

Predictive analytics

All Slides

Ca' Foscari University of Venice

Academic year 2022/2023

This Course

Syllabus

1 Introduction:

- What is predictive modeling and, more in general, statistical learning?
- General notation and background

2 Linear models I

- Model formulation and least squares
- Inference and prediction

3 Linear models II: model selection, extensions, and diagnostics

4 Generalized linear models (and extension)

- **Warning:** A lot of practicals with **R** - maybe refresh a bit.
- We will use R for data analysis and writing reports using **Rmarkdown**
- Bring your laptop: we will play with real data!

References

- Textbooks:
 - Julian J. Faraway, 2014. *Linear Models with R Second Edition*, Chapman and Hall/CRC
 - Julian J. Faraway, 2016. *Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models*, Second Edition Chapman and Hall/CRC
 - James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. An Introduction to Statistical Learning. Springer [Open access book](#) (ISLR book - new edition just out)
- Other teaching materials will be regularly posted on Moodle, i.e.
 - These slides
 - Lab sessions

Other (free) material

Books which have inspired the course material:

- [Lecture notes](#) by Julian Faraway
- [The Elements of Statistical Learning](#) by Hastie, Tibshirani and Friedman (the “big brother” of ISLR)
- [A draft textbook on data analysis methods](#) by Cosma Shalizi
- [Introduction to Data Science](#) by Rafael A. Irizarry
- [Applied Statistics with R](#) by David Dalpiaz

There is a deluge of material out there: let me know if you find something good.

The last two books have been written using `bookdown` - an R package which extends `rmarkdown`.

Course Organization

Moodle: <https://moodle.unive.it/course/view.php?idnumber=180748>

- Lessons:

Day	Time	Room
Monday	10:30 - 12:00	Zeta Aula C
Tuesday	10:30 - 12:00	Zeta Aula C

- Office hours: Tuesday 12:10 - 13:30 (book **here**)
 - In person or online - specify your choice when booking.
- Please check the notices on my university webpage/calendar for variations
- The **written exam** using the computer consists of exercises designed to measure
 - the theoretical knowledge of the course topics,
 - the ability to apply them for solving real data problems.

Disability and Specific Learning Disorders Service

Students with **disabilities** and/or **SLD** that follow this course are invited to report to the **teacher** and to **Disability and SLD Office** any need in order to optimize the preparation for the exam.

For general information about the services offered by Ca' Foscari:

Disability and SLD Office
inclusione@unive.it
041 234 7961

Statistical learning: Introduction and examples¹

¹This part is mainly based on ISLR. From this book I have extracted or rephrased sentences.

E-mail example

Goal: construct an automatic spam detector that block spam.

Data: information on 4601 emails, in particular,

- whether was it spam (spam) or not (email);
- the relative frequencies of 57 of the most common words or punctuation marks.

word	george	you	your	hp	free	hpl	!	...
spam	0.00	2.26	1.38	0.02	0.52	0.01	0.51	...
email	1.27	1.27	0.44	0.90	0.07	0.43	0.11	...

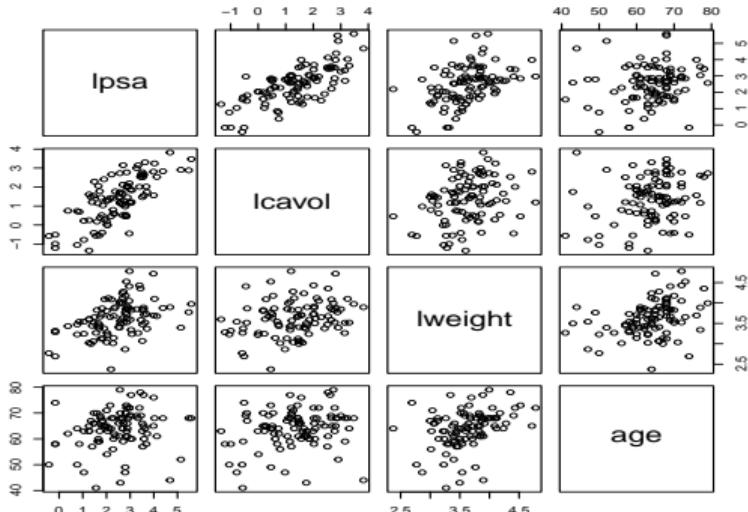
Possible rules:

if (%george < 0.6) & (%you > 1.5) then spam else email

or

if (0.2· %you - 0.3· %george) > 0 then spam else email.

Prostate cancer example



- Goal: predict the level of (log) prostate specific antigen (lpsa) from some clinical measures, such as log cancer volume (lcavol), log prostate weight (lweight), age (age), ...;
- Possible rule: $\text{lpsa} = 0.32 \times \text{lcavol} + 0.15 \times \text{lweight} + 0.20 \times \text{age}$

Statistical Learning

- The goal of statistical learning is to “get knowledge” from the data, so that the information can be used for prediction, identification, understanding, ...
- Statistical learning tools are often divided into
 - supervised
 - unsupervised
- **Supervised learning:** Given observed data $\{(x_1, y_1), \dots, (x_n, y_n)\}$, learn a statistical model to predict Y from X .
 - If y_i is a continuous numeric value, this task is called prediction (E.g., Y_i = stock price, income, survival time)
 - If y_i is a discrete or symbolic value, this task is called classification (E.g., $y_i \in \{0, 1\}$, $y_i \in \{\text{spam, email}\}$, $y_i \in \{1, 2, 3, 4\}$)
- **Unsupervised learning:** Given observed data $\{x_1, \dots, x_n\}$, identify some underlying patterns or structure in the data.

Statistical Learning: examples

- **Supervised learning**

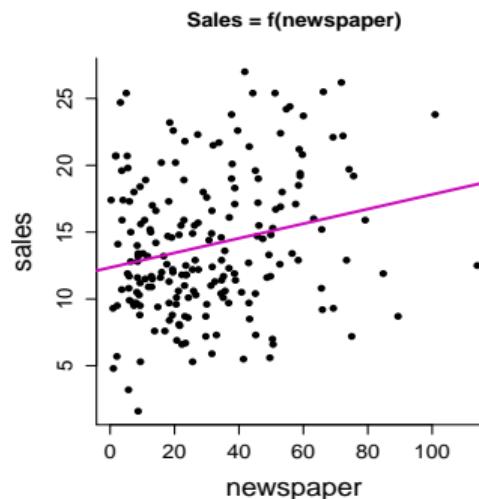
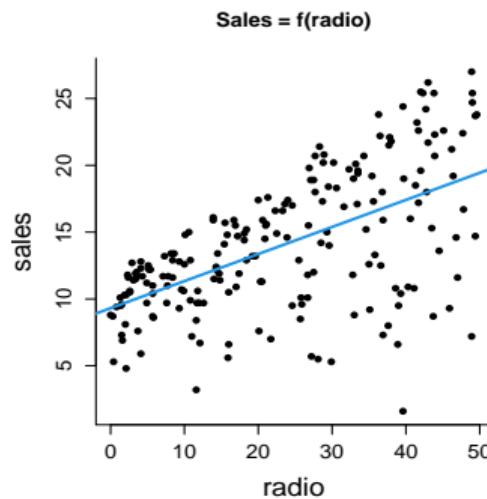
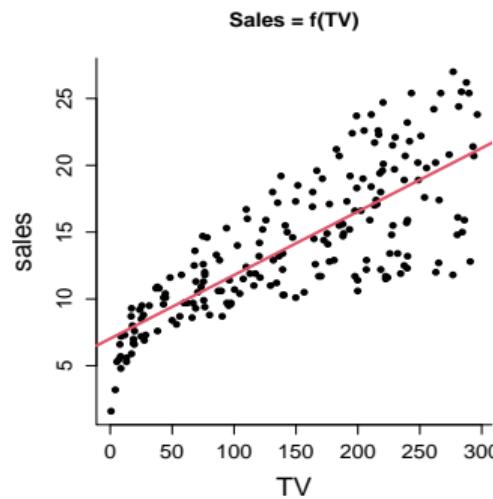
- Q: To whom should I extend credit?
- Task: Predict how likely an applicant is to repay loan.
- Q: How profitable will this subscription customer be?
- Task: Predict how long customer will remain subscribed.
- Q: What characterizes customers who are likely to churn?
- Task: Identify variables that are predictive of churn.

- **Unsupervised learning**

- Clustering customers into groups with similar spending habits
- Learning association rules: E.g., 50% of clients, who recently got promoted and had a baby, want to get a mortgage

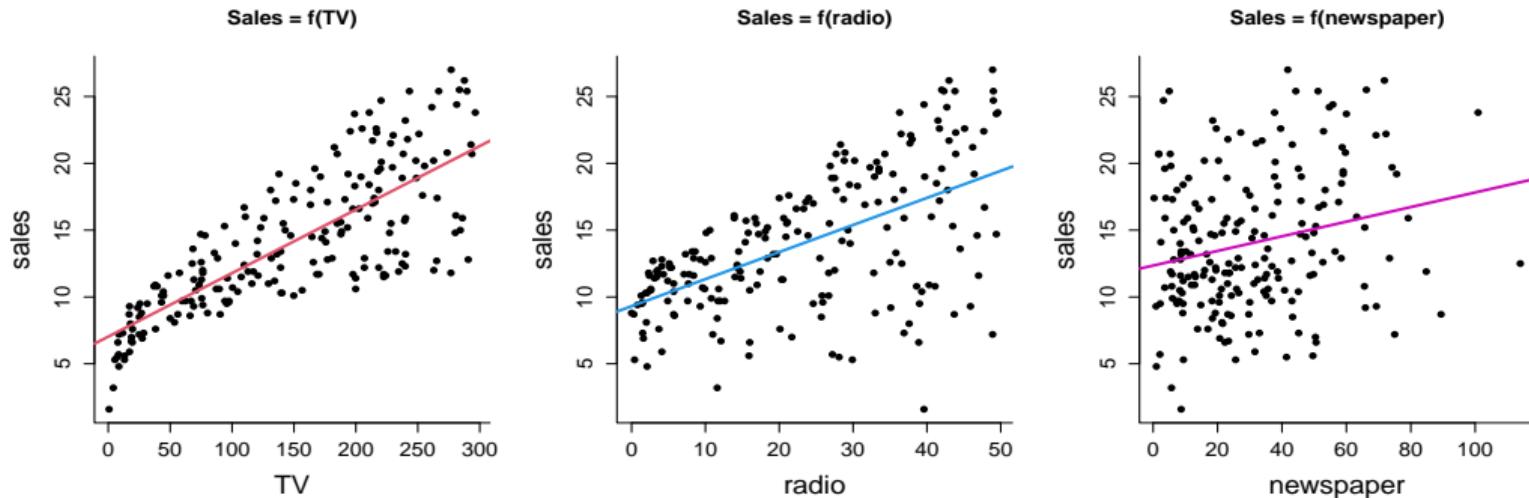
Advertising data set I

The Advertising data set contains data on $n = 200$ different markets.



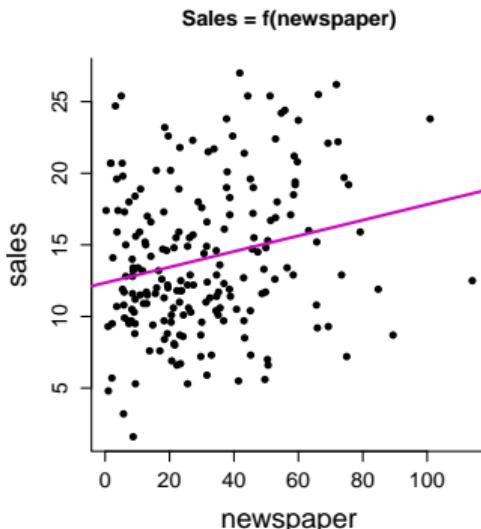
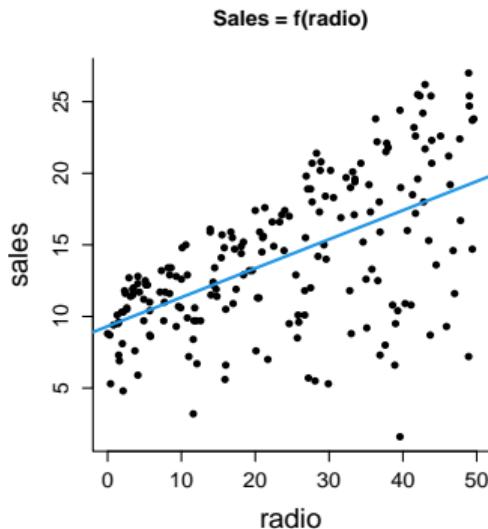
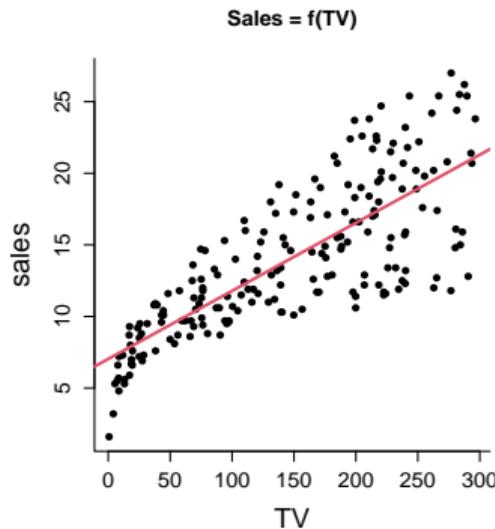
- Outcome $Y = \text{sales}$ in 1000's of units
- Inputs: advertising budgets for $X_1 = \text{TV}$, $X_2 = \text{radio}$, and $X_3 = \text{newspaper}$ in 1000's of dollars

Advertising data set II



- Ideally, we would like to have a joint model of the form $\text{sales} \approx m(\text{TV}, \text{radio}, \text{newspaper})$
- Wanted: a function m such that $m(\text{TV}, \text{radio}, \text{newspaper})$ is a good explanation of sales.

Advertising data set III



- Here restricted to linear functions.
- What does it mean to be a good predictor?

Notation and Terminology

- sales is known as the **response**, or **target**, or **outcome**, or **output**. It's the variable we wish to predict. We denote the response variable as Y .
- TV is a **feature**, or **input**, or **predictor**, or **regressor**, or **covariate** or **explanatory variable**.
- We denote TV by X_1 .
- Similarly, we denote $X_2 = \text{radio}$ and $X_3 = \text{newspaper}$
- Notice that the order of numbering the r.v. is arbitrary.
- We can put all the predictors into a single input vector $\mathbf{X} = (X_1, X_2, X_3)$
- Now we can write our model as

$$Y = m(\mathbf{X}) + \varepsilon$$

where ε captures **measurement errors** and other discrepancies between the response Y and the model m

Why estimate m ?

- In essence, statistical learning refers to a set of approaches for estimating m starting from the data on X and Y .
- In the sequel, the estimate of m will be denoted by $\hat{m}(X)$
- Two main reasons why we may wish to estimate m :
 - prediction
 - inference (interpretation)

Prediction vs inference I

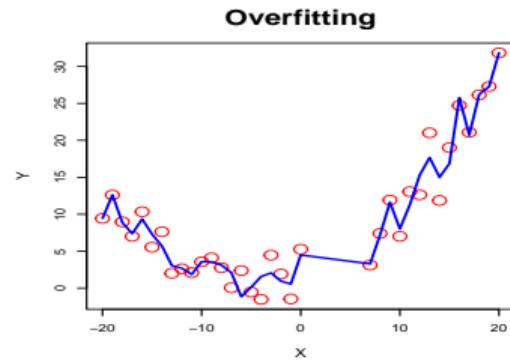
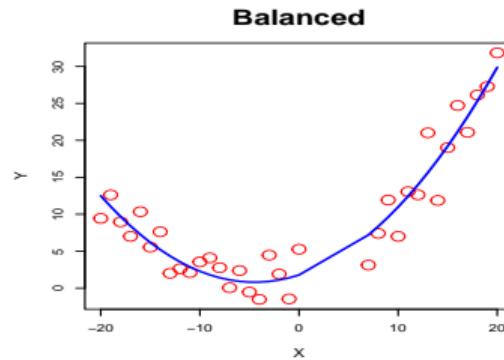
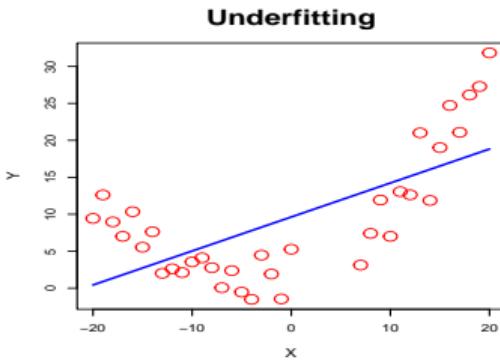
- **Prediction:** accurately projecting the chances that something will (or will not) happen.
- Prediction example: we don't really care why an e-mail filter thinks a message is spam. Rather, we only care that the filter accurately trashes spam and allows messages we care about to pass through to our mailbox.
- **Inference:** understand why something will (or will not) occur.
- Inference example:
 - Which media contribute to sales?
 - Which media generate the biggest boost in sales?
 - Size of increase in sales associated to a given increase in TV advertising?

Prediction vs inference II

- Sometimes modeling could be conducted both for prediction and inference, it depends on the question.
- Example: in a real estate setting, one may seek to relate values of houses to inputs as crime rate, zoning, distance from a river, etc...
 - Question: is the value of a home given its characteristics under- or over-valued? (prediction problem)
 - Question: what is the additional value for a house to have a view on the river? (inference problem)

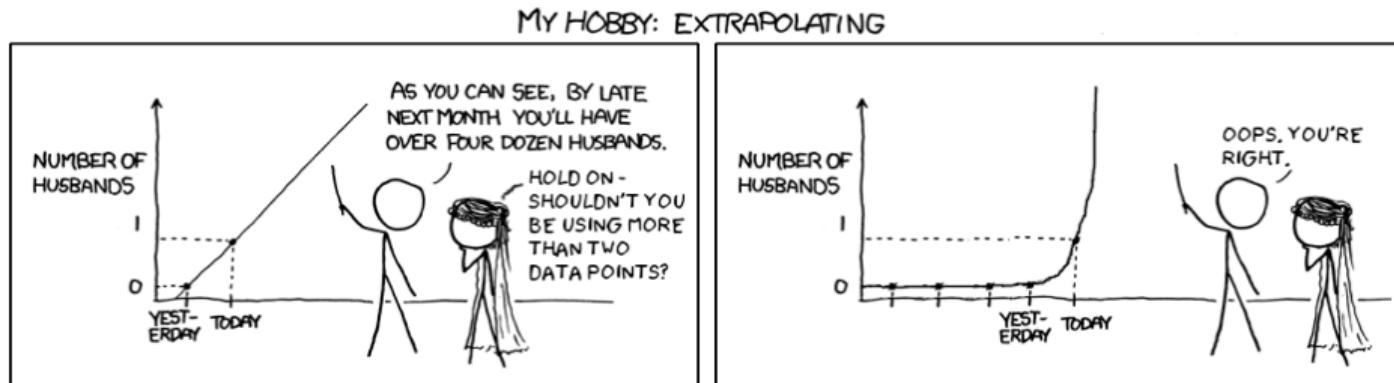
Generalizability I

- We want to construct \hat{m} that generalize well to unseen data
 - Ex. $x = 4$, $\hat{m}(4)$?
- i.e., we want predictors that:
 - Capture useful trends in the data (don't underfit)
 - Ignore meaningless random fluctuations in the data (don't overfit)



Generalizability II

- We also want to avoid unjustifiably extrapolating beyond the scope of our data

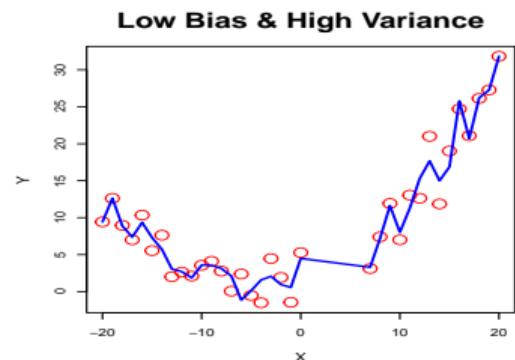
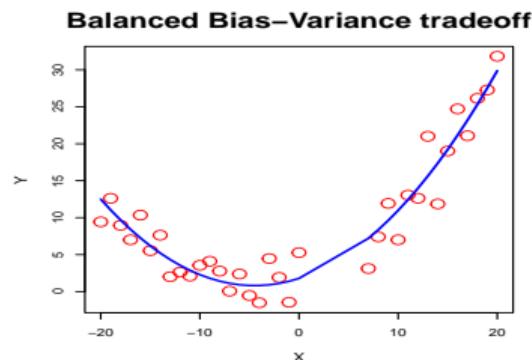
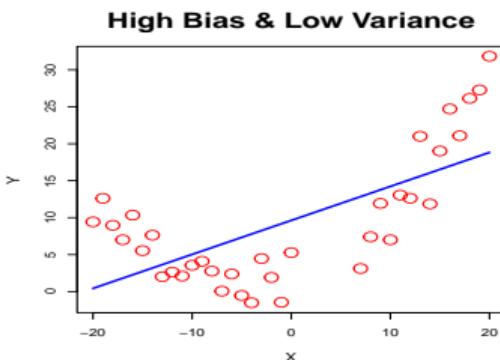


Bias-Variance Tradeoff I

- We will talk a lot about the **Bias-Variance tradeoff**, which relates to the fact that given a predictor \hat{m} ,

$$\text{Expected-prediction-error}(\hat{m}) = \text{Variance}(\hat{m}) + \text{Bias}^2(\hat{m})$$

- In the language of the previous theme



Interpretability-Flexibility Tradeoff I

- We can build highly structured, interpretable models and highly flexible models
- Actually we will focus more on the first ones.
- The best predictor for a problem may turn out to be an uninterpretable or hard-to-interpret black box
- Depending on the purpose of the prediction, we may prefer a more interpretable, worse-performing model to a better-performing "black box".

