therefore be written in the usual form $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & & & & \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

Fitting of Multiple Regression Model (cont.)

- The OLS estimates of the regression coefficients $\boldsymbol{\beta}' = (\beta_0, \beta_1, \ldots, \beta_p)$ are exactly the same as for simple linear regression from equation (4):

$$\hat{\boldsymbol{\beta}} \;=\; (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}.$$

- The general matrix equations for the predictions, residuals, R^2, hat matrix/leverages, variance of predictions/fitted values/coefficients are all the same as for simple linear regression so are not reproduced again here.

Fitting of Multiple Regression Model (cont.)

- One difference is an unbiased estimator of the error variance is now given by:

$$\begin{aligned} \hat{\sigma}^2 &= \frac{\sum(y_i - \hat{y}_i)^2}{n-(p+1)} \\ &= \frac{\boldsymbol{\epsilon}'\boldsymbol{\epsilon}}{n-(p+1)} \end{aligned} \tag{20}$$

  where $p+1$ is the number of explanatory variables and the intercept, i.e. to estimate the error variance $p+1$ degrees of freedom (from regression coefficients) have been used up.

- If we wish to predict the **mean response** or for a **response for a new observation** the results from equations (17) and (18) are the same except that all the explanatory variables are put together into the row vector $\mathbf{x} = [1\ x_1\ x_2\ \ldots\ x_p]$.

Matrix Plots

- Most commonly used plots for multidimensional data are called **matrix plots**:
  - scatterplots of each pair of variables against each other; and
  - placed side by side in order (like a matrix); and
  - common axis scale in all plots of the variable.
- Often called **pairwise scatterplots**
- Diagonals often replaced with histogram or boxplot

Example: Abrasion Resistance

- Objective: determine how the abrasion resistance is affected by the rubber hardness and tensile strength
- Study: rubber specimens (experimental units) exposed to steady abrasion for a fixed time period
- Variables:
  - Loss: measured the weight loss (grams per hour);
  - Tensile: tensile strength measured in kg/cm2; and
  - Hardness: measured in degrees Shore.
- Dataset: 30 triplets of these three measurements

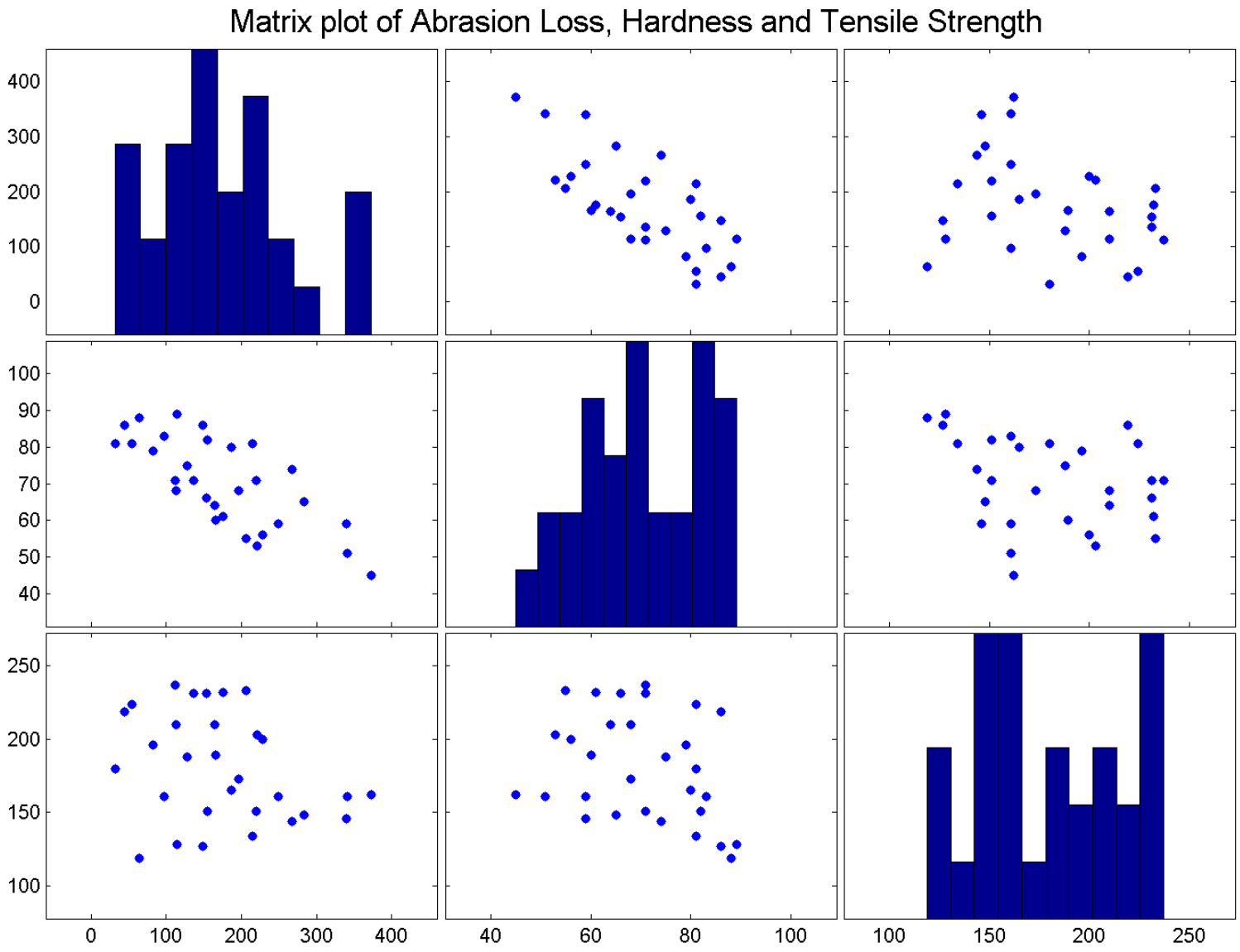
Example: Abrasion Resistance Matrix Plot Code

- Code assumes data read in one matrix called `data`
- One column per variable (response first)
- Matrix is simple to produce in MATLAB:

```
>> % Assumes all data is read in as a matrix
>> plotmatrix(data)
>> title('Matrix plot of Abrasion Loss, Hardness and
   Tensile Strength')
```

- Only issue: Painful to put axis labels on the plot
- Workaround: put them in title (in right order)

Example: Abrasion Resistance Matrix Plot



Example: Abrasion Resistance Matrix Plot Features

- Upper and lower triangles are mirror image of each other
- If response is first, then first row gives first indication of association between response and explanatory variables
- Other plots in upper triangle summarise associations between the explanatory variables
- Histogram on diagonal useful to understand spread of data
- Impression for this dataset:
  - moderate-strong negative linear association between abrasion losses and hardness
  - weak negative association between abrasion losses and tensile strength
  - weak negative association between hardness and tensile strength

Matrix Plot General Comments

- Matrix plots are excellent for:
  - examining relationships between pairs of variables;
  - detecting obvious outliers in one or two variables; and
  - displaying a large number of variables.
- However, it can be difficult to extract higher order relationships (interactions) between several variables and outlier in more than two variables.
- Other possibilities: `glyphplot`, `andrewsplot` or `parallelcoords`

Example: Abrasion Resistance Correlation and Regression

- Often worthwhile getting correlation matrix:

```
>> corr(data)
```

- which gives:

```
ans =

    1.0000   -0.7377   -0.2984
   -0.7377    1.0000   -0.2992
   -0.2984   -0.2992    1.0000
```

- Fitting multiple regression model :

```
>> % extract response vector
>> y=data(:,1);
>> % create design matrix
>> X=[ones(size(data,1),1) data(:,2:end)];
>> % fit regression model
>> [B,BINT,R,RINT,STATS] = regress(y,X);
```

Example: Abrasion Resistance Regression Results

- **Multiple coefficient of determination** R^2 (proportion of variation in abrasion loss explained by model) obtained from:

```
>> STATS(1)
```

- which gives $R^2 = 84\%$
- Regression coefficient vector (B vector) is:

```
B =

  885.1611
   -6.5708
   -1.3743
```

