PSTAT 126: Regression Analysis

Department of Statistics and Applied Probability

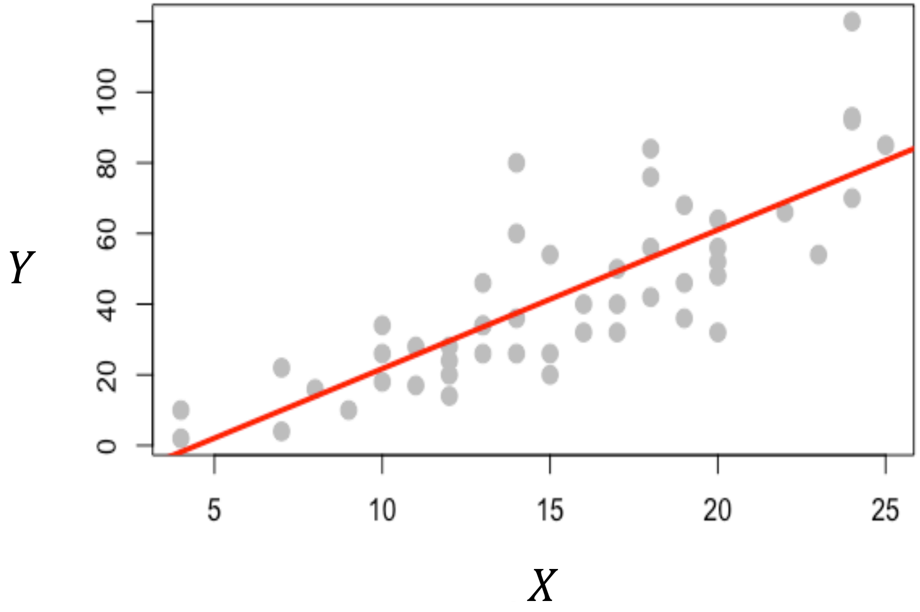
University of California, Santa Barbara

Regression and its Applications

There are many areas of human endeavor in which we would like to learn and model, from relevant but noisy data, an unknown functional relationship between a variable *X* (or variables) and a variable *Y*, the values of which we think of as dependent, in some sense, on those of *X*. The ability to do this has key applications in such areas as, among others:

- Science & Medicine
- Technology & Industry
- Economics & Finance
- Sociology & Behavioral Sciences
- Public Policy

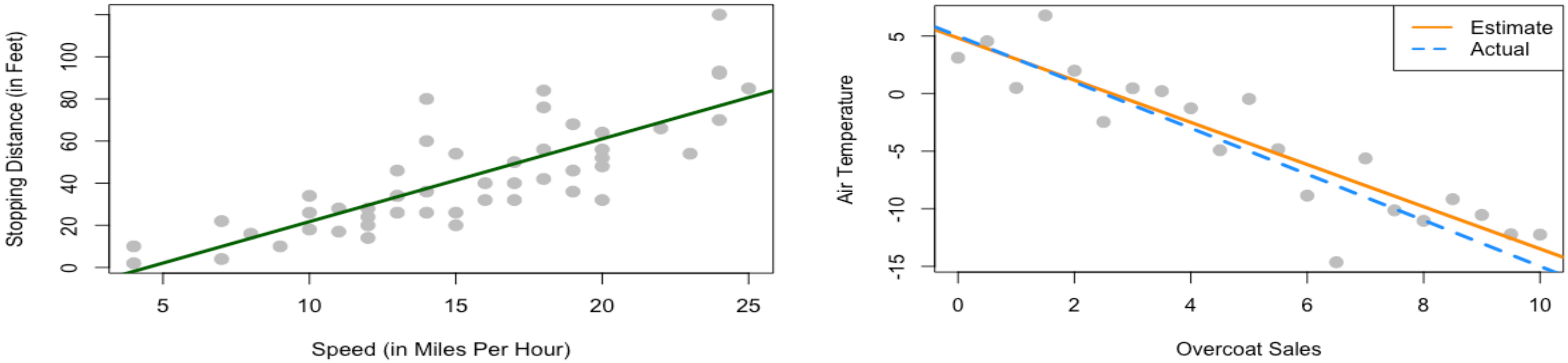
The study of how best to do this, including which mathematical and statistical methods and algorithms to use, is the subject of **Regression**.



Uses of Regression: Explanation and Insight

Explanation and insight:
Modeling the relationship between an input/inputs and an outcome, given observed, sampled data, in order to gain deeper understanding into that relationship.

What is the functional relationship between the stopping distance of a car (that is, the safe stopping distance, without the driver's loss of control) and the car's speed?



The graphic shows an example of linear regression – regression for which the functional relationship between X and Y is, or is presumed to be, linear in an appropriate sense.

Uses of Regression: Prediction

Prediction:
Given a new input value, not previously sampled, estimate the corresponding outcome/output value using the trained regression model.

Given one’s high school and/or college GPA, can SAT and/or GRE scores be predicted?

History of Regression

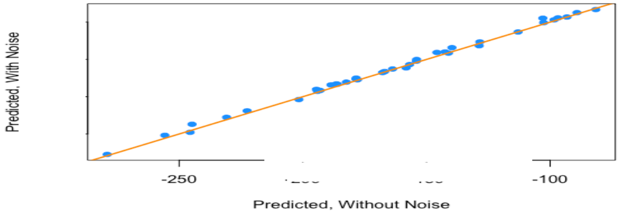
- The mathematicians Legendre (1805) and Gauss (1809) were the first known to have used the technique of statistical regression (that is, the method of least squares) as such, in order to find the best linear fit to a finite set of data points.
- They applied the method to analyze and predict planetary motion.
- Using the normal (or Gaussian) distribution to describe the behavior of errors, Gauss also developed a formula for this distribution, which plays such an important role in modeling errors in (linear) regression.
- Techniques for Linear Regression can rightly be viewed as Artificial Intelligence/Machine Learning methods and indeed as, historically speaking, perhaps the original versions of the types of Machine Learning algorithms so widely used today.

NASA / Johnson Space Center

Goals of Regression

Let P be any population. This population could be virtually any set of objects of interest, including people, cities, companies, biological cells, or stars in the night sky, for example. For any given population, we may be interested in the relationship between two variables of interest, a so-called predictor variable X -- also called the explanatory or independent variable -- and a response variable Y (also known as the dependent or target variable). For example, X and Y could be the respective

- Height and weight of people in P
- Distance from Earth of a set of stars and their corresponding brightness
- Education level and average income in the population of a given city.



In order to understand and explain the interaction between X and Y, which we think of as random variables, we would like to find an approximate functional relationship $f(X) \approx Y$ between them. Note that, for us, the function f we will attempt to learn will be assumed deterministic (non-random), and we will have

$$Y = f(X) + \epsilon,$$

with the noise term ϵ, also a random variable and such that the conditional expectation

$\boldsymbol{E}[\epsilon | X = x] = 0$, representing random error or variation in the model.

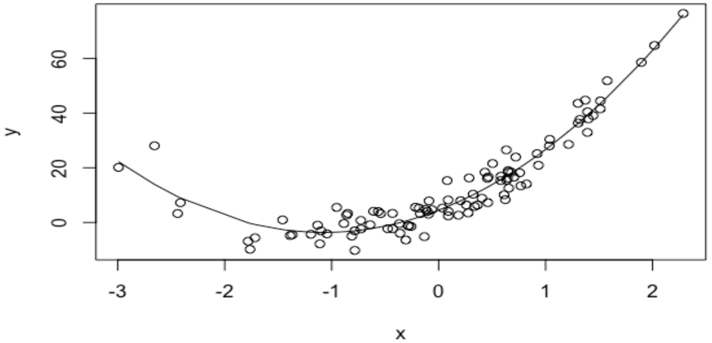
- We are essentially always interested in determining f in the context of regression models, but interest in determining more about the random error ϵ may depend on context.
- Indeed, for the purpose of explanation and insight concerning the relationship between X and Y, more information about the nature of ϵ, including its variance, may be of significant interest, whereas, when applying the model expressly for prediction, additional information about ϵ may be of less value.

Regression and the Mean Function

To determine a functional relationship between predictor X and response Y, our goal is to learn the conditional expectation function $\boldsymbol{E}[Y|X]$ – or, at least, a reasonably close approximation of it. We call $\boldsymbol{E}[Y|X]$ the **regression** or **mean function**.

Why is the mean function $\boldsymbol{E}[Y|X]$ so important here?

It clearly gives you the mean value of Y given $X = x$. But we can go further than this. We want to find a a function minimizing the difference between $f(X)$ and Y, on average. So, this would suggest looking at the absolute value of the difference $f(X) - Y$, i.e., $|f(X) - Y|$, and then considering the mean or expectation $\boldsymbol{E}[|f(X) - Y|]$. However, in part because the absolute value function is not smooth as it is not differentiable at 0 (spaces of functions defined by the square having other nice mathematical properties as well), it is more convenient to consider $\boldsymbol{E}[(f(X) - Y)^2]$.



$\boldsymbol{E}[Y|X]$ is the function that minimizes this squared error among all candidate functions f.
In fact it can be shown that

$$\boldsymbol{E}[(f(X) - Y)^2] = \boldsymbol{E}[(f(X) - \boldsymbol{E}[Y|X])^2] + \boldsymbol{E}[(Y - \boldsymbol{E}[Y|X])^2], \qquad (1)$$

for any candidate function f, where $\boldsymbol{E}[(Y - \boldsymbol{E}[Y|X])^2]$ depends on X and Y but not f. Equation (1) holds whether X is a scalar or vector-valued variable. Equation (1) says that that, for any function f, the expectation of the square of the difference between $f(X)$ and Y is equal to the expectation of the square of the difference between f and the mean function (plus a nonnegative constant, as shown in (1)).

- Since $(f(X) - Y)^2 \geq 0$ for any function f we can minimize the magnitude of the error of approximating f by Y on the left-hand side of (1) by in fact taking $f(X) = \boldsymbol{E}[Y|X]$.
- This means that the function of X=x that approximates the behavior of the response Y with the smallest error on average is in fact the mean $\boldsymbol{E}[Y|X]$ function itself.
- So, it is the mean function which gives us the ”best” representation of the functional relationship between X and Y in the sense described.

Hence, it is the mean function $\boldsymbol{E}[Y|X]$ that we would like to use regression methods and algorithms to determine or at least closely approximate in order to identify and understand any functional relationship between X and Y.

Linear Regression

Our goal in this course is to study specifically **Linear Regression**, which is regression for which $\boldsymbol{E}[Y|X]$ is or may be presumed to be closely approximated by a linear function (i.e., more technically, a function selected from a finite-dimensional, linear space of candidate functions).

The linear case is of great interest because

- from the point-of-view of mathematical structure, it is relatively simple (shades of Occam's razor)
- it robustly describes many situations arising in applications
- it is the model base case for investigations into nonlinear regression (indeed, somewhat paradoxically, the linear regression model itself encompasses many seemingly "nonlinear" cases as well, as we shall see).

So, for the first part of the course we will be considering models of the relatively simple form

$$\boldsymbol{E}[Y|X = x]=\beta_0 + \beta_1 x, \qquad (2)$$

where x is a fixed, scalar value (real number), and Y is a scalar-valued continuous random variable. The numbers β_0, β_1 are parameters which, as we shall see, it is the goal of canonical regression algorithms to compute. When the regression function can be represented as in (2) it is called **Simple Linear Regression** (see the next slide) because only one predictor variable is involved and the predictor appears within a linear term only. Later on, we will augment this framework by adding additional predictor variable terms on the right in (2). This is called **Multiple Linear Regression.** Note that any representation of the function $\boldsymbol{E}[Y|X]$ in the form as on the RHS of (2) will be unique for either simple -- or multiple – regression (at least for the kinds of typical continuous probability distributions we are interested in in this course).

Simple Linear Regression (SLR) Model

But what are the methods of regression that enable us to determine the parameters β_0 and β_1 (or close approximations of these parameters), given that in general we have no ready or direct access to the actual values of the function $\boldsymbol{E}[Y|X]$?

The answer of course involves sampling. For this, let $x_1, x_2,\ldots,x_N$ be N given fixed, real numbers. We could think of these numbers as sampled from the predictor X, but, in keeping with what seems to be fairly standard expository practice in textbooks on basic regression, we usually downplay or suppress the explicit role of the underlying variable $X.$ Now, given these N values $x_n, n = 1, \ldots, N$, write

$$Y_n = \beta_0 + \beta_1 x_n + \epsilon_n, n = 1, \ldots, N, \qquad (3)$$

$$\epsilon_n \sim N(0, \sigma^2), n = 1, \ldots, N. \qquad (4)$$

Here, the ϵ_n are N independent, real-valued, normally-distributed random variables (i.i.d.), with $N(0, \sigma^2)$ being the normal (Gaussian) distribution with mean 0 and variance σ^2 . The ϵ_n represent random variation or noise in the model, and we shall have more to say later about our assumptions concerning the ϵ_n. We call (3)-(4) are our **Simple Linear Regression (SLR) Model**. The goal of regression is it to identify the scalar parameters β_0 and β_1 and also, often, σ as well, or, more commonly, close approximations of these three parameters. The SLR model above in (3)-(4) is the formal model we will now generally work with until we get to Multiple Linear Regression.

In (3)-(4) we assume, as already noted, that each x_n is a known constant (say the outcome of an experiment after the nth trial). So, for each n, we actually can write

$$\boldsymbol{E}[Y_n] = \boldsymbol{E}[Y_n | X = x_n] = \beta_0 + \beta_1 x_n. \qquad (5)$$

Simple Linear Regression Model (cont'd)

Our SLR model: Given N values $x_n, n = 1, \dots, N$, write

$$Y_n = \beta_0 + \beta_1 x_n + \epsilon_n, n = 1, \dots, N, \qquad (6)$$

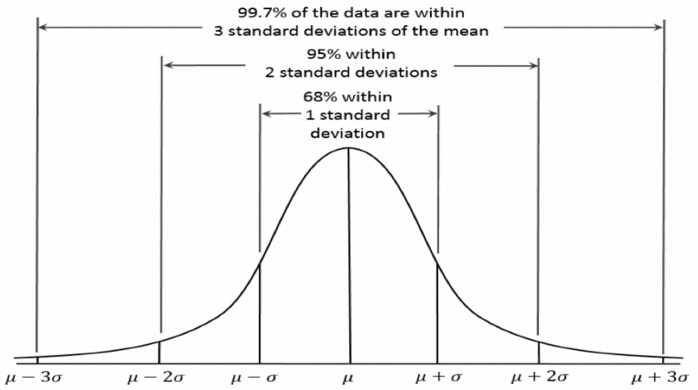
$$\epsilon_n \sim N(0, \sigma^2), n = 1, \dots, N, \qquad (7)$$

the ϵ_n being N independent, normally-distributed random variables (i.i.d.), with $N(0, \sigma^2)$ being the normal distribution with mean 0 and variance σ^2 . So independence of the ϵ_n for us means *mutual independence* so that the corresponding joint and respective individual probability density functions satisfy

$$f_{\epsilon_1, \dots, \epsilon_N}(z_1, \dots, z_N) = f_{\epsilon_1}(z_1) \dots f_{\epsilon_N}(z_N), \qquad (8)$$

where

$$f_{\epsilon_n}(z) = N(0, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{z^2}{2\sigma^2}), \text{ for each } n. \qquad (9)$$



Normal Distribution

The Y_n satisfy similar conditions but with different means. Note that the error ϵ_n is distributed symmetrically about $\boldsymbol{E}[Y_n | X = x_n] = \beta_0 + \beta_1 x_n$. We also note that the i.i.d. assumption is, while a common assumption, a strong assumption and its full strength is not always necessary in the context of regression analysis as we study in this course.

First Steps with R

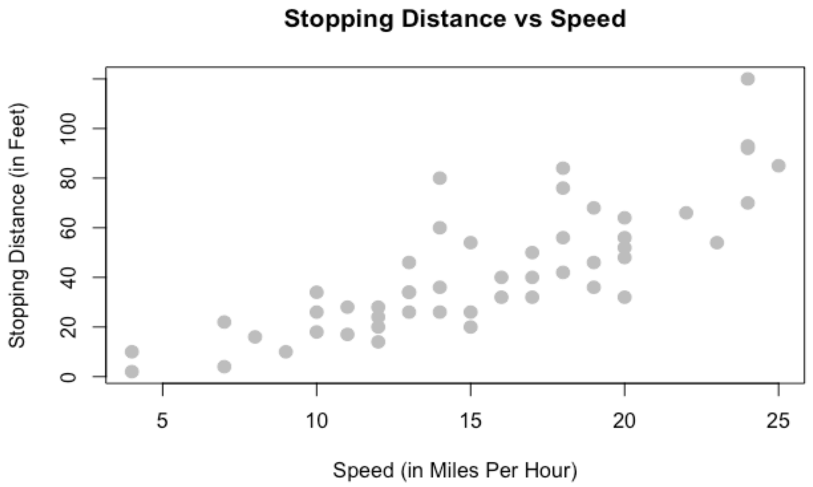
At this point, let’s see how the R language can be applied in the context of an actual data set to generate a simple linear regression model.

R is a language and environment for statistical computing and graphics, an integrated suite of software facilities for data manipulation. R is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S.

R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, etc.) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity.

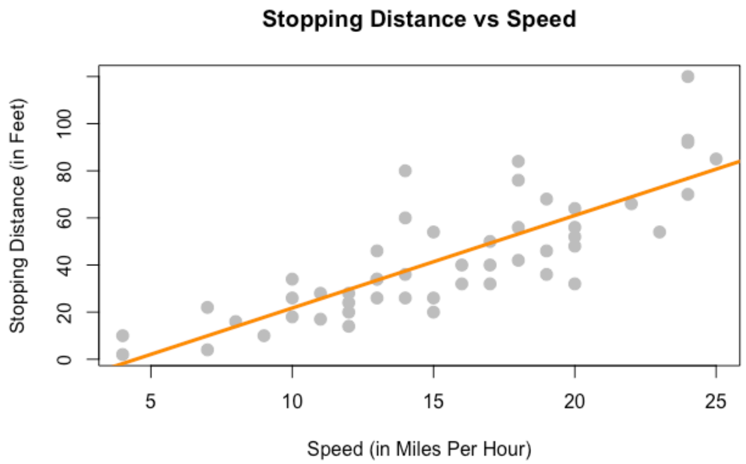
We use the “cars” data set, which is “built-in” to R. It contains data gathered during the 1920s about the speed of cars and the resulting distance it takes for the car to safely come to a stop, without loss of vehicle control.
Thinking of Speed as our predictor variable X and Stopping Distance as our Response Y, we can plot the stopping distance against the speed using the R code below.

```
plot(dist ~ speed, data = cars,
    xlab = "Speed (in Miles Per Hour)",
    ylab = "Stopping Distance (in Feet)",
    main = "Stopping Distance vs Speed",
    pch  = 20,
    cex  = 2,
    col  = "grey")
```



First Steps with R (cont’d)

```
stop_dist_model = lm(dist ~ speed, data = cars)
  stop_dist_model
## Call:
## lm(formula = dist ~ speed, data = cars)
## Coefficients:
## (Intercept)    speed
## -17.579        3.932
```



In order to compute the regression function (regression line) for the cars example we use the lm( ) function in R. The initials stand for ”linear model”, and it will be perhaps our most commonly used R function in this course. We will concern ourselves with how estimates of the model parameters are computed in forthcoming slides, but for now note that R gives

$$\beta_0 = Intercept \approx -17.579$$
$$\beta_1 = Slope \approx 3.932$$

```
plot(dist ~ speed, data = cars,
     xlab = "Speed (in Miles Per Hour)",
     ylab = "Stopping Distance (in Feet)",
     main = "Stopping Distance vs Speed",
     pch  = 20,
     cex  = 2,
     col  = "grey")
abline(stop_dist_model, lwd = 3, col = "darkorange")
```

Lecture 2 Overview

- Some computations with simulated data in R

  (it may be somewhat helpful for this to review the last part of the previous lecture video)
- Method of Least Squares for Simple Linear Regression (SLR)
- Gauss-Markov Theorem
- Behavior of the Mean Function Estimate as $N \to \infty$
- LINE Assumptions for SLR
- The residuals
- Sampling distributions for the SLR regression coefficients

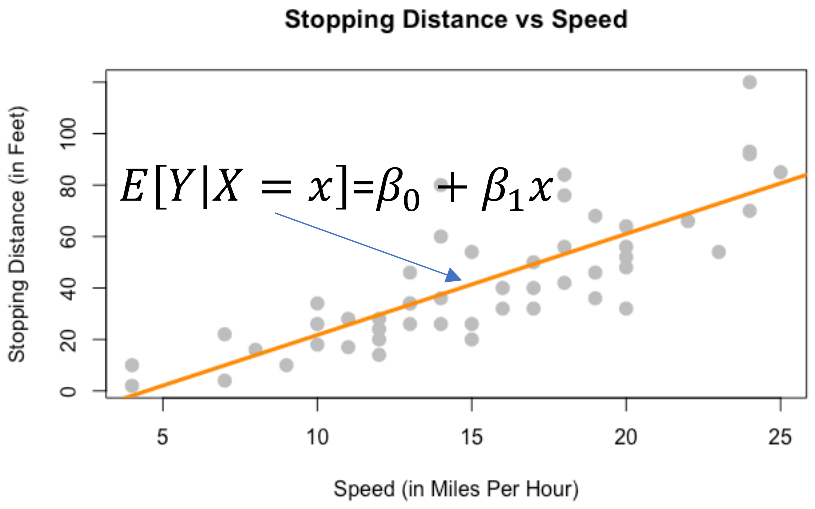
Method of Least-Squares for SLR

How do we approximate the parameters $\boldsymbol{\beta_0}$ and $\boldsymbol{\beta_1}$?