Interpretability-Flexibility Tradeoff II

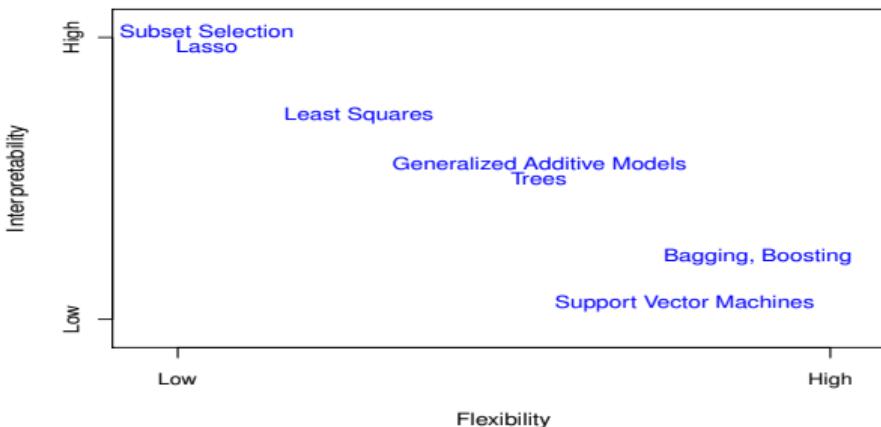


Figure: A representation of the tradeoff between flexibility and interpretability, using different statistical learning methods. In general, as the flexibility of a method increases, its interpretability decreases.
Source: ISLR book, fig. 2.7

Reminders from Basic Probability

One random variable I

- Expectation of a continuous random variable X with **probability density** function $p(x)$

$$\mathbb{E}[X] = \int xp(x)dx$$

- Expectation of a discrete random variable with probability mass function $p(x)$

$$\mathbb{E}[X] = \sum_x xp(x)$$

- Because everything is parallel for the discrete and continuous cases, I will just write out the integrals

One random variable II

- **Expectation** of any function of a random variable $h(X)$

$$\mathbb{E}[h(X)] = \int h(x)p(x)dx$$

- It is generally the case that $\mathbb{E}[h(X)] \neq h(\mathbb{E}[X])$
- $X - \mathbb{E}[X]$ is the **deviation** or **fluctuation** of X from its expected value.
- The **variance** of X is

$$\text{Var}[X] = \mathbb{E}[(X - \mathbb{E}[X])^2]$$

Bivariate (Multivariate) random variables I

- X and Y are two continuous random variables with **joint probability density function** $p_{X,Y}(x,y)$
- the marginal density of X may be recovered by

$$p_X(x) = \int p_{X,Y}(x,y) dy$$

- the **marginal density** of Y may be found with

$$p_Y(y) = \int p_{X,Y}(x,y) dx,$$

Bivariate (Multivariate) random variables II

- Two random variables are **independent** if and only if

$$p_{X,Y}(x,y) = p_X(x) \times p_Y(y)$$

- For a function g with arguments (x, y) the expectation of $g(X, Y)$ is

$$\mathbb{E}[g(X, Y)] = \int g(x, y) p_{X,Y}(x, y) dx dy,$$

- Set $g(x, y) = (x - \mathbb{E}[X])(Y - \mathbb{E}[Y])$
- The **covariance** of X and Y is

$$\text{Cov}[X, Y] = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$$

Bivariate (Multivariate) random variables III

- The covariance is positive when X and Y tend to be above or below their expected values together, and negative if X tends to be above (below) its expected value while Y tends to be below (above) its expected value.
- $\text{Cov}[X, Y] = \text{Cov}[Y, X]$ by definition.
- The **correlation** between X and Y is the covariance between X and Y rescaled to fall in the interval $[-1, 1]$. It is formally defined by

$$\text{Corr}[X, Y] = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}[X] \text{Var}[Y]}}.$$

- The correlation will be denoted by $\rho_{X,Y}$ or simply ρ if the random variables are clear from context.
- Important facts about the correlation coefficient:

Bivariate (Multivariate) random variables IV

- The range of correlation is $-1 \leq \rho_{X,Y} \leq 1$.
- Equality holds above ($\rho_{X,Y} = \pm 1$) if and only if Y is a linear function of X with probability one, i.e. $Y = a + bX$ with probability one.
- If X and Y are independent then $\rho_{X,Y} = 0$, since $\text{Cov}[X, Y] = 0$. The converse is **not** true.
- $\text{Corr}[X, Y] = \text{Corr}[Y, X]$ by definition.

Algebra with expectations, variances and covariances I

① Linearity of expectations

$$\mathbb{E}[aX + bY] = a\mathbb{E}[X] + b\mathbb{E}[Y]$$

② Variance identity

$$\text{Var}[X] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2 = \mathbb{E}[(X - \mathbb{E}[X])^2]$$



③ Covariance identity

$$\text{Cov}[X, Y] = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$$

④ Covariance is symmetric

$$\text{Cov}[X, Y] = \text{Cov}[Y, X]$$

Algebra with expectations, variances and covariances II

- ⑤ Variance is covariance with itself

$$\text{Cov}[X, X] = \text{Var}[X]$$

- ⑥ Variance is not linear

$$\text{Var}[aX + b] = a^2\text{Var}[X]$$

- ⑦ Covariance is not linear

$$\text{Cov}[aX + b, Y] = a\text{Cov}[X, Y]$$

- ⑧ Variance of a sum

$$\text{Var}[X + Y] = \text{Var}[X] + \text{Var}[Y] + 2\text{Cov}[X, Y]$$

- ⑨ Variance of a sum of n variables

$$\text{Var}\left[\sum_{i=1}^n X_i\right] = \sum_{i=1}^n \sum_{j=1}^n \text{Cov}[X_i, X_j] = \sum_{i=1}^n \text{Var}[X_i] + 2 \sum_{i=1}^{n-1} \sum_{j>i} \text{Cov}[X_i, X_j]$$

Conditional Distributions I

- If x is such that $p_X(x) > 0$, then we define the **conditional density** of $Y|X = x$, denoted $p_{Y|x}$, by

$$p_{Y|x}(y|x) = \frac{p_{X,Y}(x,y)}{p_X(x)},$$

- We define $p_{X|y}$, the **conditional density** of $X|Y = y$, in a similar fashion, i.e. for y such that $p_Y(y) > 0$

$$p_{X|y}(x|y) = \frac{p_{X,Y}(x,y)}{p_Y(y)}$$

Conditional Distributions II

- If X and Y are independent

$$p_{X|y}(x|y) = p_X(x), \quad p_{Y|x}(y|x) = p_Y(y)$$

- The **conditional expectation** of Y given $X = x$ is the expectation computed relative to the conditional distribution i.e.

$$\mathbb{E}[Y|X = x] = \int y p_{Y|x}(y|x) dy$$

- Since $\mathbb{E}[Y|X = x]$ depends on the given value x of X , we can denote the function

$$\mu(x) = \mathbb{E}[Y|X = x]$$

- The random variable $\mu(X)$ is called the conditional expected value of Y given X or **regression function** and is denoted by $\mathbb{E}[Y|X]$

Conditional Distributions III

- Similarly we can define the **conditional variance** of Y given $X = x$

$$\begin{aligned}\text{Var}[Y|X = x] &= \mathbb{E}[(Y - \mathbb{E}[Y|X = x])^2 | X = x] \\ &= \int (y - \mathbb{E}[Y|X = x])^2 p_{Y|x}(y|x) dy\end{aligned}$$

- Again, $\text{Var}[Y|X = x]$ depends on the given value x of X , we can denote the function $\nu(x) = \text{Var}[Y|X = x]$. The random variable $\nu(X)$ is denoted by $\text{Var}[Y|X]$

Conditional Distributions IV

- Useful properties:

- For a real function r

$$\mathbb{E}[r(X)\mathbb{E}[Y|X]] = \mathbb{E}[r(X)Y]$$

- **Law of total expectation.** By taking r to be the constant function 1 in the previous formula

$$\mathbb{E}[\mathbb{E}[Y|X]] = \mathbb{E}[Y]$$

- For any real function r the two random variable $Y - \mathbb{E}[Y|X]$ and $r(X)$ are uncorrelated i.e.

$$\text{Corr}[Y - \mathbb{E}[Y|X], r(X)] = \text{Cov}[Y - \mathbb{E}[Y|X], r(X)] = 0$$

- For any real function r

$$\mathbb{E}[r(X)Y|X] = r(X)\mathbb{E}[Y|X]$$

- **Law of total variance.**

$$\text{Var}[Y] = \text{Var}[\mathbb{E}[Y|X]] + \mathbb{E}[\text{Var}[Y|X]]$$

Optimal Prediction ²

²This part is mainly based on a course taught by [Cosma Shalizi](#). From his slides I have extracted or rephrased sentences.

Predict a Random Variable from Its Distribution I

- We want to predict the value of a random variable Y .
- What's the best guess we can make?
- We need to measure how good a guess is.
- Let m be our prediction (a scalar).
- The difference $Y - m$ is random and it should somehow be small.
- If we don't care about positive more than negative errors, it's traditional to care about the squared error: $(Y - m)^2$.
- We take its expected value:

$$MSE(m) = \mathbb{E}[(Y - m)^2]$$

the **Mean Squared Error** of m .

Predict a Random Variable from Its Distribution II

- We have the following **bias-variance decomposition**

$$MSE(m) = \mathbb{E} [(Y - m)^2] = (\mathbb{E}[Y - m])^2 + \text{Var}[Y - m]$$

- $(\mathbb{E}[Y - m])^2$ the squared **bias**
- Remember that $\text{Var}[Y - m] = \text{Var}[Y]$,

$$\begin{aligned} MSE(m) &= (\mathbb{E}[Y] - m)^2 + \text{Var}[Y] \quad \text{💡} \\ &= \text{Bias}^2 + \text{Variance} \end{aligned}$$

Predict a Random Variable from Its Distribution III

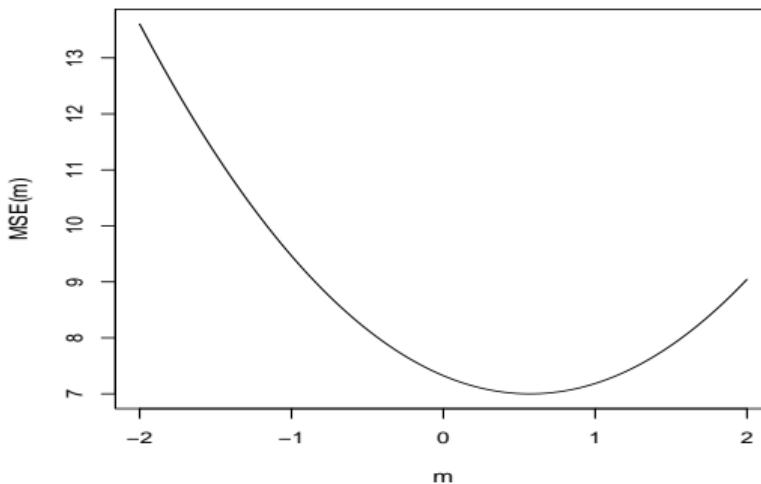


Figure: Mean squared error $\mathbb{E}[(Y - m)^2]$ as a function of the value m which we predict, when $\mathbb{E}[Y] = 0.57$, $\text{Var}[Y] = 7$.

Predict a Random Variable from Its Distribution IV

- We wish to pick m such that it minimizes $MSE(m)$ i.e. we want

$$m^* = \operatorname{argmin}_m MSE(m)$$

- Note that the variance term is irrelevant to making this small
- From basic calculus:

$$\frac{dMSE(m)}{dm} = -2(\mathbb{E}[Y] - m) \quad \text{and} \quad \frac{d^2MSE(m)}{dm^2} = 2$$

- First derivative is zero at $m^* = \mathbb{E}[Y]$, a unique minimum.
- So, the best possible one-number prediction m^* for Y is its expected value, i.e.

$$m^* = \mathbb{E}[Y]$$

Predict One Random Variable from Another I

- Two random variables, say X and Y .
- We know X and would like to use that knowledge to improve our guess about Y .
- Our guess is therefore a function of x , say $m(x)$.
- We would like

$$\mathbb{E}[(Y - m(X))^2]$$


to be small.

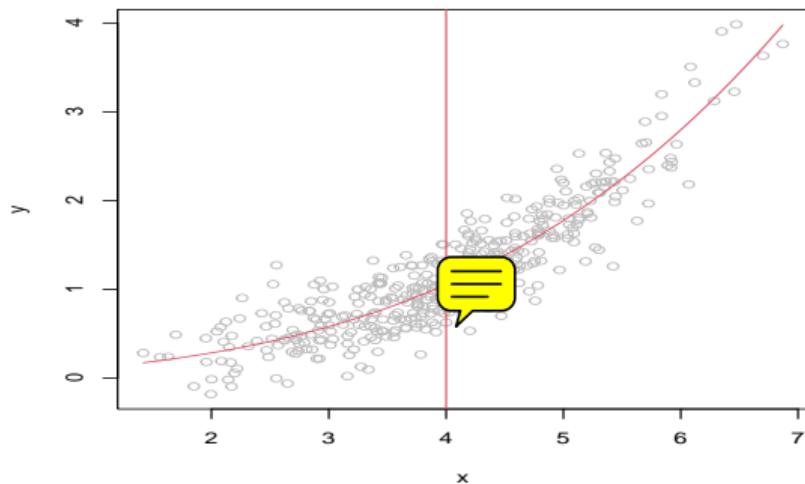
- Use conditional expectations to reduce this problem to the one already solved.

$$\mathbb{E}[(Y - m(X))^2] = \mathbb{E}[\mathbb{E}[(Y - m(X))^2 | X]]$$

- For each possible value x , the optimal value $m^*(x)$ is just the conditional expectation,
 $\mu(x) = \mathbb{E}[Y | X = x]$

Predict One Random Variable from Another II

The conditional expectation, $\mathbb{E}[Y|X = x]$ is the **regression function** of Y on X .



The Optimal Linear Predictor I

- Unfortunately, in general $\mu(x)$ is a really complicated function, for which there exists no nice mathematical expression.
- In searching optimal predictors, we restrict our attention to linear functions $m(x)$ that have the form $b_0 + b_1x$. (actually, that's an “affine” rather than a “linear” function.)
- The mean squared error of the linear model $b_0 + b_1x$ is a function of b_0 and b_1 (the parameters) and the variance of Y can be once again ignored. Specifically we find:



The Optimal Linear Predictor II

$$\begin{aligned}MSE(b_0, b_1) &= \mathbb{E}[(Y - (b_0 + b_1X))^2] \\&= \mathbb{E}[Y^2] - 2b_0\mathbb{E}[Y] - 2b_1\mathbb{E}[XY] + \mathbb{E}[(b_0 + b_1X)^2] \\&= \mathbb{E}[Y^2] - 2b_0\mathbb{E}[Y] - 2b_1(\text{Cov}[X, Y] + \mathbb{E}[X]\mathbb{E}[Y]) \\&\quad + b_0^2 + 2b_0b_1\mathbb{E}[X] + b_1^2\mathbb{E}[X^2] \\&= \mathbb{E}[Y^2] - 2b_0\mathbb{E}[Y] - 2b_1\text{Cov}[X, Y] - 2b_1\mathbb{E}[X]\mathbb{E}[Y] \\&\quad + b_0^2 + 2b_0b_1\mathbb{E}[X] + b_1^2\text{Var}[X] + b_1^2(\mathbb{E}[X])^2\end{aligned}$$

The Optimal Linear Predictor III

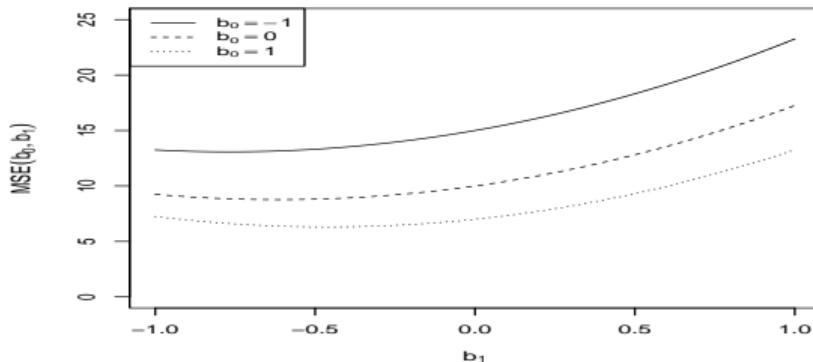


Figure: Mean squared error of linear models with different slopes (b_1) and intercepts (b_0), when $\mathbb{E}[X] = -0.5$, $\text{Var}[X] = 3$, $\mathbb{E}[Y] = 2$, $\mathbb{E}[Y^2] = 10$, $\text{Cov}[X, Y] = -1$. Each curve represents a different intercept b_0 in the linear model $b_0 + b_1 x$ for Y .

The Optimal Linear Predictor IV

- We minimize again by setting derivatives to zero

$$\frac{\partial MSE(b_0, b_1)}{\partial b_0} = -2\mathbb{E}[Y] + 2b_0 + 2b_1\mathbb{E}[X]$$

$$\frac{\partial MSE(b_0, b_1)}{\partial b_1} = -2\text{Cov}[X, Y] - 2\mathbb{E}[X]\mathbb{E}[Y] + 2b_0\mathbb{E}[X]$$

$$+ 2b_1\text{Var}[X] + 2b_1(\mathbb{E}[X])^2$$

- We denote the optimal value of b_0 and b_1 as β_0 and β_1 .

The Optimal Linear Predictor V

- From the first equation

$$\beta_0 = \mathbb{E}[Y] - \beta_1 \mathbb{E}[X]$$

Remarks

- The optimal intercept (β_0) enforces the line to go through the mean Y value at the mean X value. (To see this, add $\beta_1 \mathbb{E}[X]$ to both sides.)
- β_0 should have the same units as Y , and the right-hand side of this formula does, because β_1 has the units of Y/X .
- If the variables were “centered”, with $\mathbb{E}[X] = \mathbb{E}[Y] = 0$, we’d get $\beta_0 = 0$.

The Optimal Linear Predictor VI

- Now we plug this in to the other equation:

$$\begin{aligned} 0 &= -\text{Cov}[X, Y] - \mathbb{E}[X]\mathbb{E}[Y] + \beta_0\mathbb{E}[X] + \beta_1\text{Var}[X] + \beta_1(\mathbb{E}[X])^2 \\ &= -\text{Cov}[X, Y] - \mathbb{E}[X]\mathbb{E}[Y] + (\mathbb{E}[Y] - \beta_1\mathbb{E}[X])\mathbb{E}[X] \\ &\quad + \beta_1\text{Var}[X] + \beta_1(\mathbb{E}[X])^2 \\ &= -\text{Cov}[X, Y] + \beta_1\text{Var}[X] \end{aligned}$$

Then

$$\beta_1 = \frac{\text{Cov}[X, Y]}{\text{Var}[X]}$$



The Optimal Linear Predictor VII

$$\beta_1 = \frac{\text{Cov}[X, Y]}{\text{Var}[X]}$$

- Remarks

- The slope increases the more X and Y tend to fluctuate together, and gets pulled towards zero the more X varies.
- The expected values $\mathbb{E}[X]$ and $\mathbb{E}[Y]$ play no role in the formula, only the variance and covariance matter, and they don't change when we add or subtract constants. In particular, the optimal slope doesn't change if we use $Y - \mathbb{E}[Y]$ and $X - \mathbb{E}[X]$ (i.e. the centered variables) in the calculations.

- The line

$$\beta_0 + \beta_1 x$$

is the **optimal regression line** (of Y on X), or the **optimal linear predictor** (of Y on X).

Some important remarks I

- We have not assumed that the regression function $\mu(x)$ between X and Y really is linear.
- We have derived the optimal linear approximation to the true relationship, whatever that might be.
- This approximation works in some cases...
- Suppose that $\mu(x)$ is a “sufficiently” smooth function
- Use Taylor series around any value x_0 .

$$\mu(x) = \mu(x_0) + (x - x_0)\mu'(x_0) + \frac{1}{2}(x - x_0)^2\mu''(x_0) + \dots$$

Some important remarks II

- For x close enough to x_0 ,

$$\begin{aligned}\mu(x) &\approx \mu(x_0) + (x - x_0)\mu'(x_0) \\ &\approx \mu(x_0) - x_0\mu'(x_0) + \mu'(x_0)x \\ &\approx \beta_0 + \beta_1 x\end{aligned}$$

- How close is enough? Close enough that all the other terms don't matter, so, e.g., the quadratic term has to be negligible, meaning

$$\begin{aligned}|x - x_0|\mu'(x_0) &\gg \frac{1}{2}|x - x_0|^2\mu''(x_0) \quad \text{i.e.} \\ 2\mu'(x_0)/\mu''(x_0) &\gg |x - x_0|\end{aligned}$$

Unless the function is really straight, therefore, any linear approximation is only going to be good over very short ranges.