Example: Abrasion Resistance Regression Coefficients

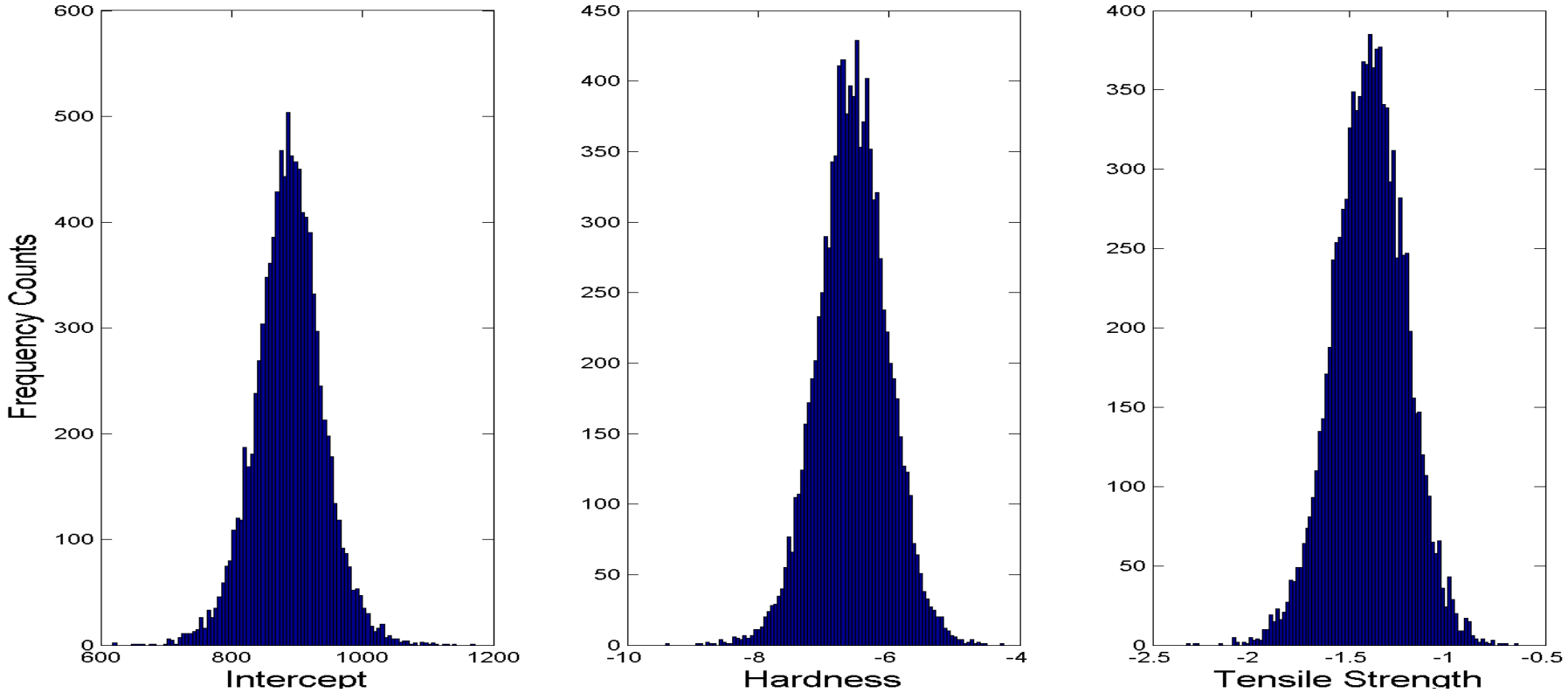
- 95% confidence interval for regression coefficients assuming normally distributed errors given by BINT matrix:

```
BINT =

  1.0e+003 *

    0.7585    1.0119
   -0.0078   -0.0054
   -0.0018   -0.0010
```

- Better to use bootstrapping (observation resampling appropriate here):

```
>> % observation resampling (random X)
>> bootbetas = bootstrp(nsim,@(y,x) regress(y,x),y,X);
>> prctile(bootbetas,[2.5 97.5],1)
>> figure;
>> subplot(1,3,1);hist(bootbetas(:,1),100);xlabel('Intercept')
>> subplot(1,3,2);hist(bootbetas(:,2),100);xlabel('Hardness')
>> subplot(1,3,3);hist(bootbetas(:,3),100);xlabel('Tensile Strength')
```

Example: Abrasion Resistance Bootstrap Coefficients



- 95% confidence intervals for regression coefficients using bootstrapping (does not require normal errors assumptions)
- Given by each column of `prctile(bootbetas,[2.5 97.5],1)`:

```
ans =

  780.9881   -7.6261   -1.7635
  992.9718   -5.4824   -1.0359
```

Hypothesis Testing of Regression Coefficients 1

"state of nature"	Don't Reject H_0	Reject H_0
$H_0 : \beta_i = 0$ is "true"	OK	Type I error
$H_1 : \beta_i \neq 0$ is "true"	Type II error	OK

- We want to reject H_0 when it is true with a small probability
- That is, we want to keep the probability of Type I error $\leq \alpha = 0.05$, say
- Similarly, we want to minimize type II error as well
- There are two ways to do a hypothesis test
- 1. Reject H_0 if 0 is not inside the $(1 - \alpha)$ confidence interval for β_i
- 2. Compute p-value – a measure of evidence against H_0

"p-value" range	evidence
< 0.01	very strong evidence against H_0
$[0.01, 0.05]$	strong evidence against H_0
$[0.05, 0.10]$	weak evidence against H_0
> 0.1	little or no evidence against H_0

Hypothesis Testing of Regression Coefficients 2

- Suppose we want to carry out hypothesis test of each coefficient β_i being zero:
  - Null hypothesis $H_0 : \beta_i = 0$
  - Alternative hypothesis $H_A : \beta_i \neq 0$
- Could use above confidence intervals which has double sided 95% confidence:
  - if zero is included within the interval, then "**there is insufficient evidence at the 95% level to reject the null hypothesis**"
  - if zero is excluded within the interval, then "**there is sufficient evidence at the 95% level to reject the null hypothesis**"
- Or you can calculate p-values for single sided tests (depending on which direction is most relevant):
  - $H_0 : \beta_i \leq 0$ and $H_A : \beta_i > 0$ (use for positive $\hat{\beta}_i$)
  - $H_0 : \beta_i \geq 0$ and $H_A : \beta_i < 0$ (use for negative $\hat{\beta}_i$)
- using boostrap simulations of coefficient (see next slide)

Example: Abrasion Resistance Testing Coefficients

- MATLAB code for calculating each type of p-value:

```
>> pnegative=sum(bootbetas>0)/nsim
>> ppositive=sum(bootbetas<0)/nsim
```

- which for this example gives:

```
>> pnegative=sum(bootbetas>0)/nsim

pnegative =

     1     0     0

>> ppositive=sum(bootbetas<0)/nsim

ppositive =

     0     1     1
```

- i.e. none of the bootstrap simulations were of opposite sign to OLS estimate in $\hat{\boldsymbol{\beta}}$
- In practice this means the p-value is less than the reciprocal of the number of bootstrap simulations
- Should be reported as $p < 1e-5$ for `nsim=100000`

Diagnostic Plots for Multiple Regression

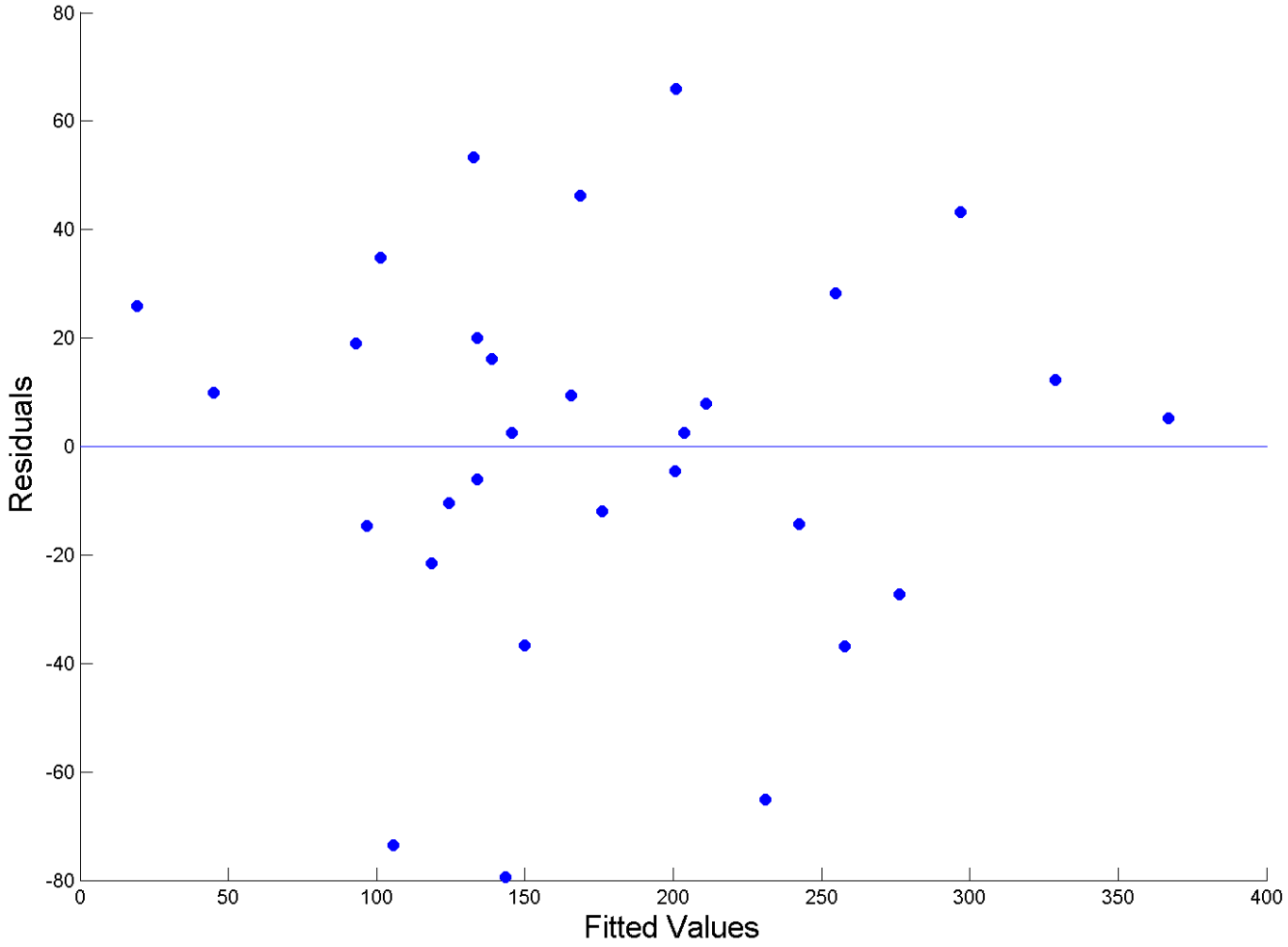
Common diagnostics for multiple regression models:

1. residuals against each explanatory variable;
2. residuals against predictions;
3. leave-one-out change in predictions (or coefficients) against leverages;
4. histogram of residuals;
5. residuals against auxiliary variables (e.g. variables left out of model, or time); and