Let $x_1, x_2,\ldots,x_N$ be *N* given fixed values as before. Now, for each *n=1,…,N*, we also sample a random value from the variable Y_n (in (3)-(4) on prior slide) corresponding to n. So denote by

$$(x_1, y_1),\ (x_2, y_3),\ldots,\ (x_N, y_N) \qquad (10)$$

the resulting *N* sample data points (*N* ordered pairs).



$E[Y|X = x]$=$\beta_0 + \beta_1 x$

To compute estimates for the true parameters β_0 and β_1 and solve for the model under the linearity assumption, we use the classic Method of Least Squares:

$$(\hat{\beta}_0, \hat{\beta}_1) = \arg\min_{(\alpha_0,\alpha_1)\epsilon\mathbb{R}^2} \sum_{n=1}^{N}\big(y_n - (\alpha_0 + \alpha_1 x_n)\big)^2 \qquad (11)$$

Numbers $\hat{\beta}_0$ and $\hat{\beta}_1$ minimizing (11) will always exist. Our approximation for the mean function $\boldsymbol{E}[Y|X = x]$=$\beta_0 + \beta_1 x$ is then $\boldsymbol{E}[Y|X = x] \approx \widehat{\boldsymbol{E}}[Y|X = x] = \hat{\beta}_0 + \hat{\beta}_1 x$, assuming we can compute $\hat{\beta}_0$ and $\hat{\beta}_1$ (more on that below).

The minimizers $\hat{\beta}_0$ and $\hat{\beta}_1$ of the function $F(\alpha_0, \alpha_1) = \sum_{n=1}^{N}\big(y_n - (\alpha_0 + \alpha_1 x_n)\big)^2$ in (11) can be determined by computing the partial derivatives of F and setting them equal to 0. The resulting system of linear equations can then be solved for $\hat{\beta}_0$ and $\hat{\beta}_1$. In fact it follows that

$$\hat{\beta}_1 = \frac{\sum_{n=1}^{N}(x_n - \bar{x})(y_n - \bar{y})}{\sum_{n=1}^{N}(x_n - \bar{x})^2} = \frac{S_{xy}}{S_{xx}}, \qquad \hat{\beta}_0 = \frac{1}{N}\left(\sum_{n=1}^{N} y_n - \hat{\beta}_1 \sum_{n=1}^{N} x_n\right), \qquad (12)$$

where $\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n$ and similarly for the *y*-variable.

Gauss-Markov Theorem

Recall our Simple Linear Regression Model. Given *N* values $x_1, x_2,\ldots,x_N$, we have

$$Y_n = \beta_0 + \beta_1 x_n + \epsilon_n, n = 1, \ldots, N, \qquad (13)$$

$$\epsilon_n \sim N(0, \sigma^2), n = 1, \ldots, N, \qquad (14)$$

where the ϵ_n are N independent, normally-distributed random variables, as well as our respective estimates $(\hat{\beta}_0, \hat{\beta}_1)$ for (β_0, β_1). In the previous slide we defined these estimators in terms of the fixed, deterministic samples (x_1, y_1), (x_2, y_3),…, (x_N, y_N) in part in order to make concrete how they can be defined and calculated. However, it can also be useful, in order to assess their performance and behavior, to view the Y_n as random in this context as well (as if they had not yet already been computed). So, using upper case *Y*-values to denote their instantiation as random variables as in (13)-(14), we rewrite (12) in the form

$$\hat{\beta}_1 = \frac{\sum_{n=1}^{N}(x_n - \bar{x})(Y_n - \bar{Y})}{\sum_{n=1}^{N}(x_n - \bar{x})^2}, \qquad \hat{\beta}_0 = \frac{1}{N}\left(\sum_{n=1}^{N} Y_n - \hat{\beta}_1 \sum_{n=1}^{N} x_n\right). \qquad (15)$$

In the setting of our simple linear regression model (13)-(14) above, the **Gauss-Markov Theorem** then asserts that

(1) The respective estimators $\hat{\beta}_0$ and $\hat{\beta}_1$for the regression parameters β_0 and β_1 are unbiased, i.e. $\boldsymbol{E}[\hat{\beta}_0] = \beta_0$ and $\boldsymbol{E}[\hat{\beta}_1] = \beta_1$.

(2) $\hat{\beta}_0$ and $\hat{\beta}_1$are of minimum variance among all unbiased, linear estimators for β_0, β_1 , respectively. This implies that, among all unbiased, linear estimators α_0, α_1, the error $\boldsymbol{E}[(\alpha_i - \beta_i)^2] = \boldsymbol{E}[(\alpha_i - \boldsymbol{E}[\alpha_i])^2], i = 1,2,$ is minimized when $(\alpha_1, \alpha_2) = (\hat{\beta}_0, \hat{\beta}_1)$.

This shows that the respective estimates $\hat{\beta}_0, \hat{\beta}_1$ are in an important sense the optimal ones for a fixed number *N* of samples.

Note that a linear estimator in this context means that both $\hat{\beta}_0$ and $\hat{\beta}_1$can be written as finite, linear combinations of the Y_n(that is, in this context, that we can write $\hat{\beta}_i = \sum_{n=1}^{N} k_{in} Y_n, i = 1,2,$ for some constant coefficients k_{in} -- which follows from (15) since the x_n are assumed to be fixed, constant values).

Behavior of the Mean Function Estimate as $N \to \infty$

But why should the solution of the SLR least-squares minimization problem (Equ. (11) in a previous slide) – an optimization problem that after all only involves minimizing over a finite number of discrete points, however many, give a good estimate of the true mean function $\boldsymbol{E}[Y|X]$ over the entire underlying distribution, if we take $N \to \infty$? If we do know or can assume a priori that $\boldsymbol{E}[Y|X]$ really is linear (and furthermore in our simplified SLR setting right now has the very simple form $\boldsymbol{E}[Y|X = x] = \beta_0 + \beta_1 x$) and we think of the ordered pairs (X_1, Y_1), (X_2, Y_3),…, (X_N, Y_N) as i.i.d.-generated from some

random process, we can give some of the underlying intuition as to why right here, without formal statements or proofs.

Under suitable, quite general conditions, the answer has to do with the Law of Large Numbers (LLN) from Probability Theory and its extensions. From so-called “uniform” versions of the LLN, it follows that, for any small number $\varepsilon > 0$ and all *N* sufficiently large, we have, for all choices α_0, α_1 of the parameters,

$$\left|\boldsymbol{E}[(Y - (\alpha_0 + \alpha_1 X))^2] - \frac{1}{N}\sum_{i=1}^{N}\big(Y_i - (\alpha_0 + \alpha_1 X_i)\big)^2\right| \le \varepsilon, \text{ with arbitrarily high probability.} \qquad (16)$$

This suggests that for a sufficiently large number *N* of random samples

$$(X_1, Y_1),\ (X_2, Y_3),\ldots,\ (X_N, Y_N)$$

the minimizing least-squares regression parameters $(\hat{\beta}_0, \hat{\beta}_1)$ in (7) also give rise to a function $f_{min}(x) = \hat{\beta}_0 + \hat{\beta}_1 x$ that, up to high probability, approximately minimizes

$$\boldsymbol{E}[(f(X) - Y)^2] = \boldsymbol{E}[(f(X) - \boldsymbol{E}[Y|X])^2] + \boldsymbol{E}[(Y - \boldsymbol{E}[Y|X])^2] \qquad (17)$$

among all functions of the form $f(x) = \alpha_0 + \alpha_1 x$ for some choice of α_0, α_1.

Since, as we have seen, the exact or true mean function $\boldsymbol{E}[Y|X]$, which we assume also has

the simple linear form $\boldsymbol{E}[Y|X = x] = \beta_0 + \beta_1 x$ with parameters β_0 and β_1, is a minimizer of (17) it follows that f_{min} is close to $\boldsymbol{E}[Y|X]$ in the sense that $\boldsymbol{E}[(f_{min}(X) - \boldsymbol{E}[Y|X])^2]$ must be small.

LINE Assumptions for Simple Linear Regression

Recall our Simple Linear Regression Model. Given *N* fixed values $x_1, x_2, \ldots, x_N$, we have

$$Y_n = \beta_0 + \beta_1 x_n + \epsilon_n, n = 1, \ldots, N, \qquad (18)$$

$$\epsilon_n \sim N(0, \sigma^2), n = 1, \ldots, N, \qquad (19)$$

where the error terms ϵ_n -- representing noise or natural stochastic (statistical) variation -- are N independent, normally-distributed random variables.

The main assumptions of this model are frequently denoted by means of the mnemonic acronym **LINE**:

Linearity: The relationship between each Y_n and each x_n, respectively, is linear, and $\boldsymbol{E}[Y_n] = \boldsymbol{E}[Y_n | X_n = x_n] = \beta_0 + \beta_1 x_n$ for all $n = 1, \ldots, N$.

Independence: The errors $\epsilon_n, n = 1, \ldots, N$, are independent random variables.

Normality: The errors $\epsilon_n, n = 1, \ldots, N$, follow a normal distribution. That is, the error across the regression line at any point x_n is described by a normal distribution.

Equal Variance: The normal distribution describing the behavior of the ϵ_n has the same variance, σ^2, for all n. This property is called *homoscedasticity.*

Note that the first or “L” assumption implies that $\boldsymbol{E}[\epsilon_n] = \boldsymbol{E}[Y_n - (\beta_0 + \beta_1 x_n)] = 0.$

Some Comments on the LINE Assumptions

Some observations/comments on the **LINE** assumptions:

- How valid is it to specify that the errors $\epsilon_n, n = 1, \dots, N$, should be normally distributed? It is known that this frequently tends to be the case for random noise as well as random natural variation. One reason could have to with the Central Limit Theorem, which says that, roughly speaking, a large sum of i.i.d. random variables, whatever distribution these individual random variables may follow, will be approximately normally distributed. This suggests that superpositions of large amounts of random noise will tend to be approximately normally-distributed.
- Gauss-Markov Theorem: Assuming the **LINE** hypotheses enables us to know that the Gauss-Markov Theorem holds, which means that we obtain unbiased, minimal variance estimators for the coefficients of the regression function.

We will see in the rest of the course that we will actually be using various methods – including formal statistical tests as well as graphical ones -- to verify or provide evidence for the **LINE** assumptions – or more precisely the latter three “I-N-E” assumptions -- on the random error terms. Successfully verifying those in a specific situation can provide strong evidence that the linearity assumption on the model itself holds as well, in particular in cases in which any knowledge one may have about the particular application domain involved does not give sufficient insight into the nature of the relationship between *X* and *Y*.

The Residuals and Residual Standard Error

Recall once again our SLR model

$$Y_n = \beta_0 + \beta_1 x_n + \epsilon_n, n = 1, \dots, N, \qquad (20)$$

$$\epsilon_n \sim N(0, \sigma^2), n = 1, \dots, N. \qquad (21)$$

We are not able to sample the errors $\epsilon_n, n = 1, \dots, N$, in any direct way, only the Y_n. However, we would want to use the error values to support the validity of our model, as pointed out in the previous slide.

So, consider the so-called **residuals** instead:

$$e_n := y_n - \hat{y}_n, \text{ where } \hat{y}_n := \hat{\beta}_0 + \hat{\beta}_1 x_n, n = 1, \dots, N, \qquad (22)$$

We will in essence use the residuals in key ways in place of the errors ϵ_n, in essence as proxies for the errors ϵ_n whose values we do not have access to, to help justify the validity of our linear regression models, as we will see.

First, we use them to define an estimator for σ^2 in the form

$$\hat{\sigma}^2 = s_e^2 = \frac{1}{N-2} \sum_{n=1}^{N} e_n^2, \qquad (23)$$

where $\hat{\sigma} = s_e$, the square root of the value in (23), is known as the **Residual Standard Error (RSE).** Note the factor $\frac{1}{N-2}$ appearing in (23).

It can be shown that this is actually the right factor to make $\hat{\sigma}^2$ an unbiased estimator for σ^2, so that $E[\hat{\sigma}^2] = \sigma^2$.

In R, we can find the value of the RSE using the following:

```
car_model=lm(dist ~ speed, data = cars)
summary(car_model)$sigma
```

The following further command outputs the residuals for this model:

```
residuals(car_model)
```

Normality of the Residuals

Note that, assuming as we wish to, that the errors ϵ_n, $n = 1, \ldots$,*N*, are normally-distributed according to some distribution $N(0, \sigma^2)$, the Y_n must be normally-distributed as well (with a different mean but the same variance). But, more interestingly, it can be shown (see Sec. 3.2.5 in Sheather (2009) reference) that, for each $n = 1, \ldots$,*N*,

$$e_n = \epsilon_n - \sum_{i=1}^{N} h_{ni}\epsilon_i = (1\text{-}h_{nn})\epsilon_n - \sum_{i=1}^{n-1} h_{ni}\epsilon_i - \sum_{i=n+1}^{N} h_{ni}\epsilon_i, \qquad (24)$$

where $h_{ni} = \frac{1}{N} + \frac{(x_n - \bar{x})(x_i - \bar{x})}{\sum_{j=1}^{N}(x_j - \bar{x})^2}$. Since a (finite) linear combination of independent normally-distributed random variables is also normally distributed, this means that the residuals e_n are themselves also normally distributed if the original noise terms ϵ_n are. Note that, by a linear combination of random variables $Z_1, \ldots, Z_J$, we mean any random variable of the form $\sum_{j=1}^{J} c_j Z_j$, where the c_j are any fixed constants.

But interestingly we can go further than this using (24). Indeed it is argued in Sheather (2009) (again see Sec. 3.2.5 Sheather) that sums of random variables as in (24) can behave approximately like normally-distributed variables even when the ϵ_i,*i=1,…,N*, are not each assumed normally-distributed. Indeed there are extensions of the classical Central Limit Theorem that assert that large, weighted sums of i.i.d. random variables (similar to the sum in (24) above) ,for which the random variables in the sum need not necessarily be normally-distributed themselves, exhibit behavior that approximately follows a normal distribution for very large values of *N.*

Note that our SLR model, as we have defined it, presupposes of course the LINE assumptions, including that of normality of the errors and/or the residuals. However, a key aim of Linear Regression is still try to check that these assumptions are indeed valid for each specific regression model we consider. We will consider methods for this.

Lecture 3 Overview

- Sampling distributions for the SLR regression coefficient estimates (review)
- Confidence intervals for intercept and slope (review)
- Distribution of new observations
- Confidence interval for new observations
- Some general background on hypothesis tests
- Interpretation of the SLR model summary output in R
- Some illustrative examples/computations with R

Sampling Distributions for $\hat{\beta}_0$ and $\hat{\beta}_1$

Explicitly thinking once again of the regression parameter estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ as random variables, we can discuss their **sampling distributions**, the sampling distribution being the probability distribution that results when a statistic is considered as a random variable. Since $\hat{\beta}_0$ and $\hat{\beta}_1$ are both finite, linear combinations of the Y_n (which are independent) and each Y_n is normally distributed, both $\hat{\beta}_0$ and $\hat{\beta}_1$ are normally distributed as well. In fact, we have

$\hat{\beta}_1 = \sum_{n=1}^{N} c_n y_n$, where $c_n = \frac{x_n - \bar{x}}{S_{xx}}$ and $S_{xx} = \sum_{n=1}^{N} (x_n - \bar{x})^2$, and $\hat{\beta}_0 = \sum_{n=1}^{N} d_n y_n$, where $d_n = \frac{1}{N} - c_n \bar{x}$.

It can be shown (see Appendix A.4 in Weisberg (2014)) that

$$\hat{\beta}_0 \sim N\left(\boldsymbol{E}[\hat{\beta}_0], \sigma_{\hat{\beta}_0}^2\right) = N\left(\beta_0, \sigma^2\left(\frac{1}{N} + \frac{\bar{x}^2}{S_{xx}}\right)\right), \quad (25)$$

$$\hat{\beta}_1 \sim N\left(\boldsymbol{E}[\hat{\beta}_1], \sigma_{\hat{\beta}_1}^2\right) = N\left(\beta_1, \frac{\sigma^2}{S_{xx}}\right), \quad (26)$$

where $S_{xx} = \sum_{n=1}^{N} (x_n - \bar{x})^2$, and $\bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n$. So, Var($\hat{\beta}_0$)= $\sigma^2 \left(\frac{1}{N} + \frac{\bar{x}^2}{S_{xx}}\right)$ and Var($\hat{\beta}_1$)= $\frac{\sigma^2}{S_{xx}}$, where σ is as in the definition of a SLR model in (3)-(4) in a prior slide. Of course as we have seen we must estimate the variance σ^2, so as already observed we can estimate it using the RSE s_e=$\hat{\sigma}$: $\hat{\sigma}^2 = \frac{1}{N-2} \sum_{n=1}^{N} e_n^2$. So we can in turn obtain estimates for the respective variances $\sigma_{\hat{\beta}_0}^2$ = Var($\hat{\beta}_0$) and

$\sigma_{\hat{\beta}_1}^2$ =Var($\hat{\beta}_1$) (that is, for the respective standard deviations, taking square roots) via:

$$\sigma_{\hat{\beta}_0} \approx \mathrm{SE}[\hat{\beta}_0] := \hat{\sigma}\left(\frac{1}{N} + \frac{\bar{x}^2}{S_{xx}}\right)^{\frac{1}{2}}, \sigma_{\hat{\beta}_1} \approx \mathrm{SE}[\hat{\beta}_1] := \frac{\hat{\sigma}}{(S_{xx})^{\frac{1}{2}}},$$

where “SE” refers to “Standard Error” and “:=“ denotes for us “is defined as” and “ ≈” denotes “is approximately equal to”.

Confidence Intervals for Intercept and Slope

We can obtain confidence intervals for the true values of the intercept and slope β_0 and β_1 as well:

$$\hat{\beta}_0 - t\left(\frac{\alpha}{2}, N-2\right)\mathrm{SE}\left[\hat{\beta}_0\right] \le \beta_0 \le \hat{\beta}_0 + t\left(\frac{\alpha}{2}, N-2\right)\mathrm{SE}\left[\hat{\beta}_0\right], \qquad (27)$$

$$\hat{\beta}_1 - t\left(\frac{\alpha}{2}, N-2\right)\mathrm{SE}\left[\hat{\beta}_1\right] \le \beta_1 \le \hat{\beta}_0 + t\left(\frac{\alpha}{2}, N-2\right)\mathrm{SE}\left[\hat{\beta}_1\right], \qquad (28)$$

with $(1-\alpha)$x100% confidence, where $\mathrm{SE}\left[\hat{\beta}_0\right] = \hat{\sigma}\left(\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}}\right)^{\frac{1}{2}}$, $\mathrm{SE}\left[\hat{\beta}_1\right] = \frac{\hat{\sigma}}{(S_{xx})^{\frac{1}{2}}}, \hat{\sigma}^2 = \frac{1}{n-2}\sum_{i=1}^{n} e_i^2$.

Here, *t*(*α*/2, *N-2*) is the value that cuts off *α*/2 × 100% in the *upper tail* of the *t*-distribution for *N-2* degrees of freedom (*N* sample data points along with the 2 parameters, intercept and slope, being estimated). The t-distribution (also called "Student's" t-distribution) is invoked because each $\hat{\beta}_i, i = 0,1$, follows a normal distribution as we saw in the previous slide, except for the fact that the corresponding variance σ^2 is unknown and an estimate $\hat{\sigma}^2$ for it must therefore be used instead. When this is done, the resulting variable follows a t-distribution instead. Note that it also goes under the name "student's" t-distribution because a statistician, William Gosset, who played a key role in developing and promoting it used the *nom de plume* "Student" (and not because it is only good for training purposes or something like that).

```
> stop_distance = lm(dist ~ speed, data = cars)
> confint(stop_distance, level = 0.95)

                  2.5 %       97.5 %
(Intercept) -31.167850   -3.990340
speed         3.096964    4.767853
```

R code for generating confidence intervals for β_0, β_1 and example output for 95% confidence.