Some important remarks III

- Think to $\mathbb{E}[Y|X = x] = e^x$, or $\mathbb{E}[Y|X = x] = \sin(x)$.
- Linear models are *computationally* convenient, and there are many situations where computation is at a premium. For instance if you have huge amounts of data, or you need predictions very quickly, or your computing hardware is very weak (say old time computing machines level weak).
- Slogan: *Getting a simple answer can be better than getting the “right” answer*
- No assumptions about the marginal distributions of the variables or about the joint distribution of the two variables
- No assumptions about the fluctuations of Y around the optimal regression line
- No assumption that X came before Y in time, or that X causes Y (what would be the slope of the optimal regression of X on Y ?)
- No assumption that X is known precisely but Y with noise, etc.

Some important remarks IV

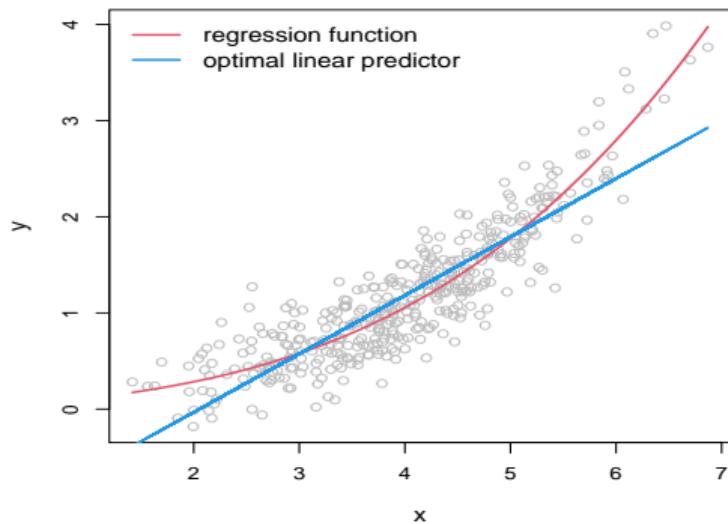


Figure: The regression function $\mathbb{E}[Y|X = x]$ and the optimal linear prediction

Empirical covariance and correlation ³

³This part is mainly based on a book written by Raphael Irizarry. From his book I have extracted or rephrased sentences.

The empirical standard deviation and variance I

- Define the empirical mean of a sample (x_1, \dots, x_n) as

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

- Define the empirical variance of a sample (x_1, \dots, x_n) as

$$s^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 = \frac{1}{n} \left(\sum_{i=1}^n x_i^2 - n\bar{x}^2 \right)$$

- The empirical standard deviation is defined as $s = \sqrt{s^2}$. Notice that the standard deviation has the same units as the data.

Normalization I

- The data defined by

$$z_i = \frac{x_i}{s}$$

have empirical standard deviation 1. This is called "scaling" the data.

- The data defined by

$$z_i = \frac{x_i - \bar{x}}{s}$$

have empirical mean zero and empirical standard deviation 1.

- The process of centering then scaling the data is called "normalizing" the data.
- Normalized data are centered at 0 and have units equal to standard deviations of the original data.

The empirical covariance I

- Consider now when we have pairs of data, (x_i, y_i) .

The empirical covariance

$$c_{XY} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) = \frac{1}{n} \left(\sum_{i=1}^n x_i y_i - n \bar{x} \bar{y} \right)$$

Empirical correlation coefficient I

- The correlation coefficient is defined for a sequence of pairs $(x_1, y_1), \dots, (x_n, y_n)$ as:

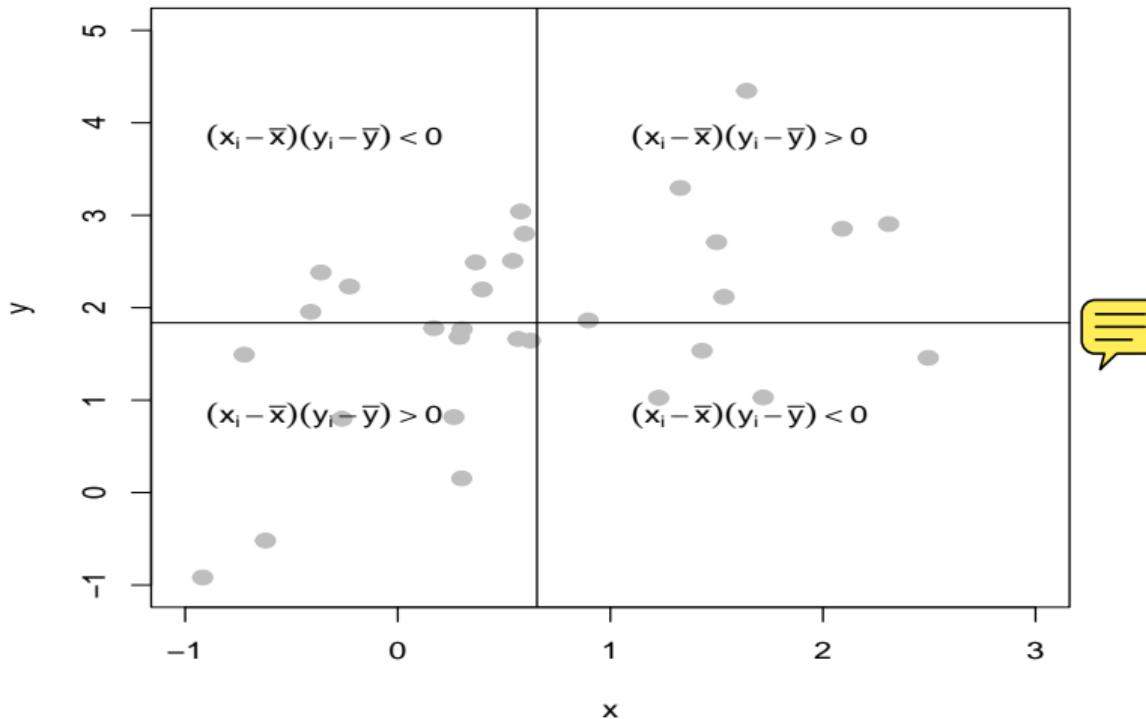
$$\begin{aligned} r_{XY} &= \frac{1}{n} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{s_X} \right) \left(\frac{y_i - \bar{y}}{s_Y} \right) \\ &= \frac{1}{s_X s_Y n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) = \frac{c_{XY}}{s_X s_Y} \end{aligned}$$



with \bar{x}, \bar{y} the averages of x_1, \dots, x_n and y_1, \dots, y_n respectively, and s_X, s_Y their standard deviations.

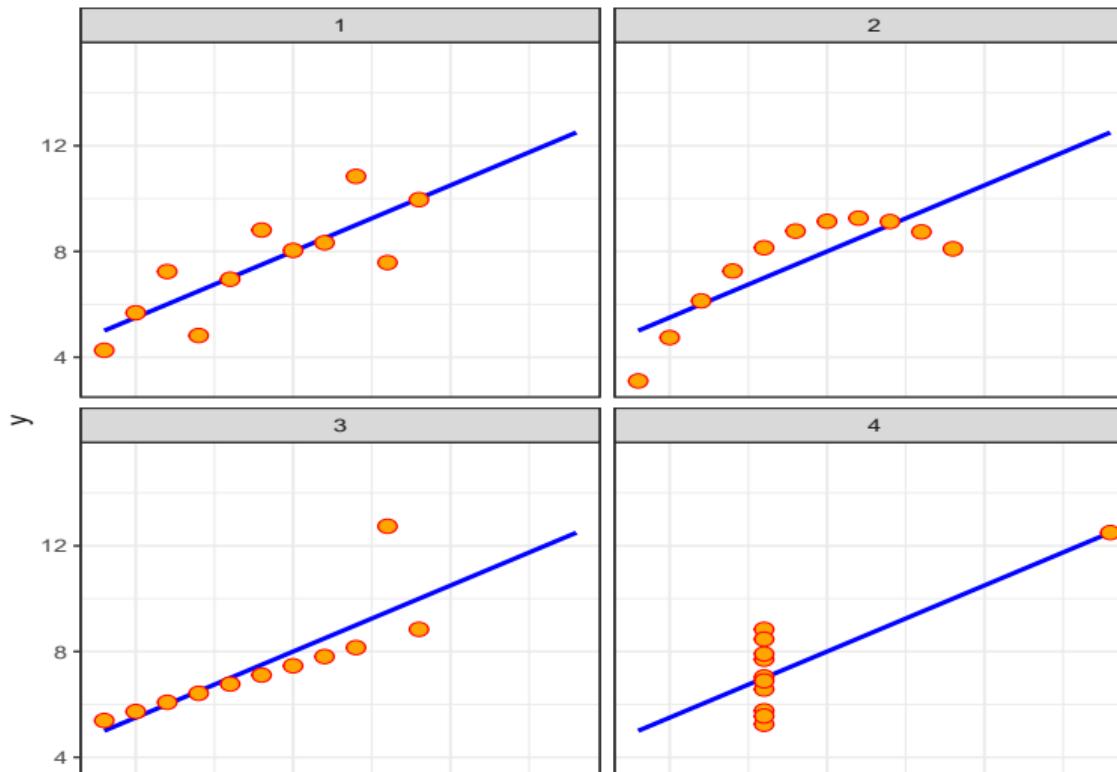
- This correlation coefficients is referred to as the Pearson's correlation coefficient and is based on deviations (i.e. $(x - \bar{x})$ and $(y - \bar{y})$). Other correlation coefficients based on rank also exist (Kendall's τ).

Empirical correlation coefficient II



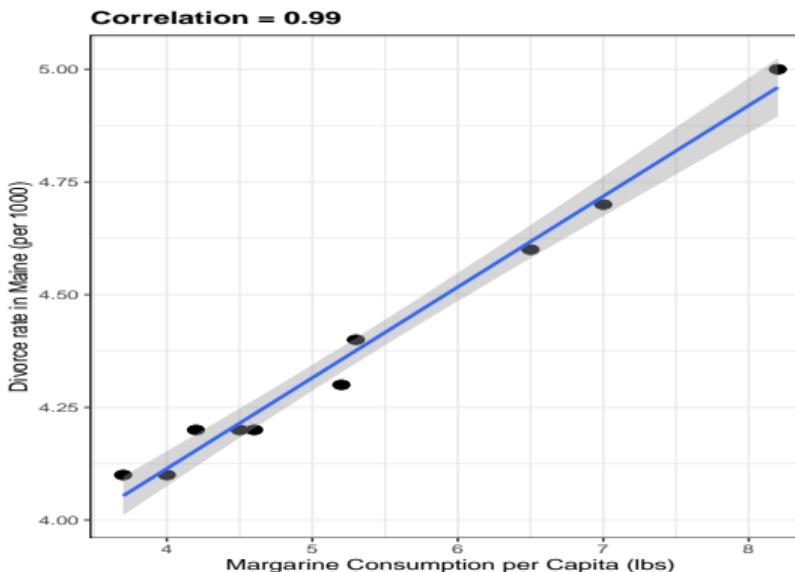
Pearson correlation's pitfalls !

- All these pairs have a correlation of 0.82:



Correlation is not causation I

- There are many reasons that a variable X can be correlated with a variable Y without either being a cause for the other.



Correlation is not causation II

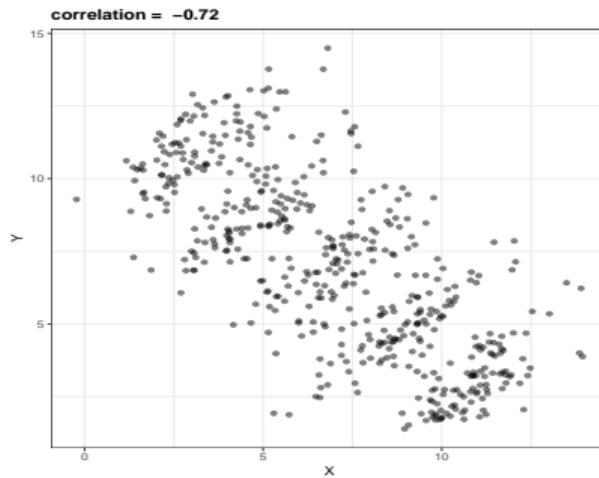
- Does this mean that margarine causes divorces? Or do divorces cause people to eat more margarine? Of course the answer to both these questions is no.
- This is just an example of what we call a **spurious correlation**.
- You can see many more absurd examples of spurious correlation on [this website](#)

Confounders

- **Confounders** are perhaps the most common reason that leads to associations being misinterpreted.
- If X and Y are correlated, we call Z a confounder if changes in Z cause changes in both X and Y .
- For example, ice cream sales and murder rates tend to rise together during hot summers. Does that mean that ice cream consumption causes murder, or that murder makes people crave ice cream?
- Of course not.
- Research shows that they both rise in the summer months because warm weather makes ice cream a particularly appealing treat and summer is a time when people are more likely to get together and to be outside, where they come into greater contact with one another.
- Take home message: determining causal relationships requires extensive research and subject matter expertise
- Take home message 2: observational data are often less than ideal when we wish to make causal statements

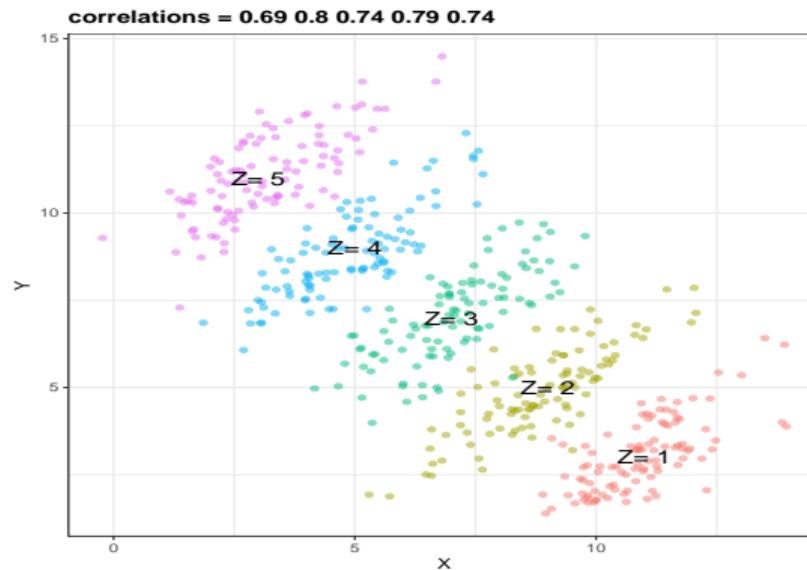
Simpson's Paradox I

- It is called a paradox because we see the sign of the correlation flip when comparing the entire dataset and specific strata.
- Suppose you have three variables X , Y and Z . Here is a scatterplot of Y versus X :



Simpson's Paradox II

- X and Y are negatively correlated. However, once we stratify by Z , shown in different colors below, another pattern emerges.



Simpson's Paradox III

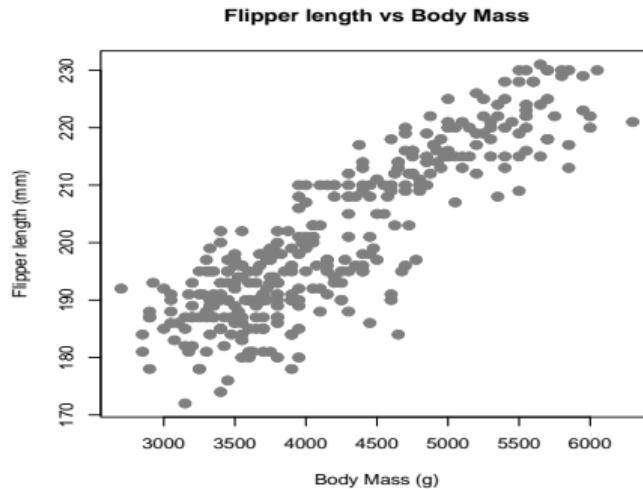
- It is really Z that is negatively correlated with X . If we stratify by Z , the X and Y are actually positively correlated

Simple linear regression model I⁴

⁴Material is based on the Lecture 4 of Cosma Shalizi's [course](#)

The Simple Linear Regression Model I

- This is a statistical model with two variables X and Y
- We wish to predict/explain Y from X .



- Of all possible functions $Y = m(X)$ we restrict ourselves to a linear (affine) function.

The Simple Linear Regression Model II

- In particular we make have the following

Assumptions of the model

- The distribution of X is arbitrary (and perhaps X is even non-random).
- $Y = \beta_0 + \beta_1 X + \varepsilon$, for some constants (“coefficients”, “parameters”) β_0 and β_1 , and some random noise variable ε .
- $\mathbb{E}[\varepsilon|X=x] = 0$ (no matter what x is), $\text{Var}[\varepsilon|X=x] = \sigma^2$ (no matter what x is).
- ε is uncorrelated across elements.
- With multiple pairs, $(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)$, then the model says that, for each $i = 1, \dots, n$,

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

where the noise variables ε_i all have the same expectation (0) and the same variance (σ^2), and $\text{Cov}[\varepsilon_i, \varepsilon_j] = 0$ (unless $i = j$, of course).

The Simple Linear Regression Model III

- The noise variable may represent measurement error, fluctuations in Y , the effect of the variables which affect Y which we have not (can not) include in the model, etc
- The assumption of **additive** noise is non-trivial — it's not absurd to imagine that either measurement error or fluctuations might change Y multiplicatively (for instance).
- The assumption of a **linear functional form** for the relationship between Y and X is non-trivial; lots of non-linear relationships actually exist.
- The assumption of **constant variance**, or **homoskedasticity**, is non-trivial; the non-correlation assumptions are non-trivial.
- But the assumption that the noise has mean 0 *is* trivial. (Why?)
- Ideally, all of the non-trivial assumptions will be checked, and we will talk later in the course about ways to check them.

Plug-in estimate I

- The optimal linear predictor of Y from X has slope and intercept

$$\beta_1 = \frac{\text{Cov}[X, Y]}{\text{Var}[X]}, \quad \beta_0 = \mathbb{E}[Y] - \beta_1 \mathbb{E}[X]$$

- $\mathbb{E}[Y]$, $\mathbb{E}[X]$ $\text{Cov}[X, Y]$ and $\text{Var}[X]$ are functions of the true distribution.
- Rather than having that full distribution, we merely have realizations, say $(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)$.
- How might we estimate β_1 from this empirical data?

Plug-in principle

$$\widehat{\beta}_1 = \frac{c_{XY}}{s_X^2}, \quad \widehat{\beta}_0 = \bar{y} - \widehat{\beta}_1 \bar{x}$$

Plug-in estimate II

- We can't hope that $\widehat{\beta}_1 = \beta_1$, but we *can* hope that as $n \rightarrow \infty$, $\widehat{\beta}_1 \rightarrow \beta_1$. i.e. that the estimator is **consistent**,
- Consider (with a slight abuse of notation, note the upper case) the estimators

$$\widehat{\beta}_1 = \frac{C_{XY}}{S_X^2}, \quad \widehat{\beta}_0 = \bar{Y} - \widehat{\beta}_1 \bar{X}$$



with

$$C_{XY} = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})$$

and

$$S_X^2 = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2$$

Plug-in estimate III

- Notice that we need $S_x^2 > 0$ for $\widehat{\beta}_1$ to be defined: if there is no variation in X we can not really say much about the effect that X has on Y
- The estimators are consistent if $\bar{Y} \rightarrow \mathbb{E}[Y]$, $\bar{X} \rightarrow \mathbb{E}[X]$, and $C_{XY} \rightarrow \text{Cov}[XY]$, and $S_X^2 \rightarrow \text{Var}[X]$.
- On the other hand, it would be nice to say more: we want to know *how far* from the truth our estimate is likely to be, whether it tends to over- or under- estimate the slope, etc.
- We will see in later lectures how the assumptions of the simple linear regression model will let us say something about all of these matters, and how the even stronger assumption that the noise is Gaussian will let us be even more precise.