- Interpretation of (1)-(3) same as for simple linear regression
- (4) should look close to normal if you want to use confidence intervals (or hypothesis testing approaches) based on normal error distribution assumption
- Remember: normality assumption avoided using bootstrap
- (5) highlights other predictors, or correlation with errors over time (thus breaking uncorrelated errors assumption)

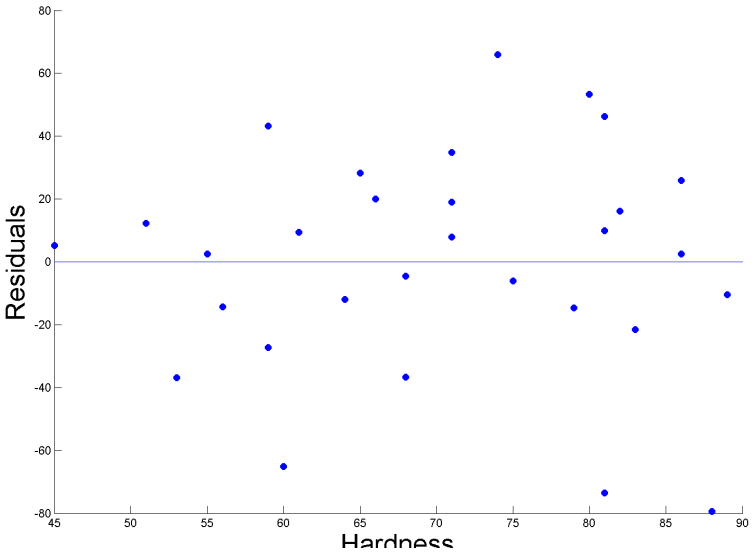
Example: Abrasion Resistance Diagnostic Plots

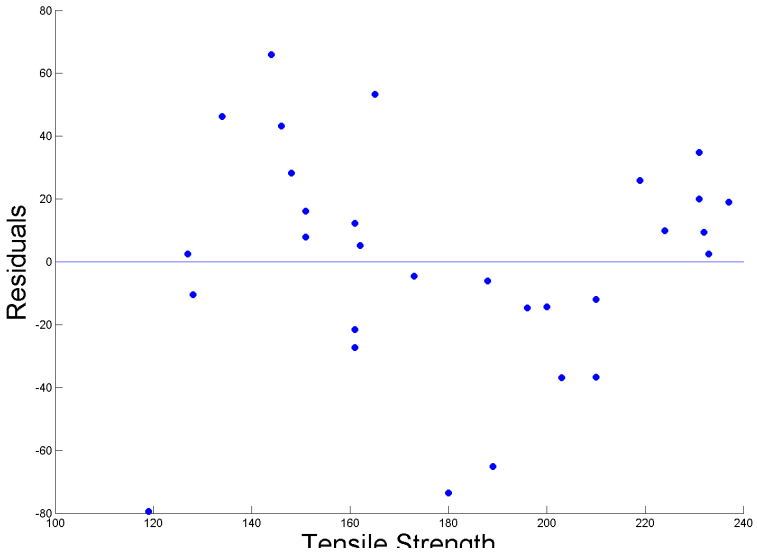
1. Residuals against predictions



Example: Abrasion Resistance Diagnostic Plots

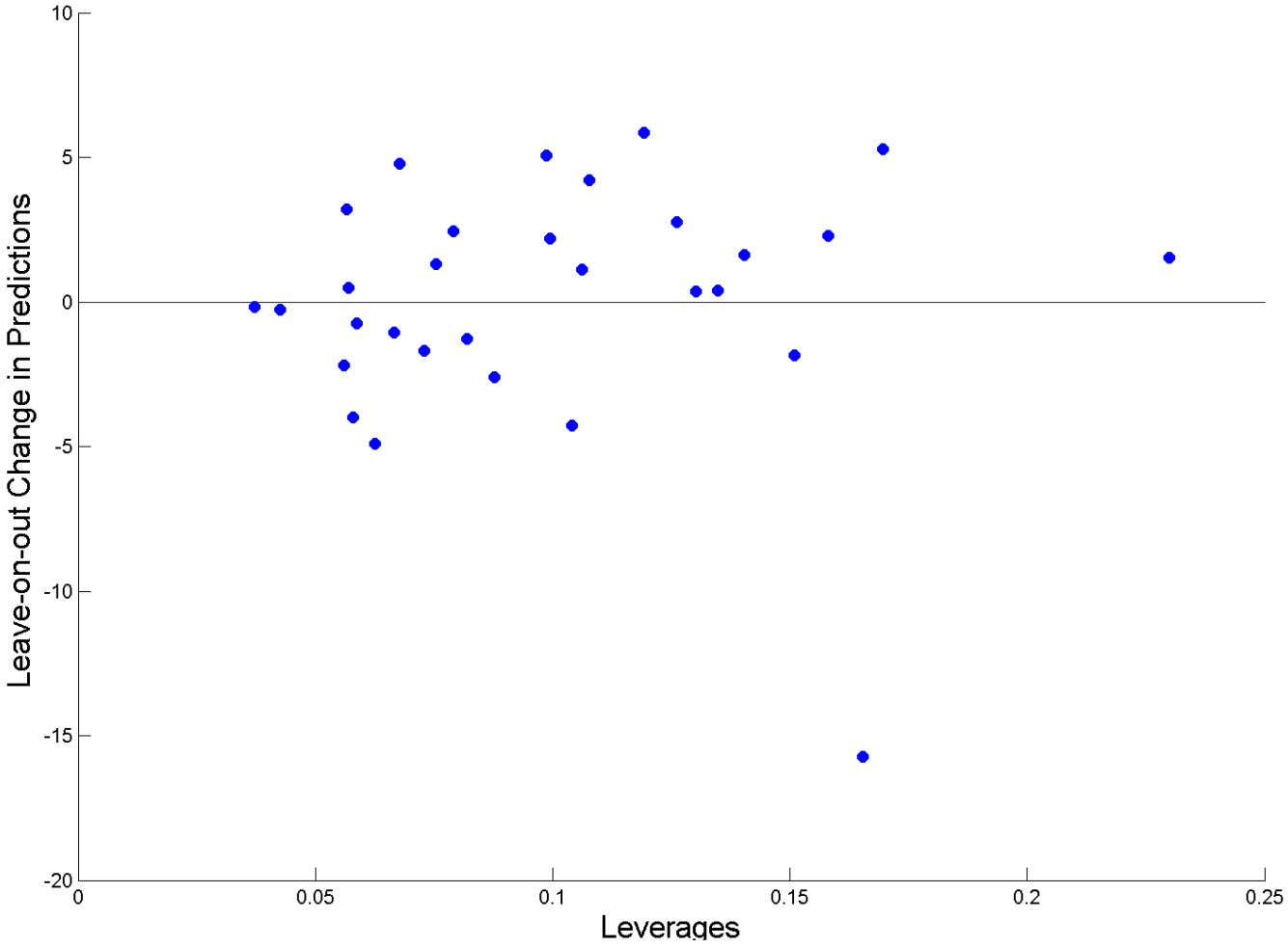
2 Residuals against both explanatory variables





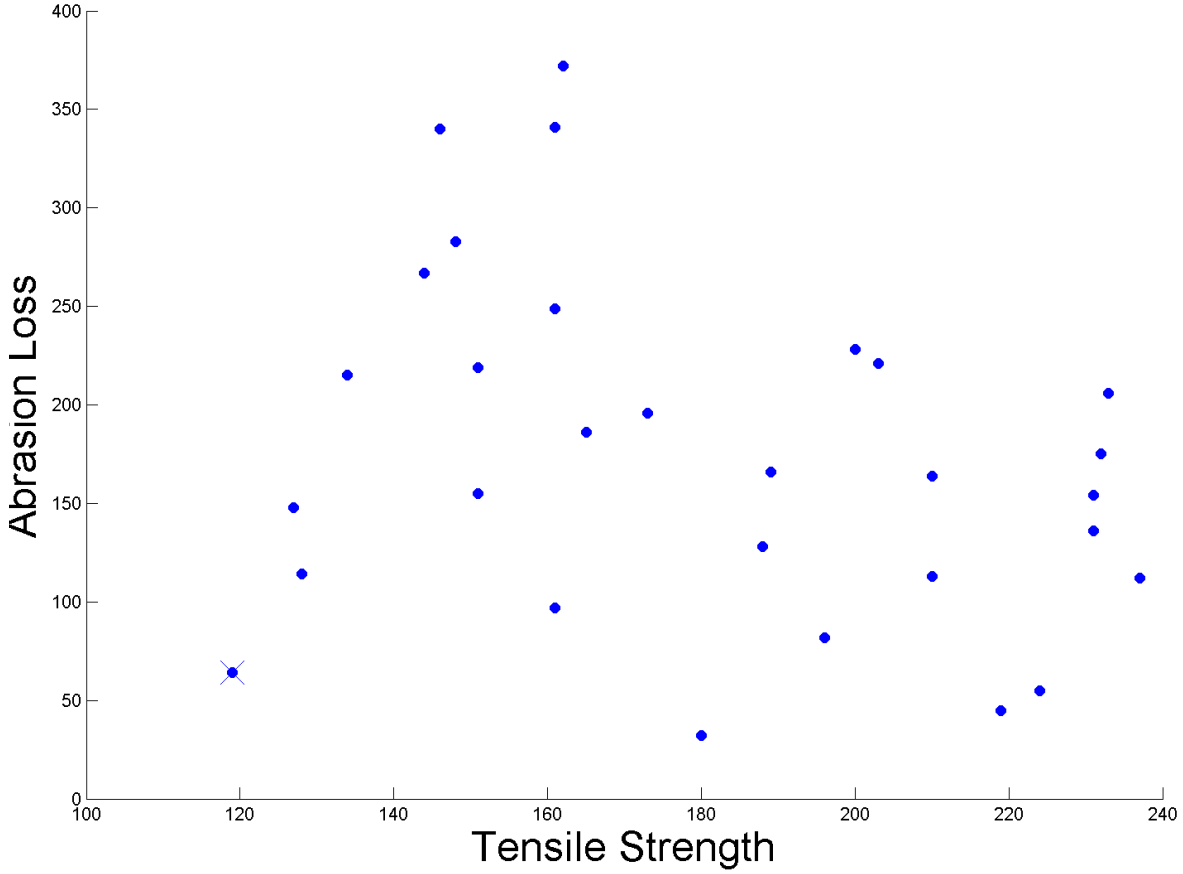
Example: Abrasion Resistance Diagnostic Plots