Distribution of New Observations

We would like to give probabilistic confidence intervals for model predictions for new observations. To do this, let's first describe the corresponding probability distribution. Given *X*=*x* for some new observation *x* (so not one of the *x* values we used for training, i.e., that we used to perform the least-squares regression minimization step in a previous slide in which $\hat{\beta}_0$ and $\hat{\beta}_1$ were defined), we would like to determine confidence intervals for the response variable $Y = \beta_0 + \beta_1 x + \epsilon$ at this new value *x*. For this we need the variance of the variable $\hat{y} + \epsilon = \hat{y}(x) + \epsilon = \hat{\beta}_0 + \hat{\beta}_1 x + \epsilon$, where once again we are now viewing $\hat{\beta}_0$ and $\hat{\beta}_1$ here as random variables, that is, as estimators.

Note that $\mathrm{Var}(\hat{y} + \epsilon)$=$\mathrm{Var}(\hat{y})$+$\mathrm{Var}(\epsilon)$, since $\hat{y}$ and ϵ may be presumed independent. Hence, having calculated $\mathrm{Var}\left(\hat{\beta}_0\right)$ and $\mathrm{Var}(\hat{\beta}_1)$ in previous slides, it follows that

$$\mathrm{Var}(\hat{y} + \epsilon)=\sigma^2\left(\frac{1}{N} + \frac{(x-\bar{x})^2}{S_{xx}}\right) + \sigma^2, \qquad (29)$$

by using the general identity

$$\mathrm{Var}(aZ + bZ') = a^2\mathrm{Var}(Z) + b^2\mathrm{Var}(Z') + 2ab\mathrm{Cov}(Z, Z'),$$

for any random variables Z, Z' and where $\mathrm{Cov}(\cdot,\cdot)$ denotes the covariance $\mathrm{Cov}(Z, Z')$=$\boldsymbol{E}[(Z - \boldsymbol{E}[Z])(Z' - \boldsymbol{E}[Z'])]$. We also use the fact that, in this case, $\mathrm{Cov}\left(\hat{\beta}_0, \hat{\beta}_1\right) = -\sigma^2\frac{\bar{x}}{S_{xx}}$ (see Equ. (2.12) in Weisberg(2014)). Hence,

$$\hat{y} + \epsilon \sim N\left(\beta_0 + \beta_1 x, \sigma^2\left(\frac{1}{N} + \frac{(x-\bar{x})^2}{S_{xx}} + 1\right)\right), \qquad (30)$$

*S*ince we also need an estimate for σ^2 as we have previously seen, we insert in (30) the RSE and can use, in place of

$\sigma\left(\frac{1}{N} + \frac{(x-\bar{x})^2}{S_{xx}} + 1\right)^{1/2}$,

$$\mathrm{SE}[\hat{y} + \epsilon] = \hat{\sigma}\left(\frac{1}{N} + \frac{(x-\bar{x})^2}{S_{xx}} + 1\right)^{1/2}. \qquad (31)$$

Confidence Interval for New Observations

We want to know a confidence interval for the response $Y = \beta_0 + \beta_1 x + \epsilon$ at a given new value *x*. We can use the probability distribution we have for $\hat{y} + \epsilon = \hat{y}(x) + \epsilon = \hat{\beta}_0 + \hat{\beta}_1 x + \epsilon$ from the previous slide for this purpose. Our corresponding confidence interval is then

$$\hat{y} + \epsilon - t\left(\frac{\alpha}{2}, N-2\right) \text{SE}[\hat{y} + \epsilon] \le Y \le \hat{y} + \epsilon + t\left(\frac{\alpha}{2}, N-2\right) \text{SE}[\hat{y} + \epsilon], \quad (32)$$

with $(1-\alpha)$x100% probabilistic confidence, $0 \le \alpha \le 1$ and chosen as desired (e.g., 0.05), and where $t(\cdot,\cdot)$ once again corresponds to a t-distribution (and with *N*− 2 degrees of freedom) as we saw for confidence intervals for β_0, β_1 as well.

R code and output for the 95% confidence interval for a new observation, in this case 16 mph.

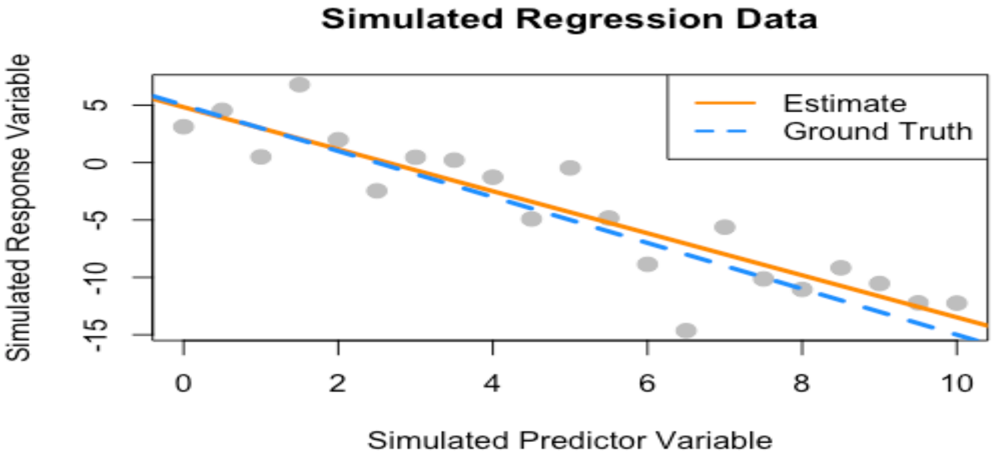
```
stop_distance = lm(dist ~ speed, data = cars)
 speedsabc = data.frame(speed = c(16))
  predict(stop_distance, newdata = speedsabc,
    interval = c("prediction"), level = 0.95)
              fit          lwr          upr
   45.339445   14.10499  76.57390
```

Simple Linear Regression Simulation Study

We simulate “artificial” data samples

$$Y_n = 5 - 2x_n + \epsilon_n, n = 1, \ldots, 21,$$
$$\epsilon_n \sim N(0,9), n = 1, \ldots, 21,$$

and then can see how closely our computed least-squares regression model can approximate the true, artificially-generated model. Corresponding R code is to the right with some of the output below (in the form of a graphic).



```
num_obs = 21
beta_0  = 5
beta_1  = -2
sigma   = 3
set.seed(1)
epsilon = rnorm(n = num_obs, mean = 0, sd = sigma)
x_vals = seq(from = 0, to = 10, length.out = num_obs)
y_vals = beta_0 + beta_1 * x_vals + epsilon
sim_fit = lm(y_vals ~ x_vals)
coef(sim_fit)


sim_slr = function(x, beta_0 = 5, beta_1 = -2, sigma = 3) {
  n = length(x)
  epsilon = rnorm(n, mean = 0, sd = sigma)
  y = beta_0 + beta_1 * x + epsilon
  data.frame(predictor = x, response = y)
}
set.seed(1)
sim_data = sim_slr(x = x_vals, beta_0 = 5, beta_1 = -2, sigma = 3)
sim_fit = lm(response ~ predictor, data = sim_data)
coef(sim_fit)
plot(response ~ predictor, data = sim_data,
     xlab = "Simulated Predictor Variable",
     ylab = "Simulated Response Variable",
     main = "Simulated Regression Data",
     pch  = 20,
     cex  = 2,
     col  = "grey")
abline(sim_fit, lwd = 3, lty = 1, col = "darkorange")
abline(beta_0, beta_1, lwd = 3, lty = 2, col = "dodgerblue")
legend("topright", c("Estimate", "Ground Truth"), lty = c(1, 2), lwd = 2,
       col = c("darkorange", "dodgerblue"))
```

Hypothesis Tests

In Hypothesis Tests, we translate a scientific question into a statement concerning hypotheses regarding parameters in a statistical model.

Examples:

1. Are the chances of having a heart attack the same for males and females?

$$H_0 : p_m = p_f \text{ and } H_1 : p_m \neq p_f$$

where p_m is the probability to have a heart attack for males and p_f is the probability to have a heart attack for females.

2. In clinical trials, is the effect of a treatment no different from that of a placebo? Or is the effect different?

Given a possible least squares SLR regression model

$$Y_n = \beta_0 + \beta_1 x_n + \epsilon_n, n = 1, \dots, N,$$

where x_n is the treatment level, e.g., 0, 1, 2, 3, 4, where 0 means the placebo

level, and Y_n is the observed response. The hypothesis test for significance of the regression relative to an "intercept-only" model is

$$H_0 : \beta_1 = 0 \text{ and } H_1 : \beta_1 \neq 0$$

Steps to Conduct a Hypothesis Test

1) Set up two competing hypotheses: Null hypothesis (H_0) and the alternative hypothesis (H_1 or H_a).

2) Set some significance level called α: the most common α is 0.05 or 5%.

3) Calculate a test statistic (t⋆): a function of the data whose distribution

depends only on the parameter(s) being tested.

4) Calculate probability value (p-value), or equivalently find a rejection region. A p-value is the probability of seeing data at least as extreme as t⋆ under the assumption of H_0.

5) The rejection region is found by using α to find a critical value; the rejection region is the area that is more extreme than the critical value.

6) Make a test decision about the null hypothesis.
Reject H_0 if the p-value < α.

Interpretation of SLR Model Summary Output in R

Residuals: If the residuals are normally distributed with mean 0 (and constant variance), this should be consistent with the values reported here. The median would likely be close to 0 and the symmetry of the distribution would likely be reflected in the values of the other four numbers here as well, which would be expected to approximately balance.

Estimate: Computed estimates for the intercept and x-variable coefficient (slope).

Std. Error: This is the value SE$\left[\hat{\beta}_i\right]$,$i = 0{,}1$, for the corresponding estimator for the standard deviation of $\hat{\beta}_i$ that we have introduced in previous slides.

$$\text{SE}\left[\hat{\beta}_0\right] = \hat{\sigma}\left(\frac{1}{N} + \frac{x^2}{S_{xx}}\right)^{\frac{1}{2}},\ \text{SE}\left[\hat{\beta}_1\right] = \frac{\hat{\sigma}}{(S_{xx})^{\frac{1}{2}}}, \text{where } \hat{\sigma}^2 = \frac{1}{N-2}\sum_{n=1}^{N} e_n^2$$

t value: This is actually the Estimate as above divided by the Std. Error. A larger value implies more confidence in the corresponding parameter estimate.

Pr(>|t|): This is the p-value – probability value – for the associated t-test on the linear model parameters. Since for SLR this test is essentially equivalent to the F-test below, we defer more discussion of it until we consider Multiple Regression.

Signif. Codes: The significance codes indicate how certain we can be that the coefficient has an impact on the dependent variable. For example, a significance level of 0.001 indicates that there is less than a 0.1% chance that the coefficient might be equal to 0 and thus be insignificant. Stated differently, we can be 99.9% sure that it is significant. The significance codes (shown by asterisks) are intended for quickly ranking the significance of each variable.

Residual Standard Error: This is our estimate $\hat{\sigma}$ (introduced in previous slides) for the standard deviation σ of our exact or true SLR regression model:

$$\text{RSE-squared is } \hat{\sigma}^2 = \frac{1}{N-2}\sum_{n=1}^{N} e_n^2.$$

(Multiple) R-squared(Coefficient of Determination): Measures, in a suitable sense, the proportion of the variance in the dependent variable that is predictable from the independent variable(s). Defined as

$$R^2 = 1 - \frac{\sum_{n=1}^{N} e_n^2}{\sum_{n=1}^{N}(y_n - y)^2}\ \text{,where } \bar{y} = \frac{1}{N}\sum_{n=1}^{N} y_n.$$

Adjusted R-squared: We defer discussion on this until we get to multiple regression.

F-statistic: This, here 89.57, is the value of the F-statistic corresponding to an F Test (F Hypothesis Test) for the SLR model intended to assess whether the regression model is significant, that is, whether a non-zero value for $\hat{\beta}_1$ yields a model that better explains the data than simply taking β_1=0, which corresponds to an "intercept-only" model. Note that "1 and 48 DF" corresponds to the difference in the number of parameters between the SLR model and an intercept-only one – which is 1 – and the number of degrees of freedom in the SLR model, which here is 50-2=48.