Least Squares Estimates I

- An alternative way of estimating the simple linear regression model
- We keep the assumption that we seek a linear relationship between X and Y :
$$Y = f(X) = \beta_0 + \beta_1 X$$
- Find β_0 and β_1 which minimize the sum of squared difference between the observed Y_i and the value of Y_i under the model $b_0 + b_1 X_i$, i.e.



$$(\beta_0, \beta_1) = \arg \min_{(b_0, b_1)} \sum_{i=1}^n (Y_i - (b_0 + b_1 X_i))^2$$



- The sum of squared deviations corresponds to the

The empirical MSE

$$\widehat{MSE}(b_0, b_1) = \frac{1}{n} \sum_{i=1}^n (y_i - (b_0 + b_1 x_i))^2$$

Least Squares Estimates II

- If our samples are all independent, for any fixed (b_0, b_1) , the law of large numbers tells us that $\widehat{MSE}(b_0, b_1) \rightarrow MSE(b_0, b_1)$ as $n \rightarrow \infty$.
- So it doesn't seem unreasonable to try minimizing the in-sample error

$$\frac{\partial \widehat{MSE}}{\partial b_0} = \frac{1}{n} \sum_{i=1}^n (y_i - (b_0 + b_1 x_i))(-2)$$

$$\frac{\partial \widehat{MSE}}{\partial b_1} = \frac{1}{n} \sum_{i=1}^n (y_i - (b_0 + b_1 x_i))(-2x_i)$$

Set these to zero at the optimum $(\widehat{\beta}_0, \widehat{\beta}_1)$:

Least Squares Estimates III

The estimating equations

$$\frac{1}{n} \sum_{i=1}^n (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)) = 0$$

$$\frac{1}{n} \sum_{i=1}^n (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)) x_i = 0$$

i.e.

$$\bar{y} - \hat{\beta}_0 - \hat{\beta}_1 \bar{x} = 0$$

$$\bar{xy} - \hat{\beta}_0 \bar{x} - \hat{\beta}_1 \bar{x^2} = 0$$

Least Squares Estimates IV

The first equation, re-written, gives

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

Substituting this into the remaining equation,

$$\begin{aligned} 0 &= \bar{xy} - \bar{y}\bar{x} + \hat{\beta}_1\bar{x}\bar{x} - \hat{\beta}_1\bar{x}^2 \\ 0 &= c_{XY} - \hat{\beta}_1 s_X^2 \\ \hat{\beta}_1 &= \frac{c_{XY}}{s_X^2} \end{aligned}$$

The least square estimates

$$\hat{\beta}_1 = \frac{c_{XY}}{s_X^2}, \quad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

Least Squares Estimates V

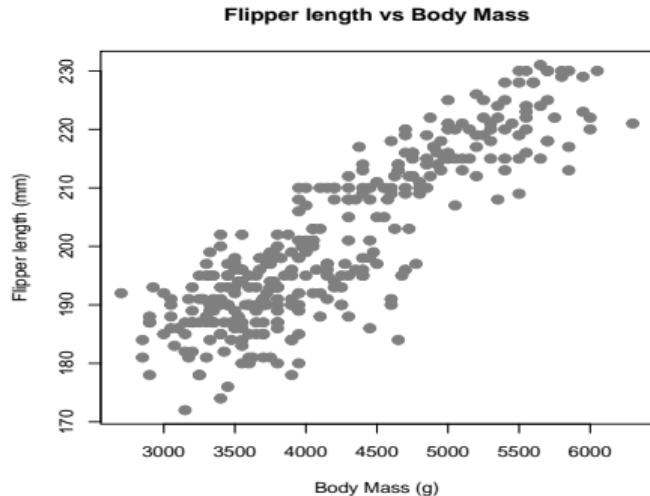
- The equivalence between the plug-in estimates and the least-squares estimates is a bit of a special case for linear models. In some non-linear models, least squares is quite feasible (though the optimum can only be found numerically, not in closed form); in others, plug-in estimates are more useful than optimization.

Penguins I

- Palmer's penguins dataset
- How does the size of the penguin (body ass) affects the length of the flipper?
- Let's have a look

```
plot(flipper_length_mm ~ body_mass_g, data = penguins,  
      ylab = "Flipper length (mm)",  
      xlab = "Body Mass (g)",  
      main = "Flipper length vs Body Mass",  
      pch = 20, cex = 2, col = "grey50")
```

Penguins II



- We define regressor (flipper length) and predictor (mass) with a notation consistent with above mathematics

Penguins III

```
x <- penguins$body_mass_g  
y <- penguins$flipper_length_mm
```

- We calculate the statistics

```
my<-mean(y);    mx<-mean(x)  
Cxy <- mean((x - mx) * (y - my))  
Sxx <- mean((x - mx) ^ 2)
```

- We finally calculate $\hat{\beta}_0$ and $\hat{\beta}_1$.

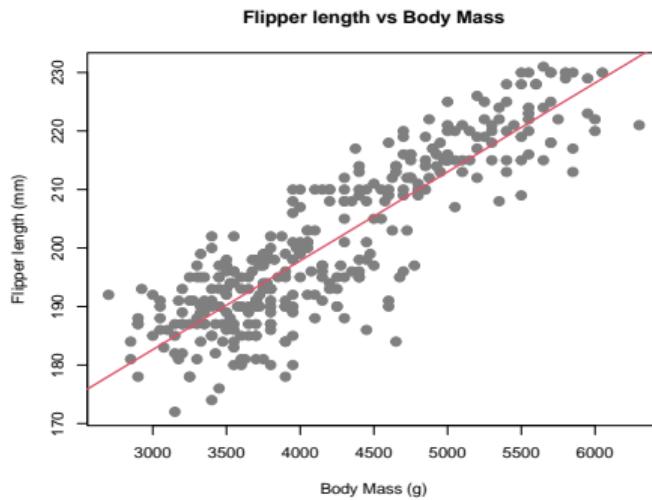
```
beta_1_hat <- Cxy / Sxx  
beta_0_hat <- my - beta_1_hat * mx  
c(beta_0_hat, beta_1_hat) ## vector of estimated betas  
[1] 137.03962089    0.01519526
```

Penguins IV

We can now write the **fitted** or estimated line,

$$\hat{m}(x) = \hat{\beta}_0 + \hat{\beta}_1 x.$$

In this case $\hat{m}(x) = 137.04 + 0.02x$.



Penguins V

Is this the least square?

```
sum((y - (beta_0_hat+beta_1_hat*x))^2)
[1] 15516.03
## take some other values of beta_0 and beta_1
sum((y - (138+0.015*x))^2)
[1] 15530.66
sum((y - (137+0.016*x))^2)
[1] 19383.48
### try any other value
# the estimates minimize the sum of squares
```

Penguins VI

Let's make an estimation for the mean of the flipper of a penguin who weights 3100gr using the estimated model.

The observed value is

```
penguins[penguins$body_mass_g == 3100,]
```

```
  flipper_length_mm body_mass_g
```

```
43           186          3100
```

The estimate is: $\hat{m}(3100) = \hat{\beta}_0 + \hat{\beta}_1 \cdot 8 = 137.04 + 0.02 \cdot 3100$

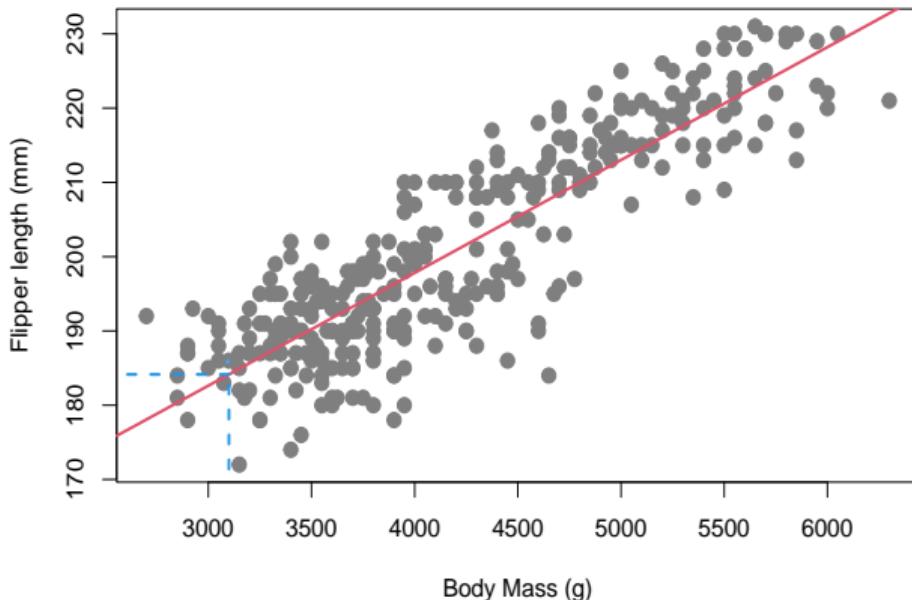
```
beta_0_hat + beta_1_hat * 3100
```

```
[1] 184.1449
```

Not too far off

Penguins VII

Flipper length vs Body Mass



Penguins VIII

Now: an estimation for the mean of the flipper of a penguin who weights 3420gr.

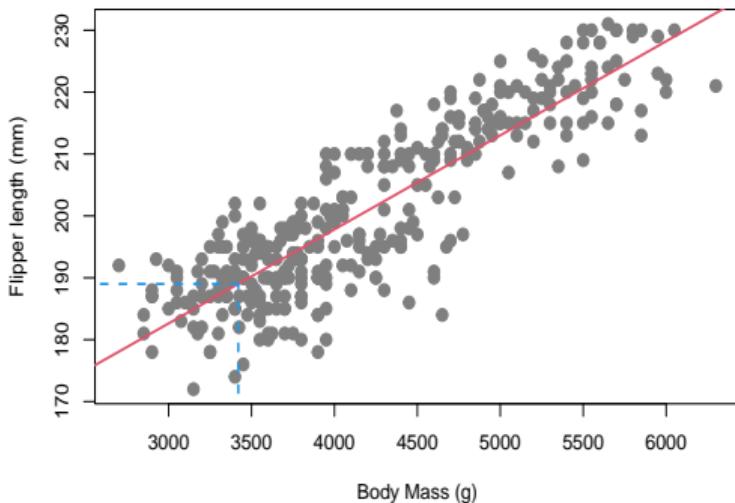
```
penguins[penguins$body_mass_g == 3420,]  
[1] flipper_length_mm body_mass_g  
<0 rows> (or 0-length row.names)  
## no observations
```

This is considered **interpolation** as 3420 is not an observed value of x . (But is in the data range.)

```
beta_0_hat + beta_1_hat * 3420  
[1] 189.0074
```

Penguins IX

Flipper length vs Body Mass



Penguins X

Lastly, we can make a prediction for the mean of the flipper of a penguin who weights 7500gr. This is considered **extrapolation** as 7500 is not an observed value of x and is outside data range.

```
beta_0_hat + beta_1_hat * 7500
```

```
[1] 251.0041
```

We should be less confident in estimates of this type, there are no penguins of that body size.

We will see how to quantify this higher uncertainty in a formal way (under some assumptions).

Penguins XI



Function lm |

- In R we can obtain the least squares estimates using the `lm` function
- Syntax

```
lm(formula = response ~ predictor, data = data)
```

where `response` and `predictor` are the names of two variables/columns in the data frame `data` (see `?lm`).

```
lm(flipper_length_mm ~ body_mass_g, data = penguins)
```

Call:

```
lm(formula = flipper_length_mm ~ body_mass_g, data = penguins)
```

Coefficients:

(Intercept)	body_mass_g
137.0396	0.0152

Function lm ||

- The lm object

```
fit<-lm(flipper_length_mm ~ body_mass_g, data = penguins); fit
```

Call:

```
lm(formula = flipper_length_mm ~ body_mass_g, data = penguins)
```

Coefficients:

(Intercept)	body_mass_g
137.0396	0.0152

- fit is a list of objects whose names are

```
typeof(fit)
```

```
[1] "list"
```

Function lm III

```
names(fit)
[1] "coefficients"   "residuals"      "effects"
[4] "rank"           "fitted.values" "assign"
[7] "qr"             "df.residual"   "xlevels"
[10] "call"          "terms"         "model"
```

- We can access these elements by \$

```
fit$coefficients
```

```
(Intercept) body_mass_g
137.03962089  0.01519526
```

```
fit$call
```

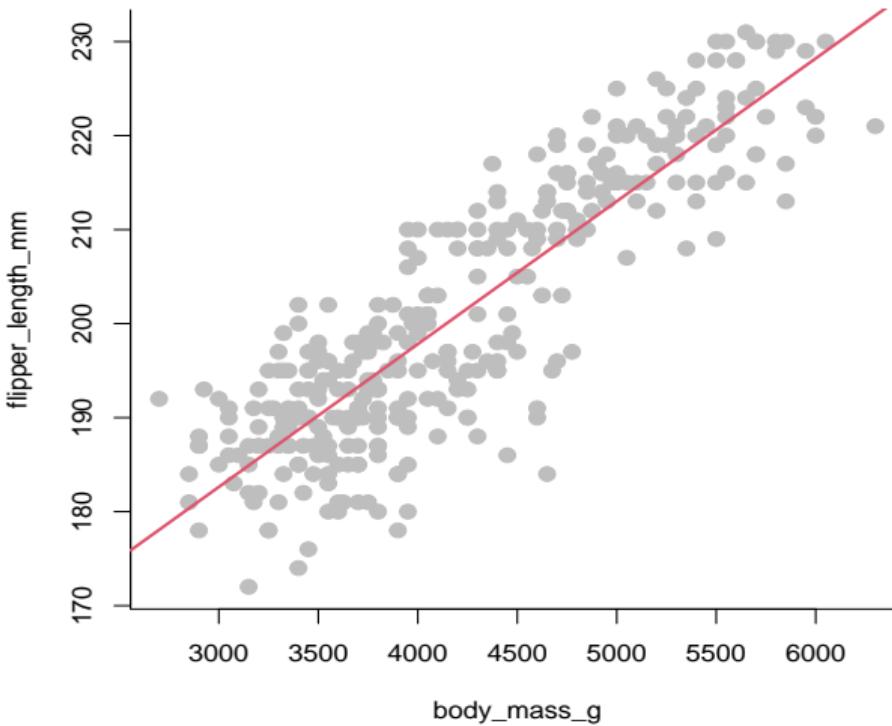
```
lm(formula = flipper_length_mm ~ body_mass_g, data = penguins)
```

Function lm IV

- We can produce a plot with the linear fit easily

```
plot(flipper_length_mm ~ body_mass_g, data = penguins,  
      # some cosmetic changes  
      pch = 20, cex = 2, # filled dots of twice the size  
      col = "grey", ## color  
      bty = "l" ## box for the plot  
)  
## draw a line: first number is intercept second is slope  
abline(coef = fit$coefficients, col = 2, lwd = 2)
```

Function lm V



Estimating the slope - a small note

We have seen before that

$$\widehat{\beta}_1 = \frac{c_{XY}}{s_X^2}$$



But we know from the definition of the covariance function that: $c_{XY} = r_{XY} s_X s_Y$ (where r_{XY} is the empirical pearson correlation coefficient).

We can then write:

$$\widehat{\beta}_1 = \frac{r_{XY} s_X s_Y}{s_X^2} = r_{XY} \frac{s_Y}{s_X}$$



The regression slope is a re-weighting of the Pearson's correlation coefficient which accounts for the variability of the individual variables.

Since s_X and s_Y are positive the sign of $\widehat{\beta}_1$ is the same as the sign of r_{XY} .

Bias and Variance of Parameter Estimates I

- It can be shown that the least square estimates are unbiased and have a certain specified variance
- The results are obtained for “designed” or “controlled” experiments, where we get to chose the X values. We therefore derive the conditional expectations $\mathbb{E} [\hat{\beta}_0 | x_1, \dots, x_n]$, $\mathbb{E} [\hat{\beta}_1 | x_1, \dots, x_n]$ and conditional variances $\text{Var} [\hat{\beta}_0 | x_1, \dots, x_n]$, $\text{Var} [\hat{\beta}_1 | x_1, \dots, x_n]$. In randomized experiments or in observational studies, obviously the x_i aren't necessarily fixed; however, the results are still correct.
- We have:

$$\mathbb{E} [\hat{\beta}_0 | x_1, \dots, x_n] = \beta_0 \quad \text{and} \quad \mathbb{E} [\hat{\beta}_1 | x_1, \dots, x_n] = \beta_1$$

Bias and Variance of Parameter Estimates II

- first remember that:

$$\begin{aligned}\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) &= \sum (x_i y_i + \bar{x}\bar{y} - \bar{x}y_i - \bar{y}x_i) = \\ \sum x_i y_i + n\bar{x}\bar{y} - \bar{x} \sum y_i - \bar{y} \sum x_i &= n\bar{x}\bar{y} - n\bar{x}\bar{y} + \sum (x_i y_i - \bar{x})y_i = \\ \sum (x_i - \bar{x})y_i &= \sum (y_i - \bar{y})x_i\end{aligned}$$

so

$$\begin{aligned}\widehat{\beta}_1 &= \frac{c_{xy}}{s_x^2} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2} = \frac{\sum (x_i - \bar{x})y_i}{\sum (x_i - \bar{x})^2} = \frac{\sum (x_i - \bar{x})(\beta_0 + \beta_1 x_i + \varepsilon_i)}{(x_i - \bar{x})^2} \\ &= \frac{\beta_0 \sum (x_i - \bar{x}) + \beta_1 \sum (x_i - \bar{x})x_i + \sum (x_i - \bar{x})\varepsilon_i}{\sum (x_i - \bar{x})^2} = \frac{0 + \beta_1 \sum (x_i - \bar{x})^2 + \sum (x_i - \bar{x})\varepsilon_i}{\sum (x_i - \bar{x})^2} \\ &= \beta_1 + \frac{\sum (x_i - \bar{x})\varepsilon_i}{\sum (x_i - \bar{x})^2}\end{aligned}$$

Bias and Variance of Parameter Estimates III

Since $\mathbb{E}[\varepsilon_i | X_i] = 0$, we have

$$\mathbb{E}[\widehat{\beta}_1 | x_1, \dots, x_n] = \mathbb{E}\left[\beta_1 + \frac{\sum(x_i - \bar{x})\varepsilon_i}{\sum(x_i - \bar{x})^2} | x_1, \dots, x_n\right] = \beta_1.$$

It follows that

$$\mathbb{E}[\widehat{\beta}_0 | x_1, \dots, x_n] = \mathbb{E}[\bar{y} - \widehat{\beta}_1 \bar{x}] = \mathbb{E}\left[\beta_0 + \beta_1 \bar{x} - \widehat{\beta}_1 \bar{x} | x_1, \dots, x_n\right] = \beta_0$$

Bias and Variance of Parameter Estimates IV

- and we can show that

$$\text{Var} \left[\hat{\beta}_0 | x_1, \dots, x_n \right] = \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{ns_X^2} \right] \quad \text{and} \quad \text{Var} \left[\hat{\beta}_1 | x_1, \dots, x_n \right] = \frac{\sigma^2}{ns_X^2}$$

$$\begin{aligned}\text{Var} \left[\hat{\beta}_1 \right] &= \text{Var} \left[\beta_1 + \frac{\sum (x_i - \bar{x})\varepsilon_i}{\sum (x_i - \bar{x})^2} \right] = \text{Var} \left[\frac{\sum (x_i - \bar{x})\varepsilon_i}{\sum (x_i - \bar{x})^2} \right] = \\ &= \text{Var} \left[\frac{\sum (x_i - \bar{x})\varepsilon_i}{\sum (x_i - \bar{x})^2} \right] = \frac{\sum (x_i - \bar{x})^2 \text{Var} [\varepsilon_i]}{(\sum (x_i - \bar{x})^2)^2} \\ &= \frac{ns_X^2 \sigma^2}{(ns_X^2)^2} = \frac{\sigma^2}{ns_X^2}\end{aligned}$$



Bias and Variance of Parameter Estimates V

and

$$\text{Var} [\hat{\beta}_0] = \text{Var} [\bar{y} - \hat{\beta}_1 \bar{x}] = \frac{\sigma^2}{n} + \frac{\sigma^2}{ns_X^2} \bar{x}^2$$

Bias and Variance of Parameter Estimates VI

- Both parameter estimates are unbiased: we can hope that when n is large we recover their real values.
- Looking at the variances: they both have σ^2 at the numerator; and s_x^2 and n at the denominator.
- So, the variability increases when the noise around the regression line increases.
- And the variability decreases as we have more observations (n), which are further spread out along the horizontal axis (s_x^2).

Bias and Variance of Parameter Estimates VII

- Finally, the **standard error** of an estimator is just its standard deviation, or the square root of its variance:

$$se(\hat{\beta}_1) = \frac{1}{\sqrt{ns_X^2}} \quad \text{and} \quad se(\hat{\beta}_0) = \sigma \sqrt{\frac{1}{n} + \frac{\bar{x}^2}{ns_X^2}}$$

- It can be shown that the least square estimates are in some sense optimal: they are the minimum-variance mean-unbiased estimators for the simple linear model (when the assumptions are valid).