3 Leave-one-out change in predictions against leverages



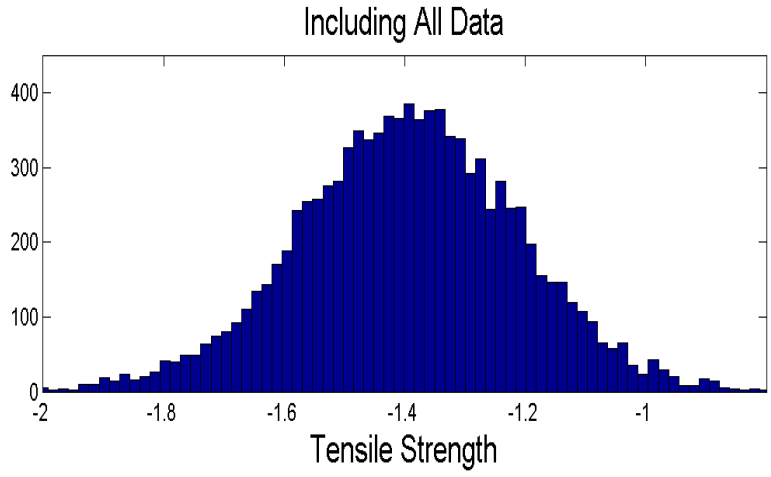
Example: Abrasion Resistance Diagnostic Plots

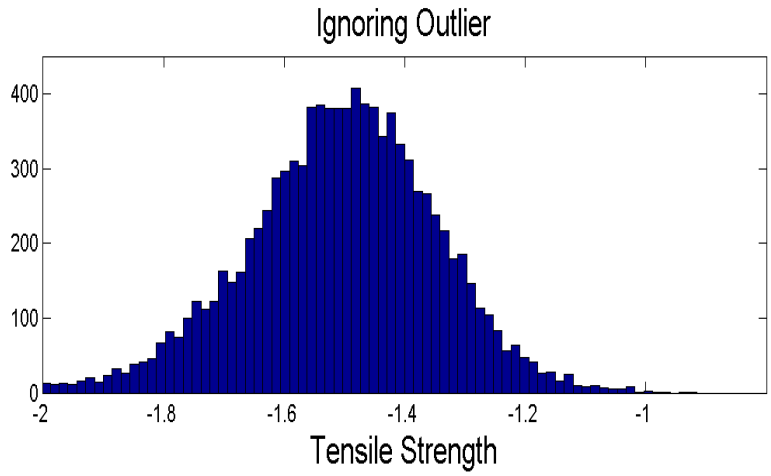
- Previous plot suggests an outlier (did you spot it?)
- They are often not easy to spot in large multiple regression problems (shows power of diagnostic plot)
- Looking again at the original abrasion loss and tensile strength data suggest a possible outlier:



Example: Abrasion Resistance Diagnostic Plots

- Bootstrapped "`Tensile Strength`" coefficients show effect of outlier
- Notice wider spread and possible second mode (around -1.25) when using all data
- Much less spread (uncertainty) in coefficient when outlier is ignored
- **Bootstrap simulations provide another useful diagnostic for influential outliers**





Summary of Diagnostic Plots

Summary diagnostic plots to evaluate OLS assumptions:

Assumption	Diagnostic Plots	Check For
Random and representative of population. (e.g. no outliers)	Scatterplot (Matrix plot) Histogram of residuals Leverages against leave-one-out statistics Boostrap correlations/coefficients	No outlying responses Random scatter No outlying residuals No outliers with high leverage Multimodel behaviour or large change if outlier ignored
Linearity	Residual against explanatory variables	No remaining pattern Constant mean of zero
Constant Variance	Residual against explanatory variables Residuals against predicted values	Constant spread Constant spread
Normality	Histogram of residuals	Normal shape (if assumed)
Uncorrelated	Residuals against explanatory variables or time (if relevant)	No clustering of residuals

Model Choice Statistics

- Abrasion model has **small number of explanatory variables** and it is clear that the models provide an **adequate fit**
- If there are a large number of explanatory variables an **objective procedure** is needed to decide how many explanatory variables (and which ones) need to be included in the model to provide an "adequate fit".
- Need to outline **model choice statistics** to assess model adequacy
- If there are a **moderate number of potential explanatory variables**: compare the performance of all 2^p models (called the **all possible regressions selection procedure**)
- For large models: **methodical algorithms** commonly used to efficiently search for terms to be included in the model

Overfitting

- Key principle: model should not **overfit** the data
- An overfitted model has more terms in the model than is required to provide an "adequate fit" to the data
- Consequence of overfitting:
  - model will **perform well on the observed sample** of data used to fit the model
  - but will **perform poorly for predictions on future data**
- Extrapolation of an overfitted model will also tend provide very poor predictions
- An overfitted model explains the random errors in the sample data rather than capture the true underlying relationship
- Thus future predictions will be poor as the overfitted model tries to predict the random noise, which will be different in future observations as it is random!

Parsimony

- It is desirable to find the simplest model required for the application, which is captured by the concept of **parsimony**:
  - *If two competing models have statistically the same predictive ability then the* **parsimonious model** *is the one with the smaller number of parameters*

All Possible Regressions

- Considers models containing all possible combinations of the explanatory variables
- For example, if there are 3 explanatory variables X_1, X_2 and X_3
- There are $2^3 = 8$ subsets of the explanatory variables:
  1. intercept only
  2. X_1
  3. X_2
  4. X_3
  5. X_1 and X_2
  6. X_1 and X_3
  7. X_2 and X_3
  8. all three variables
- Lots of summary statistics to evaluate performance
- In practice, the choice of the criteria is left upto the modeler
- But good practice to make judgement based on many statistics

Adjusted R^2

- (Multiple) coefficient of determination R^2 always increases (or at least stays the same) if more explanatory variables are included in the model.
- So useful criteria for model choice
- Adjusted R^2 statistics overcomes this issue:

$$R^2_{adj} = 1 - (1 - R^2)\frac{n-1}{n-(p+1)} \tag{21}$$

- Penalizes models with large numbers of parameters p
- Many, many! other statistics
- A **very general approach** is to use cross-validation

Leave-one-out is Cross-validation

- We have already met the idea of leave-one-out predictions (or coefficients) to investigate outliers
- The leave-one-out idea can also be used to provide a measure of the overall model fit, but which penalizes against overfitting
- Leave-one-out is just a special case of more general **cross-validation** and approach
- So far, the sample of data has been used both to:
  1. estimate the model, and then
  2. assess the model performance.
- In some sense, the information contained in the data is being used twice, which can lead to **overoptimistic estimates of the model performance**.
- A model that performs well on the original sample may perform poorly on future predictions

Holdout Method

- Simple way to ameliorate this problem, is to randomly split the entire sample into two non-overlapping subsets:
  - **training set** used to fit model
  - **test set** used to evaluate performance
- Known as the **holdout method**
- Typical ratio is 2:1 in favour of training set
- Advantage of this method is that it is less prone to overestimating the model performance due to overfitting and does not require any extra computations
- However, the regression estimates and performance can have a high variance, due to the reduced sample size in each subset.
- The performance evaluation also depends heavily on split into training and test sets, particularly for small datasets
- Different "runs" will likely give different results!

Cross-Validation

- **K-fold cross validation** improves on holdout method
- Dataset is randomly divided into K subsets and the holdout method is repeated in K trials
- In each trial, one of the K subsets is used as the test set and the other $K-1$ subsets are used as the training set
- Mean sum of square (MSE) of the errors (or similar performance measure) across all K test set trials is computed
- Every observation is in a test set exactly once, and gets to be in a training set $K-1$ times
- Principal advantages are (a) variance of the regression estimates is reduced as the training sample is larger than in holdout method and (b) result is less dependent on initial random allocation to K sets
- Principal disadvantages are (a) the extra computations involved and (b) results can change on each run
- Typically 10-fold cross-validation is the default

Leave-one-out Cross-Validation and PRESS

- Special case of K-fold cross validation is **leave-one-out cross-validation**
- i.e. where one observation is left out at a time or n-fold
- Same idea is same as leave-one-out concept we met before
- Mean sum of square of the errors (MSE) for the leave one predictions is then calculated
- Some computer packages give the total sum of square of errors instead, which is called the Prediction REsidual Sum of Squares (PRESS) statistic:

$$\text{PRESS} = \sum_{i=1}^{n} (y_i - \hat{y}_{-i})^2, \tag{22}$$

  where y_{-i} is the prediction for observation i obtained when the regression model is estimated with observation i left out
- Main advantage is that the results will be same on every run, as there is no random allocation

Cross-Validation and PRESS Comments

- No matter which approach you take, you select the model with the smallest prediction error (total or mean) sum of squares
- PRESS sounds computationally expensive, as you need to fit the regression model n times to each training set
- However, PRESS can be calculated directly from the regression model estimated using the complete dataset
- By adjusting for the influence of each observation (i.e. using the leverage h_{ii}):

$$\text{PRESS} = \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i}{1 - h_{ii}} \right)^2 \tag{23}$$