p-value: This is the p-value for the F-test (Significance of Regression test).

```
stop_distance_model = lm(dist ~ speed, data = cars)
summary(stop_distance_model)
Call:
lm(formula = dist ~ speed, data = cars)
Residuals:
    Min      1Q  Median      3Q     Max
-29.069  -9.525  -2.272   9.215  43.201

Coefficients:
            Estimate     Std. Error    t value    Pr(>|t|)
(Intercept) -17.5791     6.7584    -2.601    0.0123 *
speed         3.9324      0.4155     9.464     1.49e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.38 on 48 Degrees of
Freedom
Multiple R-squared:  0.6511,  Adjusted R-squared:
0.6438
F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

Lecture 4 Overview

- Coefficient of Determination
- F-test for significance of Simple Linear Regression
- Multiple Linear Regression
- Least-squares solution of Multiple Regression
- Various computational examples and implementations in R

Coefficient of Determination (R^2)

The **Coefficient of Determination R^2** (R-squared) is a goodness-of-fit measure of the proportion of the variance in the response variable that is explained from the predictor variable(s). The R^2 can be viewed as a measure of regression model accuracy. Define

$$R^2 = 1 - \frac{\sum_{n=1}^{N} e_n^2}{\sum_{n=1}^{N}(y_n - \bar{y})^2} = 1 - \frac{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{\sum_{n=1}^{N}(y_n - \bar{y})^2}, \text{where } \bar{y} = \frac{1}{N}\sum_{n=1}^{N} y_n. \qquad (33)$$

Now the **Decomposition of Variation (DoV)** equation, which holds in the context of Simple Linear Regression models (as well as more generally for Multiple Linear Regression, which we discuss later on) is the following:

$$\sum_{n=1}^{N}(y_n - \bar{y})^2 = \sum_{n=1}^{N}(y_n - \hat{y}_n)^2 + \sum_{n=1}^{N}(\bar{y} - \hat{y}_n)^2 \quad \text{(requires proof)} \qquad (34)$$

So, DoV shows that $0 \leq R^2 \leq 1$. Larger values of R^2 are generally viewed as more promising for the validity of the model. Part of the reason for this is that a lower value for $\sum_{n=1}^{N}(y_n - \hat{y}_n)^2$ clearly implies a better model. But with the DoV we can say more. From (33) and (34), we see that

$R^2 = 1 - \frac{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{\sum_{n=1}^{N}(y_n - \bar{y})^2} = \frac{\sum_{n=1}^{N}(\hat{y}_n - \bar{y})^2}{\sum_{n=1}^{N}(y_n - \bar{y})^2}$. Since $\sum_{n=1}^{N}(z_n - \bar{z})^2$ is an estimate (up to a constant factor) of the variance of any given random variable Z and we also have $R^2 \leq 1$, a high value for R^2 appears consistent with a model whose predictor variable *X* (or variables for multiple regression) explains more (or much) of the variance or variation in the response *Y.*

F-test for Significance of Simple Linear Regression

The *F*—test (results of which are at the bottom of the output summary report for the R lm( ) function, see a previous slide) concerns the following statistical Hypothesis Test for SLR:

$$\mathrm{H}_0 : \beta_1 = 0 \text{ and } \mathrm{H}_1 : \beta_1 \neq 0$$

So, this test in the case of SLR tells us whether the predictor variable adds any explanatory value to the model at all. That is, it tells us whether employing the predictor variable makes the model more complex than it needs to be and simply including the β_0 parameter alone would suffice -- or not. The *F*-statistic for the F-test is defined by

$$F = \frac{\sum_{n=1}^{N}(\hat{y}_n - \bar{y})^2}{\left(\frac{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{N-2}\right)} \text{ where } \hat{y}_n \coloneqq \hat{\beta}_0 + \hat{\beta}_1 x_n. \qquad (35)$$

Using Decomposition of Variation, we can rewrite this in the form

$$F = \frac{\sum_{n=1}^{N}(y_n - \bar{y})^2 - \sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{\left(\frac{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{N-2}\right)}, \qquad (36)$$

and this representation appears consistent with the magnitude of the statistic F rising the better H_1 explains the distribution of the sample data points, i.e., the better taking $\hat{y}_n \coloneqq \hat{\beta}_0 + \hat{\beta}_1 x_n$ explains it. This is because the better H_1 explains the distribution of the sample data points the smaller the denominator is (making F larger) and the smaller in absolute the second term in the numerator is as well (again tending to make *F* larger as the term considered in absolute value is subtracted).

Under the Null Hypothesis, it is known that the *F*-statistic should follow an $F(\cdot,\cdot)$ probability distribution with respective parameters $(d_2 - d_1, N - d_2)$, where $d_2 - d_1$ is the difference in the number of regression function parameters between H_1 and H_0, 2-1=1 in this case, and $N - d_2$ is the number of degrees of freedom for H_1 , $N - 2$ in this case. From the summary report we obtain the associated *p*-value corresponding to the model and the data. Indeed, under the assumption that the Null Hypothesis is true, the *p*-value is the probability of the value of the F -statistic being as large as it is or larger. Hence, in essence, a very low p-value (for example, one less or even much less than 0.05, which corresponds to a 95% confidence interval) implies that we should reject the Null Hypothesis H_0, and hence also implies the significance of the SLR model in this case with both parameters – intercept and slope – significant as well.

Multiple Linear Regression

Of course many datasets feature multiple predictor variables. Indeed a response variable may naturally depend on a number (> 1) of explanatory variables. So we extend our current linear model to allow a response to depend on *multiple* predictors. This is called **Multiple Linear Regression**, or simply **Multiple Regression**. Many aspects of our Simple Linear Regression (SLR) model extend fairly naturally to the multiple-predictor setting, and this does in fact hold true for the general definition of the multiple regression model itself:

For any given sample $(x_1, x_2, \dots, x_M)$, we have the general underlying model

$$\boldsymbol{E}[Y] = \boldsymbol{E}[Y|X_1 = x_1, X_2 = x_2, \dots, X_M = x_M] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_M x_M, \qquad (37)$$

where Y is a continuous random variable. For N fixed $(\text{non} - \text{random})$ sample vectors $(x_{11}, \dots, x_{1M})$, $(x_{21},, \dots, x_{2M})$,…,$(x_{N1}, \dots, x_{NM})$ we have, in analogy with SLR, the corresponding set of equations (with random noise terms)

$$Y_n = \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + \cdots + \beta_M x_{nM} + \epsilon_n, \qquad (38a)$$

where $\epsilon_n \sim N(0, \sigma^2), n = 1, \dots, N,$ with independent random noise terms ϵ_n. (38b)

Least-Squares Solution of Multiple Linear Regression

Given our Multiple Regression (MR) model

$$Y_n = \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + \cdots + \beta_M x_{nM} + \epsilon_n,\ n = 1, \ldots, N, \qquad (39)$$

$$\epsilon_n \sim N(0, \sigma^2), n = 1, \ldots, N, \text{where the terms } \epsilon_n \text{ are independent r.v.'s} \qquad (40)$$

we can, as in the case of simple linear regression, compute estimators

$(\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_M)$ corresponding to the coefficients $(\beta_0, \beta_1, \ldots, \beta_M)$ as above. This we do, once again by means of least-squares optimization (minimization):

$$(\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_M) = \arg\min_{(\alpha_0, \alpha_1, \ldots, \alpha_M) \epsilon \mathbb{R}^{M+1}} \sum_{i=1}^{N} \left(y_i - (\alpha_0 + \alpha_1 x_{i1} + \cdots + \alpha_M x_{iM})\right)^2, \qquad (41)$$

Perhaps the most natural way to compute a solution to this minimization problem is to rewrite our MR model equations in terms of matrices, so define

$$\mathbb{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \mathbb{X} = \begin{bmatrix} 1 & x_{11} & \ldots & x_{1M} \\ 1 & x_{21} & \ldots & x_{2M} \\ \vdots & \vdots & & \vdots \\ 1 & x_{N1} & \ldots & x_{NM} \end{bmatrix}, \mathbb{B} = \begin{bmatrix} \beta_0 \\ \beta_2 \\ \vdots \\ \beta_M \end{bmatrix}, \mathbb{E} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{bmatrix}, \text{ and } \widehat{\mathbb{B}} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_2 \\ \vdots \\ \hat{\beta}_M \end{bmatrix} \qquad (42)$$

and (39)-(40) can be rewritten as $\mathbb{Y} = \mathbb{X}\mathbb{B} + \mathbb{E}$. It can then be shown, using differential calculus, that a solution $\widehat{\mathbb{B}}$ to (41) always exists and must satisfy

$$(\mathbb{X}^T\mathbb{X})\widehat{\mathbb{B}} = \mathbb{X}^T\mathbb{Y}. \qquad (43)$$

Lecture 5 Overview

Topics in this lecture include:

- Basic Concepts and Results in Multiple Linear Regression
- Sampling Distribution for the $\hat{\beta}_m$ for Multiple Regression
- Confidence Intervals for β_m for Multiple Regression

Multiple Linear Regression

Of course many datasets feature multiple predictor variables. Indeed a response variable may naturally depend on a number (> 1) of explanatory variables. So we extend our current linear model to allow a response to depend on *multiple* predictors. This is called **Multiple Linear Regression**, or simply **Multiple Regression**. Many aspects of our Simple Linear Regression (SLR) model extend fairly naturally to the multiple-predictor setting, and this does in fact hold true for the general definition of the multiple regression model itself:

For any given sample $(x_1, x_2, \ldots, x_M)$, we have the general underlying model

$$\boldsymbol{E}[Y] = \boldsymbol{E}[Y|X_1 = x_1, X_2 = x_2, \ldots, X_M = x_M] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_M x_M, \qquad (37)$$

where Y is a continuous random variable. For N fixed $(\mathrm{non} - \mathrm{random})$ sample vectors $(x_{11}, \ldots, x_{1M})$, $(x_{21},, \ldots, x_{2M})$,…,$(x_{N1}, \ldots, x_{NM})$ we have, in analogy with SLR, the corresponding set of equations (with random noise terms), which is our general **Multiple Linear Regression Model**:

$$Y_n = \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + \cdots + \beta_M x_{nM} + \epsilon_n, \qquad (38a)$$

where the $\epsilon_n \sim N(0, \sigma^2), n = 1, \ldots, N,$ are independent random noise terms ϵ_n. (38b)

Least-Squares Solution of Multiple Linear Regression

Given our Multiple Linear Regression, or simply Multiple Regression (MR), model

$$Y_n = \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + \cdots + \beta_M x_{nM} + \epsilon_n,\ n = 1, \dots, N, \qquad (39)$$

$$\epsilon_n \sim N(0, \sigma^2), n = 1, \dots, N, \text{where the terms } \epsilon_n \text{ are independent r.v.'s} \qquad (40)$$

we can, as in the case of simple linear regression, compute estimators

$(\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_M)$ corresponding to the coefficients $(\beta_0, \beta_1, \dots, \beta_M)$ as above. This we do, once again by means of least-squares optimization (minimization):

$$(\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_M) = \arg\min_{(\alpha_0, \alpha_1, \dots, \alpha_M) \epsilon \mathbb{R}^{M+1}} \sum_{i=1}^{N} \left(y_i - (\alpha_0 + \alpha_1 x_{i1} + \cdots + \alpha_M x_{iM})\right)^2, \qquad (41)$$

Perhaps the most natural way to compute a solution to this minimization problem is to rewrite our MR model equations in terms of matrices, so define

$$\mathbb{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \mathbb{X} = \begin{bmatrix} 1\, x_{11} & \dots x_{1M} \\ 1\, x_{21} & \dots x_{2M} \\ \vdots \quad \vdots & \vdots \\ 1\, x_{N1} & \dots x_{NM} \end{bmatrix}, \mathbb{B} = \begin{bmatrix} \beta_0 \\ \beta_2 \\ \vdots \\ \beta_M \end{bmatrix}, \mathbb{E} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{bmatrix}, \text{ and } \widehat{\mathbb{B}} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_2 \\ \vdots \\ \hat{\beta}_M \end{bmatrix} \qquad (42)$$

and (39)-(40) can be rewritten as $\mathbb{Y} = \mathbb{X}\mathbb{B} + \mathbb{E}$. It can then be shown, using differential calculus, that a solution $\widehat{\mathbb{B}}$ to (41) always exists and must satisfy

$$(\mathbb{X}^T\mathbb{X})\widehat{\mathbb{B}} = \mathbb{X}^T\mathbb{Y}. \qquad (43)$$

Least-Squares Solution of Multiple Regression (cont'd)

Given our Multiple Regression model with sample data $\mathbb{X} = \begin{bmatrix} 1\, x_{11} \ldots x_{1M} \\ 1\, x_{21} \ldots x_{2M} \\ \vdots \quad \vdots \qquad \vdots \\ 1\, x_{N1} \ldots x_{NM} \end{bmatrix}$, $\mathbb{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$ and solution vector $\widehat{\mathbb{B}}$ of estimators $\widehat{\mathbb{B}} = \begin{bmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ \vdots \\ \hat{\beta}_M \end{bmatrix}$ that satisfies $(\mathbb{X}^T\mathbb{X})\widehat{\mathbb{B}}$=$\mathbb{X}^T\mathbb{Y}$ from the previous slide, note that we can quite simply take

$$\widehat{\mathbb{B}}=(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{Y} \qquad (44)$$

provided the matrix inverse of $\mathbb{X}^T\mathbb{X}$ exists. One key case in which it does exist is when the columns of $\mathbb{X}$ are linearly independent. However, even if $\mathbb{X}^T\mathbb{X}$ is not invertible -- or is not invertible in the conventional sense but may require the concept of the "generalized inverse" of a matrix -- there still exist efficient algorithms for computing a matrix $\widehat{\mathbb{B}}$ satisfying $(\mathbb{X}^T\mathbb{X})\widehat{\mathbb{B}}$=$\mathbb{X}^T\mathbb{Y}$ in any case. For the most part in this course, we will generally make the assumption that the inverse $\mathbb{X}^T\mathbb{X}$ exists or that there does exist a "generalized inverse" that satisfies the key properties of the true inverse that we need here in the context of multiple regression.

Expectation of the $\widehat{\mathbb{B}}$ matrix

Making the assumption that $\mathbb{X}^T\mathbb{X}$ is an invertible matrix, we want to compute $\mathbf{E}(\widehat{\mathbb{B}})=\mathbf{E}((\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{Y})$, where we define the expectation or mean of a random vector V=(V_1,…,V_d) of length d, denoted $\mathbf{E}$(V), to also be a vector of the same length, having $\mathbf{E}$(V_i) as its ith component. Also, with respect to such a random vector V, we have, for any matrix A whose entries are fixed (non-random) and with d columns, the identity:

$$\mathbf{E}(AV)=A\mathbf{E}(V), \tag{45}$$

Then,

$$\begin{aligned}\mathbf{E}(\widehat{\mathbb{B}})&=(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbf{E}(\mathbb{Y})\\ &=(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T(\mathbb{X}\mathbb{B})\\ &=\mathbb{B}.\end{aligned} \tag{46}$$

Variance of the $\widehat{\mathbb{B}}$ matrix

Want to compute

$$\mathrm{Var}(\widehat{\mathbb{B}})=\mathrm{Var}((\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{Y}\ ), \qquad (47)$$

where we define the variance of a random vector V=(V_1,…,V_d) length d to be a dxd matrix, denoted Var(V), having Var(V_i) as its (i,i) entry and Cov(V_i,V_j)= Cov(V_j,V_i) as both its (i,j) and its (j,i) entries. We have, for any matrix A whose entries are fixed (non-random) and with d columns, the identity:

$$\mathrm{Var(AV)=AVar(V)\ A^T}, \qquad (48)$$

A^T denoting the transpose of A. Hence,

$$\begin{aligned}\mathrm{Var}(\widehat{\mathbb{B}})&=\mathrm{Var}((\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{Y}\ )\\ &= (\mathbb{X}^T\mathbb{X})^{-1}\mathrm{Var}(\mathbb{X}^T\mathbb{Y})\ ((\mathbb{X}^T\mathbb{X})^{-1})^{\ T}\\ &=(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathrm{Var}(\mathbb{Y})\ \ \mathbb{X}((\mathbb{X}^T\mathbb{X})^{-1})^{\ T} \qquad (49)\\ &=(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\sigma^2 I\ \mathbb{X}((\mathbb{X}^T\mathbb{X})^{-1})^{\ T}\\ &= \sigma^2\ (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}\\ &= \sigma^2(\mathbb{X}^T\mathbb{X})^{-1}\end{aligned}$$

using such properties as $(A^{-1})^{\ T}= (A^T)^{\ -1}$, $(AB)^{-1}=B^{-1}A^{-1}$, and $(AB)^T=B^TA^T$, as well as the fact that the entries of $\mathbb{Y}$ are independent (meaning, here, independence in the probabilistic/statistical sense).

Sampling Distribution for the $\hat{\beta}_m$ for Multiple Regression

We update our estimate for σ^2 in the case of Multiple Regression, and now take our estimate for it to be

$$\hat{\sigma}^2 = s_e^2 = \frac{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{N-M-1}, \qquad (50)$$

where we had to reduce the value of the denominator to reflect the additional regression parameters. We have $\boldsymbol{E}[s_e^2]= \sigma^2$, so s_e^2 is an unbiased estimator for σ^2.

We have that $\hat{\mathbb{B}}$=$(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{Y}$ where as usual we assume that the $\mathbb{X}$ matrix is non-random (deterministic), which means that since the entries of $\mathbb{Y}$ are i.i.d. and normally-distributed, the entries of the $\hat{\mathbb{B}}$ matrix are normally-distributed as well since they are finite, linear combinations of the components of $\mathbb{Y}$.

So, for each $m = 0, \dots, M$, we have, using (46) and (49), respectively, from prior slides,

$$\hat{\beta}_m \sim N(\beta_m, \sigma^2((\mathbb{X}^T\mathbb{X})^{-1})_{mm}). \qquad (51)$$

Moreover, in analogy with the case of SLR, we have the following estimate SE$(\hat{\beta}_m)$ for the standard deviation $\sigma\sqrt{((\mathbb{X}^T\mathbb{X})^{-1})_{mm}}\big(\text{which is the square root of the variance value appearing in } (51)\big)$ of $\hat{\beta}_m$:

$$\text{SE}[\hat{\beta}_m]=s_e\sqrt{((\mathbb{X}^T\mathbb{X})^{-1})_{mm}}\,, \qquad (52)$$

where $s_e^2 = \frac{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{N-M-1}$ is as above.

Confidence Interval for β_m for Multiple Regression

Similarly to the case for SLR, since we must estimate σ^2 and we do so using s_e^2, $\hat{\beta}_m$ as a random variable is known to follow in practice a t-distribution once σ^2 is replaced with the approximation s_e^2 . The t-distribution, like the normal distribution, is symmetric and bell-shaped, but has heavier tails than its more famous cousin so is more apt to generate values that fall further from its mean. It is a special case of the generalized hyperbolic distribution. So we have that, for each, $m = 0, \dots, M$,

$$\frac{\hat{\beta}_m - \beta_m}{\text{SE}[\hat{\beta}_m]} \sim t_{(N-M-1)} \qquad (53)$$

So, we can obtain confidence intervals for the true values of the β_m as well:

$$\hat{\beta}_m - t\left(\frac{\alpha}{2}, N-M-1\right)\text{SE}[\hat{\beta}_m] \leq \beta_m \leq \hat{\beta}_m + t\left(\frac{\alpha}{2}, N-M-1\right)\text{SE}[\hat{\beta}_m], \quad (54)$$

with $(1-\alpha)$x100% confidence, where, as with SLR, $t\left(\frac{\alpha}{2}, N-M-1\right)$ is the value that cuts off $\alpha/2$ × 100% in the *upper tail* of the *t*-distribution for $N-M-1$ degrees of freedom.

R commands:

```
model_147= lm(hp ~ wt + cyl, data = mtcars)
confint(model_147, level = 0.99)
```

Result:

```
                 0.5 %   99.5 %
(Intercept) -124.10755 20.49642
wt           -30.57995 33.24087
cyl           13.90506 48.87074
```

Lecture 6 Overview

Topics in this lecture include:

- The Residuals in Multiple Regression
- F-test for significance of Multiple Regression
- Single parameter significance test for Multiple Regression
- Computational examples with R
- Review of Solutions to Homework Assignment #1

The Residuals for Multiple Regression

Recall of course our Multiple Regression (MR) model

$$Y_n = \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + \cdots + \beta_M x_{nM} + \epsilon_n, n = 1, \dots, N, \qquad (55)$$

$$\epsilon_n \sim N(0, \sigma^2), n = 1, \dots, N, \qquad (56)$$

We are not able to sample the errors $\epsilon_n, n = 1, \dots, N$, in any direct way, only the Y_n. However, we would want to use the error values to support the validity of our model, as pointed out for SLR.

So, consider the so-called **residuals** instead:

$$e_n := y_n - \hat{y}_n, \text{ where } \hat{y}_n := \hat{\beta}_0 + \hat{\beta}_1 x_{n1} + \cdots + \hat{\beta}_M x_{nM}, \; n = 1, \dots, N, \qquad (57)$$

We will in essence use the residuals in key ways in place of the errors ϵ_n, in essence as proxies for the errors ϵ_n whose values we do not have access to, to help justify the validity of our linear regression models, as we will see.

First, we use them to define an estimator for σ^2 in the form

$$\hat{\sigma}^2 = s_e^2 = \frac{1}{N-M-1} \sum_{n=1}^{N} e_n^2, \qquad (58)$$

where $\hat{\sigma}$=s_e (as also previously defined in (50)) is known as the **Residual Standard Error (RSE).** Note the factor $\frac{1}{N-M-1}$ appearing in (58).

It can be shown that this is actually the right factor to make $\hat{\sigma}^2$ an unbiased estimator for σ^2, so that $E[\hat{\sigma}^2]$= σ^2.

Computing the RSE with R:

```
model_147= lm(hp ~ wt + cyl, data = mtcars)
summary(model_147)$sigma
```

F-test for Significance of Multiple Regression

The *F*—test (results of which are at the bottom of the output summary report for the R lm( ) function) concerns the following statistical Hypothesis Test for Multiple Regression:

H$_0$: β_1 =… = β_M $=$0 (intercept-only model) and H$_1$: $\beta_m \neq 0$ for at least one *m=1,…,M*

So, this test tells us whether the predictor variables add any explanatory value to the model at all. That is, it tells us whether including any predictor variables in the regression model makes the model more complex than it needs to be and simply including the β_0 parameter alone would suffice -- or not. The *F*-statistic for the F-test is defined by

$$F = \frac{\frac{\sum_{n=1}^{N}(\hat{y}_n - \bar{y})^2}{M}}{\left(\frac{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{N-M-1}\right)} \text{ where } \hat{y}_n \coloneqq \hat{\beta}_0 + \hat{\beta}_1 x_{n1} + \cdots + \hat{\beta}_M x_{nM},\ n = 1, \dots, N, \quad (59)$$

Using Decomposition of Variation as we have introduced in a previous slide, we can rewrite this in the form

$$F = \frac{\frac{\sum_{n=1}^{N}(y_n - \bar{y})^2 - \sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{M}}{\left(\frac{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{N-M-1}\right)}, \qquad (60)$$

and this representation appears consistent with the magnitude of the statistic F rising the better H_1 explains the distribution of the sample data points, i.e., the better taking $\hat{y}_n$ explains it. This is because the better H_1 explains the distribution of the sample data points the smaller the denominator is (making F larger) and the smaller in absolute the second term in the numerator is as well (again tending to make *F* larger as the term considered in absolute value is subtracted).

Under the Null Hypothesis, it is known that the *F*-statistic should follow an $F(\cdot,\cdot)$ probability distribution with respective parameters $(d_2 - d_1, N - d_2)$, where $d_2 - d_1$ is the difference in the number of regression function parameters between H$_1$ and H$_0$, and $N - d_2$ is the number of degrees of freedom for H$_1$. From the summary report we obtain the associated *p*-value corresponding to the model and the data. Indeed, under the assumption that the Null Hypothesis is true, the *p*-value is the probability of the value of the F -statistic being as large as it is or larger. Hence, in essence, a very low p-value (for example, one less or even much less than 0.05, which corresponds to a 95% confidence interval) implies that we should reject the Null Hypothesis H$_{0,}$ and hence also implies the significance of the non-intercept only Multiple Regression model in this case.

Single-parameter significance test (t-test) for multiple regression

The R language lm( ) function summary report contains the results of the following Hypothesis Test in the multiple regression context:

For any given $m = 1,2,\dots,M$, $H_0: \beta_m = 0$ vs. $H_1: \beta_m \neq 0$.

So, this test assesses whether the response variable Y depends linearly on the *m*-th predictor x_m in any significant way. The null hypothesis H_0, if true, implies no significant dependence on x_m.

We apply a t-test with test statistic defined by

$$t_{(N-M-1)} = \frac{\widehat{\beta}_m - \beta_m}{\mathrm{SE}[\widehat{\beta}_m]} = \frac{\widehat{\beta}_m}{s_e\sqrt{\left((\mathbb{X}^T\mathbb{X})^{-1}\right)_{mm}}}, \text{ where } s_e^2 = \frac{\sum_{n=1}^{N}(y_n - \bar{y})^2}{N-M-1}. \qquad (61)$$

(in keeping with the test statistic typically satisfying (estimate – hypothesis)/standard error)

which, under the null hypothesis, follows a t-distribution with *N−M-1* degrees of freedom. The p-value for the corresponding t-test is the probability that a corresponding t-distributed random variable would take on a value greater than or equal to the absolute value of the t_0 statistic as above.

Lecture 7 Overview

Topics in this lecture include:

- Interpretation of Multiple Regression Model Summary Output for R’s lm function
- Confidence interval for the mean function for Multiple Regression
- Confidence interval for new values/predictions for MR
- Adjusted Coefficient of Determination
- Computational examples using R

Interpretation of MR Model Summary Output in R

Residuals: If the residuals are normally distributed with mean 0 (and constant variance), this should be consistent with the values reported here. The median would likely be close to 0 and the symmetry of the distribution would likely be reflected in the values of the other four numbers here as well, which would be expected to approximately balance.

Estimate: The estimates $\hat{\beta}_m$, $m = 0,1,\dots,M$ for the model parameters β_M, $m = 0,1,\dots,M$, respectively.

Std. Error: This is the value SE$[\hat{\beta}_m]$,$m = 0,1,\dots,M$. for the corresponding estimator for the standard deviation of $\hat{\beta}_m$ that we have introduced in previous slides (see (52), slide 35)

$$\mathrm{SE}[\hat{\beta}_m] = s_e\sqrt{((\mathbb{X}^T\mathbb{X})^{-1})_{mm}},\ \ s_e^2 = \frac{1}{N-M-1}\sum_{n=1}^N e_n^2.$$

t value: This is actually the Estimate as above divided by the Std. Error. A larger value implies more confidence in the corresponding parameter estimate.

Pr(>|t|): This is the p-value – probability value – for the associated t-test for the individual predictor variables. This t-test assesses whether the corresponding predictor has significant predictive influence on the response variable within the regression model.

Signif. Codes: The significance codes indicate how certain we can be that the coefficient has an impact on the dependent variable. For example, a significance level of 0.001 indicates that there is less than a 0.1% chance that the coefficient might be equal to 0 and thus be insignificant. Stated differently, we can be 99.9% sure that it is significant. The significance codes (shown by asterisks) are intended for quickly ranking the significance of each variable.

Residual Standard Error: This is our estimate $s_e = \hat{\sigma}$ (introduced in previous slides) for the exact standard deviation σ in our MR regression model:

$$\text{the RSE is the square root of } \hat{\sigma}^2 = s_e^2 = \frac{1}{N-M-1}\sum_{n=1}^N e_n^2.$$

(Multiple) R-squared(Coefficient of Determination): Measures, in a suitable sense, the proportion of the variance in the dependent variable that is predictable from the independent variable(s). Defined as

$$R^2 = 1 - \frac{\sum_{n=1}^N e_n^2}{\sum_{n=1}^N (y_n - y)^2}\ \text{,where } \bar{y} = \frac{1}{N}\sum_{n=1}^N y_n.$$

Adjusted R-squared: The value for the Adjusted Coefficient of Determination (see corresponding slide about this).

F-statistic: F-statistic value corresponding to an F Test (F Hypothesis Test) for the MR model, intended to assess whether the overall regression model is significant or is no better than simply taking all $\beta_m = 0$, $m = 1,\dots,M$, (see corresponding slides on the $\mathrm{F-Test}$ for both Multiple Regression and SLR as well).

p-value: This is the p-value for the F-test (Significance of Regression test).