Unconditional-on-X Properties

- We need to find $\mathbb{E}[\hat{\beta}_1]$ and $\text{Var}[\hat{\beta}_1]$, not just $\mathbb{E}[\hat{\beta}_1|x_1, \dots, x_n]$ and $\text{Var}[\hat{\beta}_1|x_1, \dots, x_n]$.
- We use the **law of total expectation**:

$$\begin{aligned}\mathbb{E}[\hat{\beta}_1] &= \mathbb{E}[\mathbb{E}[\hat{\beta}_1|X_1, \dots, X_n]] \\ &= \mathbb{E}[\beta_1] = \beta_1\end{aligned}$$

- That is, the estimator is *unconditionally* unbiased.
- To get the unconditional variance, we use the **law of total variance**:

$$\begin{aligned}\text{Var}[\hat{\beta}_1] &= \mathbb{E}[\text{Var}[\hat{\beta}_1|X_1, \dots, X_n]] + \text{Var}[\mathbb{E}[\hat{\beta}_1|X_1, \dots, X_n]] \\ &= \mathbb{E}\left[\frac{\sigma^2}{nS_X^2}\right] + \text{Var}[\beta_1] = \frac{\sigma^2}{n}\mathbb{E}\left[\frac{1}{S_X^2}\right]\end{aligned}$$

Estimator Properties by simulation I

- A numerical illustration in which we repeat $m = 1000$ simulation of $n = 100$ observations (x_i, y_i) , $i = 1, \dots, n$ from

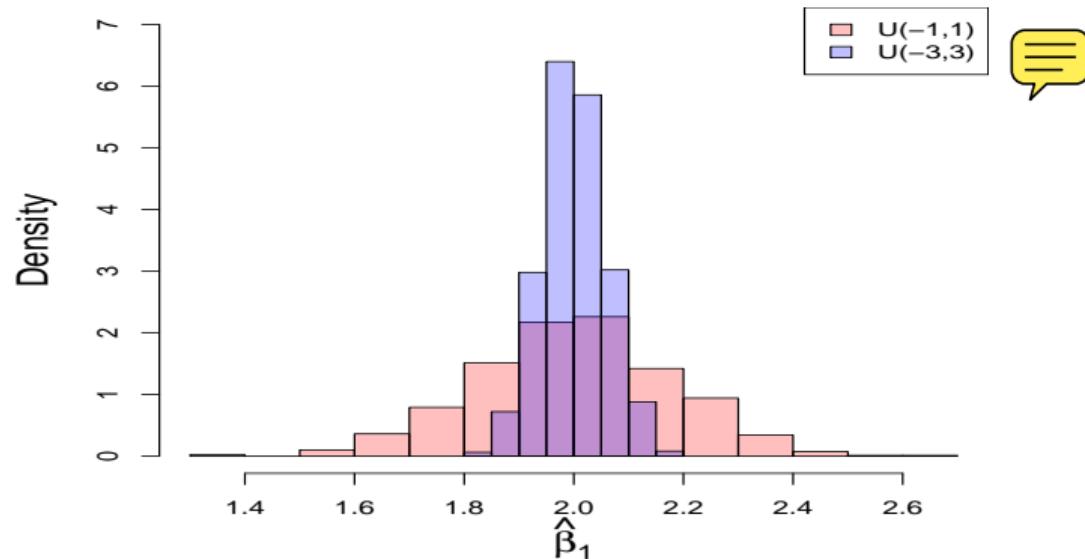
$$Y = \beta_0 + \beta_1 X + \varepsilon$$

where $\beta_0 = 1$, $\beta_1 = 2$ and $\varepsilon \sim \text{Exp}(1)$, i.e. $\mathbb{E}[\varepsilon] = 0$. We consider two setup namely

- $X \sim U(-1, 1)$
 - $X \sim U(-3, 3)$
- Exercise: Implement this example in R.

Simulation is a powerful way to study the behavior of estimates and models - we will explore this in the R practical sessions.

Estimator Properties by simulation II



Estimator Properties by simulation III

$\text{mean}(\widehat{\beta}_0) = 1, \text{sd}(\widehat{\beta}_0) = 0.1; \text{mean}(\widehat{\beta}_1) = 2, \text{sd}(\widehat{\beta}_1) = 0.18$
 $\text{mean}(\widehat{\beta}_0) = 1, \text{sd}(\widehat{\beta}_0) = 0.1; \text{mean}(\widehat{\beta}_1) = 2, \text{sd}(\widehat{\beta}_1) = 0.06$

Predictions I

- If we knew β_0 and β_1 , at $X = x$ our prediction for Y would be

$$\beta_0 + \beta_1 x.$$

- If the simple linear regression model is *true*, i.e. $\mathbb{E}[Y|X = x] = \beta_0 + \beta_1 x$, this is the best prediction we can make.
- However β_0 and β_1 are unknown and we predict that on average Y at an *arbitrary* value of X , say x will be

Point prediction

$$\hat{m}(x) = \hat{\beta}_0 + \hat{\beta}_1 x$$

- But $\hat{m}(x)$ is an *estimate* of $\mathbb{E}[Y|X = x]$, i.e. a function of our data, which are random, hence $\hat{m}(x)$ is also random.

Predictions II

- We analyze the randomness in $\hat{m}(x)$.
- First we ask what is its expected value:

$$\mathbb{E} [\hat{m}(x) | x, x_1, \dots, x_n] = \mathbb{E} [\widehat{\beta}_0 + \widehat{\beta}_1 x | x, x_1, \dots, x_n] = \beta_0 + \beta_1 x.$$

- The predictions are unbiased.
- Next we check the variance. To do that we rewrite $\hat{m}(x)$ as:

$$\hat{m} = (\bar{Y} - \widehat{\beta}_1 \bar{x}) + \widehat{\beta}_1 x = \bar{Y} + (x - \bar{x}) \widehat{\beta}_1$$

Predictions III

- so we can write:

$$\begin{aligned}\text{Var} [\hat{m}(x)|x, x_1, \dots, x_n] &= \text{Var} [\widehat{\beta}_0 + \widehat{\beta}_1 x | x, x_1, \dots, x_n] \\&= \text{Var} [\bar{Y} + (x - \bar{x})\widehat{\beta}_1 | x, x_1, \dots, x_n] \\&= \frac{\sigma^2}{n} + (x - \bar{x})^2 \frac{\sigma^2}{ns_x^2} \\&= \frac{\sigma^2}{n} \left(1 + \frac{(x - \bar{x})^2}{s_x^2} \right).\end{aligned}$$

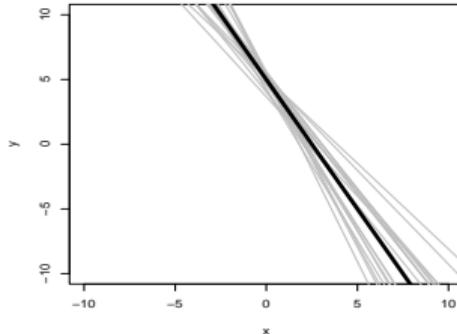

Predictions IV

$$\text{Var} [\hat{m}(x)|x, x_1, \dots, x_n] = \frac{\sigma^2}{n} \left(1 + \frac{(x - \bar{x})^2}{s_X^2} \right).$$

- The variance grows as σ^2 grows: the noisier the data the more variable the estimated regression line, and the more of that noise will propagate into our predictions.
- The larger n is, the smaller the variance: the more points we see, the more exactly we can pin down the line, and so our predictions.
- The variance of our predictions is the sum of two terms. The first term is σ^2/n : the variance of \bar{Y} . Since our line has to go through the center of the data, this just how much noise there is in the height of that center.
- The other term does change with x , specifically with $(x - \bar{x})^2$: the further our operating point x is from the center of the data \bar{x} , the bigger our uncertainty.

Predictions V

- The Figure illustrates the spread in point predictions as we move away from \bar{x} . We plot several estimated least-squares regression lines (grey) and the true regression line (black).



- Notice how the estimated lines become more spread out as we move away from the mean of the distribution of X (here set at 0)
- Again, the previous equation are conditional on the x_i . If those are random, we need to use the law of total variance to get the unconditional variance of $\hat{m}(X)$.

Estimating σ^2 : Sum of Squared Errors I

- Under the simple linear regression model, it is easy to show that

$$\mathbb{E} \left[(Y - (\beta_0 + \beta_1 X))^2 \right] = \sigma^2 \quad (1)$$

- This suggests that the minimal value of the in-sample MSE,

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{m}(x_i))^2 \quad (2)$$

is a natural estimator for σ^2 .

- This is, in fact, a **consistent** estimator. (You can prove this using the consistency of $\hat{\beta}_0$ and $\hat{\beta}_1$, and continuity.)

Estimating σ^2 : Sum of Squared Errors II

- It is, a slightly **biased** estimator. Specifically

$$s_e^2 = \frac{n}{n-2} \hat{\sigma}^2.$$

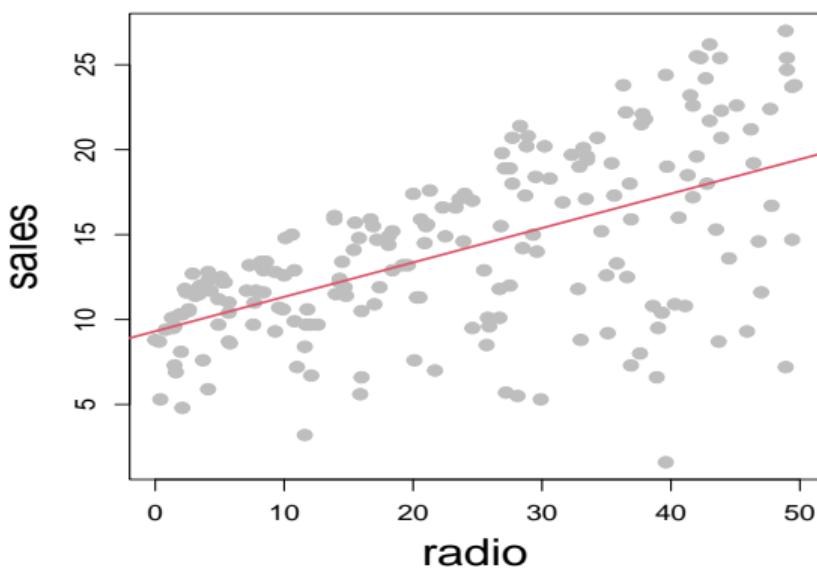

It turns out that S_e^2 is an *un*-biased estimator of σ^2 , though one with a larger variance.

Sum of squared errors

$$SSE = \sum_{i=1}^n (y_i - \hat{m}(x_i))^2 = \sum_{i=1}^n (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2$$

Advertising data

- Let's use the Advertising data set and study the relationship between radio and Sales.



Residuals I

The residuals



$$e_i = y_i - \hat{m}(x_i) = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i) \quad (3)$$

- This may look like re-arranging the basic equation for the linear regression model,

The errors

$$\varepsilon_i = Y_i - (\beta_0 + \beta_1 X_i) \quad (4)$$

and it *is* similar, but it's *not* the same.

- The right-hand side of Eq. 4 involves the true parameters.
- The right-hand side of Eq. 3 involves the *estimated* parameters, which are different.
- Therefore: The residuals are not the noise terms; $e_i \neq \varepsilon_i$, but ...

Residuals II

- ... some ways in which the residuals are *like* the noise terms.
 - The residuals are always uncorrelated with the x_i :

$$\frac{1}{n} \sum_{i=1}^n e_i(x_i - \bar{x}) = 0 \quad (5)$$

However, this is a consequence of the estimating equations: it is true *whether or not* the simple linear regression model is actually true.

$$\frac{1}{n} \sum_{i=1}^n e_i = 0 \quad (6)$$

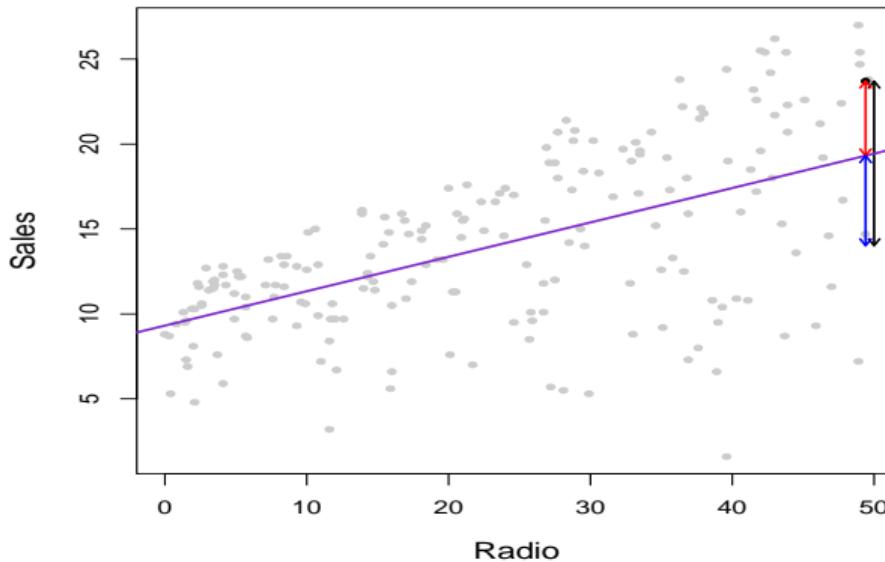
Also a consequence of the estimating equations. This is *reminiscent* of $\mathbb{E}[\varepsilon] = 0$, but generally $n^{-1} \sum_{i=1}^n \varepsilon_i \neq 0$.

Residuals III

- Despite these differences, there is enough of a relationship between the ε_i and the e_i that a lot of our model-checking and diagnostics will be done in terms of the residuals.

Decomposition of the sum of squares I

- How to judge the goodness of fit in a synthetic way?



Decomposition of the sum of squares II

- Decomposition of the sum of squares; $SS_{\text{tot}} = SS_{\text{reg}} + SS_{\text{res}}$

Decomposition of the sum of squares III

- Ingredients (define $\hat{y}_i = \hat{m}(x_i) = \hat{\beta}_0 + \hat{\beta}_1 x_i$):

- the **total sum of squares**

$$SS_{\text{tot}} = \sum_{i=1}^n (y_i - \bar{y})^2$$

this is n times the variance of y , $s_Y^2 = SS_{\text{tot}}/n$

- the **explained sum of squares**

$$SS_{\text{reg}} = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2$$

- the **residual sum of squares**

$$SS_{\text{res}} = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n e_i^2 = (n - 2)s_e^2$$

Decomposition of the sum of squares IV

Derived from:

$$\begin{aligned} SS_{tot} &= \sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n (y_i - \bar{y} + \hat{y}_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^n \left\{ (y_i - \hat{y}_i)^2 + (\hat{y}_i - \bar{y})^2 + 2(y_i - \hat{y}_i)(\hat{y}_i - \bar{y}) \right\} = \\ &= SS_{Res} + SS_{Reg} + 2 \sum_{i=1}^n (y_i - \hat{y}_i)(\hat{\beta}_0 + \hat{\beta}_1 x_i - \bar{y}) \\ &= SS_{Res} + SS_{Reg} + 2(\hat{\beta}_0 - \bar{y}) \sum_{i=1}^n (y_i - \hat{y}_i) + 2\hat{\beta}_1 \sum_{i=1}^n (y_i - \hat{y}_i)x_i \\ &= SS_{Res} + SS_{Reg} \end{aligned}$$

where the last two terms cancel out because of the estimating equations.

The coefficient of determination

- The **coefficient of determination** R^2 is the proportion of variability in the response that is accounted for by the statistical model

R^2 index

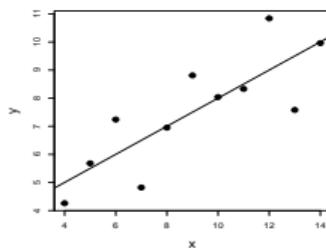
$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}} = \frac{SS_{\text{reg}}}{SS_{\text{tot}}}$$

- Properties of R^2 :
 - R^2 assumes values between 0 and 1
 - the better the model, the smaller the residual sum of squares
- In our example, the R^2 is 0.332
- The latter value is quite high but is not enough for judging the quality of the goodness of fit.

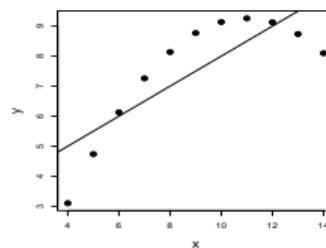
Exercise: show that for the simple linear regression R^2 is exactly the squared value of the correlation coefficient (r)

Inspecting the residuals I

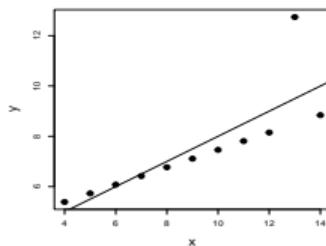
The same "quality" of fitting ...



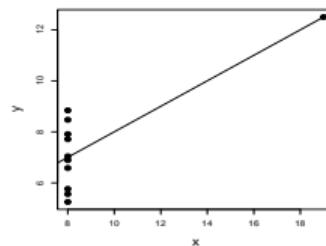
$$R^2 = 0.667$$



$$R^2 = 0.666$$



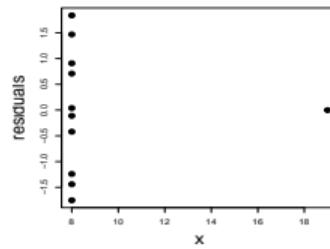
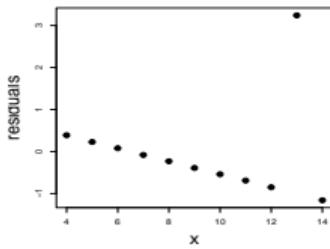
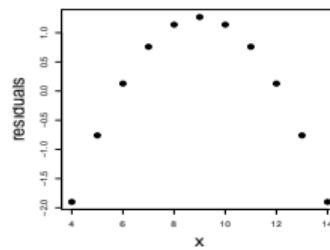
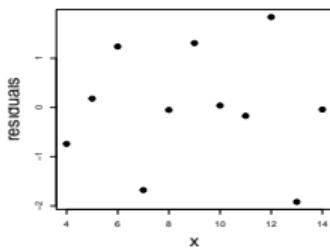
$$R^2 = 0.666$$



$$R^2 = 0.667$$

Inspecting the residuals II

... but different residuals



Summary on properties of the residuals

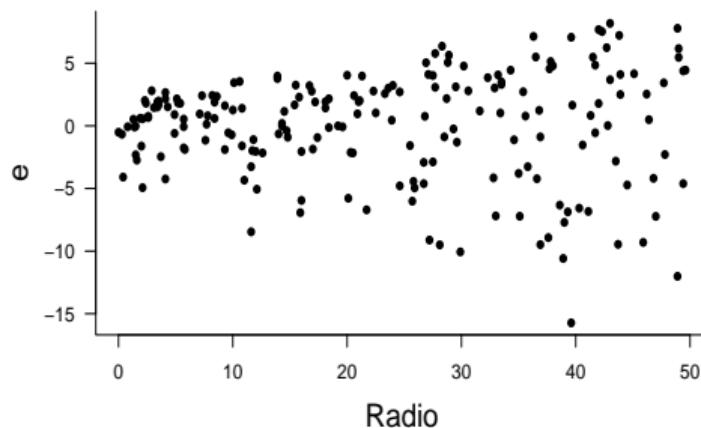
Let's sum up the most relevant observations from the last slides:

- ① The residuals should have expectation zero, conditional on x , $\mathbb{E}[e_i|X=x] = 0$. (The residuals should also have an over-all sample mean of exactly zero.)
- ② The residuals should show a constant variance, unchanging with x .
- ③ The residuals can't be completely uncorrelated with each other, but the correlation should be extremely weak, and grow negligible as $n \rightarrow \infty$.

Each one of these points leads to a diagnostic, and something to check for in a model.

Plot of the residuals against the predictor X_1

- This allows us to zoom in on instances of poor model fit. Whenever we look at a residual plot, we are searching for systematic patterns of any sort.
- Here's the plot for Advertisement data:



Plot of the residuals against the predictor X II

- Some issues with the constant variance assumption
- The covariance between e_i and x :

$$\begin{aligned}\frac{1}{n} \sum_{i=1}^n (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)) (x_i - \bar{x}) &= \\ \frac{1}{n} \sum_{i=1}^n (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)) x_i &- \frac{1}{n} \sum_{i=1}^n (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)) \bar{x}\end{aligned}$$

- Because of the estimating equation, these are 0 by definition, so

$$\frac{1}{n} \sum_{i=1}^n (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)) (x_i - \bar{x}) = \frac{1}{n} \sum_{i=1}^n e_i (x_i - \bar{x}) = 0$$

i.e. the residuals are uncorrelated with the predictor, but it doesn't mean that the residuals can not exhibit non linear dependence with the predictor. If non-linearities are visible - the relationship between X and Y might be more complex than assumed.