- Hence, the regression model need only be estimated once using the complete dataset, and PRESS is simply the sum of square of the residuals corrected for the leverage

MATLAB Implementation of Cross-validation

- Thankfully MATLAB implements the tedious cross-validation process using `crossval` function
- First need to create function handle to fit model to training data and predict on test data:

```
>> regf = @(Xtrain, ytrain, Xtest)(Xtest * regress(ytrain,Xtrain))
```

- Then apply cross-validation to dataset:

```
>> cvMSE = crossval('mse',X,y,'predfun',regf)
```

- MATLAB defaults to 10-fold cross-validation
- First input `'mse'` tells `crossval` function to calculate MSE
- Following `X` and `y` inputs are design matrix and response vector, to which cross-validation is to be applied
- Last input `'predfun'` sets prediction function handle to `regf`

Some Explanation of MATLAB Cross-validation Options

- If you specify one of `'Kfold'`, `'Holdout'` or `'Leaveout'` options to `crossval` function then default behaviour is overridden:

```
>> cvMSE = crossval('mse',X,y,'predfun',regf,'Kfold',5) % 5-fold
>> cvMSE = crossval('mse',X,y,'predfun',regf,'Holdout',1/3) % holdout 2:1
>> cvMSE = crossval('mse',X,y,'predfun',regf,'Leaveout',1) % Leave-one-out
```

- Note: only specify one of these options!
- Value of `'Holdout'` as 1/3, specifies the proportion of data in test set (total number in test set can be specified instead)
- Value of `'Leaveout'` can only be 1
- `crossval` has many options, see: `help crossval` or `doc crossval`

Example: Abrasion Resistance Cross-validation

- Repeat cross-validation for all model subsets:

```
>> regf = @(Xtrain, ytrain, Xtest)(Xtest * regress(ytrain,Xtrain))
>> cvMSE = crossval('mse',X,y,'predfun',regf)
>> cvMseNoStrength = crossval('mse',X(:,1:2),y,'predfun',regf)
>> cvMSENoHardness = crossval('mse',X(:,[1,3]),y,'predfun',regf)
```

- which gives:

```
>> [cvMse cvMseNoStrength cvMseHardness]

ans =

  1.0e+003 *

    1.5259    3.8825    7.5040
```

- Adjusted R^2 available from `regstats` function:
- Conclusion: all variables needed in model as 10-fold cross-validation MSE is smallest for this model

Example: Abrasion Resistance Adjusted-R^2

- Adjusted R^2 available from `regstats` function:

```
>> regstats(y,data(:,2:end),'linear',{'adjrsquare'})
```

- Which gives $R^2_{adj} = 83\%$ (and 53% for Hardness only model and 6% for Strength only model)
- Overall conclusion: all variables needed in model according to both 10-fold cross-validation and adjusted-R^2 statistics

- NOTE: **no assumptions about the distribution of the errors have been required**

Comparsion of Nested Models

- Two models are nested if the set of variables included in the simpler model (**restricted model**) are a subset of those used in more complex model (**unrestricted model**)
- An ANalysis Of the VAriance (ANOVA) explained by adding the extra terms into the model enables a statistical test as to whether they significantly improve the model fit
- Consider a restricted model ($\beta_{k+1} = \beta_{k+2} = \ldots = \beta_p = 0$):

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \epsilon. \tag{24}$$

- with $(k+1)$ terms, and it's unrestricted form:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \beta_{k+1} x_{k+1} + \ldots + \beta_p x_p + \epsilon \tag{25}$$

- with $(p+1)$ terms, where $p > k$

Hypothesis test for Nested Models

- Wish to test the null hypothesis:

$$H_0 : \beta_{k+1} = \beta_{k+2} = \ldots = \beta_p = 0,$$

- against the alternative hypothesis:

  H_1 : At least one of the coefficients $\beta_{k+1}, \beta_{k+2}, \ldots, \beta_p$ is nonzero.

- It is possible to apply bootstrapping (i.e. using confidence intervals or p-values) to evaluate the evidence to reject the null hypothesis based on an appropriate statistic
- **Unfortunately, applying this in a stepwise algorithm for model selection is challenging to implement in MATLAB**
- So we will revert to traditional testing ideas under normality assumption for errors

F-test Statistic for Nested Models

- Denote the Residual Sum of Squares as $RSS = \sum(y_i - \hat{y}_i)^2$
- The test statistic is given by:

$$F = \frac{\text{Explained Variance}}{\text{Unexplained Variance}} = \frac{(RSS_{k+1} - RSS_{p+1})/(p-k)}{RSS_{p+1}/[n-(p+1)]}$$

where:

$$\begin{aligned}
RSS_{p+1} &= \text{RSS of unrestricted model} \\
RSS_{k+1} &= \text{RSS of the restricted model} \\
n &= \text{sample size} \\
p+1 &= \#\text{ of terms in unrestricted model} \\
p-k &= \#\text{ of terms constrained to zero in restricted model.}
\end{aligned}$$

F-test Statistic Properties

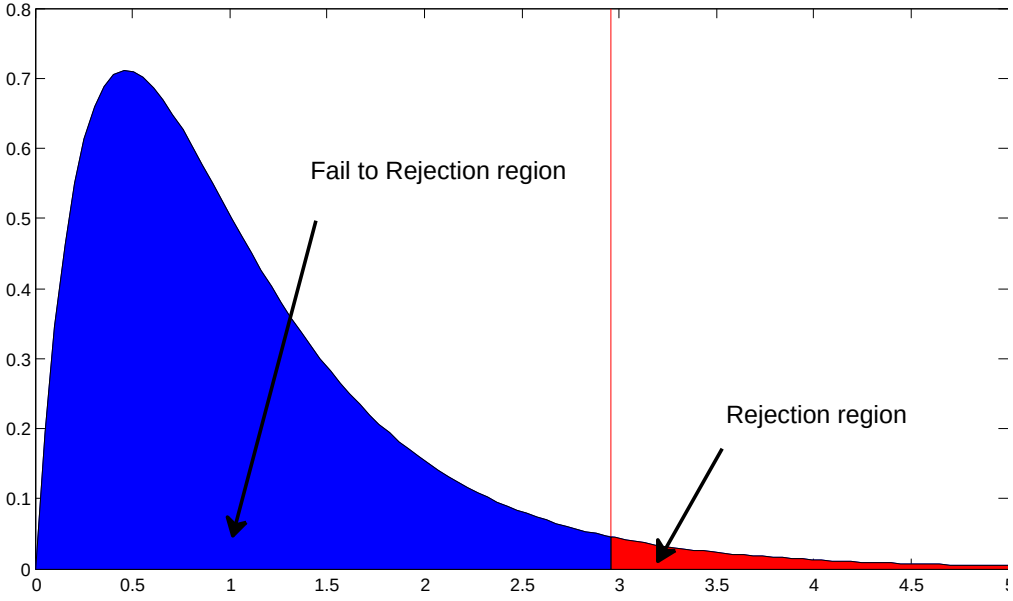
- Essentially, the F-test is ratio of the additional variance explained by the extra terms in the more complex model (accounting for the degrees of freedom used up when including them) to that left over in the residuals:
  - If the extra variables are useful for prediction then they will "**explain more variability on average**" than in the random errors (i.e. F statistic will be large)
  - If the extra variables are not useful for prediction the they will "**explain little or the same variability on average**" than in random errors (i.e. F statistic will be small)
- Notice that the denominator in the F-test statistic is just the estimated error variance for unrestricted model:

$$\text{Unexplained Variance} = \hat{\sigma}^2_{p+1} = \frac{RSS_{p+1}}{n-(p+1)}$$

F-test Statistic Under Normal Assumption

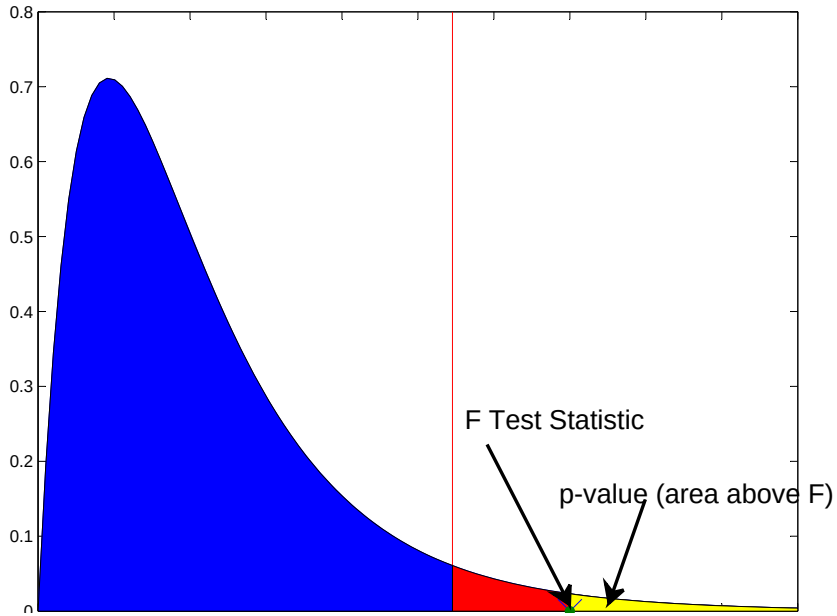
- If we assume the errors in the regression model are normally distributed, in addition to satisfying the usual OLS error assumptions
- Then the F-test statistic is known to follow a particular distribution **under the null hypothesis**:
  - F distribution on $v_1 = p - k$ numerator degrees of freedom and $v_2 = n - (p + 1)$ denominator degrees of freedom
  - The null hypothesis is rejected if the F statistic is within the rejection region $F > F_{v_1, v_2}(\alpha)$ for a $100(1 - \alpha)\%$ significance test
  - Or equivalently, we reject the null hypothesis if the p-value for the F statistics is smaller than the significance level
- See wikipedia for formal definition of F-distribution - not expected for this course