```
stop_distance_model = lm(cyl ~ mpg + disp + wt, data =
mtcars)
summary(stop_distance_model)

Call:
lm(formula = cyl ~ mpg + disp + wt, data = mtcars)

Residuals:
     Min       1Q   Median       3Q      Max
-1.15191 -0.41402 -0.02361  0.50677  1.30771

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.064118   1.656523   4.868 3.98e-05 ***
mpg         -0.132156   0.044950  -2.940  0.00651 **
disp         0.011774   0.002363   4.983 2.91e-05 ***
wt          -0.602417   0.319520  -1.885  0.06979.
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.706 on 28 degrees of freedom
Multiple R-squared:  0.8589,	Adjusted R-squared:  0.8437
F-statistic: 56.79 on 3 and 28 DF,  p-value: 5.03e-12
```

Confidence interval for the mean function for multiple regression

We can also, for example, obtain a confidence interval for the value of the mean function

$$\boldsymbol{E}[Y|X = x] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_M x_M. \qquad (62)$$

at new values $x = (x_1, x_2, \ldots, x_M) \in \mathbb{R}^M$. Our unbiased estimate for (62) is of course

$$\hat{y}(x) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_M x_M. \qquad (63)$$

Our confidence interval for $\boldsymbol{E}[Y|X = x]$ at a new value $x = (x_1, x_2, \ldots, x_M) \in \mathbb{R}^M$ is then

$$\hat{y}(x) \ - t\left(\frac{\alpha}{2}, N - M - 1\right) \mathrm{SE}[\hat{y}(x)] \le \boldsymbol{E}[Y|X = x] \le \hat{y}(\mathrm{x}) + t\left(\frac{\alpha}{2}, N - M - 1\right) \mathrm{SE}[\hat{y}(x)], \quad (64)$$

where $\mathrm{SE}[\hat{y}(x)] = s_e\sqrt{x^T(\mathbb{X}^T\mathbb{X})^{-1}x}$ and with $(1-\alpha)$x100% confidence, where, as with SLR, $t\left(\frac{\alpha}{2}, N - M - 1\right)$ is the value that cuts off $\alpha/2$ × 100% in the *upper tail* of the *t*-distribution for $N - M - 1$ degrees of freedom.

R code:
```
model_148= lm(hp ~ mpg + wt, data = mtcars)
 new_cars17 = data.frame(mpg = c(25), wt = c(4))
new_cars17
predict(model_148, newdata = new_cars17, interval = "confidence", level = 0.95)
```

Result:
```
fit          lwr        upr
97.19416   43.25282  151.1355
```

Confidence interval for new values/predictions for multiple regression

We can also, for example, obtain a confidence interval for the random value of the random variable

$$\boldsymbol{E}[Y|X = x] +\epsilon = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_M x_M + \epsilon, \qquad (65)$$

where $\epsilon \sim N(0, \sigma^2)$, at new values $x = (x_1, x_2, \ldots, x_M) \in \mathbb{R}^M$. Our unbiased estimate for (65) is of course

$$\hat{y}(x) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_M x_M \qquad (66)$$

Our confidence interval for $\boldsymbol{E}[Y|X = x] +\epsilon$ at a new value $x = (x_1, x_2, \ldots, x_M) \in \mathbb{R}^M$ can be shown to be

$$\hat{y}(x) - t\left(\frac{\alpha}{2}, N - M - 1\right) \mathrm{SE}[\hat{y}(x) + \epsilon] \leq \boldsymbol{E}[Y|X = x] + \epsilon \leq \hat{y}(\mathrm{x}) + t\left(\frac{\alpha}{2}, N - M - 1\right) \mathrm{SE}[\hat{y}(x) + \epsilon], \quad (67)$$

where $\mathrm{SE}[\hat{y}(x) + \epsilon] = s_e\sqrt{1 + x^T(\mathbb{X}^T\mathbb{X})^{-1}x}$. with $(1 - \alpha)$x100% confidence, where, as with SLR, $t\left(\frac{\alpha}{2}, N - M - 1\right)$ is the value that cuts off $\alpha/2$ × 100% in the *upper tail* of the *t*-distribution for $N - M - 1$ degrees of freedom.

R code:

```
model_148= lm(hp ~ mpg + wt, data = mtcars)
new_cars17 = data.frame(mpg = c(25), wt = c(4))
new_cars17
predict(model_148, newdata = new_cars17, interval = "prediction", level = 0.99)
```

produces the 99% confidence interval

```
        fit        lwr        upr
 97.19416  -45.73935   240.1277
```

Adjusted Coefficient of Determination (R_a^2)

The Coefficient of Determination, R^2 (R-squared), which we covered for SLR applies naturally to the case of Multiple Regression as well, and our exposition of it in a previous slide in connection with SLR applies virtually unchanged for the case of multiple regression. However, one problem with standard $\boldsymbol{R^2}$ as we have defined it there is that it typically and automatically tends to become larger simply by adding more predictor variables. Hence, all else being equal, models with fewer independent variables are penalized, even when they explain the behavior of the response relatively well.

For these reasons the related **Adjusted Coefficient of Determination $\boldsymbol{R_a^2}$** was developed to address some of these issues. Both the R^2 and R_a^2 can be viewed as measures of regression model accuracy. The Adjusted Coefficient of Determination R_a^2 is a goodness-of-fit measure defined via

$$R_a^2 \; = 1 - (1 - R^2)\,\frac{N-1}{N-M-1}\,, \qquad (68)$$

where $R^2 = 1 - \frac{\sum_{n=1}^{N} e_n^2}{\sum_{n=1}^{N}(y_n - \bar{y})^2} = 1 - \frac{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{\sum_{n=1}^{N}(y_n - \bar{y})^2}$,where $\bar{y} = \frac{1}{N}\sum_{n=1}^{N} y_n$. Because of the way R_a^2 is defined in terms of R^2 in (68), much of the behavior of R^2 -- discussed in the slide introducing R^2 is reflected in that of R_a^2. However, the presence of the factor $\frac{N-1}{N-M-1}$ corrects to an extent for the fact that R^2 automatically rises in value if the number of predictors does.

Lecture 8 Overview

Topics in this lecture include:

- Some computational examples/simulations in Multiple Linear Regression with R
- Gauss-Markov Theorem for Multiple Regression
- Comparison of Nested Regression Models
- LINE Assumptions for Multiple Regression
- Initial Steps in Linear Regression Model Diagnostics

Gauss-Markov Theorem for Multiple Regression

Recall our Multiple Regression Model. Given N values $x_1, x_2,\ldots,x_N$, we have

$$Y_n = \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + \cdots + \beta_M x_{nM} + \epsilon_n, n = 1, \ldots, N, \qquad (69)$$

$$\epsilon_n \sim N(0, \sigma^2), n = 1, \ldots, N, \qquad (70)$$

where the ϵ_n are N independent, normally-distributed random variables, as well as our respective estimates $\hat{\beta}_m$ for β_m:

$$(\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_M) = \arg\min_{(\alpha_0,\alpha_1,\ldots,\alpha_M)\epsilon\mathbb{R}^{M+1}} \sum_{i=1}^{N}\left(y_i - (\alpha_0 + \alpha_1 x_{i1} + \cdots + \alpha_M x_{iM})\right)^2, \qquad (71)$$

In the setting of this multiple linear regression model, the **Gauss-Markov Theorem** then asserts that

(1) $\boldsymbol{E}[\hat{\beta}_m] = \beta_m$ (the estimators $\hat{\beta}_m$ for the regression parameters β_m are unbiased)

(2) The $\hat{\beta}_m$ are of minimum variance among all unbiased, linear estimators for β_m. This implies that, among all unbiased, linear estimators α_m, the error $\boldsymbol{E}[(\alpha_m - \beta_m)^2] = \boldsymbol{E}[(\alpha_m - \boldsymbol{E}[\alpha_m])^2]$ is minimized when $\alpha_m = \hat{\beta}_m$.

This shows that the respective estimates $\hat{\beta}_m$ are in an important sense the optimal ones for a fixed number *N* of samples.

Note that a linear estimator in this context means that $\hat{\beta}_m$ can be written as a finite, linear combination of the Y_n(that is, in this context, that we can write $\hat{\beta}_i = \sum_{n=1}^{N} k_{in} Y_n, i = 1{,}2$, for some constant coefficients k_{in}.

Comparison of Nested Regression Models

The significance of regression test (F-test) is actually a special case of comparing what are known as **nested models**. Indeed we can compare two linear regression models, where one model is “nested” inside the other, meaning the set of predictors of one model is a subset of the set of predictors of the larger one.

Consider the following full model

$$Y_n=\beta_0+\beta_1 x_{n1}+\beta_2 x_{n2}+\cdots+\beta_M x_{iM}+\epsilon_{n,}\ n=1,\ldots,N.$$

This model has M predictors, for a total of (M+1) β-parameters. We will denote the fitted values of this model as $\hat{y}_{1n}$,n=1,…,N.

We can denote the “smaller”, null-hypothesis model by

$$Y_n=\beta_0+\beta_1 x_{n1}+\beta_2 x_{n2}+\cdots+\beta_H x_{nH}+\epsilon_n,\ n=1,\ldots,N,$$

where H<M. We denote the fitted values of this model as $\hat{y}_{0n}$,n=1,…,N.

The difference between these two models can be codified by a statistical test:

H_0:$\beta_{H+1}=\cdots=\beta_M=0$. H_1: NOT ($\beta_{H+1}=\cdots=\beta_M=0$).

We can then perform the test using an F-test with the F-statistic defined as

$$F=\frac{\sum_{i=1}^{N}(\hat{y}_{1i}-\hat{y}_{0i})^2/(M-H)}{\sum_{i=1}^{N}(y_i-\hat{y}_{1i})^2/(N-M)}.$$

In R:

```
null_mpg_model = lm(mpg ~ wt + disp, data = mtcars)
full_mpg_model = lm(mpg ~ wt + disp + hp, data = mtcars)
anova(null_mpg_model, full_mpg_model)
```

Output:

```
Analysis of Variance Table
Model 1: mpg ~ wt + disp
Model 2: mpg ~ wt + disp + hp
  Res.Df    RSS Df  Sum of Sq    F  Pr(>F)
1     29 246.68
2     28 194.99  1    51.692  7.4228    0.01097 *
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

LINE Assumptions for Multiple Linear Regression

Recall our Simple Linear Regression Model. Give N fixed sample vectors $(x_{11}, \dots, x_{1M})$, $(x_{21},, \dots, x_{2M})$,…,$(x_{N1}, \dots, x_{NM})$ we have

$$Y_n = \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + \cdots + \beta_M x_{nM} + \epsilon_n, \qquad (72)$$

$$\epsilon_n \sim N(0, \sigma^2), n = 1, \dots, N \qquad (73)$$

where the noise terms ϵ_n are N independent, normally-distributed random variables.

The main assumptions of this model are frequently denoted by means of the mnemonic acronym **LINE**:

Linearity: The relationship between each Y_n and each x_n, respectively, is linear, and $\boldsymbol{E}[Y_n] = \boldsymbol{E}[Y_n | X_n = x_n] = \beta_0 + \beta_1 x_n + \beta_2\, x_{n2} + \cdots + \beta_M x_{nM}$for all $n = 1, \dots, N$.

Independence: The errors $\epsilon_n, n = 1, \dots, N$, are independent random variables.

Normality: The errors $\epsilon_n, n = 1, \dots, N$, follow a normal distribution. That is, the error across the regression line at any point x_n is described by a normal distribution.

Equal Variance: The normal distribution describing the behavior of the ϵ_n has the same variance, σ^2, for all n. This property is called *homoscedasticity.*

Note that the first or “L” assumption implies that $\boldsymbol{E}[\epsilon_n] = \boldsymbol{E}[Y_n - (\beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + \cdots + \beta_M x_{nM})] = 0$.

Regression Model Diagnostics

At this point in the course we address Regression Model Diagnostics. In Model Diagnostics, we check the validity of the regression model we use to fit the data in question as well as assess underlying model assumptions using graphical visualizations as well as (a little later on) formal statistical tests.

- The basic tools here are the residuals.
- If the fitted model does not give a set of residuals that appear to be reasonable, then some characteristic of the model may be called into doubt.
- Such characteristics of the model may include the assumed mean function or assumptions concerning the variance function.

This will set the stage for our treatment of Model Transformations, with respect to which we attempt to modify the regression model so that it better fits the data.

Lecture 9 Overview

Topics in this lecture include:

- Linear Regression Model Diagnostics
- Variable Transformations: Variance-Stabilizing Transformations
- Intro to Logarithmic Transformation Computational Example

Regression Model Diagnostics

At this point in the course we address Regression Model Diagnostics. In Model Diagnostics, we check the validity of the regression model we use to fit the data in question as well as assess underlying model assumptions using graphical visualizations as well as (a little later on) formal statistical tests.

- The basic tools here are the residuals.
- If the fitted model does not give a set of residuals that appear to be reasonable, then some characteristic of the model may be called into doubt.
- Such characteristics of the model may include the assumed mean function or assumptions concerning the variance function.

This will set the stage for our treatment of Model Transformations, with respect to which we attempt to modify the regression model so that it better fits the data.

Transformations: Variance-Stabilizing Transformations

It seems likely from the plots of the salary vs. seniority data (see Lec 9/Lec 10 videos or the following slides) that the variance of Y actually depends on the x-value, and so the model as is not homoscedastic:

$$\mathrm{Var}[Y|X = x] = f(x),$$

for some non-constant function f. Indeed, f appears to be an increasing function of x. To remedy this so that we can reconcile our LINE assumptions with these data, we seek a function g for which

$$\mathrm{Var}[g(Y)|X = x] = c, \text{a constant.}$$

A function g achieving this is called a **Variance Stabilizing Transformation (VST).**

A common VST to apply when we see increasing variance in a fitted versus residuals plot is log(Y)=log$_e$(Y). Note that, if the values of a variable range over more than one order of magnitude and the variable is strictly positive, then replacing the variable by its logarithm is likely to be helpful. So we apply a Logarithmic Transformation to the response variable within the standard Simple Linear Regression framework:

$$\log(Y_n) = \beta_0 + \beta_1 x_n + \epsilon_n, n = 1, \ldots, N,$$

Transformations: Variance-Stabilizing Transformations (cont’d)

We modify our code in R in the following way to solve the model with the variance-stabilizing transformation:

```
> model_fit_log = lm(log(salary) ~ years, data = model)
```

Note that while log(y) is considered the new response variable, we do not actually create a new variable in R, but simply transform the variable inside the model formula. By plotting the data on the original scale, and adding the fitted regression, we see an exponential relationship. However, this is still a *linear* model, since the new transformed response, $\log(y)=\exp^{-1}(y)$, is still a *linear* combination of the predictors.

Logarithmic Transformations: Example

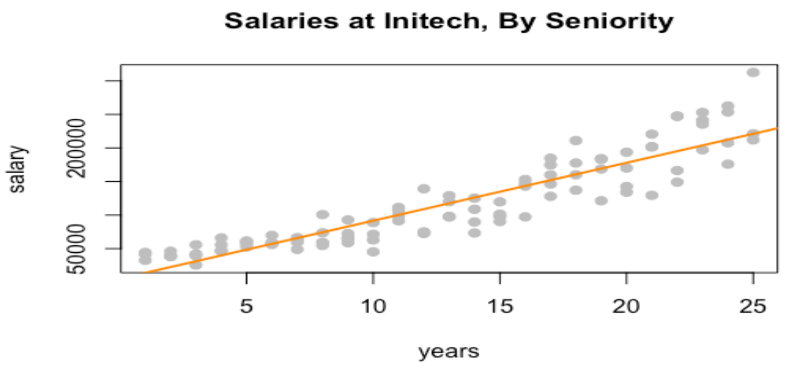
We can plot salary vs. years at a company with a suitable dataset:

```
initech67 = read.csv("data/initech.csv")
```

(This dataset is the same as the “alltech” dataset in the lecture videos.)

First, solve (for the parameters in) the model and generate the summary report (at right).

According to the R-squared the fit is a reasonably close or accurate one and the F-test shows that the regression model to be significant overall as well. But, we wish to examine model assumptions.



From the plot with the fitted orange regression line, we see that the linear relationship at least appears (very roughly) perhaps approximately correct.

Summary Report:

```
initech67_fit = lm(salary ~ years, data = initech67)
summary(initech_fit)
```

Produces:

```
 Residuals: Min  1Q  Median  3Q   Max
            -57225  -18104  241  15589  91332
              Estimate Std. Error  t value  Pr(>|t|)
 (Intercept) 5302  5750  0.922  0.359
Years           8637  389  22.200  <2e-16 ***
 Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 27360 on 98 degrees of freedom
Multiple R-squared: 0.8341, Adjusted R-squared: 0.8324
 F-statistic: 492.8 on 1 and 98 DF, p-value: < 2.2e-16
```

Lecture 10 Overview

Topics in this lecture include:

- Variable Transformations: Variance-Stabilizing Transformations
- Logarithmic Transformation Computational Example(s) with R

Transformations: Variance-Stabilizing Transformations

It seems likely from the plots of the salary vs. seniority data (see Lec 9/Lec 10 videos or the following slides) that the variance of Y actually depends on the x-value, and so the model as is not homoscedastic:

$$\mathrm{Var}[Y|X = x] = f(x),$$

for some non-constant function f. Indeed, f appears to be an increasing function of x. To remedy this so that we can reconcile our LINE assumptions with these data, we seek a function g for which

$$\mathrm{Var}[g(Y)|X = x] = c, \text{a constant.}$$

A function g achieving this is called a **Variance Stabilizing Transformation (VST).**

A common VST to apply when we see increasing variance in a fitted versus residuals plot is log(Y)=log$_e$(Y). Note that, if the values of a variable range over more than one order of magnitude and the variable is strictly positive, then replacing the variable by its logarithm is likely to be helpful. So we apply a Logarithmic Transformation to the response variable within the standard Simple Linear Regression framework:

$$\log(Y_n) = \beta_0 + \beta_1 x_n + \epsilon_n, n = 1, \dots, N,$$

Transformations: Variance-Stabilizing Transformations (cont’d)

We modify our code in R in the following way to solve the model with the variance-stabilizing transformation:

```
> model_fit_log = lm(log(salary) ~ years, data = model)
```

Note that while log(y) is considered the new response variable, we do not actually create a new variable in R, but simply transform the variable inside the model formula. By plotting the data on the original scale, and adding the fitted regression, we see an exponential relationship. However, this is still a *linear* model, since the new transformed response, log(y)=exp^{-1}(y), is still a *linear* combination of the predictors.

Logarithmic Transformations: Example

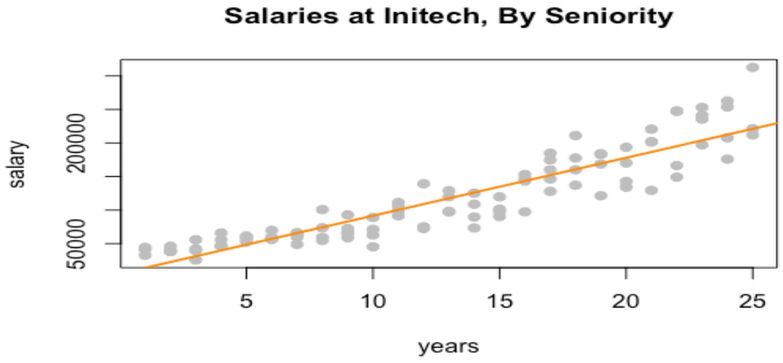
We can plot salary vs. years at a company with a suitable dataset:

```
initech67 = read.csv("data/initech.csv")
```

(This dataset is the same as the “alltech” dataset in the lecture videos.)

First, solve (for the parameters in) the model and generate the summary report (at right).

According to the R-squared the fit is a reasonably close or accurate one and the F-test shows that the regression model to be significant overall as well. But, we wish to examine model assumptions.



From the plot with the fitted orange regression line, we see that the linear relationship at least appears (very roughly) perhaps approximately correct.