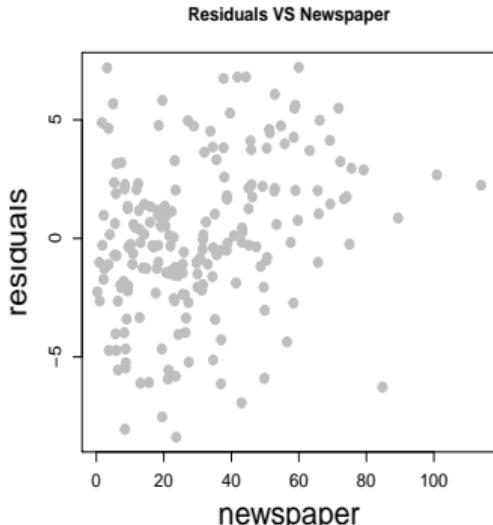
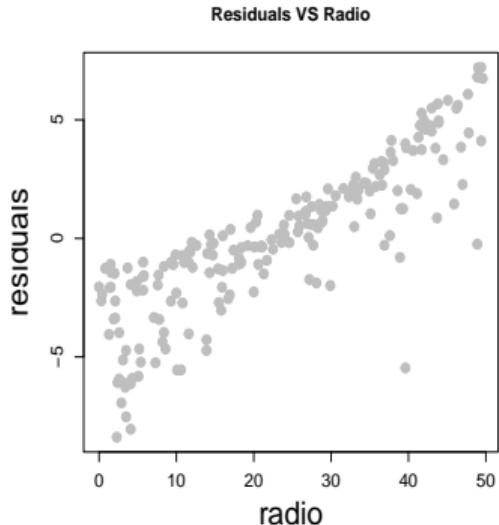
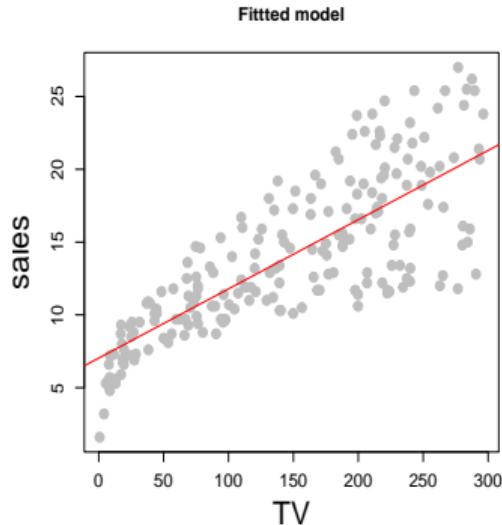
Plotting against another variable... I

- If you have other potential predictor variables, you should be able to plot the residual against them and see a flat line around zero. If not, that is an indication that the other variable might indeed help predict the response, so it should probably be incorporated in your model.
- In particular, if you make such a plot and you see the points in it fall around a straight line, that's a strong indication that you need a **multiple linear regression model**.
- Example: advertising data. We want to predict sales using TV

$$\text{sales} = \beta_0 + \beta_1 \text{TV} + \varepsilon$$

- ... but other variables (radio, newspaper) could be potential predictors

Plotting against another variable... II

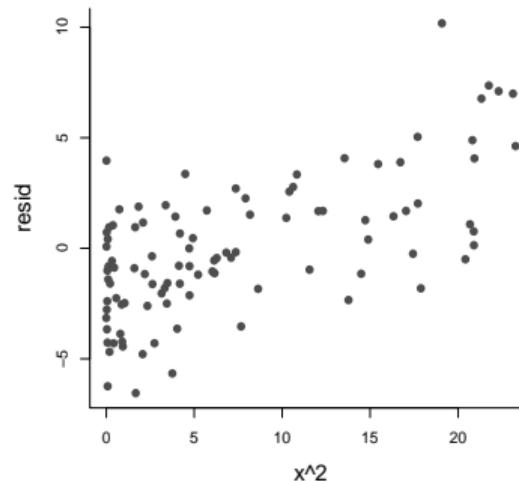
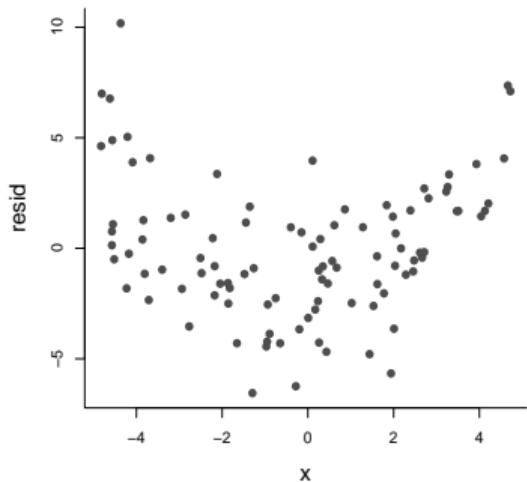


- In this case radio could be considered, i.e.

$$\text{sales} = \beta_0 + \beta_1 \text{TV} + \beta_2 \text{radio} + \varepsilon$$

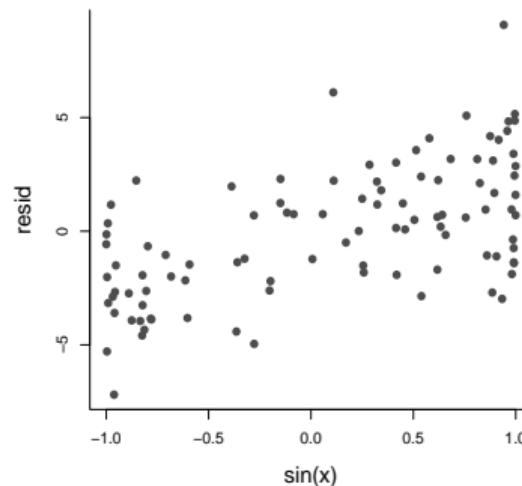
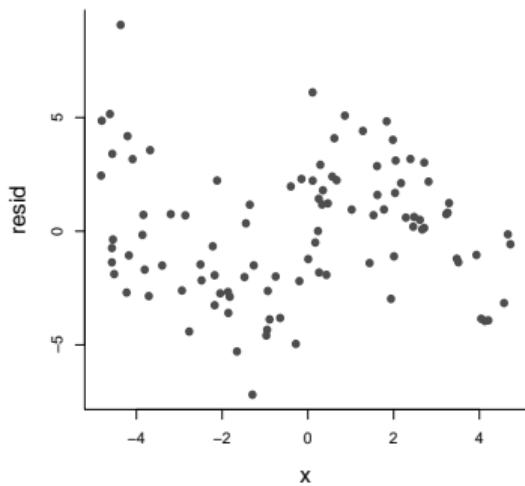
Plotting against another variable... III

- The other variable might be a transformation of the original variable (fictional data)



Plotting against another variable... IV

- The other variable might be a transformation of the original variable (fictional data)



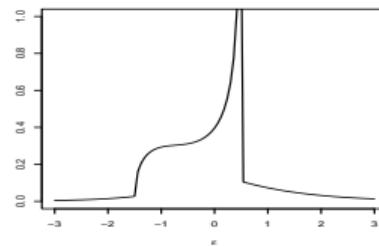
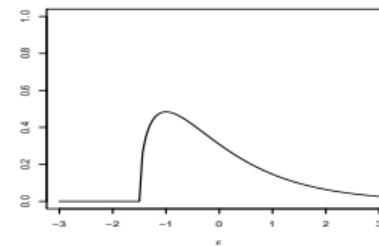
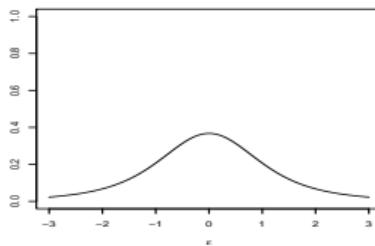
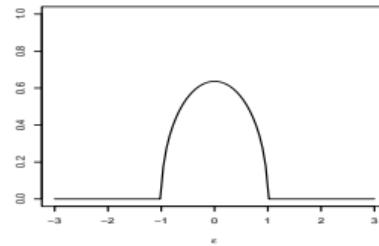
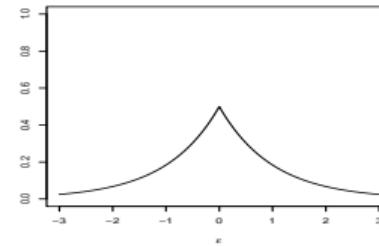
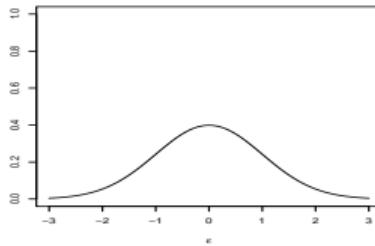
- We discuss more on this later

The Gaussian-noise simple linear regression model⁵

⁵Material is based on the Lecture 4 of Cosma Shalizi's [course](#)

The Gaussian-noise simple linear regression model I

- If we made more detailed assumptions about the distribution of ϵ , we could make more precise inference.
- Some possible noise distributions for the simple linear regression model, i.e. $\mathbb{E}[\epsilon] = 0$



The Gaussian-noise simple linear regression model II

- The most common choice is to assume ϵ follows a Gaussian/Normal distribution.
 - The distribution of X is arbitrary (and perhaps X is even non-random).
 - If $X = x$, then $Y = \beta_0 + \beta_1 x + \epsilon$, for some constants ("coefficients", "parameters") β_0 and β_1 , and some random noise variable ϵ .
 - $\epsilon \sim \mathcal{N}(0, \sigma^2)$, independent of X .
 - ϵ is independent across observations.
- Under this condition then $Y|X = x \sim \mathcal{N}(\beta_0 + \beta_1 x, \sigma^2)$, and Y_i and Y_j are independent given X_i and X_j , so we have

The likelihood

$$L(\beta_0, \beta_1, \sigma^2) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - (\beta_0 + \beta_1 x_i))^2}{2\sigma^2}}$$

and

The Gaussian-noise simple linear regression model III

The log-likelihood

$$\begin{aligned}\ell(\beta_0, \beta_1, \sigma^2) &= \log(L(\beta_0, \beta_1, \sigma^2)) \\ &= -\frac{n}{2} \log 2\pi - \frac{n}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_i))^2\end{aligned}$$

- Maximum likelihood estimates are derived by taking derivatives and set them to zero.

$$\frac{\partial \ell}{\partial \beta_0} = -\frac{1}{2\sigma^2} \sum_{i=1}^n 2(y_i - (\beta_0 + \beta_1 x_i))(-1)$$

$$\frac{\partial \ell}{\partial \beta_1} = -\frac{1}{2\sigma^2} \sum_{i=1}^n 2(y_i - (\beta_0 + \beta_1 x_i))(-x_i)$$

The Gaussian-noise simple linear regression model IV

$$\sum_{i=1}^n y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i) = 0$$

$$\sum_{i=1}^n (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)) x_i = 0$$

- These are, up to a factor of $1/n$, exactly the equations we got from the method of least squares
- **The least squares solution is the maximum likelihood estimate under the Gaussian noise model**
- The derivative with respect to σ^2 :

$$\frac{\partial \ell}{\partial \sigma^2} = -\frac{n}{\sigma^2} + \frac{1}{2\sigma^4} \sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_i))^2$$

The Gaussian-noise simple linear regression model V

- Setting this to 0 we get

$$\widehat{\sigma^2} = \frac{1}{n} \sum_{i=1}^n (y_i - (\widehat{\beta}_0 + \widehat{\beta}_1 x_i))^2 \quad (7)$$

- The solution is the in-sample mean squared error.

An useful result

Suppose that $Y_i, i = 1, \dots, n$ are n independent random variables with $Y_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ and $w_i, i = 0, \dots, n$, are real constants.

Then

$$w_0 + \sum_{i=1}^n w_i Y_i \sim \mathcal{N}\left(w_0 + \sum_{i=1}^n w_i \mu_i, \sum_{i=1}^n w_i^2 \sigma_i^2\right)$$

Sampling distributions I

- Under the Gaussian noise hypothesis we can discuss the sampling distribution of the estimates for $\hat{\beta}_0$ and $\hat{\beta}_1$
- Note that

$$\begin{aligned}\hat{\beta}_1 &= \frac{\sum_{i=1}^n (x_i - \bar{x})(Y_i - \bar{Y})}{\sum_{i=1}^n (x_i - \bar{x})^2} \\ &= \frac{\sum_{i=1}^n (x_i - \bar{x}) Y_i}{\sum_{i=1}^n (x_i - \bar{x})^2} \\ &= \sum_{i=1}^n w_i Y_i\end{aligned}$$

Sampling distributions II

and

$$\begin{aligned}\hat{\beta}_0 &= \bar{Y} - \hat{\beta}_1 \bar{x} \\&= \frac{1}{n} \sum_{i=1}^n Y_i - \bar{x} \sum_{i=1}^n w_i Y_i \\&= \sum_{i=1}^n \left(\frac{1}{n} - \bar{x} w_i \right) Y_i \\&= \sum_{i=1}^n w_i^* Y_i\end{aligned}$$

- Since both $\hat{\beta}_0$ and $\hat{\beta}_1$ are a linear combination of the Y_i and each Y_i is Gaussianly distributed, then both $\hat{\beta}_0$ and $\hat{\beta}_1$ also follow a Gaussian distribution.

Sampling distributions III

- For $\hat{\beta}_0$,

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{x} \sim \mathcal{N} \left(\beta_0, \sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \right) \right).$$

- For $\hat{\beta}_1$ we have,

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x}) Y_i}{\sum_{i=1}^n (x_i - \bar{x})^2} \sim \mathcal{N} \left(\beta_1, \frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \right).$$

Sampling distributions IV

- Then by standardizing these results using the standard deviation of $\hat{\beta}_i$, $SD[\hat{\beta}_i]$, we find that

$$\frac{\hat{\beta}_0 - \beta_0}{SD[\hat{\beta}_0]} \sim \mathcal{N}(0, 1) \quad \text{and} \quad \frac{\hat{\beta}_1 - \beta_1}{SD[\hat{\beta}_1]} \sim \mathcal{N}(0, 1)$$

where

$$SD[\hat{\beta}_0] = \sigma \sqrt{\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}$$

and

$$SD[\hat{\beta}_1] = \sigma \frac{1}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}}$$

Sampling distributions V

- We don't know σ but we can estimate it using s_e where

$$s_e^2 = \frac{\sum_{i=1}^n e_i^2}{n-2} = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n-2}$$

- These two new expressions are called

Standard errors

$$\text{SE}[\hat{\beta}_0] = s_e \sqrt{\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}$$

$$\text{SE}[\hat{\beta}_1] = s_e \frac{1}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}}$$

i.e. the **estimated** standard deviations of the sampling distributions.

Sampling distributions VI

- If we divide by the standard error, instead of the standard deviation, we obtain the following results which allows us to make confidence intervals and perform hypothesis testing:

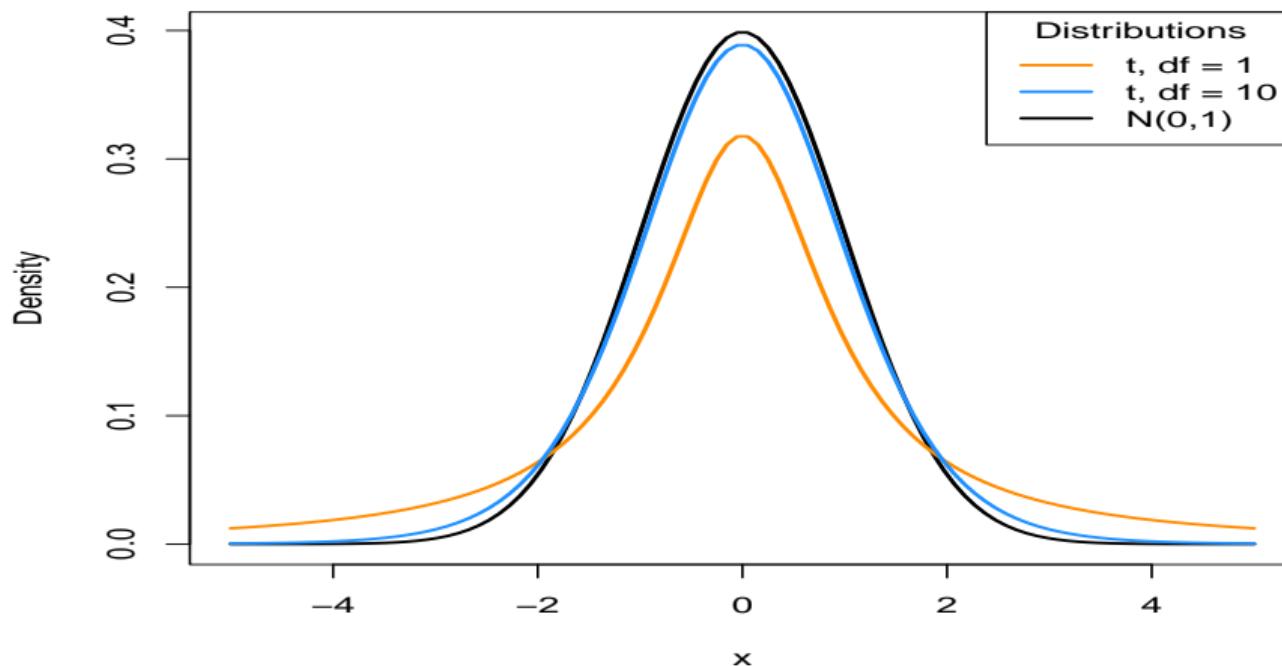
$$\frac{\hat{\beta}_0 - \beta_0}{\text{SE}[\hat{\beta}_0]} \sim t_{n-2} \quad \text{and} \quad \frac{\hat{\beta}_1 - \beta_1}{\text{SE}[\hat{\beta}_1]} \sim t_{n-2}$$

where t_d is the t distribution with d degrees of freedom.

- A t distribution is similar to a Gaussian, but with heavier tails.
- As the degrees of freedom increases, the t distribution becomes more and more like a standard Gaussian.
- We can use this result to make inference for the model parameters and the regression function (more on this later)

Sampling distributions VII

Gaussian vs t distribution



Multiple Linear Regression ⁶

⁶Material in these slides was heavily influenced by David Dalpiaz *Applied Statistics with R!*, the book is under active development.

Multiple predictors I

- It is rarely the case that a response variable will only depend on a single variable.
- For example, flipper length might be related to both body mass and bill length.

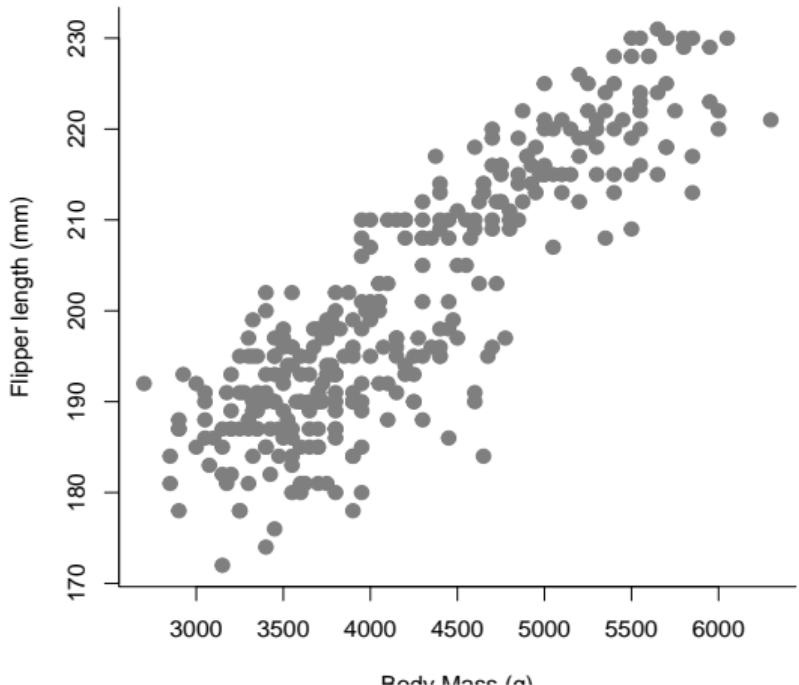
```
autompq = read.table(fl, quote = "\"",  
                      comment.char = "", stringsAsFactors = FALSE)  
  
# give the dataframe headers  
colnames(autompq) <- c("mpg", "cyl", "disp", "hp", "wt",  
                      "acc", "year", "origin", "name")  
  
# remove missing data, which is stored as "?"  
autompq <- subset(autompq, autompq$hp != "?")  
  
# remove the plymouth reliant, as it causes some issues  
autompq <- subset(autompq, autompq$name != "plymouth reliant")  
  
# give the dataset row names, based on the engine, year and name  
rownames(autompq) <- paste(autompq$cyl, "cylinder",
```

Multiple predictors II

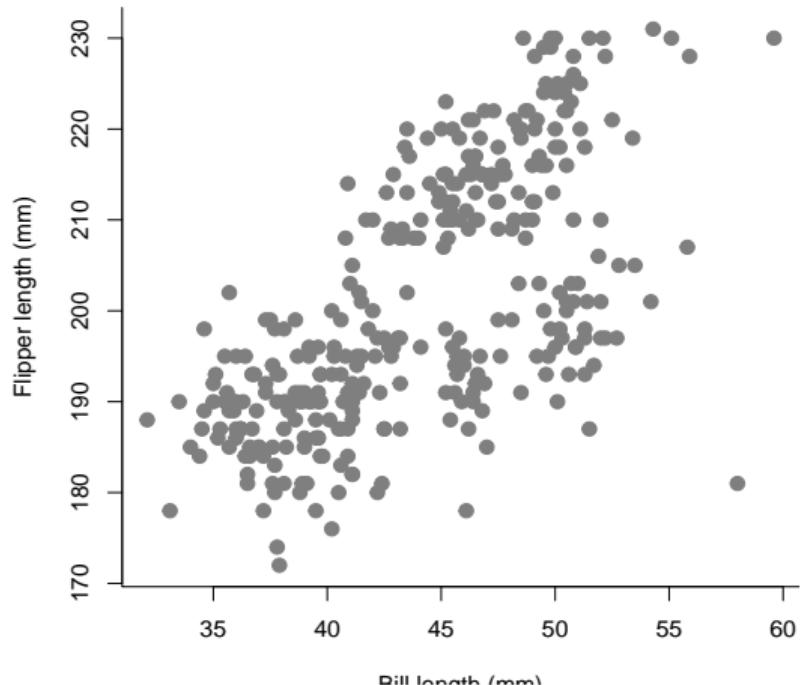
```
autompg$year, autompg$name)  
# remove the variable for name, as well as origin  
autompg <- subset(autompg,  
                    select = c("mpg", "cyl", "disp", "hp", "wt", "acc", "year"))  
# change horsepower from character to numeric  
autompg$hp <- as.numeric(autompg$hp)
```

Multiple predictors III

Flipper length vs Body Mass



Flipper length vs Bill Length



Multiple predictors IV

- We can use a linear model:

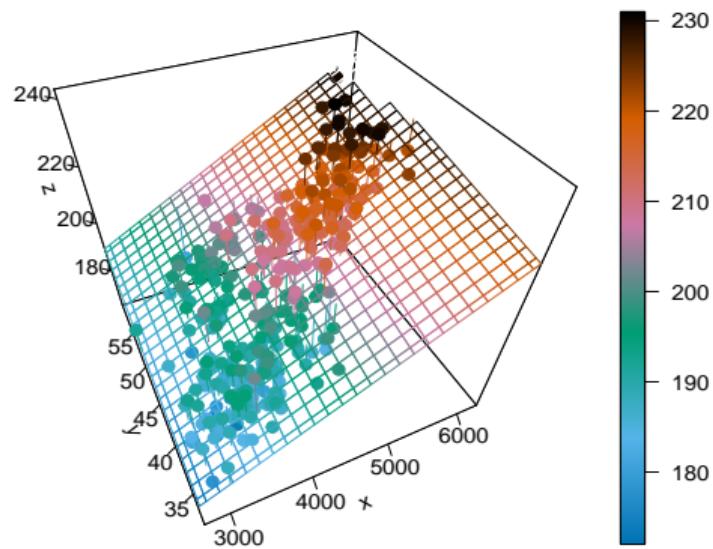


$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i, \quad i = 1, 2, \dots, n$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ (and independent).