F-test Distribution and Rejection Region



- If F statistic calculated for two models is in red region where $F > F_{v_1,v_2}(\alpha)$ then "there is sufficient evidence at the α% significance level to reject the null hypothesis of zero coefficients (i.e. extra terms should be included as they significantly improve model fit)
- If F statistic is in blue region where $F < F_{v_1,v_2}(\alpha)$ then "there is insufficient evidence at the α% significance level to reject the null hypothesis of zero coefficients (i.e. extra terms should be left out as they do not significantly improve model fit)

F-test Distribution and p-value



- p-value is probability of getting the particular F test statistic or something more unusual under the null hypothesis (yellow region)
- Reject null hypothesis if p-value for F-test statistic is smaller than α and fail to reject otherwise
- Using p-value or rejection region ideas are completely equivalent

General Model Descriptors

- Before discussing variable selection algorithms it is useful to know a few descriptors used for very large models:

Maximal Model	Contains all explanatory variables that may be of interest Most complicated model to consider Many terms likely to be insignificant
Minimal Adequate Model	Likely a simplified model with less terms than the maximal All terms significantly improve the model fit
Null Model	A single parameter, i.e. intercept only Equivalent to having using mean $y = \bar{y}$ Usually, not a good fit and no explanatory power

- Maximal and null models are useful benchmarks with which to judge the performance of others
- Note: in general there isn't a unique minimal adequate model

Variable Selection Algorithms

- **Aim of model variable selection** is to find the **minimal adequate model** in an efficient manner
- Essentially, these algorithms try to avoid fitting all 2^p possible models, by building models (or deconstructing them!) in a hierarchical fashion (one term at a time)
- There are many statistical procedures to accomplish this, but here we will consider the three most commonly used:
- Namely: **forward, backward and stepwise selection**
- We will also discuss some of the key pitfalls/dangers with using these algorithms (they are very controversial)[3]
- **Main piece of advice: NEVER solely rely on these algorithms and always validate the chosen model with subject matter expertise**

[3] See Wikipedia on "stepwise regression"

Forward Selection

- Forward selection algorithm:
  1. Start with the **null model** (first restricted model)
  2. Create list of all **new potential explanatory variables**
  3. For each of these variables in turn, add only this variable to give a **new unrestricted model**
  4. Evaluate the **performance gain** of each new unrestricted model separately using the above F-test statistic (or similar performance statistic)
     - 5a If all the F-tests are insignificant (e.g. $p > 0.05$) then retain the restricted model and STOP, this gives the **minimal adequate model**
     - 5b Otherwise, add the single variable that has the largest F-test statistic and goto step 6
  6. Start the process again at step 2, with the significant variable from step 5 added to the restricted model
- No terms are dropped from the model during forward model selection

Backward Selection

- Backward selection algorithm:
  1. Start with the **maximal model** (first unrestricted model)
  2. Create list of all **potentially removable explanatory variables**
  3. For each of these variables in turn, remove only this variable to give a **new restricted model**
  4. Evaluate the **performance drop** of each new restricted model separately using the above F-test statistic (or similar performance statistic)
     - 5a If all the F-tests are significant (e.g. $p < 0.05$) then retain the unrestricted model and STOP, this gives the **minimal adequate model**
     - 5b Otherwise, add the single variable that has the smallest F-test statistic and goto step 6
  6. Start the process again at step 2, with the least useful variable from step 5 removed from the unrestricted model
- No terms are added to the model during backward model selection

Stepwise Selection

- Stepwise selection is essentially a combination of the forward and backward procedures.
- Stepwise is essentially forward selection, but each "**entry step**" is immediately followed by a "**deletion step**"
- Deletion step re-evaluates variables entered at previous steps
- If all the variables in the current model significantly improve the fit, then none are deleted
- The procedure is stopped when no new variables can significantly improve the model fit
- The reason for considering a **Deletion Step** is that a variable that may have been useful at an early stage in the model build may be superfluous at a later stage after further variables have entered (may be duplication of explanatory information)
- Generally, the significance levels for the entry and deletion steps are the same
- Condition to prevent infinite loops are needed

General Comments

- Don't use selection algorithms unless you have to!
- Stepwise selection is generally preferred
- Worthwhile trying all approaches you have available
- If considering a polynomial model of order p, then usually best to retain the terms of lower order
- Further, if the investigator knows that a certain variable is physically important for prediction then generally this variable should be included in the model (even if the F-test indicates it is not important)
- Physical relevance generally overrides statistical significance
- Clearly, a lot of hypothesis tests are being carried out in variable selection algorithms (called "**multiple testing**" problem)
- Therefore, there is a very high probability of making at least one Type I error (including some irrelevant variables) or Type II error (not including important variables)

Further General Comments

Some of the problems attributed with selection algorithms:

- Generally model selection is ignored, so multiple testing effects and degrees of freedom used are ignored;
- Essentially there will be more uncertainty in our model, than suggested if we ignore model selection stage
- Performance measures like R^2 values will be biased to be high;
- Confidence intervals for coefficients and predictions are falsely narrow;
- They have severe problems in the presence of multicollinearity (we'll come to that next)

General advice: always validate the physical validity of the chosen model using subject matter expertise

Multicollinearity in Regression

- In the ideal world all explanatory variables would be uncorrelated with each other (each one then spans an orthogonal dimension in the explanatory vector space)
- Then it is possible to interpret each coefficient on their own, whilst keeping all the other variables fixed
- In real world applications this is rarely the case and we have some level of "**multicollinearity**" between the explanatory variables (see correlation between Hardness and Tensile Strength in Abrasion example)
- Essentially there is an overlap in the linear information content of two or more variables (i.e. two or more explanatory vectors span similar dimensions)
- The presence of multi-collinearity complicates interpretation, analysis and fitting of models

Multicollinearity in Regression

- Firstly, it is impossible to disentangle the relative contributions of effects of collinear variables on the response variable
- Therefore, the coefficients for collinear terms will be related and cannot be interpreted on their own
- Common to see inappropriate signs of coefficients (i.e. suggesting effect of explanatory variable is opposite to reality)
- Uncertainty estimates (e.g. confidence intervals) are much higher for collinear variables, as effects cannot be disentangled they are very uncertain
- Dependence causes problems for model selection algorithms
- Extremely highly collinear variables can make design matrix ill-conditioned (due to linear dependence in columns) making evaluation of matrix inverse $(\mathbf{X}'\mathbf{X})^{-1}$ numerically unstable

Multicollinearity - The Good News

- Despite all these problems there is some good news...
- Provided you are only interested in predictions, and not trying to interpret individual coefficients in model, then you don't have to do anything (as long as matrix inverse $(\mathbf{X}'\mathbf{X})^{-1}$ exists!) as the predictions are unaffected by multicollinearity

Multicollinearity - How to Avoid or Ameliorate?

- Multicollinearity is sometimes avoided by screening variables before modelling commences using subject matter expertise
- Generally, drop the least important collinear variable
- Mean correction of the explanatory variables (particular power terms, $x, x^2, x^3, \ldots$) can substantially aid in the numerical conditioning
- There are various approaches to avoid or ameliorate the effects of multicollinearity, e.g. ridge regression or principal component analysis, but these are beyond the scope of this course
- If the collinearity is between just two variables, then this can be highlighted using the correlation matrix between all the explanatory variables
- But if the collinearity is between three or more variables then alternative statistics, e.g. variation inflation factors (VIF), can be used but these are beyond the scope of this course[4]

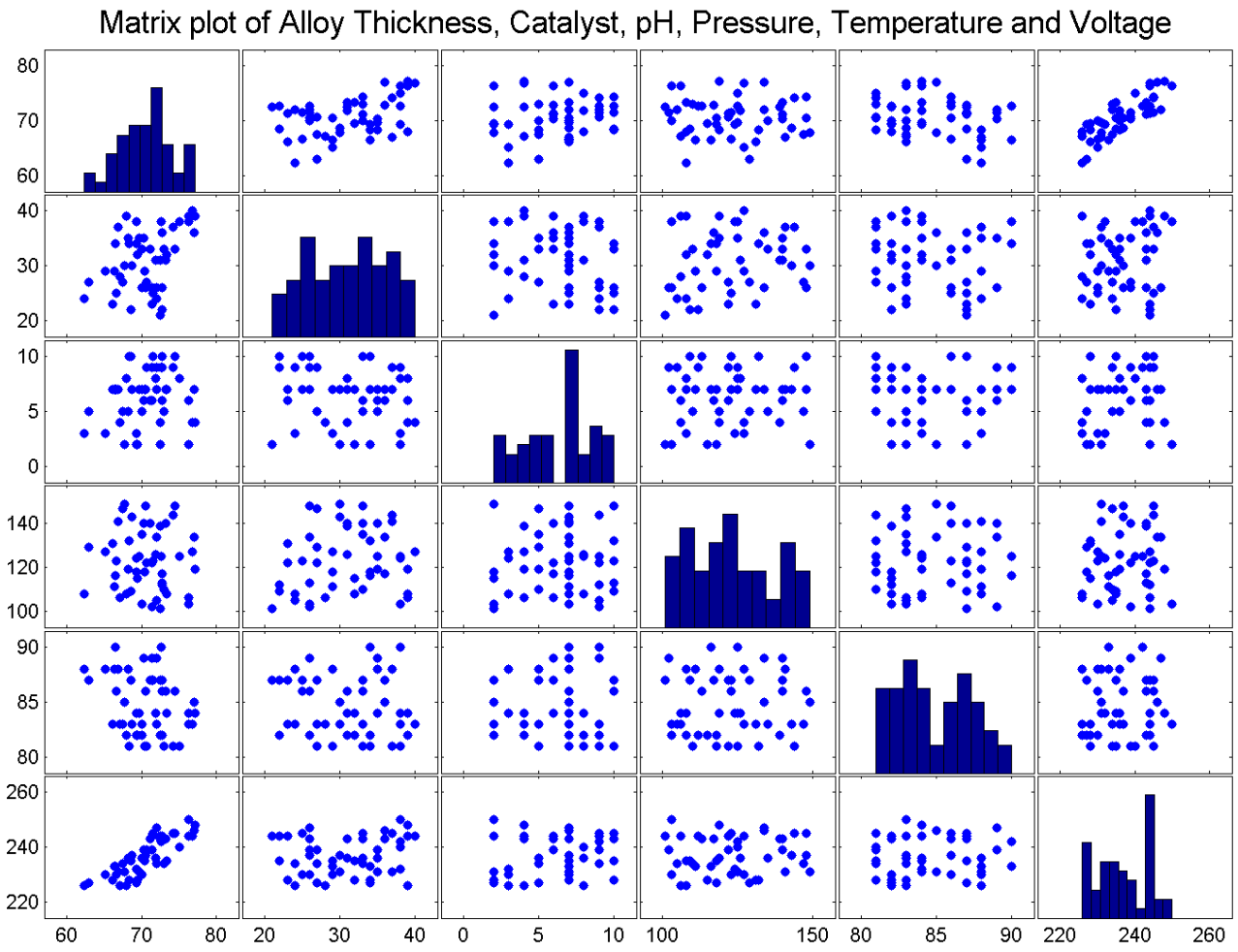
[4] See Wikipedia on "Multicollinearity"