Summary Report:

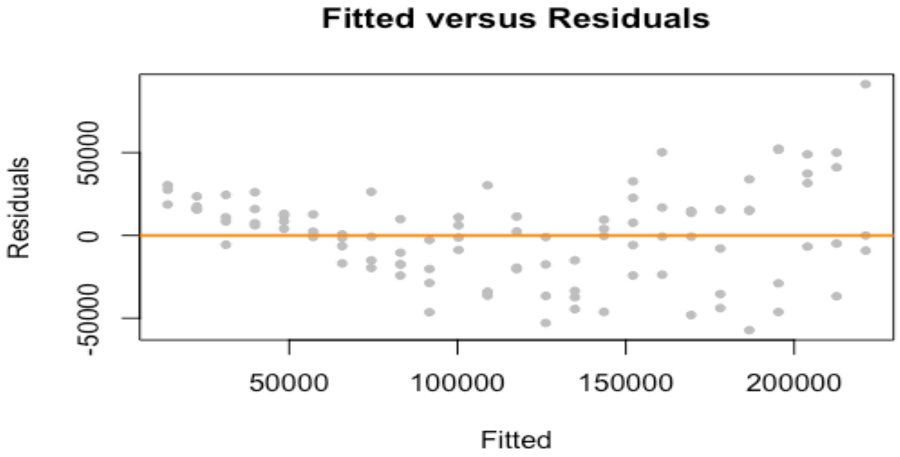
```
initech67_fit = lm(salary ~ years, data = initech67)
summary(initech_fit)
```

Produces:

```
 Residuals: Min  1Q  Median  3Q   Max
            -57225  -18104  241  15589  91332
              Estimate Std. Error  t value  Pr(>|t|)
 (Intercept) 5302  5750  0.922  0.359
Years           8637  389  22.200  <2e-16 ***
 Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 27360 on 98 degrees of freedom
Multiple R-squared: 0.8341, Adjusted R-squared: 0.8324
 F-statistic: 492.8 on 1 and 98 DF, p-value: < 2.2e-16
```

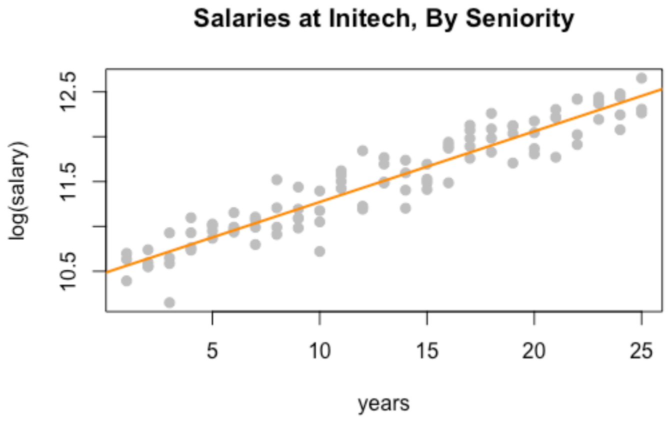
Logarithmic Transformations: Example

```
plot(fitted(initech67_fit), resid(initech67_fit), col = "grey", pch = 20,
xlab = "Fitted", ylab = "Residuals", main = "Fitted versus Residuals")
abline(h = 0, col = "darkorange", lwd = 2)
```



However, from the fitted versus residuals plot it appears there is non-constant variance. Specifically, the variance increases as the fitted value increases.

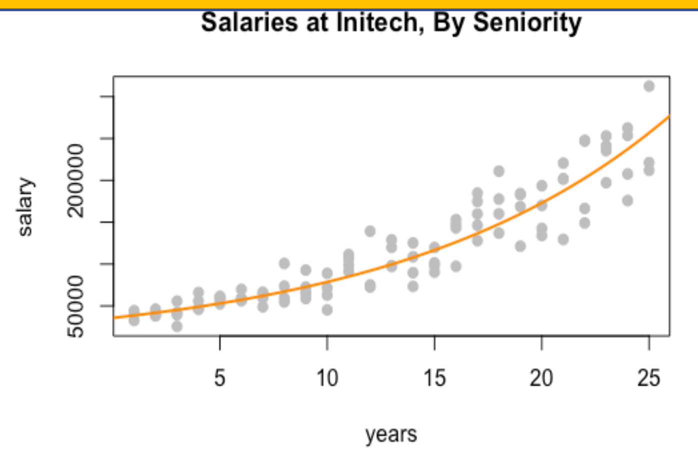
```
initech67_fit_log = lm(log(salary) ~ years, data = initech67)

plot(log(salary) ~ years, data = initech67, col = "grey", pch = 20, cex = 1.5,
    main = "Salaries at Initech, By Seniority")
abline(initech67_fit_log, col = "darkorange", lwd = 2)
```



Plotting the data on the transformed log scale and adding the fitted line, the relationship again appears linear, and we can already see that the variation about the fitted line looks constant.

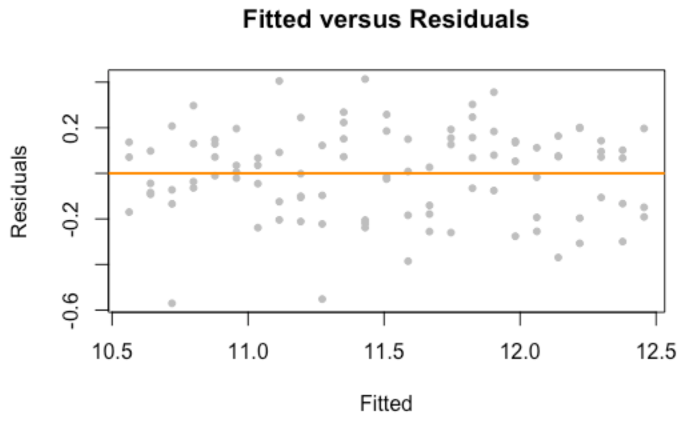
Logarithmic Transformations: Example

```
plot(salary ~ years, data = initech67, col = "grey", pch = 20, cex = 1.5,
   main = "Salaries at Initech, By Seniority")
curve(exp(initech67_fit_log$coef[1] + initech67_fit_log$coef[2] * x),
    from = 0, to = 30, add = TRUE, col = "darkorange", lwd = 2)
```



By plotting the data on the original scale, and adding the fitted regression, we see an exponential relationship. However, this is still a *linear* model, since the new transformed response, log(y), is still a *linear* combination of the predictors.

```
plot(fitted(initech67_fit_log), resid(initech67_fit_log), col = "grey", pch = 20,
    xlab = "Fitted", ylab = "Residuals", main = "Fitted versus Residuals")
abline(h = 0, col = "darkorange", lwd = 2)
```



The fitted versus residuals plot looks much better. It appears the constant variance assumption is no longer violated.

(This example as well as some of the others is based on one described at https://daviddalpiaz.github.io/appliedstats/simple-linear-regression.html#decomposition-of-variation.)

Logarithmic Transformations: Example – RMSE Assessment

Comparing the RMSE (Root-Mean-Square Error),

where $\mathrm{RMSE} = \sqrt{\frac{\sum_{n=1}^{N} e_n^2}{N}} = \sqrt{\frac{\sum_{n=1}^{N}(\hat{y}_n - y_n)^2}{N}}$,

of the original and transformed responses of the previous slides, we also see that the log transformed model simply fits better, with a smaller average squared error. In R code:

- **sqrt**(**mean**(**resid**(initech_fit) ^ 2))

  27080.16
- **sqrt**(**mean**(**resid**(initech_fit_log) ^ 2))

  0.1934907

This difference is due to a great extent to the different scales being used. But consider that

- **sqrt**(**mean**((initech$salary - **fitted**(initech_fit)) ^ 2))

  27080.16
- **sqrt**(**mean**((initech$salary - **exp**(**fitted**(initech_fit_log))) ^ 2))
- 24280.36

Transforming the fitted values of the log model back to the data scale, we do indeed see that the transformed model legitimately fits better.

Lecture 11 Overview

- Short review of polynomial (quadratic) regression model diagnostics
- So-called Box-Cox variable transformations, which provide an algorithmic approach based on maximum likelihood estimation for identifying a power transformation of the response variable that offers a potentially optimal fit.
- Formal statistical hypothesis test for identifying Heteroscedasticity
- Formal statistical hypothesis test for assessing normality of the residuals

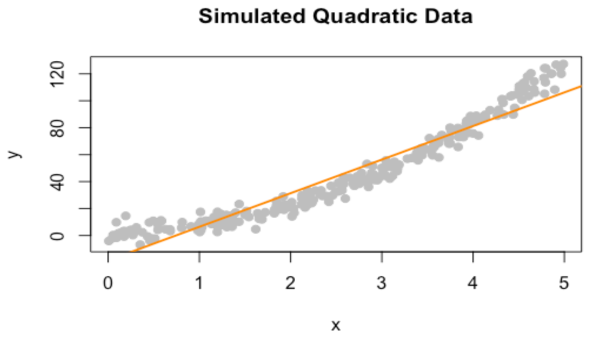
Example: Polynomial Regression Model Diagnostics

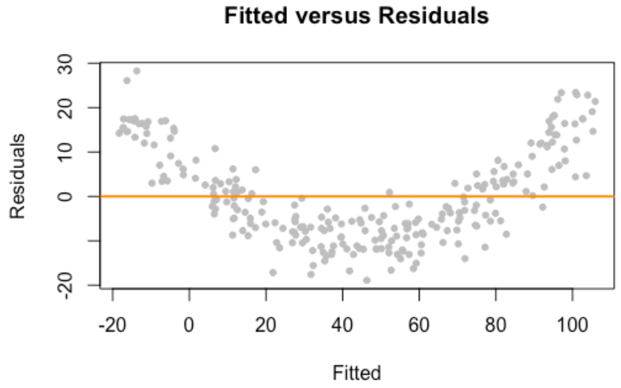
We generate synthetic ("artificial") data according to the following polynomial (quadratic) regression model:

$$Y_n = \beta_0 + \beta_1 x_n^2 + \epsilon_n, n = 1, \dots, N,$$
$$\epsilon_n \sim N(0, \sigma^2), n = 1, \dots, N.$$

The R code for generating such data with β_0=3, β_1=5, N=250 and also solving the corresponding linear regression model is

```
sim_quad = function(sample_size = 250) {
    x = runif(n = sample_size) * 5
    y = 3 + 5 * x ^ 2 + rnorm(n = sample_size, mean = 0, sd = 5)
    data.frame(x, y)
}
set.seed(314)
quad_data = sim_quad(sample_size = 250)
lin_fit = lm(y ~ x, data = quad_data)
plot(y ~ x, data = quad_data, col = "grey", pch = 20, cex = 1.5,
  main = "Simulated Quadratic Data")
abline(lin_fit, col = "darkorange", lwd = 2)
plot(fitted(lin_fit), resid(lin_fit), col = "grey", pch = 20,
  xlab = "Fitted", ylab = "Residuals", main = "Fitted versus Residuals")
abline(h = 0, col = "darkorange", lwd = 2)
```

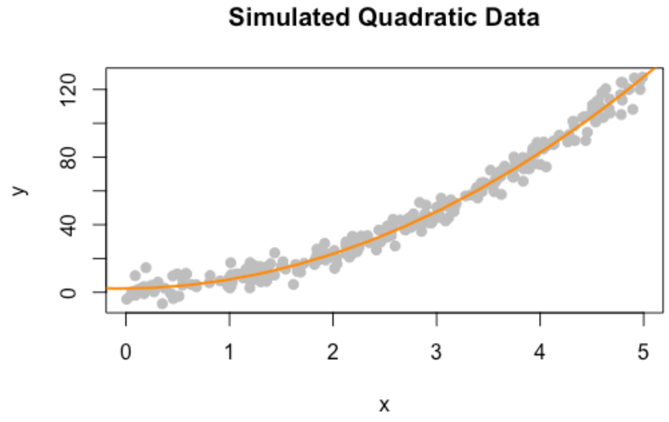


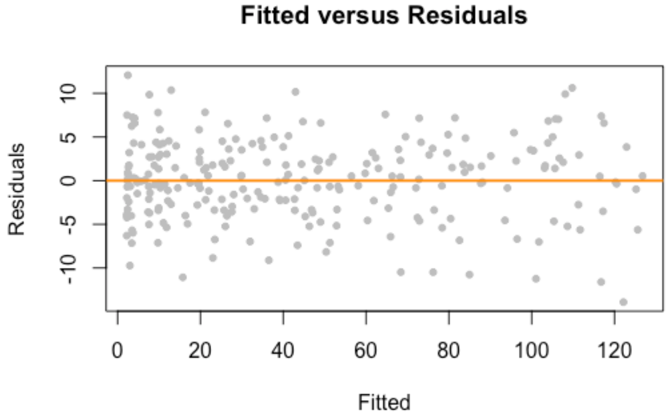


The linear (SLR) model on this slide appears a poor fit from the direct x-y plot at right, but even more so from the fitted vs. residuals plot. Of course this is as we would expect since the data was generated based on a quadratic model as above.

Example: Polynomial Regression Model Diagnostics (cont’d)

```
quad_fit = lm(y ~ x + I(x^2), data = quad_data)


plot(y ~ x, data = quad_data, col = "grey", pch = 20, cex = 1.5,
    main = "Simulated Quadratic Data")
curve(quad_fit$coef[1] + quad_fit$coef[2] * x + quad_fit$coef[3] * x ^ 2,
     from = -5, to = 30, add = TRUE, col = "darkorange", lwd = 2)


plot(fitted(quad_fit), resid(quad_fit), col = "grey", pch = 20,
    xlab = "Fitted", ylab = "Residuals", main = "Fitted versus Residuals")
abline(h = 0, col = "darkorange", lwd = 2)
```





Both the direct x-y plot but in particular the new fitted vs. residuals plot at right show a much better fit, which of course is to be expected because we are now attempting to fit data generated according to a quadratic polynomial model with a quadratic regression model. Note that the R^2 value for the linear model is 0.9207, but for the quadratic one it rises to 0.9846. Of course we can address data generated according to cubic (or higher) polynomial models in analogous ways.

Transformations: Box-Cox Approach

It is often difficult to determine from diagnostic plots which transformation of the response *Y* is most appropriate for correcting skewness of the distributions of error terms, unequal error variances, and nonlinearity of the regression function. The Box-Cox procedure automatically identifies a transformation from a family of power transformations on *Y.*

The Box-Cox transformation approach considers a family of transformations on

strictly positive response variables:

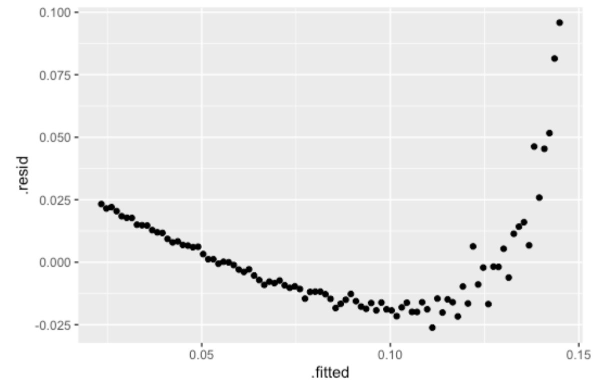
$$g_\lambda(\mathrm{y}) = \begin{cases} \frac{y^\lambda - 1}{\lambda}, \lambda \neq 0 \\ \log(y)\,, \lambda = 0, \end{cases}$$

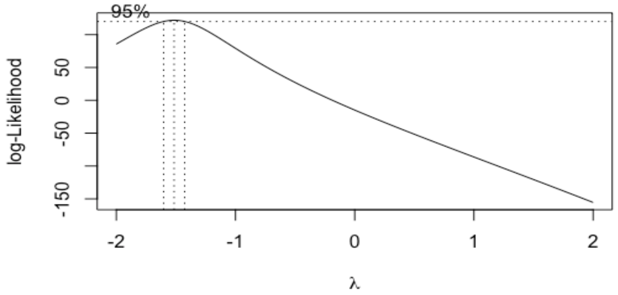
where the λ parameter is chosen from the data samples by numerically maximizing a suitable log-likelihood function.

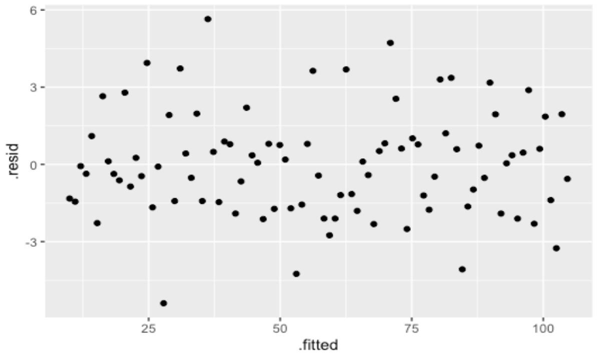
Box-Cox Power Transformations: Example

```
library(MASS)
set.seed(9)
x <- 10:100
eps <- rnorm(length(x), sd = 2)
y <- (x + eps) ^ (-1 / 1.5)
datafr72 <- data.frame(y,x)
m <- lm(y ~ x)
summary(m)
library(broom)
augmented_m <- augment(m)
library(tidyverse)
ggplot(augmented_m, aes(x = .fitted, y = .resid)) +
geom_point()
bc <- boxcox(m)
lambda <- bc$x[which.max(bc$y)]
lambda
z <- y ^ lambda
m2 <- lm(z ~ x)
summary(m2)
m2 <- lm(I(y ^ lambda) ~ x)
augmented_m2 <- augment(m2)
ggplot(augmented_m2, aes(x = .fitted, y = .resid)) +
 geom_point()
```

One can typically use the Box-Cox method to identify a suitable power transform $h(y) = y^{\lambda}$ to apply to the response variable to obtain a good or better regression model fit to data. For example, we use the first lines of the R code to the left to generate "artificial" data samples according to y = (x + ϵ) ^ (-1 / 1.5), so that here $\lambda = -1.5$. Modeling the data, incorrectly, with a standard simple linear regression model gives the fitted vs. residual plot at upper right, which shows a clear, curved pattern as well as heteroskedasticity – not what we want in a such plot accurately capturing the dynamics of a suitable underlying model. So with the rest of the code we seek to test the Box-Cox method to see if it can recover this exponent value. In this case Box-Cox generates the λ vs. log-likelihood plot to the right graphically showing the approximate value (along with a confidence interval) for the optimal λ (according to Box − Cox). We can also generate it numerically with the R command(s) lambda <- bc$x[which.max(bc$y)]\lambda. We find here that Box-Cox gets $\lambda =$ -1.515152. Replotting the fitted vs. residual graph with this value of λ results in the plot at bottom right, with an even, symmetric distribution of points about the residual axis 0 line. We note too that if one looks at the respective Coefficient of Determination R^2 values in the summary output reports for the two models (the standard simple linear one vs. Box − Cox with $\lambda =$ -1.515152), there is also a significant improvement with respect to the latter model.







Formal Test for Homoscedasticity

While a fitted versus residuals plot can give us an idea about homoscedasticity, sometimes we would prefer a more formal test. There are many tests for constant variance, but here we apply one, the **Breusch-Pagan Test**, which is derived from the Lagrange multiplier test from basic calculus. The null and alternative hypotheses for this statistical hypothesis test can respectively be considered to be:

- H_0: Homoscedasticity. The errors have constant variance about the true model.
- H_1: Heteroscedasticity. The errors have non-constant variance about the true model.

```
> #install.packages("lmtest")
> library(lmtest)
> model_1 = lm(y ~ x, data = sim_data_1)
> bptest(model_1)
```

The p-value of 0.05 may be used as a threshold for determining acceptance or rejection of the null hypothesis.

Formal Test for Normality

We can also describe a formal statistical hypothesis test we can use to determine whether the residuals are in fact normally distributed. This is the **Shapiro-Wilk test**:

- H_0: Normality. The data is consistent with having been sampled according to a normal distribution.
- H_1: Non-Normality. The data is not consistent with having been generated according to a normal distribution.

For example, we can apply this normality test to the residuals associated to a regression model:

```
>model_1 = lm(y ~ x, data = sim_data_1)
>shapiro.test(resid(model_1))
```

The p-value of 0.05 may be used as a threshold for determining acceptance or rejection of the null hypothesis.

Lecture 12 Overview

- More on the application (in the context of empirical examples with R) of certain formal statistical tests for identifying heteroscedasticity and assessing normality of the residuals
- Introduction to ANOVA (so-called “Analysis of Variance”), which can be used for example to assess whether population means from different groups really are different in a statistically significant way
- A look back at aspects of computationally justifying the sampling distributions which we had previously asserted are followed by the estimators $\hat{\beta}_0$ and $\hat{\beta}_1$ (in particular, their mean and variance values)

ANOVA: Comparison of the Response Means of Different Groups

Analysis of Variance (ANOVA) is a collection of statistical models and their associated estimation procedures used to analyze the differences among group means in a sample. ANOVA is based on the the law of total variance, where the observed variance in a particular variable is partitioned into components attributable to different sources of variation. In its simplest form, ANOVA provides a statistical test of whether two or more population means are equal, and therefore can involve generalizing the *t*-test beyond two means.

To illustrate ANOVA with two populations or groups, we consider the following example. One group may be called the **control**, while the other the **treatment**. Subjects (samples) are randomly assigned to one of the two groups. After being assigned to a group, each subject has some quantity measured, which is the response variable.

Mathematically we consider the following model:

$$y_{ij} \sim N(\mu_i, \sigma^2), i = 1,2,$$

where *j=1,2,…n*$_{i,}$ with *n*$_i$ being the number of subjects in group *i* and with the y_{ij} being independent.

ANOVA: Comparison of the Response Means of Different Groups (cont’d)

Measurements of subjects in group 1 we can say follow a normal distribution with mean μ_1:

$$y_{1j} \sim N(\mu_1, \sigma^2),$$

Measurements of subjects in group 2 we can say follow a normal distribution with mean μ_2:

$$y_{2j} \sim N(\mu_2, \sigma^2).$$

A natural question to ask is: Are the means of the two groups different? Mathematically, that can be interpreted as:

$$H_0: \mu_1 = \mu_2 \text{ vs. } H_1: \mu_1 \neq \mu_2$$

For the stated model and assuming the null hypothesis is true, the corresponding t-test statistic would follow a t-distribution with degrees of freedom n_1+n_2-2. As an example, suppose we are interested in the effect of the compound melatonin on sleep duration. A researcher obtains a random sample of 20 adult males. Of these subjects, 10 are randomly chosen for the control group, which will receive a placebo. The remaining 10 will be given 5mg of melatonin before bed. The sleep duration in hours of each subject is then measured. The researcher chooses a significance level of $\alpha=0.10$. Was sleep duration affected by the melatonin?