Example: Alloy Thickness Data

- Article investigating metal deposition using electroplating: Conklin (June, 2009) *3.4 per million: It's a Marathon, Not a Sprint*, Quality Process Journal
- An alloy is layered onto a metal substrate in an acid bath
- For simplicity, assume a single layer is applied
- The key output (response variable) is thickness of the deposit. A minimum thickness is required to ensure the performance.
- A team of engineers and technicians is studying the process, with the goals of reducing variation and fine tuning the key input levels for best effect.
- The key inputs (explanatory variables) are:
  - Catalyst - acid bath catalyst concentration;
  - pH - acid bath pH level;
  - Pressure - pressure in the acid bath tank;
  - Temperature - temperature in the acid bath; and
  - Voltage - voltage applied.

Example: Alloy Thickness Matrix Plot

```
>> plotmatrix(data)
>> title('Matrix plot of Alloy Thickness, Catalyst, pH,
   Pressure, Temperature and Voltage')
```



Example: Alloy Thickness Correlation Matrix

```
>> corr(data)

    1.0000    0.3935    0.1468   -0.0407   -0.2522    0.8201
    0.3935    1.0000   -0.1594    0.1671   -0.1545    0.1567
    0.1468   -0.1594    1.0000    0.0639    0.0486    0.1853
   -0.0407    0.1671    0.0639    1.0000    0.0412    0.1203
   -0.2522   -0.1545    0.0486    0.0412    1.0000    0.1276
    0.8201    0.1567    0.1853    0.1203    0.1276    1.0000
```

Example: Alloy Thickness Subjective Impression

- There is a strong positive linear association between `Voltage` and `Thickness`
- There is a moderate positive linear association between `Catalyst` concentration and `Thickness`
- There is a weak (positive/negative) linear association between `pH/Temperature` and `Thickness`
- There is essentially no association between `Pressure` and `Thickness`
  - There is very little association between the explanatory variables
  - So we expect no issues associated with multicollinearity
  - Sample data provide good coverage across range of sensible values
  - Observation study - data collected whilst in operation, so explanatory variable values are not fixed
  - Hence observation resampling relevant bootstrap approach here

Example: Alloy Thickness Stepwise Selection

- Easy to do stepwise selection in MATLAB

```
>> [B,SE,PVAL,INMODEL,STATS,NEXTSTEP,HISTORY]=stepwisefit(data(:,2:end),data(:,1))
```

- Note: `data(:,2:end)` matrix has only explanatory variables, not columns of ones for intercept
- `data(:,1)` is response data vector
- There are lots of options for outputs and inputs, see `doc stepwisefit`
- Commonly used input options are:
  - `penter` - significance level for entry step
  - `premove` - significance level for deletion step
- which are set to 5% below:

```
>> [B,SE,PVAL,INMODEL,STATS,NEXTSTEP,HISTORY]=stepwisefit(data(:,2:end),data(:,1),
                                                        'penter',0.05,'premove',0.05);
```

Example: Alloy Thickness Stepwise MATLAB Output 1

- Default display output is then:

```
Initial columns included:  none
Step 1, added column 5, p=3.19733e-013
Step 2, added column 4, p=1.3158e-006
Step 3, added column 1, p=0.000830253
Step 4, added column 3, p=0.00315094
Final columns included:  1 3 4 5
    'Coeff'      'Std.Err.'    'Status'    'P'
    [ 0.1548]    [  0.0356]    'In'        [7.7744e-005]
    [ 0.0864]    [  0.0800]    'Out'       [     0.2862]
    [-0.0420]    [  0.0135]    'In'        [     0.0032]
    [-0.4036]    [  0.0698]    'In'        [6.6247e-007]
    [ 0.4288]    [  0.0278]    'In'        [1.4798e-019]
```

- Step 0: Start with null model (intercept/mean only)
- Step 1 Entry: Column 5 (`Voltage`) entered with $p = 3.2e - 13 < 5\%$
- Step 2 Entry: Column 4 (`Temperature`) entered with $p = 1.3e - 6 < 5\%$
- Step 2 Deletion: None have $p > 5\%$
- Step 3 Entry: Column 1 (`Catalyst`) entered with $p = 0.00083 < 5\%$
- Step 3 Deletion: None have $p > 5\%$

Example: Alloy Thickness Stepwise MATLAB Output 2

- Default display output is then:

```
Initial columns included:  none
Step 1, added column 5, p=3.19733e-013
Step 2, added column 4, p=1.3158e-006
Step 3, added column 1, p=0.000830253
Step 4, added column 3, p=0.00315094
Final columns included:  1 3 4 5
    'Coeff'       'Std.Err.'     'Status'     'P'
    [ 0.1548]     [  0.0356]     'In'         [7.7744e-005]
    [ 0.0864]     [  0.0800]     'Out'        [     0.2862]
    [-0.0420]     [  0.0135]     'In'         [     0.0032]
    [-0.4036]     [  0.0698]     'In'         [6.6247e-007]
    [ 0.4288]     [  0.0278]     'In'         [1.4798e-019]
```

- Step 4 Entry: Column 3 (`Pressure`) entered with $p = 0.0032 < 5\%$
- Step 4 Deletion: None have $p > 5\%$
- Step 5 Entry: Remaining term has $p = 0.286 > 5\%$ so "**minimal adequate model**" found at Step 4:

$$y = \beta_0 + \beta_1 \mathit{Voltage} + \beta_2 \mathit{Temperature} + \beta_3 \mathit{Catalyst} + \beta_4 \mathit{Pressure} + \epsilon$$

Example: Alloy Thickness Stepwise MATLAB Output 3

- Default display output is then:

```
Initial columns included:  none
Step 1, added column 5, p=3.19733e-013
Step 2, added column 4, p=1.3158e-006
Step 3, added column 1, p=0.000830253
Step 4, added column 3, p=0.00315094
Final columns included:  1 3 4 5
    'Coeff'      'Std.Err.'    'Status'    'P'
    [ 0.1548]    [  0.0356]    'In'        [7.7744e-005]
    [ 0.0864]    [  0.0800]    'Out'       [     0.2862]
    [-0.0420]    [  0.0135]    'In'        [     0.0032]
    [-0.4036]    [  0.0698]    'In'        [6.6247e-007]
    [ 0.4288]    [  0.0278]    'In'        [1.4798e-019]
```

- Lower table gives summary of minimal adequate model
- First column gives coefficients:

$$y = \beta_0 + 0.4288\,Voltage - 0.4036\,Temperature + 0.1548\,Catalyst - 0.0420\,Pressure + \epsilon$$

- Also gives coefficients if non-included term(s) were entered individually:

$$y = \beta_0 + \beta_1\,Voltage + \beta_2\,Temperature + \beta_3\,Catalyst + \beta_4\,Pressure + 0.0864\,pH + \epsilon$$

- Notice: other coefficients will generally change due to dependence (correlation) between them:

$$\begin{aligned} y &= 4.4543 + 0.4224\,Voltage - 0.4026\,Temperature + 0.1627\,Catalyst \\ &\quad - 0.0431\,Pressure + 0.0864\,pH + \epsilon \end{aligned}$$

Example: Alloy Thickness Stepwise MATLAB Output 4

- Default display output is then:

```
Initial columns included:  none
Step 1, added column 5, p=3.19733e-013
Step 2, added column 4, p=1.3158e-006
Step 3, added column 1, p=0.000830253
Step 4, added column 3, p=0.00315094
Final columns included:  1 3 4 5
    'Coeff'      'Std.Err.'    'Status'    'P'
    [ 0.1548]    [  0.0356]    'In'        [7.7744e-005]
    [ 0.0864]    [  0.0800]    'Out'       [     0.2862]
    [-0.0420]    [  0.0135]    'In'        [     0.0032]
    [-0.4036]    [  0.0698]    'In'        [6.6247e-007]
    [ 0.4288]    [  0.0278]    'In'        [1.4798e-019]
```