ANOVA: One-way ANOVA for an Arbitrary Number of Groups

The One-Way ANOVA Test is used in the context of just a single independent variable – hence “one way”. This test can be used to check whether the various means of the response variables associated to data sampled from an arbitrary number of group populations are equal.

H_0:μ_1=μ_2=…$\mu_{g.}$ vs. H_1: Not all μ_i are equal.

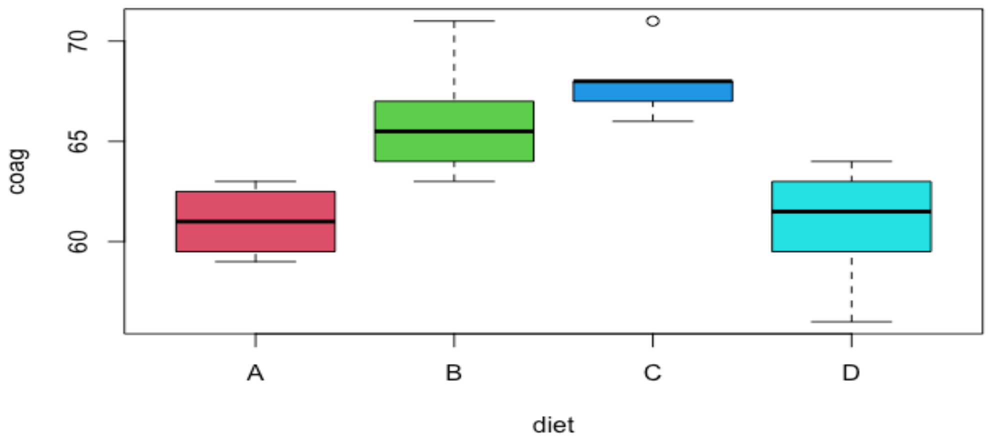
Notice that the alternative simply indicates the some of the means are not equal, not specifically which are not equal.

This test is called **Analysis of Variance**. Analysis of Variance (ANOVA) compares the variation due to specific sources (between groups) with the variation among individuals who should be similar (within groups). In particular, ANOVA tests whether several populations have the same mean by comparing how far apart the sample means are with how much variation there is within the samples. We use variability of means to test – using a statistical F-test similar to the one we already used for significance of regression – to address equality of means, thus the use of *variance* in the name. We first load the data and create the relevant boxplot. The plot alone suggests a difference of means.

The aov() function is used to obtain the relevant sums of squares. Using the summary() function on the output from aov() creates the desired ANOVA table.

Notice that the p-value of this test is incredibly low, so using any reasonable significance level we would reject the null hypothesis. Thus we believe the diets had an effect on blood coagulation time.

```
> library(faraway)
  plot(coag ~ diet, data = coagulation, col = 2:5)
  coag_aov = aov(coag ~ diet, data = coagulation)
  summary(coag_aov)
```



```
#> Df   Sum Sq   Mean    Sq  F value   Pr(>F)
   3    228       76.0     13.57         4.66e-05 ***
```

Lecture 13 Overview

- **More on Polynomial Regression models (Polynomial Transformations) using empirical examples with R**

Polynomial Regression Example(s)

We work with the following polynomial regression model

$$Y_n = \beta_0 + \beta_1 x_n + \beta_2 x_n^2 + \cdots + \beta_M x_n^M + \epsilon_n, n = 1, \ldots, N, \qquad \text{(A)}$$

for some positive integral order M, which as can be seen is a sum of predictor variable (polynomial) transformations and can be interpreted as in essence a form of multiple regression.

For *M*=4, for example, we can solve the model (A) above using the lm command in the following way:

```
> fit_perf = lm(y ~ x + I(x ^ 2) + I(x ^ 3) + I(x ^ 4))          (B)
```

In the next slide (and the corresponding lecture videos, by the way) we go into more depth with models of this kind and analyze their performance based on the results of statistical hypothesis tests and/or other metrics. The goal in part is to consider what order of polynomial might be best to model the data of interest.

Polynomial Regression: Example

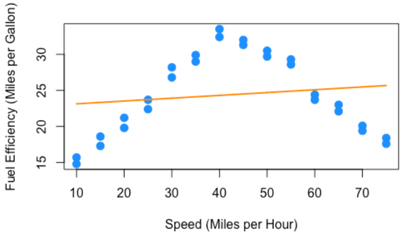
R code:

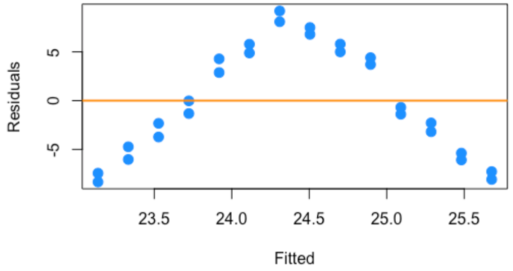
```
econ = read.csv("data/fuel_econ.csv")
plot_econ_curve = function(model) {
  plot(mpg ~ mph, data = econ, xlab = "Speed (Miles per Hour)",
       ylab = "Fuel Efficiency (Miles per Gallon)", col = "dodgerblue",
       pch = 20, cex =2)
xplot = seq(10, 75, by = 0.1)
 lines(xplot, predict(model, newdata = data.frame(mph = xplot)),
       col = "darkorange", lwd = 2, lty = 1)
 }
fit1 = lm(mpg ~ mph, data = econ)
Summary(fit1)
plot_econ_curve(fit1)
plot(fitted(fit1), resid(fit1), xlab = "Fitted", ylab = "Residuals", col = "dodgerblue",
pch = 20, cex =2)
 abline(h = 0, col = "darkorange", lwd = 2)
fit4 = lm(mpg ~ mph + I(mph ^ 2) + I(mph ^ 3) + I(mph ^ 4), data = econ)
summary(fit4)
plot_econ_curve(fit4)
plot(fitted(fit4), resid(fit4), xlab = "Fitted", ylab = "Residuals",
    col = "dodgerblue", pch = 20, cex =2)
abline(h = 0, col = "darkorange", lwd = 2)
```

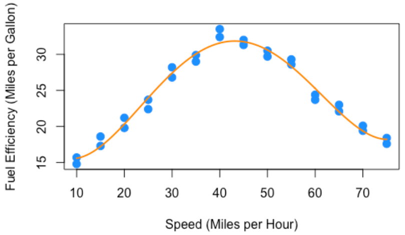
A standard simple linear regression model does a poor job in modeling the data at upper right (also see the fitted vs. residuals chart just below that backs this assertion up). Moreover, we have, for this model,
Multiple R-squared: 0.02039,
Adjusted R-squared: -0.01729,
very low values indicating a poor fit. The residuals distribution in the summary output report as well as the t- and F-test hypothesis test values also look poor. However using a polynomial regression model instead with

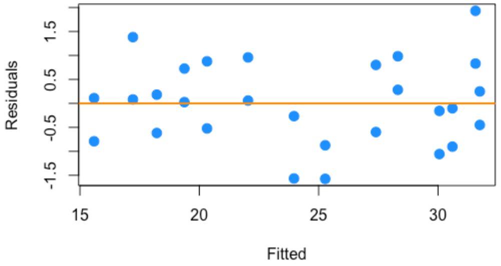
$$Y=\beta_4x^4+ \beta_3x^3+ \beta_2x^2+ \beta_1x+ \beta_0,$$

we get the much better second pair of graphs to the right, with an improvement of the R-squared numbers to
Multiple R-squared: 0.9766,
Adjusted R-squared: 0.9726,
as well as for the other statistics in the summary report.









Lecture 14 Overview

- Some aspects of dealing with categorical variables in linear regression (and in particular with R)
- A few words concerning multiple linear regression with an interaction term
- Variable selection and model building (this corresponds to Chapter 10 in the Weisberg(2014) text)

Some aspects of the use of Categorical Variables in R

How to deal with **categorical variables?**

Most or all of the data we have encountered in this course thus far have been numeric. Indeed, more than that, the data have been metric data – variables/data for which a notion of numerical distance is naturally inherent (e.g., people's heights or, also, car speed data).

Categorical variables are those which may be non-numeric or, if numeric, the numerical values which they take on are essentially arbitrary in nature. So, for example, a variable taking on, say, the values "T" or "F" for "True" or "False" is categorical, but such a variable would still be categorical – and not a metric variable -- even if it took on the values "1" or "0" in place of "T" or "F".

A categorical variable given (an essentially arbitrary) numerical value is said to be transformed into a "**dummy variable**". In R, the "**factor"** variable-type allows for essentially automatic handling of categorical variables as dummy variables.

Multiple Linear Regression with Interaction

Loosely speaking, interaction effects occur in regression when the influence of an independent variable on the target variable depends on the behavior of another independent variable.

Mathematically, a **Multiple Linear Regression model with an Interaction term** is described in the following way:

$$Y_n = \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + \beta_3 x_{n1} x_{n2} + \epsilon_n, n = 1, \ldots, N,$$
$$\epsilon_n \sim N(0, \sigma^2), n = 1, \ldots, N.$$

Variable Selection and Model Building

- Our goal here is to describe systematic algorithmic strategies that can optimize the trade-off between goodness-of-fit and excess model complexity.
- Indeed, as we have seen, ordinary least-squares regression is used to choose optimal model coefficient parameters once a regression model has been chosen. Our objective now (and with for example Stepwise Regression which we describe in what follows) is to identify an (approximately) optimal regression model itself -- that is, for example, to identify which predictor variables and/or regressors can be used to build a (multiple) regression model that best fits the data and eliminate those that can’t be.

The **Akaike Information Criterion (AIC)** can be applied to promote goodness-of-fit of a candidate regression model while limiting excess model complexity, which can in turn lead to model overfitting (overdetermination):

$$\mathrm{AIC} = N\log((\textstyle\sum_{n=1}^{N}(y_n - \hat{y}_n)^2)/N) + 2M,$$

where M+1 is the number of $\beta -$ coefficients in the (multiple) regression model, and N is as usual the number of data samples.

The related **Bayesian Information Criterion (BIC)** is defined as follows:

$$\mathrm{BIC} = N\log((\textstyle\sum_{n=1}^{N}(y_n - \hat{y}_n)^2)/N) + \log(N)\, M.$$

Lecture 15 Overview

- **More on Variable selection and model building (this corresponds to Chapter 10 in the Weisberg(2014) text)**
- **Description in some detail of the Stepwise Regression method**

Variable Selection and Model Building

- Our goal here is to describe systematic algorithmic strategies that can optimize the trade-off between goodness-of-fit and excess model complexity.
- Indeed, as we have seen, ordinary least-squares regression is used to choose optimal model coefficient parameters once a regression model has been chosen. Our objective now (and with for example Stepwise Regression which we describe in what follows) is to identify an (approximately) optimal regression model itself -- that is, for example, to identify which predictor variables and/or regressors can be used to build a (multiple) regression model that best fits the data and eliminate those that can't be.

The **Akaike Information Criterion (AIC)** can be applied to promote goodness-of-fit of a candidate regression model while limiting excess model complexity, which can in turn lead to model overfitting (overdetermination):

$$\mathrm{AIC} = N\log\left(\left(\sum_{n=1}^{N}(y_n - \hat{y}_n)^2\right)/N\right) + 2M,$$

where M+1 is the number of $\beta -$ coefficients in the (multiple) regression model, and N is as usual the number of data samples.

The related **Bayesian Information Criterion (BIC)** is defined as follows:

$$\mathrm{BIC} = N\log\left(\left(\sum_{n=1}^{N}(y_n - \hat{y}_n)^2\right)/N\right) + \log(N)\, M.$$

Stepwise Regression

Stepwise Regression is a procedure we can use to build a regression model from a set of candidate predictor variables by adding and removing predictors in a stepwise manner into the model until there is no statistically valid reason to enter or remove anymore.

The goal of stepwise regression is to build a regression model that includes all of the predictor variables that are statistically significantly with respect to the response variable.

Backward and Forward Search are similar and related to the Stepwise Regression algorithm. In Forward Search we just add predictors and in Backward Search we just remove. Exhaustive Search checks all combinations of candidate predictors but consequently requires more computational processing time.

Stepwise Regression: Understanding the Stepwise Regression Procedure

The general procedure for stepwise regression is as follows:

- **Step1:**
- Start with the intercept-only model. That is, start with no predictors in the model.
- **Step 2:**
- Fit each of the one-predictor models and choose the one that produces the lowest AIC (Akaike information criterion), which is a measure of the quality of the regression model relative to all other models. That is, fit the model $y \sim x_1$, then fit the model $y \sim x_2$, then fit the model $y \sim x_3$, then fit the model $y \sim x_4$, and keep going until you have fit all one-predictor models.
- Choose the model that produces the lowest AIC value. If no model produces an AIC value that is significantly different from the intercept-only model, then stop.

Stepwise Regression: Understanding the Stepwise Regression Procedure (cont'd)

Step 3:

- Suppose $y \sim x_1$ produced the model with the lowest AIC. Next, fit each of the two-predictor models that includes x_1 as a predictor. That is, fit the model $y \sim x_1 + x_2$, then fit the model $y \sim x_1 + x_3$, and keep going until you have fit all two-predictor models.
- Choose the model that produces the lowest AIC value. If no model produces an AIC value that is significantly different from the one-predictor model, then stop. The one-predictor model is your final model.
- But, suppose x_2 turned out to be the best second-predictor to add, and is thus entered into the model. Now, see if entering x_2 into the model somehow affected the significance of the x_1 predictor. If x_1 is no longer a significant predictor, remove x_1 from the model.

Stepwise Regression: Understanding the Stepwise Regression Procedure (cont'd)

Step 4:

- Suppose $y \sim x_1 + x_2$ produced the model with the lowest AIC. Next, fit each of the three-predictor models that includes x_1 and x_2 as predictors. That is, fit the model $y \sim x_1 + x_2 + x_3$, then fit the model $y \sim x_1 + x_2 + x_4$, , and keep going until you have fit all three-predictor models.
- Choose the model that produces the lowest AIC value. If no model produces an AIC value that is significantly different from the two-predictor model, then stop. The two-predictor model is your final model.
- But, suppose x_3 turned out to be the best second predictor to add, and is thus entered into the model. Now, see if entering x_3 into the model somehow affected the significance of the x_1 predictor and the x_2 predictor. If either predictor is no longer a significant predictor, remove that predictor from the model.

Now simply continue this process until adding additional predictors no longer significantly reduces the AIC. When no additional predictor significantly reduces the AIC, you have arrived at your final model.

R Code for Backward and Forward Search (Related and Similar to Stepwise Regression)

```
library(faraway)
hipcenter_mod = lm(hipcenter ~ ., data = seatpos)

# Backward Search: The Backward selection procedure starts with all possible
# predictors in the model, then considers how deleting a single predictor
# will effect a chosen metric -- in this case the AIC.

hipcenter_mod_back_aic = step(hipcenter_mod, direction = "backward")

# Forward Search: Forward selection is the exact opposite of backwards
# selection. Here we tell R to start with a model using no predictors,
# that is hipcenter ~ 1, then at each step R will attempt to add a predictor
# until it finds a good model or reaches hipcenter ~ Age + Weight + HtShoes
# + Ht + Seated + Arm + Thigh + Leg.

hipcenter_mod_start = lm(hipcenter ~ 1, data = seatpos)
hipcenter_mod_forw_aic = step(
  hipcenter_mod_start,
  scope = hipcenter ~ Age + Weight + HtShoes + Ht + Seated + Arm + Thigh + Leg,
  direction = "forward")
```

Stepwise Regression (Stepwise Search) R Code

```
# Stepwise Search/Stepwise Regression: Stepwise regression checks going both
# backwards and forwards at every step. It considers the addition of any variable
 # not currently in the model, as well as the removal of any variable currently in
# the model. Here we perform stepwise search using AIC as our metric. We start
# with the model hipcenter ~ 1 and search up to hipcenter ~ Age + Weight +
# HtShoes + Ht + Seated + Arm + Thigh + Leg. Notice that at many of the steps,
# some rows begin with -, while others begin with +.

hipcenter_mod_both_aic = step(
  hipcenter_mod_start,
  scope = hipcenter ~ Age + Weight + HtShoes + Ht + Seated + Arm + Thigh + Leg,
  direction = "both")

# Stepwise Regression with BIC criterion. The output will still appear to indicate
# the use of the AIC, however.

n = length(resid(hipcenter_mod))

hipcenter_mod_both_bic = step(
  hipcenter_mod_start,
  scope = hipcenter ~ Age + Weight + HtShoes + Ht + Seated + Arm + Thigh + Leg,
  direction = "both", k = log(n))
```

Output with AIC Criterion:

```
Start:  AIC=311.71

hipcenter ~ 1


          Df Sum of Sq    RSS    AIC

+ Ht       1     84023  47616 275.07

+ HtShoes  1     83534  48105 275.45

+ Leg      1     81568  50071 276.98

+ Seated   1     70392  61247 284.63

+ Weight   1     53975  77664 293.66

+ Thigh    1     46010  85629 297.37

+ Arm      1     45065  86574 297.78

<none>                 131639 311.71

+ Age      1      5541 126098 312.07


Step:  AIC=275.07

hipcenter ~ Ht


          Df Sum of Sq    RSS    AIC

+ Leg      1      2781  44835 274.78

<none>                  47616 275.07

+ Age      1      2354  45262 275.14

+ Weight   1       196  47420 276.91

+ Seated   1       102  47514 276.99

+ Arm      1        76  47540 277.01

+ HtShoes  1        26  47590 277.05

+ Thigh    1         5  47611 277.06

- Ht       1     84023 131639 311.71
```

```
Step:  AIC=274.78
hipcenter ~ Ht + Leg

          Df Sum of Sq   RSS    AIC
+ Age      1    2896.6 41938 274.24
<none>                 44835 274.78
- Leg      1    2781.1 47616 275.07
+ Arm      1     522.7 44312 276.33
+ Weight   1     445.1 44390 276.40
+ HtShoes  1      34.1 44801 276.75
+ Thigh    1      33.0 44802 276.75
+ Seated   1       1.1 44834 276.78
- Ht       1    5236.3 50071 276.98

Step:  AIC=274.24
hipcenter ~ Ht + Leg + Age

          Df Sum of Sq   RSS    AIC
<none>                 41938 274.24
- Age      1    2896.6 44835 274.78
- Leg      1    3324.2 45262 275.14
- Ht       1    4238.3 46176 275.90
+ Thigh    1     372.7 41565 275.90
+ Arm      1     257.1 41681 276.01
+ Seated   1     121.3 41817 276.13
+ Weight   1      46.8 41891 276.20
+ HtShoes  1      13.4 41925 276.23
```

Lecture(s) 16 (and 17) Overview

- Introduction to Generalized Linear Models
- Logistic Regression as an Example of a Generalized Linear Model

Generalized Linear Models

So far in this course, we've dealt with response variables that, conditioned on the predictors, were modeled using a normal distribution with a mean that is some linear combination of (functions of) the predictor variables.

Now we'll allow for two modifications of this framework. Instead of using a normal distribution for the response conditioned on the predictors, we'll allow for other distributions. Also, instead of the conditional mean being a linear combination of the predictors, it can be some function of a linear combination of the predictors.

All of this will, for example, allow us, for the case of logistic regression, to conveniently model categorical response variables *Y* (for example, as a binary 0-1 variable indicating whether someone may have a particular disease or not), based on a number of suitable predictor variables and, moreover, quite readily produce an estimated probability assessing which value (0 or 1, say) *Y* takes on, given an *X* value or values.

A Generalized Linear Model: What is it?

In general, a **Generalized Linear Model (GLM)** is said to have three elements:

- A **probability distribution** for the response, conditioned on the predictors. Technically this distribution will be from the exponential family of distributions.