- Standard error (square root of variance) given in second column
- Third column gives `In/Out` status
- Last column is p-value for testing null hypothesis $H_0 : \beta_i \leq 0$ (or $\beta_i \geq 0$ whichever is relevant) under final model
- Notice: p-value different to entry/deletion p-value
- Output p-value assumes normal errors, you will calculate this again using bootstrap without normal assumption
- Other outputs (`STATS`, `NEXTSTEP` and `HISTORY`) not relevant in this course

Example: Alloy Thickness Minimal Adequate Model

- Now go back to using `regress` function to fit final model:

```
>> y=data(:,1);
>> X=[ones(size(data,1),1) data(:,[6 5 2 4])];
>> [B,BINT,R,RINT,STATS] = regress(y,X);
```

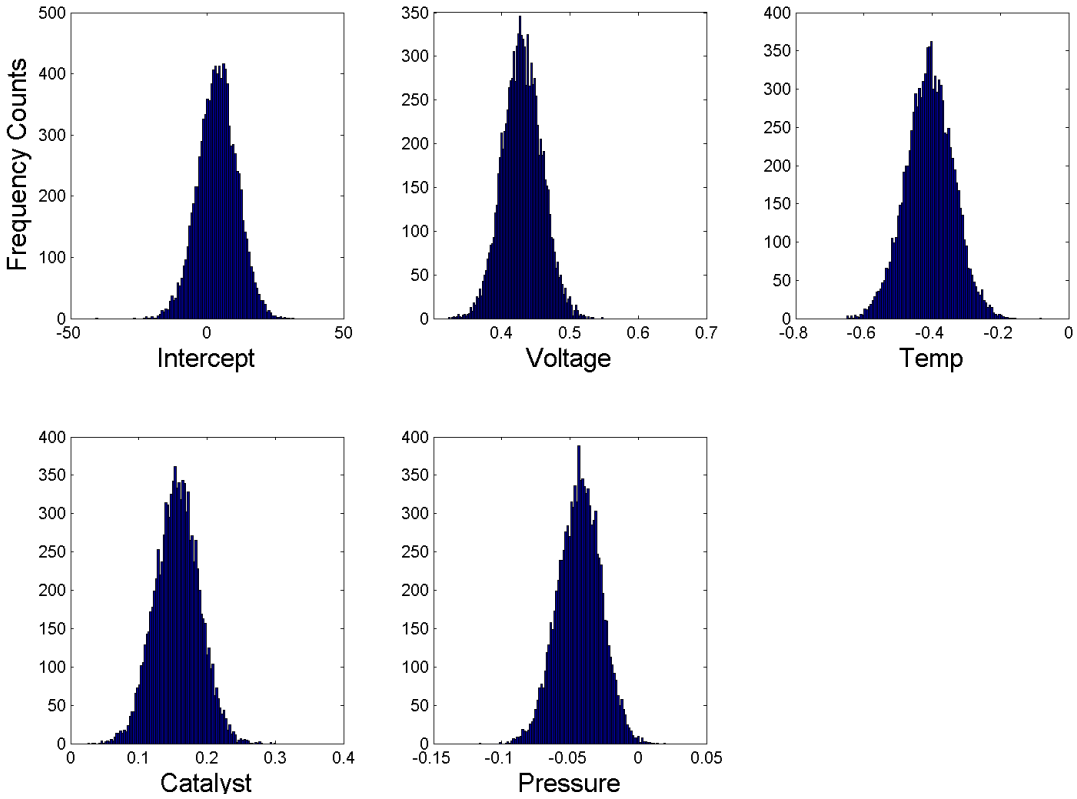
- which gives:

$$y = 3.6833 + 0.4288\,Voltage - 0.4036\,Temperature + 0.1548\,Catalyst - 0.0420\,Pressure + \epsilon$$

- Final $R^2 = 87.3\%$ (compared to $R^2 = 87.6\%$ if pH also included)
- So little drop in performance from ignoring pH

Example: Alloy Thickness Bootstrapped Coefficients

- Now go back to using `regress` function to fit final model:



- It is clear that bootstrap coefficients are well away from zero, so expect p-values (under null hypothesis) to be close to zero

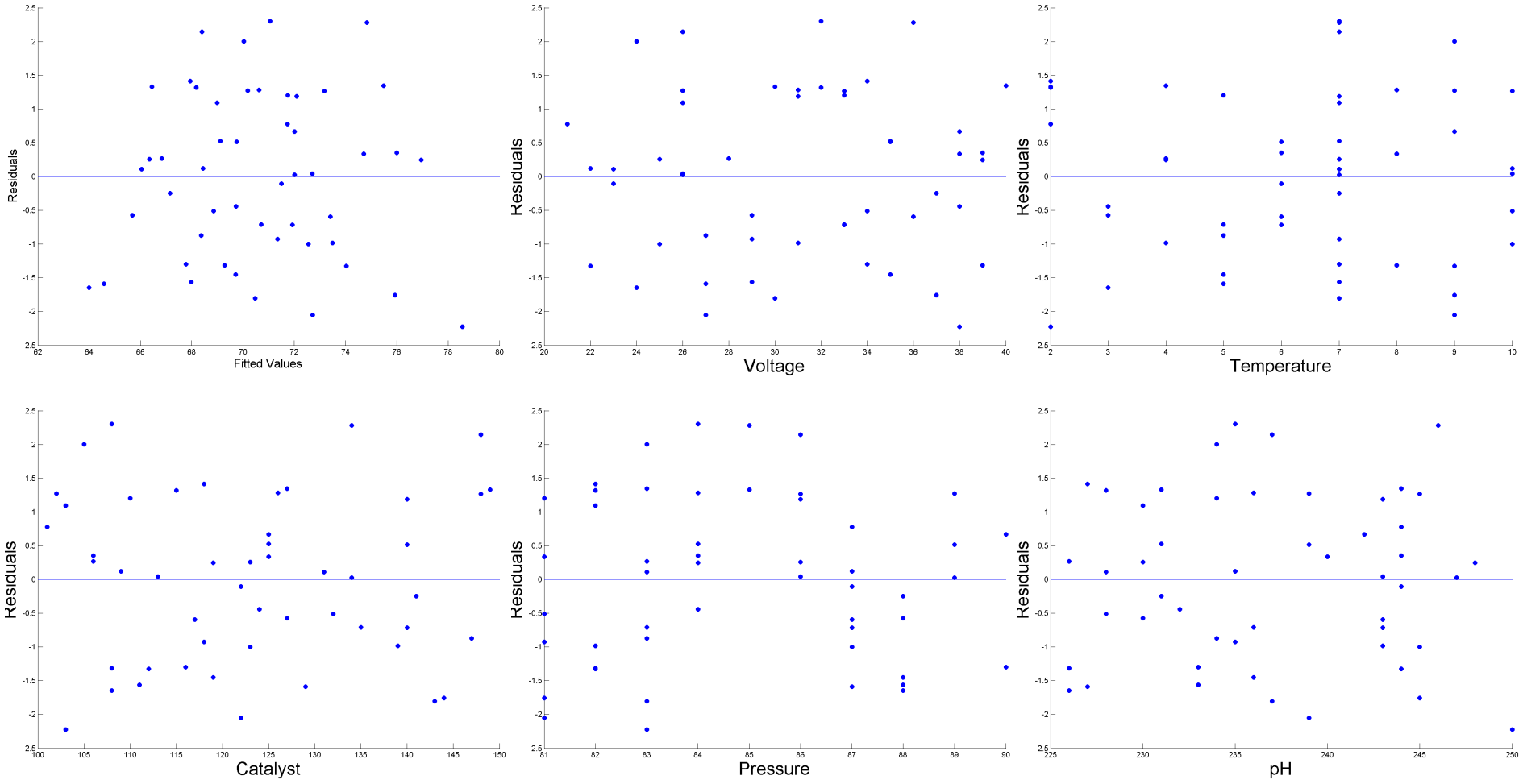
Example: Alloy Thickness Bootstrap p-values

- Calculating the p-values for each of the 5 coefficients:

```
>> ppositive=sum(bootbetas<0)/nsim

ppositive =

    0.2921         0    1.0000         0    0.9977

>> pnegative=sum(bootbetas>0)/nsim

pnegative =

    0.7079    1.0000         0    1.0000    0.0023
```

- So p-values are $p_{intercept} = 0.29$, $p_{voltage} < 1e-5$, $p_{temperature} < 1e-5$, $p_{catalyst} < 1e-5$ and $p_{pressure} = 0.0023$
- As expected the p-values are all less than 5% significance level (else they would have been dropped during Step 4 Deletion)
- Notice these are very similar to last column of `stepwisefit` results, which assume normally distributed errors

Example: Alloy Thickness Regression Diagnostics



- OLS assumptions look fine, except possibly quadratic with `Pressure` may be needed

Example: Alloy Thickness Model Performance

- Adjusted-$R^2 = 86.1\%$ (compared to adjusted-$R^2 = 86.2\%$ if `pH` also included)
- So adjusted-R^2 suggest `pH` could be included, but only very slight improvement
- Leave-one-out cross-validation MSE for final model is 1.89 (compared to 1.88 if `pH` also included)
- What does this mean?
- The choice of F-test statistic for deciding which variables to include in model influences which variables are chosen
- Different model choice statistics can lead to different results
- So try out all available statistics in package (`MATLAB` is limited as it only allows F-test) you are using for model fitting.

Further General Comments

- These model selection algorithms are not failsafe
- There is no guarantee they give "best model" overall
- Always try range of the model selection procedures, and look for consistency in the results and where the differences lie
- **Only use these algorithms when you have to**, i.e. when you have a huge number of explanatory variables
- There are a lot of subjective choices (i.e. test statistic, significance level, algorithm) required in model building, all of which can influence the final results; so be careful!
- Always have in mind the physical sensibility of the model and it's proposed application
- Remember: there is a big difference between physical significance and statistical significance