- A **linear combination** of M predictors, written as η(x):

$$\eta(x)=\beta_0+\beta_1x_1+\beta_2x_2+\ldots+\beta_Mx_M.$$

- A so-called **link** function, g, that defines how η(x), the linear combination of the predictors, is related to the mean of the response conditioned on the predictors, E[Y|X=x]:

$$\eta(x)=g(E[Y|X=x]).$$

Generalized Linear Models: Comparison

	Linear Regression	**Poisson Regression**	**Logistic Regression**
$Y\|X=x$	$N(\mu(x),\sigma^2)$	$Pois(\lambda(x))$	$Bern(p(x))$
Distribution Name	Normal	Poisson	Bernoulli (Binomial)
$E[Y\|X=x]$	$\mu(x)$	$\lambda(x)$	$p(x)$
Support	Real: $(-\infty,\infty)$	Integer: 0,1,2,…	Integer: 0,1
Usage	Numeric Data	Count (Integer) Data	Binary (Class) Data
Link Name	Identity	Log	Logit
Link Function	$\eta(x)=\mu(x)$	$\eta(x)=\log(\lambda(x))$	$\eta(x)=\log(p(x)/(1-p(x)))$
Mean Function	$\mu(x)=\eta(x)$	$\lambda(x)=e^{\eta(x)}$	$p(x)=e^{\eta(x)}/(1+e^{\eta(x)})=1/(1+e^{-\eta(x)})$

Like ordinary linear regression, we will in for example Logistic Regression seek to “fit” the model by estimating the β parameters. To do so, we use the method of Maximum Likelihood Estimation.

Generalized Linear Models: Logistic Regression Model

To illustrate the use of a GLMs we'll focus on the case of a binary response variable *Y* coded as either 0 or 1. In practice, such binary codings will allow for two classes such as yes/no, true/false, sick/healthy, etc. *X* is still our predictor variable as before. We may use bold notation ***X*** to indicate a vector (or random vector) of predictor variables.

First, we define some notation that we will use throughout. Define the (conditional) probability

$$p(\mathbf{x})=P[Y=1|\mathbf{X}=\mathbf{x}],$$

with a binary (Bernoulli) response random variable[1], we'll mostly focus on the case Y=1, since it is of course trivial in this setting to obtain probabilities when Y=0:

$$P[Y=0|\mathbf{X}=\mathbf{x}]=1-p(x)$$

We now define the **Logistic Regression Model:**

$$\log\left(\frac{p(\mathbf{x})}{1-p(\mathbf{x})}\right)=\beta_0+\beta_1 x_1+\ldots+\beta_M x_M$$

----

[1] Recall the Bernoulli probability distribution: A Bernoulli random variable Z satisfies Pr (Z=1) = p for some probability value p between 0 and 1, inclusive, and Pr (Z=0) = 1- p. Note that in this case **E[**Z]=p.

Generalized Linear Models: Logistic Regression Model

We have the Logistic Regression Model:

$$\log(p(\mathbf{x})/(1-p(\mathbf{x})))=\beta_0+\beta_1 x_1+\ldots+\beta_M x_M$$

Immediately we notice some similarities to ordinary linear regression, in particular, the right hand side. This is our usual linear combination of the predictors. We have our usual predictors for a total of M+1 ``β'' parameters. The left hand side is called the **log odds**, which is the log of the odds. The odds are the probability for a positive event (Y=1) divided by the probability of a negative event (Y=0). So when the odds are 1, the two events have equal probability. Odds greater than 1 favor a positive event. Essentially, the log odds are the logit transform applied to p(x).

$$\text{logit}(\xi)=\log(\xi/(1-\xi))$$

It will also be useful to define the inverse logit transform, otherwise known as the “logistic” or sigmoid function:

$$\text{logit}^{-1}(\xi)=e^{\xi}/(1+e^{\xi})=1/(1+e^{-\xi}).$$

Note that this function always outputs a number between 0 and 1.

Generalized Linear Models: Logistic Regression

Logistic regression model.

$$\log(p(\mathbf{x})/(1-p(\mathbf{x})))=\beta_0+\beta_1 x_1+\ldots+\beta_M x_M$$

- One might ask, where is the error term? The answer is that it is actually something specific to the normal model. First notice that since a logistic regression model uses the Bernoulli distribution, we only need to estimate the Bernoulli distribution’s single parameter p(x), which happens to be its mean. So even though we introduced ordinary linear regression first, in some ways, logistic regression is actually simpler.

- Note that applying the inverse logit transformation allows us to obtain an expression for the probability p(**x**):

$$p(\mathbf{x})=P[Y=1|\mathbf{X}=\mathbf{x}]=\exp(\beta_0+\beta_1 x_1+\cdots+\beta_M x_M)/(1+\exp(\beta_0+\beta_1 x_1+\cdots+\beta_M x_M)),$$

which is a probability representing the numerical confidence that the response Y takes on the value 1 (as opposed to 0), given (that is, conditioned on) **X**=**x**.

Generalized Linear Models: Logistic Regression

With N observations, we write the logistic regression model with a second index to note that it is being applied to each observation.

$$\log(p(\mathbf{x}_i)/(1-p(\mathbf{x}_i)))=\beta_0+\beta_1 x_{i1}+\ldots+\beta_M x_{iM,\ i=1,\ldots,N.}$$

We can apply the inverse logit transformation to obtain the probability $P[Y_i=1|X_i=x_i]$ for each observation. As these are probabilities, note that we use a function that returns values between 0 and 1.

$$p(\mathbf{x}_i)=P[Y_i=1|X_i=x_i]=\exp(\beta_0+\beta_1 x_{i1}+\cdots+\beta_M x_{iM})/(1+\exp(\beta_0+\beta_1 x_{i1}+\cdots+\beta_M x_{iM}))$$

$$1\text{-}p(\mathbf{x}_i)=P[Y_i=0|X=x_i]=1/1+\exp(\beta_0+\beta_1 x_{i1}+\cdots+\beta_M x_{iM})$$

To fit this model, that is to estimate the β-parameters, $\boldsymbol{\beta}=[\beta_0,\beta_1,\ldots,\beta_M]$, by means of corresponding estimators $\widehat{\boldsymbol{\beta}}=[\hat{\beta}_0,\hat{\beta}_1,\ldots,\hat{\beta}_M]$,respectively, computed via logistic regression using Maximum Likelihood Estimation.

Generalized Linear Models: Maximum Likelihood Estimation

To fit the GLM Logistic Regression model, we use Maximum Likelihood Estimation. We first write the likelihood given the observed data.

$$\mathrm{L}(\beta)=\prod_{i=1}^{N} \mathrm{P}[Y_i=y_i|\boldsymbol{X}_i=\boldsymbol{x}_i]$$

This is already technically a function of the β parameters, but more rearrangement can be done to make things more explicit,

$\mathrm{L}(\beta)=\prod_{i=1}^{N} \mathrm{p}(\mathbf{x}_\mathrm{i})^{\mathrm{y_i}}(1-\mathrm{p}(\mathbf{x}_\mathrm{i}))^{(1-\mathrm{y_i})}$

$\mathrm{L}(\beta)= \prod_{i=1,\mathrm{y_i}=1}^{N} \mathrm{p}(\mathbf{x}_\mathrm{i}) \prod_{j=1,\mathrm{y_j}=0}^{N} (1-\mathrm{p}(\mathbf{x}_\mathrm{j}))$

$\mathrm{L}(\beta)=\prod_{i=1,\mathrm{y_i}=1}^{N} \exp(\beta_0+\beta_1 \mathrm{x}_{\mathrm{i1}+\cdots+}\beta_\mathrm{M}\mathrm{x}_{\mathrm{iM}})/(1+\exp(\beta_0+\beta_1\mathrm{x}_{\mathrm{i1}}+\cdots+\beta_\mathrm{M}\mathrm{x}_{\mathrm{iM}})) \prod_{j=1,y_j=0}^{N} 1/(1+\exp(\beta_0+\beta_1\mathrm{x}_{\mathrm{j1}+\cdots+}\beta_\mathrm{M}\mathrm{x}_{\mathrm{jM}}\ ))$

Unfortunately, unlike ordinary linear regression, there is no analytical solution for this maximization problem. Instead, it will need to be solved numerically. Fortunately, R will take care of this using an iteratively reweighted least squares algorithm.

Generalized Linear Models: Logistic Regression (cont’d)

In part because both involve a **linear** combination

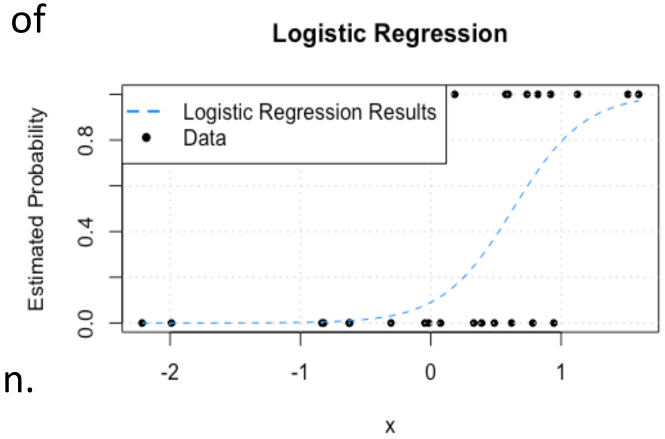
$$\eta(x)=\beta_0+\beta_1x_1+\beta_2x_2+\ldots+\beta_Mx_M$$

of the predictors, logistic regression is not so dissimilar from linear regression. Much of what was done with ordinary linear regression can be done with logistic regression in an analogous fashion. For example,

- Testing with respect to a single β parameter
- Testing with respect to a set of β parameters
- Interpretation of parameters and estimates
- Confidence intervals for parameters
- Confidence intervals for mean response
- Variable selection

Generalized Linear Models: Example R Code for Logistic Regression

```
sim_logistic_data = function(sample_size = 25, beta_0 = -2, beta_1 = 3) {
 x = rnorm(n = sample_size)
 eta = beta_0 + beta_1 * x
 p = 1 / (1 + exp(-eta))
 y = rbinom(n = sample_size, size = 1, prob = p)
 data.frame(y, x)
}
set.seed(1)
example_data = sim_logistic_data()
head(example_data)
fit_glm = glm(y ~ x, data = example_data, family = binomial)
plot(y ~ x, data = example_data,
    pch = 20, ylab = "Estimated Probability",
    main = "Logistic Regression")
grid()
curve(predict(fit_glm, data.frame(x), type = "response"),
     add = TRUE, col = "dodgerblue", lty = 2)
legend("topleft", c("Logistic Regression Results", "Data"), lty = c(2, 0),
      pch = c(NA, 20), lwd = 2, col = c("dodgerblue", "black"))

newdata89 = data.frame(x = 0.184)
predict(fit_glm, newdata89, type="response")
```



In the R code to the left, some synthetic sample data are first produced. Then, using these data, a logistic regression model is then fitted. Note, at left, the use of the glm(·) function for this purpose in place of the lm(·) function. The use of the glm(·) function for logistic regression is quite analogous to that of the lm(·) function, as we have seen, for standard linear regression. However, we must also specify a distribution under the argument for "family", which should be "binomial" to implement logistic regression. Indeed, as we know

fit_lm=lm(y ~ x_1+…+x_M , data = dataset1)

implements a familiar simple linear regression model. Similarly,

fit_glm = glm(y ~ x_1+…+x_M , data = dataset2, family = binomial)

implements a logistic regression model of the form

$$\log(p(\mathbf{x}_i)/(1-p(\mathbf{x}_i)))=\beta_0+ \beta_1 x_{i1} +\ldots+ \beta_M x_{iM,}\ \ i=1,\ldots,N,$$

where $\mathbf{x}_i = (x_{i1} ,\ldots, x_{iM})$. The case in the code at left is of course the case here with M=1. The summary function, for example, can be applied to models fitted via glm(·) as here, giving some types of model information analogous to those it does for lm(·).

We plot output of the model at left in the graph at upper right, indicating estimated probabilities, respectively, for given values of x, using the code at left. Indeed the code to the lower left also outputs, given a single input x-value (here, x=0.184), simply the associated single probability value.

Wald Test: Analogue of Single-Variable Significance t-test for Logistic Regression

In ordinary linear regression, we performed, for any j=1,…,M, the significance-of-regression test

$$H_0:\beta_j=0 \text{ vs. } H_1:\beta_j\neq 0\text{, using the t-distribution.}$$

For the logistic regression model,

$$\log(p(x)/(1-p(x)))=\beta_0+\beta_1 x_1+\ldots+\beta_M x_M$$

we can again perform a test of

$$H_0:\beta_j=0 \text{ vs. } H_1:\beta_j\neq 0$$

However, the test statistic and its distribution are no longer of the t form. The test statistic here takes the same form

$$z=\frac{\widehat{\beta}_j-\beta_j}{SE[\widehat{\beta}_j]} \sim N(0,1) \quad \text{(approximately)}$$

but now we are performing what we can call a “z-test”, as the test statistic is approximated by a standard normal distribution, *provided we have a large enough sample size*. (The t-test for ordinary linear regression, assuming the assumptions were correct, had an exact distribution for any sample size.) The use of this test will be extremely similar to the t-test for ordinary linear regression. Essentially the only thing that changes is the distribution of the test statistic.

Likelihood Ratio Test: Analogue of an F-test for Logistic Regression

Consider the following model,

$\log(p(x_i)/(1-p(x_i)))=\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots+\beta_M x_{iM}$

This model has M predictors, for a total of M+1 β-parameters. Denote the Maximum Likelihood Estimator (MLE) of these β-parameters by $\hat{\beta}_{Full}$.

Now consider a **null** (or **reduced**) model,

$\log(p(x_i)/(1-p(x_i)))=\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots+\beta_L x_{iL}$

where $L<M$. This model has L predictors, for a total of L+1 β-parameters. Denote the MLE estimator of these β-parameters by $\hat{\beta}_{Null}$. The differential value, if you will, between these two models can be assessed by the Null Hypothesis of a statistical test.

$H_0:\beta_{(L+1)}=\beta_{(L+2)}=\cdots=\beta_M=0$, H_1: NOT$(\beta_{(L+1)}=\beta_{(L+2)}=\cdots=\beta_M=0)$.

We can define a test statistic *D*

$$D=-2\log\left(\frac{L(\hat{\beta}_{Null})}{L(\hat{\beta}_{Full})}\right)=2\log\left(\frac{L(\hat{\beta}_{Full})}{L(\hat{\beta}_{Null})}\right)=2(\ell(\hat{\beta}_{Full})-\ell(\hat{\beta}_{Null})),$$

Where L denotes a likelihood and ℓ denotes a log-likelihood. For a large enough sample this test statistic has an approximate Chi-square distribution, indexed by *k*:

$$D \sim \chi_k^2 \text{ (approximate)}$$

where $k=M-L$, the difference in number of parameters of the two models, in this case. This test, which is called the **Likelihood-Ratio Test**, can be viewed as an analogue of a standard linear regression F-test (of the kind we have seen) for the case of logistic regression.

Generalized Linear Models: Logistic Regression Example

(Refer to the lecture videos to help interpret this slide.)

Data here comes from a retrospective sample of males in a heart-disease high-risk region of the Western Cape, South Africa.

The chd variable, which we will use as a response, indicates whether or not coronary heart disease is present in an individual. Note that this is coded as a numeric 0 / 1 variable. Using this as a response with glm() it is important to indicate family = binomial, otherwise ordinary linear regression will be fit instead.

The predictors are various measurements for each individual, many related to heart health. For example sbp, systolic blood pressure, and ldl, low density lipoprotein cholesterol.

Begin by attempting to model the probability of coronary heart disease based on low density lipoprotein cholesterol. That is, fit the model

$$\log(\Pr[chd=1]/(1-\Pr[chd=1]))=\beta_0+ \beta_{ldl}x_{ldl}$$

Generalized Linear Models: Logistic Regression Empirical Example

(Refer to the lecture videos to help interpret this slide.)

As before, we plot the data in addition to the estimated probabilities. Note that we have “jittered” the data to make it easier to visualize, but the data do only take values 0 and 1. As we would expect, this plot indicates that as ldl increases, so does the probability of chd.

To perform the test

$$H_0: \beta_{ldl} = 0,$$

use the summary() function as we have done so many times before. Like the t-test for ordinary linear regression, this returns the estimate of the parameter, its standard error, the relevant test statistic (z), and its p-value. Here we have an incredibly low p-value, so we reject the null hypothesis. The ldl variable appears to be a significant predictor.

Logistic Regression Empirical Example

(Refer to the lecture videos to help interpret this slide.)

When fitting logistic regression, we can use the same formula syntax as ordinary linear regression. So, to fit an additive model using all available predictors, we use:

```
chd_mod_additive = glm(chd ~ ., data = SAheart, family = binomial)
```

We can then use the likelihood-ratio test to compare the two models. Specifically, we are testing

$H_0: \beta_{sbp} = \beta_{tobacco} = \beta_{adiposity} = \beta_{famhist} = \beta_{typea} = \beta_{obesity} = \beta_{alcohol} = \beta_{age} = 0.$

We could manually calculate the test statistic,

```
2 * as.numeric(logLik(chd_mod_ldl) - logLik(chd_mod_additive))
```

```
## [1] 92.13879
```

While we prefer the additive model compared to the model with only a single predictor, do we actually need all of the predictors in the additive model? To select a subset of predictors, we can use a stepwise procedure as we did with ordinary linear regression.

Logistic Regression: Confidence Intervals

For logistic regression a confidence interval for the beta parameter is

$$\hat{\beta}_m \pm z_{\alpha/2} \cdot \mathrm{SE}[\hat{\beta}_m],$$

which gives an approximate (1−α)% confidence interval.

We can get confidence intervals for both η(x) and the probability value p(x) as well. With a “large enough” sample, we have

$$(\hat{\eta}(x) - \eta(x))/\mathrm{SE}[\hat{\eta}(x)] \sim N(0,1) \quad \text{(approximate)}.$$

Here, $\hat{\eta}(x) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \ldots + \hat{\beta}_M x_{M.}$

Then we can create an approximate (1−α)% confidence interval for η(x) using $\hat{\eta}(x) \pm z_{\alpha/2} \cdot \mathrm{SE}[\hat{\eta}(x)]$where $z_{\alpha/2}$ is the critical value such that $P(Z > z_{\alpha/2}) = \alpha/2$.

Moreover we can get an interval for the mean response, p(x). To obtain a confidence interval for p(x), we simply apply the inverse logit transform to the endpoints of the interval for η:

$$(\mathrm{logit}^{-1}(\hat{\eta}(x) - z_{\alpha/2} \cdot \mathrm{SE}[\hat{\eta}(x)]),\ \mathrm{logit}^{-1}(\hat{\eta}(x) + z_{\alpha/2} \cdot \mathrm{SE}[\hat{\eta}(x)]))$$

Logistic Regression: Classification (in the sense of Machine Learning)

So far we've mostly used logistic regression to estimate class probabilities. The somewhat obvious next step is to use these probabilities to make "predictions," which in this context, we would call **classification**. Based on the values of the predictors, should an observation be classified as Y=1 or as Y=0? Suppose we didn't need to estimate probabilities from data, and instead, we actually knew both

$$p(x)=P[Y=1|X=x] \text{ and } 1-p(x)=P[Y=0|X=x].$$

With this information, classifying observations based on the values of the predictors is actually very, very easy. Simply classify an observation to the class (0 or 1) with the larger probability.

In general, this rule is called the **Bayes Classifier**,

$$C^B(x)=\text{arg max}_k\ P[Y=k|X=x].$$

For a binary response, this is

$$C^B(x)=\begin{cases}1, & p(x) > 0.5\\ 0, & p(x) \le 0.5\end{cases}$$

Simply put, the Bayes classifier (not to be confused with the Naive Bayes Classifier) minimizes the probability of misclassification by classifying each observation to the class with the highest probability. Unfortunately, in practice, we won't generally know the necessary probabilities to directly use the Bayes classifier. Instead we'll have to use estimated probabilities. So to create a classifier that seeks to minimize misclassifications, we would use,

$$\hat{C}_B(x)=\text{arg max}_k\ \hat{P}[Y=k|X=x].$$

where, of course, the probabilities are estimated via logistic regression. For a binary response, that is,

$$\hat{C}_B(x)=\begin{cases}1, & \hat{p}(x) > 0.5\\ 0, & \hat{p}(x) \le 0.5.\end{cases}$$

Logistic Regression for Classification (in the sense of Machine Learning)

A natural metric that may be used to assess the overall performance of a classifier is the **misclassification rate**. (Sometimes, instead, the accuracy is invoked, which is instead the proportion of correct classifications, so both figures of merit serve the same purpose.) The Misclassification Rate is defined via

$$\text{Misclass}(\hat{C},\text{Data})=\frac{1}{N}\sum_{i=1}^{N}\text{I}(\text{yi}\neq\hat{C}(\text{xi}))|$$

$$\text{I}(\text{y}_\text{i}\neq \hat{C}(\text{x}_\text{i}))|=\begin{cases}0, \text{y}_\text{i}= \hat{C}(\text{xi})\\ 1, \text{y}_\text{i}\neq\hat{C}(\text{xi}).\end{cases}$$

To assess the overall performance of a logistic regression classifier we can use 5-fold cross-validation, a procedure defined by repeating the following steps 5 times:

- Randomly set aside a fifth of the data (each observation will only be held-out once)
- Train model on remaining data
- Evaluate misclassification rate on held-out data

The 5-fold cross-validated misclassification rate will be the average of these individual misclassification rates.