- In this notation we will define:
 - x_{i1} as the body mass of the i th penguin.
 - x_{i2} as the bill length of the i th penguin.
 - Extend the least square idea to higher dimension
 - Now \widehat{MSE} depends on $(\beta_0, \beta_1, \beta_2)$
- The data points (x_{i1}, x_{i2}, y_i) now exist in 3-dimensional space, so instead of fitting a line to the data, we will fit a plane

Multiple predictors V



Multiple predictors VI

- How do we find such a plane?
- We would like to minimize

$$f(\beta_0, \beta_1, \beta_2) = \sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}))^2$$

with respect to β_0 , β_1 , and β_2 .

- We take a derivative with respect to each of β_0 , β_1 , and β_2 and set them equal to zero, then solve the resulting system of equations.

$$\begin{cases} \frac{\partial f}{\partial \beta_0} = 0 \\ \frac{\partial f}{\partial \beta_1} = 0 \\ \frac{\partial f}{\partial \beta_2} = 0 \end{cases}$$

Multiple predictors VII

- We obtain the

estimating equations

$$n\beta_0 + \beta_1 \sum_{i=1}^n x_{i1} + \beta_2 \sum_{i=1}^n x_{i2} = \sum_{i=1}^n y_i$$



$$\beta_0 \sum_{i=1}^n x_{i1} + \beta_1 \sum_{i=1}^n x_{i1}^2 + \beta_2 \sum_{i=1}^n x_{i1}x_{i2} = \sum_{i=1}^n x_{i1}y_i$$

$$\beta_0 \sum_{i=1}^n x_{i2} + \beta_1 \sum_{i=1}^n x_{i1}x_{i2} + \beta_2 \sum_{i=1}^n x_{i2}^2 = \sum_{i=1}^n x_{i2}y_i$$

- (Notice that from the definition of β_0 we can see that the plane will pass through $(\bar{x}_1, \bar{x}_2, \bar{y})$).

Multiple predictors VIII

- We let R solve for us:

```
pfit <- lm(flipper_length_mm~body_mass_g+bill_length_mm, data = penguins)
coef(pfit)
```

	(Intercept)	body_mass_g	bill_length_mm
	122.26867040	0.01301733	0.54403527

$$\hat{y} = 122.27 + (0.54x_1 + 0.54x_2)$$

- β_2 represent the change in flipper length (y) for two penguins with identical body mass (x_1) which only differ by one unit of the bill length (x_2) variable.
- The model is **additive**.

Matrix approach to regression I

- For an arbitrary number $p - 1$ of predictor variables we can consider the (additive and linear) model

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_{p-1} x_{i(p-1)} + \epsilon_i, \quad i = 1, 2, \dots, n$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$.

- there are $p - 1$ predictor variables, x_1, x_2, \dots, x_{p-1} .
- There are a total of p β -parameters and a single parameter σ^2 for the variance of the errors

Matrix approach to regression II

If we were to stack together the n linear equations that represent each Y_i into a column vector, we get the following.

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1(p-1)} \\ 1 & x_{21} & x_{22} & \cdots & x_{2(p-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{n(p-1)} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{p-1} \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

$$Y = X\beta + \epsilon$$

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}, \quad X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1(p-1)} \\ 1 & x_{21} & x_{22} & \cdots & x_{2(p-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{n(p-1)} \end{bmatrix}$$

Matrix approach to regression III

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{p-1} \end{bmatrix}, \quad \epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

Matrix approach to regression IV

- With the observed data:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

we can estimate β by minimizing

$$f(\beta_0, \beta_1, \beta_2, \dots, \beta_{p-1}) = \sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_{p-1} x_{i(p-1)}))^2,$$

which would require taking p derivatives, which result in the following

Matrix approach to regression V

estimating equations

$$\begin{bmatrix} \sum_{i=1}^n x_{i1} & \sum_{i=1}^n x_{i1}^2 & \cdots & \sum_{i=1}^n x_{i1}x_{i(p-1)} \\ \vdots & \vdots & & \vdots \\ \sum_{i=1}^n x_{i(p-1)} & \sum_{i=1}^n x_{i(p-1)}x_{i1} & \cdots & \sum_{i=1}^n x_{i(p-1)}^2 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_{i1}y_i \\ \vdots \\ \sum_{i=1}^n x_{i(p-1)}y_i \end{bmatrix}$$

- The estimating equations can be written in matrix notation,

$$X^\top X \beta = X^\top y.$$

- We can then solve this expression by multiplying both sides by the inverse of $X^\top X$, which exists, provided the columns of X are linearly independent.

$$\hat{\beta} = (X^\top X)^{-1} X^\top y$$

Matrix approach to regression VI

- In R, we create an X matrix. Note that the first column is all 1s, and the remaining columns contain the data.

```
n <- nrow(penguins)
p <- length(coef(pfit))
X <- cbind(rep(1, n), penguins$body_mass_g, penguins$bill_length_mm)
y <- penguins$flipper_length_mm
beta.hat <- solve(t(X) %*% X) %*% t(X) %*% y; t(beta.hat)

[1]      [,1]      [,2]      [,3]
[1,] 122.2687 0.01301733 0.5440353

# same value
coef(pfit)

(Intercept)    body_mass_g bill_length_mm
122.26867040     0.01301733     0.54403527
```

Matrix approach to regression VII

- Actually R builds the X matrix from the formula (and then does some more optimised linear algebra to obtain the estimates):

```
head(model.matrix(pfit))
```

	(Intercept)	body_mass_g	bill_length_mm
1	1	3750	39.1
2	1	3800	39.5
3	1	3250	40.3
5	1	3450	36.7
6	1	3650	39.3
7	1	3625	38.9

```
head(X)
```

	[,1]	[,2]	[,3]
[1,]	1	3750	39.1
[2,]	1	3800	39.5
[3,]	1	3250	40.3
[4,]	1	3450	36.7
[5,]	1	3650	39.3
[6,]	1	3625	38.9

- There is also a 'fit\$model' object - what do you think it is?

Matrix approach to regression VIII

- In our new notation, the predicted values can be written

$$\hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix} = X\hat{\beta} (= X(X^\top X)^{-1}X^\top y),$$

while the vector for the residual values is

$$e = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} - \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix}.$$



Matrix approach to regression IX

- The estimate for σ^2 :

$$s_e^2 = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n - p} = \frac{\mathbf{e}^\top \mathbf{e}}{n - p}$$

Recall, we like this estimate because it is unbiased, that is,

$$\mathbb{E}[S_e^2] = \sigma^2$$

- Note that the change from the SLR estimate to now is in the denominator. Specifically we now divide by $n - p$ instead of $n - 2$.
- (Or actually, we should note that in the case of SLR, there are two β parameters and thus $p = 2$.)

Matrix approach to regression X

- Also note that if we fit the model $Y_i = \beta + \epsilon_i$ that $\hat{y} = \bar{y}$ and $p = 1$ and s_e^2 would become

$$s_e^2 = \frac{\sum_{i=1}^n (y_i - \bar{y})^2}{n - 1}$$

which is likely to be the very first definition of sample standard deviation you saw in the previous statistics courses.

- In this case we only fit one parameter β , so we lose one degree of freedom and divide by $n - 1$. In general, we are estimating p parameters, the β parameters, so we lose p degrees of freedom.
- Typically we are interested in s_e (which R calls the residual standard error)

$$s_e = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n - p}}.$$

Matrix approach to regression XI

- In R, we can directly access s_e for a fitted model, as before:

```
summary(pfit)$sigma
```

```
[1] 6.419285
```

- We can now verify that our math above is indeed calculating the same quantities.

```
y_hat = X %*% solve(t(X) %*% X) %*% t(X) %*% y
```

```
e      = y - y_hat
```

```
sqrt(t(e) %*% e / (n - p))
```

```
[,1]
```

```
[1,] 6.419285
```

```
sqrt(sum((y - y_hat) ^ 2) / (n - p))
```

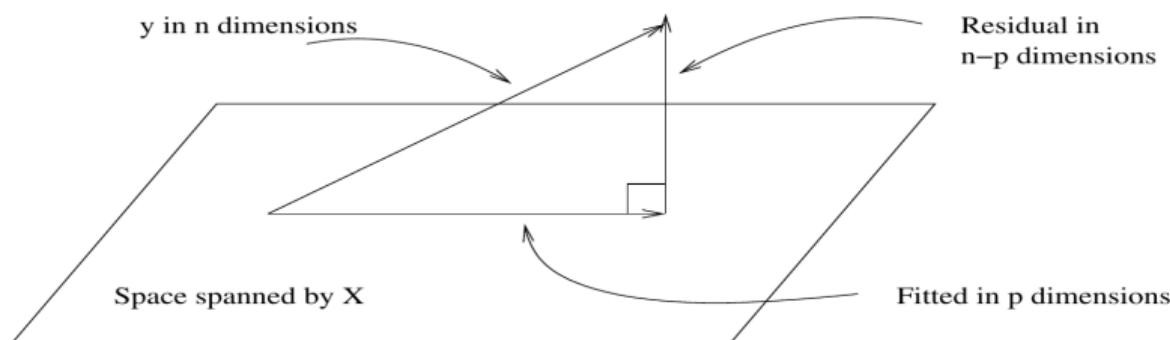
```
[1] 6.419285
```

Geometrical interpretation I

- For n observations we estimate β by minimizing,

$$\sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_{p-1} x_{i(p-1)}))^2 = (y - X\beta)^\top (y - X\beta)$$

- Geometrically speaking, $y \in \mathbb{R}^n$ and the p columns of X are also vectors of \mathbb{R}^n



Geometrical interpretation II

- The problem is to find β such that vector $X\beta$ (i.e. a linear combination of the column vectors in X) is close to y .
- The solution

$$\hat{y} = X\hat{\beta} = X(X^\top X)^{-1}X^\top y = Hy$$

where

$$H = X(X^\top X)^{-1}X^\top$$

is the **orthogonal projection matrix**.

- The conceptual purpose of the model is to represent, as accurately as possible, something complex - y which is n -dimensional - in terms of something much simpler - the model which is p -dimensional.

Geometrical interpretation III

- If our model is successful, the structure in the data should be captured in those p dimensions, leaving just random variation in the residuals which lie in an $n - p$ dimensional space.

$$\begin{aligned}\text{Data} &= \text{Systematic structure} + \text{Random variation} \\ n \text{ dimensions} &= p \text{ dimensions} + n - p \text{ dimensions}\end{aligned}$$

Multivariate Gaussian distribution I

- We will consider a random vector X of n random variables i.e. $X = (X_1, X_2, \dots, X_n)^\top$
- A random vector is said to be n -variate Gaussian distributed if every linear combination of its n components has a univariate Gaussian distribution.
- The multivariate Gaussian distribution of a n -dimensional random vector can be written in the following notation: $X \sim \mathcal{N}_n(\mu, \Sigma)$ with

$$\mu = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_n \end{bmatrix} \quad \Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn} \end{bmatrix}.$$

where $\mu_j = \mathbb{E}[X_j]$, $\sigma_{ij} = \text{Cov}[X_i, X_j]$

Multivariate Gaussian distribution II

- The multivariate normal distribution is said to be "non-degenerate" when the symmetric covariance matrix Σ is positive definite.
- In this case the distribution has density

$$f_X(x) = (2\pi)^{n/2} |\Sigma|^{-1/2} \exp \left\{ -\frac{1}{2} (x - \mu)^\top \Sigma^{-1} (x - \mu) \right\}$$

- If $Y = AX + b$ is an **affine transformation** of $X \sim \mathcal{N}(\mu, \Sigma)$ where b is an $m \times 1$ vector of constants and A is a constant $m \times n$ matrix, then Y has a multivariate Gaussian distribution with expected value $A\mu + b$ and variance-covariance matrix $A\Sigma A^\top$ i.e., $Y \sim \mathcal{N}_m(A\mu + b, A\Sigma A^\top)$
- In particular, any subset of the Y_i has a marginal distribution that is also multivariate normal.

Sampling distribution for $\hat{\beta}$ |

- We want to obtain the distribution of the $\hat{\beta}$ vector

$$\hat{\beta} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \\ \vdots \\ \hat{\beta}_{p-1} \end{bmatrix}$$

- We consider $\hat{\beta}$ to be a random vector, thus we use Y instead of the data vector y :

$$\hat{\beta} = (X^\top X)^{-1} X^\top Y$$

- As for SLR we assume that Y is (conditionally) normal

Sampling distribution for $\hat{\beta}$ II

- We can prove that

$$\hat{\beta} \sim \mathcal{N}_p \left(\beta, \sigma^2 (X^\top X)^{-1} \right).$$

- We then have $E[\hat{\beta}] = \beta$ and for any $\hat{\beta}_j$ we have $E[\hat{\beta}_j] = \beta_j$.
- We also have

$$\text{Var}[\hat{\beta}] = \sigma^2 (X^\top X)^{-1}$$

and for any $\hat{\beta}_j$ we have

$$\text{Var}[\hat{\beta}_j] = \sigma^2 C_{jj}$$

where

$$C = (X^\top X)^{-1}$$

Sampling distribution for $\hat{\beta}$ III

and the elements of C are denoted

$$C = \begin{bmatrix} C_{00} & C_{01} & C_{02} & \cdots & C_{0(p-1)} \\ C_{10} & C_{11} & C_{12} & \cdots & C_{1(p-1)} \\ C_{20} & C_{21} & C_{22} & \cdots & C_{2(p-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ C_{(p-1)0} & C_{(p-1)1} & C_{(p-1)2} & \cdots & C_{(p-1)(p-1)} \end{bmatrix}.$$

- The standard error for the $\hat{\beta}$ vector is given by

$$\text{SE}[\hat{\beta}] = s_e \sqrt{\text{diag}(X^\top X)^{-1}}$$

and for a particular $\hat{\beta}_j$

$$\text{SE}[\hat{\beta}_j] = s_e \sqrt{C_{jj}}.$$

Sampling distribution for $\hat{\beta}$ IV

- In R

```
C <- solve(t(X) %*% X)
s<-sqrt(sum((y - y_hat) ^ 2) / (n - p))
std.err<-s*sqrt(diag(C))
std.err
[1] 2.8636189207 0.0005416258 0.0797498764
t(summary(pfit)$coef[,2])
  (Intercept) body_mass_g bill_length_mm
[1,] 2.863619 0.0005416258      0.07974988
sqrt(diag(vcov(pfit)))
  (Intercept) body_mass_g bill_length_mm
2.8636189207 0.0005416258 0.0797498764
```

Sampling distribution for $\hat{\beta}$ V

- Each of the $\hat{\beta}_j$ follows a Gaussian distribution

$$\hat{\beta}_j \sim \mathcal{N}(\beta_j, \sigma^2 C_{jj})$$

and

$$\frac{\hat{\beta}_j - \beta_j}{s_e \sqrt{C_{jj}}} \sim t$$

Single Parameter Tests I

- The first test we will see is a test for a single β_j .

$$H_0 : \beta_j = 0 \quad \text{vs} \quad H_A : \beta_j \neq 0$$

- The test statistic (TS) takes the form

$$\text{TS} = \frac{\text{EST} - \text{HYP}}{\text{SE}},$$

i.e.

$$t = \frac{\hat{\beta}_j - \beta_j}{\text{SE}[\hat{\beta}_j]} = \frac{\hat{\beta}_j - 0}{s_e \sqrt{C_{jj}}},$$

which, under the null hypothesis, follows a t distribution with $n - p$ degrees of freedom.

Single Parameter Tests II

- Example: for penguins

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i, \quad i = 1, 2, \dots, n$$

where $\epsilon_i \sim N(0, \sigma^2)$.

- x_{i1} as the body mass of the i th penguin.
- x_{i2} as the model bill length of the i th penguin.
- Then the test

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

can be found in the summary function, in particular:

Single Parameter Tests III

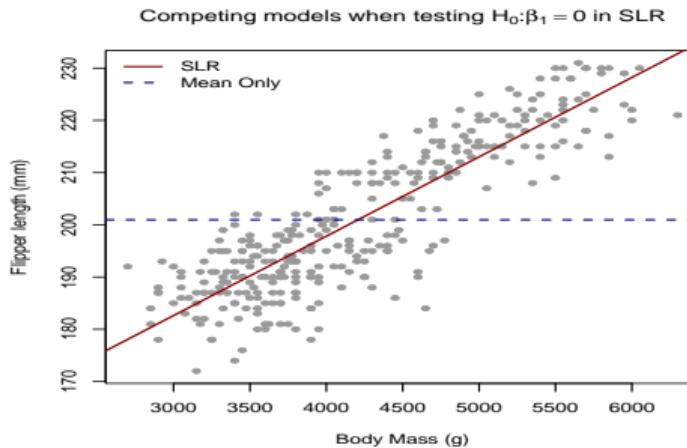
```
summary(pfit)$coef
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	122.26867040	2.8636189207	42.697256	1.900760e-136
body_mass_g	0.01301733	0.0005416258	24.033815	1.737931e-74
bill_length_mm	0.54403527	0.0797498764	6.821769	4.306811e-11

- The p-value (last column) given here is specifically for a two-sided test, where the value of the parameter under H_0 is 0.
- Also note in this case, under the hypothesis that $\beta_1 = 0$, the null and alternative essentially specify two different models:
 - $H_0: Y = \beta_0 + \beta_2 x_2 + \epsilon$
 - $H_1: Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$
- We are not simply testing whether or not there is a relationship between body mass and flipper length. We are testing if there is a relationship between body mass and flipper length, given that a term for bill length is in the model.

Single Parameter Tests IV

- Note: when we are in the simple regression case ($p=2$) testing for $\beta_1 = 0$ corresponds to testing for the need to carrying out a regression in general: if we can not reject $H_0 : \beta_1 = 0$ it means we are better off assuming the variation in Y is better explained by a unique value: the mean. A slightly different interpretation



Confidence Intervals - parameters I

- We have already shown that

$$\frac{\hat{\beta}_j - \beta_j}{s_e \sqrt{C_{jj}}} \sim t_{n-p}.$$

- From this we can construct confidence intervals for each of the $\hat{\beta}_j$:

$$\hat{\beta}_j \pm t_{\alpha/2, n-p} \cdot s_e \sqrt{C_{jj}}$$

- In R

```
confint(pfit, level = 0.99)
```

	0.5 %	99.5 %
(Intercept)	114.84957959	129.68776122
body_mass_g	0.01161408	0.01442058
bill_length_mm	0.33741855	0.75065200

Confidence Intervals - multivariate parameters I

- We use the diagonal elements of $\hat{\beta}$ to construct the confidence intervals for β_j
- On the hand we can estimate the covariance between the estimates:

`vcov(pfit)`

	(Intercept)	body_mass_g	bill_length_mm
(Intercept)	8.200313323	-1.140710e-04	-0.1726797535
body_mass_g	-0.000114071	2.933585e-07	-0.0000254611
bill_length_mm	-0.172679754	-2.546110e-05	0.0063600428

Confidence Intervals - multivariate parameters II

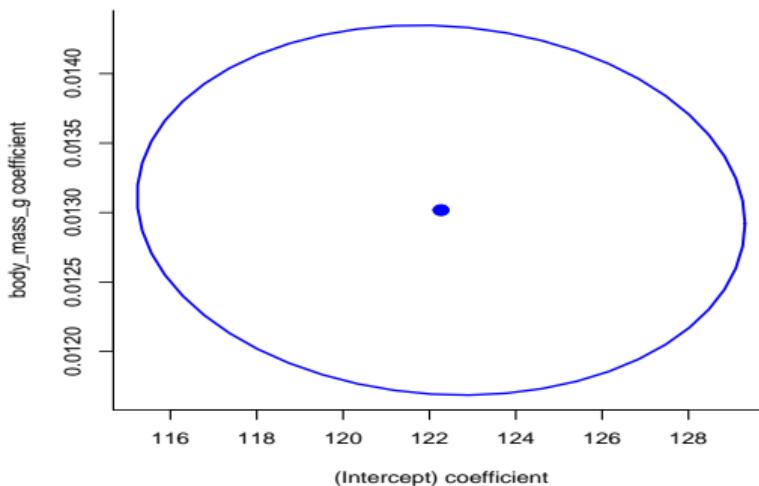
- Might be easier to look at the correlation:

```
cov2cor(vcov(pfit))
```

	(Intercept)	body_mass_g	bill_length_mm
(Intercept)	1.0000000	-0.07354629	-0.7561295
body_mass_g	-0.07354629	1.0000000	-0.5894511
bill_length_mm	-0.75612949	-0.58945111	1.0000000

- If we are interested in the confidence interval of **several** parameters we should take into account the correlation between estimates
- For a pair of parameters, say (β_0, β_1) we can construct bivariate confidence intervals which are constructed based on the bivariate normal distribution

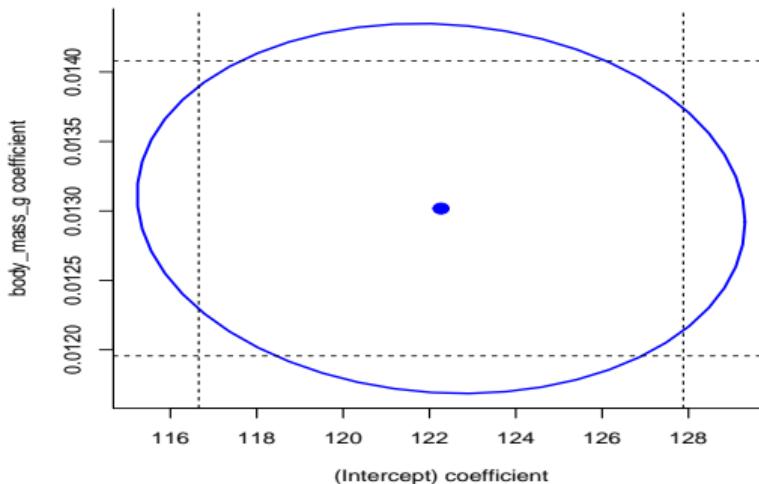
Confidence Intervals - multivariate parameters III



- the ellipse displays a 95% joint confidence region for the regression coefficients β_0 and β_1

Confidence Intervals - multivariate parameters IV

- the orientation of the ellipse reflects the correlation between the estimates (when estimates are not correlated the ellipse will look like a circle).



Confidence Intervals - multivariate parameters V

- individual confidence intervals are often overconfident
- with bi-variate intervals we can notice pairs of (β_0, β_1) whose values in combination are not within the confidence intervals

Confidence Intervals - the regression function I

- We can create confidence intervals for the regression function $E[Y | X = x]$.
- We define the vector x_0 to be

$$x_0 = \begin{bmatrix} 1 \\ x_{01} \\ x_{02} \\ \vdots \\ x_{0(p-1)} \end{bmatrix}.$$

- The estimate of $E[Y | X = x_0]$ is given by

$$\begin{aligned}\hat{y}(x_0) &= x_0^\top \hat{\beta} = \\ &= \hat{\beta}_0 + \hat{\beta}_1 x_{01} + \hat{\beta}_2 x_{02} + \cdots + \hat{\beta}_{p-1} x_{0(p-1)}.\end{aligned}$$

Confidence Intervals - the regression function II

- This is an unbiased estimate:



$$\begin{aligned}\mathbb{E}[\hat{y}(x_0)] &= \mathbb{E}[x_0^\top \hat{\beta}] = x_0^\top \beta = \\ &= \beta_0 + \beta_1 x_{01} + \beta_2 x_{02} + \cdots + \beta_{p-1} x_{0(p-1)}\end{aligned}$$

- To make an interval estimate, we will also need its standard error:

$$SE[\hat{y}(x_0)] = SE[x_0^\top \hat{\beta}] = s_e \sqrt{x_0^\top (X^\top X)^{-1} x_0}$$

Putting it all together, we obtain (notice we use the t distribution because we are using s_e rather than σ):

the confidence interval

$$\hat{y}(x_0) \pm t_{\alpha/2, n-p} \cdot s_e \sqrt{x_0^\top (X^\top X)^{-1} x_0}$$

- In R we use predict.

Confidence Intervals - the regression function III

- We create a data frame for two additional penguins: both with a body mass of 5000 grams and a bill length of 35 and 45 mm.

```
new.peng <- data.frame(body_mass_g = c(5000, 5000),
                        bill_length_mm = c(45, 35)); new.peng
```

	body_mass_g	bill_length_mm
1	5000	45
2	5000	35

- We use the predict function with interval = "confidence" to obtain intervals for the mean flipper length of these two penguins.

```
(cint <- predict(pfit, newdata = new.peng,
                  interval = "confidence", level = 0.99))
```

	fit	lwr	upr
1	211.8369	210.4808	213.1930
2	206.3966	203.5755	209.2176

Confidence Intervals - the regression function IV

- R reports the estimate $\hat{y}(x_0)$ (fit) for each, as well as the lower (lwr) and upper (upr) bounds for the interval at a desired level (99%).
- One of these estimates is good while one is suspect (i.e. very wide)

`cint[,3] - cint[,2]`

1 2

2.712218 5.642139

- The new observations are within the range of observed values

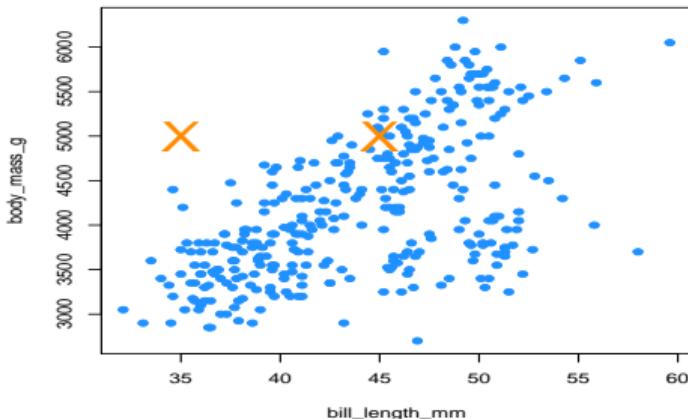
`range(penguins$body_mass_g)`

[1] 2700 6300

`range(penguins$bill_length_mm)`

[1] 32.1 59.6

Confidence Intervals - the regression function V



- However, we have to consider mass and bill length **together**. Based on the above plot, one of the new penguins is within the "blob" of observed values, while the other, is noticeably outside of the observed values. This is a hidden extrapolation which you should be aware of when using multiple regression.

Confidence Intervals - the regression function VI

- A quick verification of some of the mathematics in R.

```
x0 <- c(1,5000,45)
```

```
x0 %*% beta.hat
```

```
[,1]
```

```
[1,] 211.8369
```

```
x0[1]*beta.hat[1] + x0[2]*beta.hat[2]+ x0[3]*beta.hat[3]
```

```
[1] 211.8369
```

Confidence Intervals - the regression function VII

$$x_0 = \begin{bmatrix} 1 \\ 5000 \\ 45 \end{bmatrix}, \quad \hat{\beta} = \begin{bmatrix} 122.27 \\ 0.013017 \\ 0.54404 \end{bmatrix}$$

$$\hat{y}(x_0) = x_0^\top \hat{\beta} = \begin{bmatrix} 1 & 5000 & 45 \end{bmatrix} \begin{bmatrix} 122.27 \\ 0.013017 \\ 0.54404 \end{bmatrix} = 211.8369$$

Confidence Intervals - the regression function VIII

- Note that, using a particular value for x_0 , we can essentially extract certain $\hat{\beta}_j$ values.

```
x0 <- c(0, 0, 1)
```

```
x0 %*% beta.hat
```

```
[,1]
```

```
[1,] 0.5440353
```

- With this in mind, confidence intervals for the individual $\hat{\beta}_j$ are actually a special case of a confidence interval for mean response.
- Notice that the confidence interval gives information on the $E[Y|X = x_0]$: this is the typical value we can expect at some predictor value x_0
- What about the actual values of Y ? → We have $Y|X = x_0 = X_0\beta + \varepsilon$: prediction intervals

Prediction Intervals I

- We use $\hat{y}(x_0)$ to predict Y_0 , a new observation of Y at the predictor vector x_0 .

$$\begin{aligned}\hat{y}(x_0) &= x_0^\top \hat{\beta} \\ &= \hat{\beta}_0 + \hat{\beta}_1 x_{01} + \hat{\beta}_2 x_{02} + \cdots + \hat{\beta}_{p-1} x_{0(p-1)}\end{aligned}$$

$$\begin{aligned}E[\hat{y}(x_0)] &= x_0^\top \beta \\ &= \beta_0 + \beta_1 x_{01} + \beta_2 x_{02} + \cdots + \beta_{p-1} x_{0(p-1)}\end{aligned}$$

- We need to account for the additional variability of an observation about its mean:

$$SE[\hat{y}(x_0) + \epsilon] = s_e \sqrt{1 + x_0^\top (X^\top X)^{-1} x_0}$$

Prediction Intervals II

- Then we arrive at

the prediction interval

$$\hat{y}(x_0) \pm t_{\alpha/2, n-p} \cdot s_e \sqrt{1 + x_0^\top (X^\top X)^{-1} x_0}$$

- in R

```
predict(pfit, newdata = new.peng,  
        interval = "prediction", level = 0.99)
```

	fit	lwr	upr
1	211.8369	195.1506	228.5233
2	206.3966	189.5279	223.2653

Confidence and Prediction Intervals in SLR I

In the SLR case the design matrix X is

$$\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \quad \text{and} \quad (X^\top X) = \begin{bmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{bmatrix}$$

From this one can derive:

$$(X^\top X)^{-1} = \frac{1}{n^2 s_x^2} \begin{bmatrix} \sum x_i^2 & -\sum x_i & -\sum x_i & n \end{bmatrix}$$

And obtain the same value for the $\text{SD}(\beta_0)$ and $\text{SD}(\beta_1)$ seen for SLR.

Confidence and Prediction Intervals in SLR II

Furthermore, for the SLR we can derive that the variability of $\hat{m}(x_0)$ as

$$\text{SD}^2(\hat{m}(x_0)) = s_e^2 * x_0^\top \left(X^\top X \right)^{-1} x_0 = s_e^2 \left(\frac{1}{n} + \frac{(x - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \right)$$

From this we can derive

the confidence interval for $m(x)$ in SLR

$$\hat{m}(x_0) \pm t_{\alpha/2, n-2} \times s_e \sqrt{\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}$$

and

Confidence and Prediction Intervals in SLR III

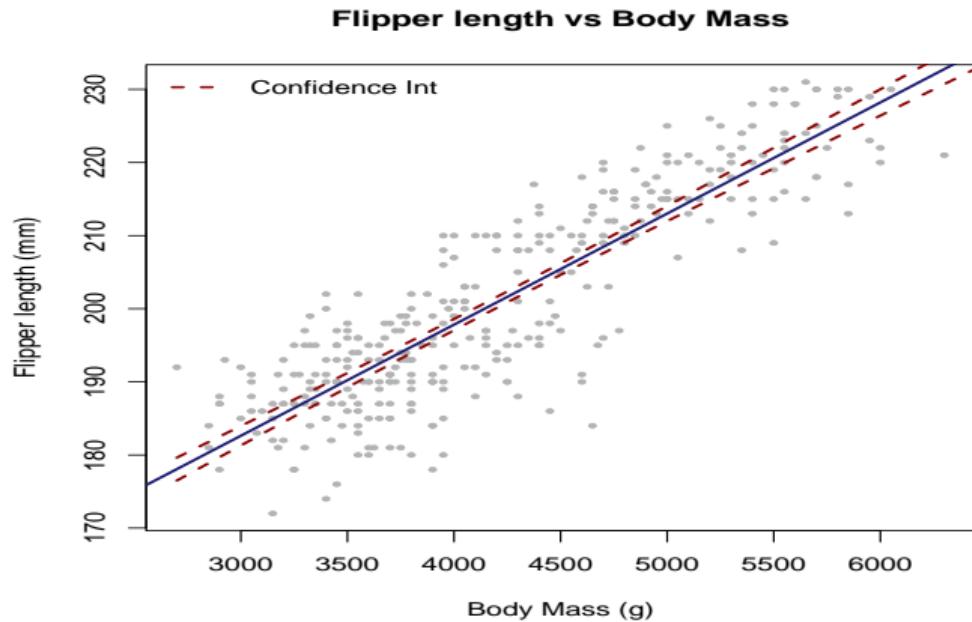
the prediction interval for $m(x)$ in SLR

$$\hat{m}(x_0) \pm t_{\alpha/2, n-2} \times s_e \sqrt{1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}.$$

Notice how these depend on s_e , $(x_0 - \bar{x})$ and n .

Take the SLR case of predicting flipper length from body mass [only](#).

Confidence and Prediction Intervals in SLR IV



Confidence and Prediction Intervals in SLR V

```
plot(flipper_length_mm ~ body_mass_g, data = penguins,
      ylab = "Flipper length (mm)",
      xlab = "Body Mass (g)",
      main = "Flipper length vs Body Mass",
      pch = 20, cex = 0.8, col = "grey70")
abline(slrfit, col = "midnightblue", lwd = 2)
nd <- data.frame(body_mass_g = seq(2700, 6500, by = 200))
pint <- predict(slrfit, newdata = nd, interval = "pred")
lines(nd$body_mass_g, cint[, "lwr"], lty = 2, col = "darkred", lwd = 2)
lines(nd$body_mass_g, cint[, "upr"], lty = 2, col = "darkred", lwd = 2)
lines(nd$body_mass_g, pint[, "lwr"], lty = 4, col = "deeppink3", lwd = 2)
lines(nd$body_mass_g, pint[, "upr"], lty = 4, col = "deeppink3", lwd = 2)
legend("topleft", bty = "n", col = c("darkred", "deeppink3"),
      lty = c(2, 4), legend = c("Confidence Int", "Prediction Int"), lwd = 2)
```

Model selection ⁷

⁷Material in these slides was heavily influenced by David Dalpiaz *Applied Statistics with R!*, the book is under active development.

Some important questions

- Is at least one of the predictors X_1, X_2, \dots, X_{p-1} useful in predicting the response?
- Do all the predictors help to explain Y , or is only a subset of the predictors useful?
- How well does the model fit the data?
- We can still use the decomposition of variation that we had seen in the simple linear regression:

$$\sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \sum_{i=1}^n (\hat{y}_i - \bar{y})^2$$

That is:

$$SS_{tot} = SS_{res} + SS_{reg}.$$



Significance of regression I

- This means that, we can still calculate R^2 in the same manner as before, which R continues to do automatically.

```
summary(mpg.fit)$r.squared
```

```
[1] 0.8082355
```

- The interpretation changes slightly compared to the simple linear regression. In the multiple linear regression case, we say that 80.82% for the observed variation in miles per gallon is explained by the linear relationship with the two predictor variables, weight and year.

Significance of regression II

- But R^2 is not something that can give a clear indication that the regression captures a considerable proportion of the variability - there is no obvious cut-off to know when the value is "large"
- We wish to have a way to declare that the model is useful and have a test to say whether the model is significantly different from taking the much simpler model in which we include no predictor variable
- In multiple regression, the **significance** of regression is tested using 

$$H_0 : \beta_1 = \beta_2 = \cdots = \beta_{p-1} = 0.$$

- The "model under the null hypothesis" is

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

i.e. none of the predictors have a significant linear relationship with the response.

Significance of regression III

- We will denote the predicted values of this model as $\hat{y}_{0,i}$ and we have

$$\hat{y}_{0,i} = \bar{y}.$$

- The alternative hypothesis here is that

$$H_A : \text{At least one of } \beta_j \neq 0, j = 1, 2, \dots, (p-1)$$

- The "model under the alternative hypothesis" is

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_{(p-1)} x_{i(p-1)} + \epsilon_i$$

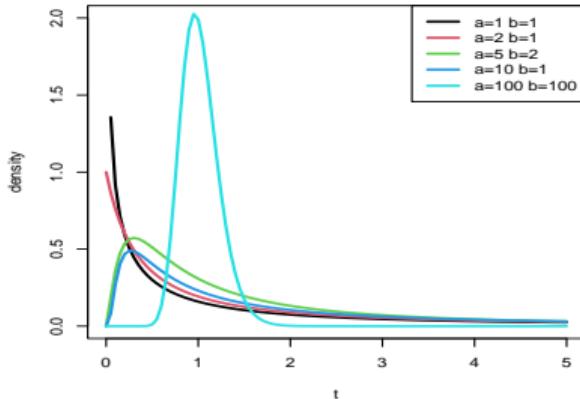
i.e. there is some linear relationship between y and the predictors, x_1, x_2, \dots, x_{p-1} .

- We will denote the predicted values of this model as $\hat{y}_{A,i}$.

F distribution



- $f(t) = \frac{\sqrt{\frac{(at)^a b^b}{(at+b)^{a+b}}}}{t B\left(\frac{a}{2}, \frac{b}{2}\right)}$
- a and b **degrees of freedom**
- $B(x, y) = \int_0^1 t^{x-1} (1-t)^{y-1} dt$ is the **beta function**
- In R df, pf, qf and rf (*f)



The F-distribution arises as the ratio of two independent scaled chi-square distributions. So if $X_1 \sim \chi_{d1}^2$ and $X_2 \sim \chi_{d2}^2$:

$$\frac{X_1/d_1}{X_2/d_2} \sim F_{d1, d2}$$

The domain is the positive reals.

ANOVA table I

- To develop the test for the significance of the regression, we will arrange the variance decomposition into an **ANalysis Of VAriance (ANOVA)** table:

Source	Sum of Squares	Degrees of		
		Freedom	Mean Square	F
Reg.	$\sum_{i=1}^n (\hat{y}_{A,i} - \bar{y})^2$	$p - 1$	$SS_{reg}/(p - 1)$	MS_{reg}/MS_{res}
Error	$\sum_{i=1}^n (y_i - \hat{y}_{A,i})^2$	$n - p$	$SS_{res}/(n - p)$	
Total	$\sum_{i=1}^n (y_i - \bar{y})^2$	$n - 1$		

- We calculate the F-statistic

$$F = \frac{\sum_{i=1}^n (\hat{Y}_{A,i} - \bar{Y})^2 / (p - 1)}{\sum_{i=1}^n (Y_i - \hat{Y}_{A,i})^2 / (n - p)},$$

under H_0 the distribution of the statistics is a **F-distribution**, also known as **Snedecor's F distribution** or the **Fisher-Snedecor distribution** with degrees of freedom $p - 1$ and $n - p$ (Notation $F \sim F_{p-1, n-p}$)

ANOVA table II

- A large value of the statistic corresponds to a large portion of the variance being explained by the regression. We reject H_0 for large values of F and the p-value is calculated as

$$\Pr(F > F_{obs})$$

- In R, we first explicitly specify the two models in R and save the results in different variables.
- We then use `anova` to compare the two models
 - Model under H_0 : $Y_i = \beta_0 + \epsilon_i$
 - Model under H_A : $Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i$

ANOVA table III

- In R

```
null_mpg.fit = lm(mpg ~ 1, data = autompg)
full_mpg.fit = lm(mpg ~ wt + year, data = autompg)
anova(null_mpg.fit, full_mpg.fit)
```

Analysis of Variance Table

Model 1: mpg ~ 1

Model 2: mpg ~ wt + year

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	389	23761.7				
2	387	4556.6	2	19205	815.55 < 2.2e-16 ***	

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

- We see that the value of the F statistic is 815.55, and the p-value is extremely low, so we reject the null hypothesis at any reasonable α and say that the regression is significant. At least one of wt or year has a useful linear relationship with mpg.

ANOVA table IV

- Now consider the summary

```
mpg.fit <- lm(mpg ~ wt + year, data = autompg)
summary(mpg.fit)
```

Call:

```
lm(formula = mpg ~ wt + year, data = autompg)
```

Residuals:

Min	1Q	Median	3Q	Max
-8.852	-2.292	-0.100	2.039	14.325

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-14.6376419	4.0233914	-3.638	0.000312 ***
wt	-0.0066349	0.0002149	-30.881	< 2e-16 ***
year	0.7614020	0.0497266	15.312	< 2e-16 ***

Signif. codes:	0 ‘***’	0.001 ‘**’	0.01 ‘*’	0.05 ‘.’
	0.1 ‘ ’	1		

Residual standard error: 3.431 on 387 degrees of freedom

ANOVA table V

```
Multiple R-squared:  0.8082,      Adjusted R-squared:  0.8072  
F-statistic: 815.6 on 2 and 387 DF,  p-value: < 2.2e-16
```

- Notice that the value reported in the row for F-statistic is indeed the F test statistic for the significance of regression test, and additionally it reports the two relevant degrees of freedom.
- Verify the sums of squares and degrees of freedom directly in R.

```
sum((fitted(full_mpg.fit) - fitted(null_mpg.fit)) ^ 2) # SSreg  
[1] 19205.03  
  
sum(resid(full_mpg.fit) ^ 2) # SSres  
[1] 4556.646  
  
sum(resid(null_mpg.fit) ^ 2) # SStot  
[1] 23761.67  
  
#  
# Degrees of Freedom: Regression  
length(coef(full_mpg.fit)) - length(coef(null_mpg.fit))  
[1] 2
```

ANOVA table VI

```
# Degrees of Freedom: Error  
length(resid(full_mpg.fit)) - length(coef(full_mpg.fit))  
[1] 387  
  
# Degrees of Freedom: Total  
length(resid(null_mpg.fit)) - length(coef(null_mpg.fit))  
[1] 389
```

Nested Models I

- The significance of regression test is actually a special case of testing what we will call **nested models**
- One model is "nested" inside the other means that one model contains a subset of the predictors from only the larger model.
- Consider the following full model,

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_{(p-1)} x_{i(p-1)} + \epsilon_i$$

This model has $p - 1$ predictors, so p β -parameters. We will denote the predicted values of this model as $\hat{y}_{A,i}$.

Nested Models II

- Let the null model be

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_{(q-1)} x_{i(q-1)} + \epsilon_i$$

where $q < p$. This model has $q - 1$ predictors, so q β -parameters. We will denote the predicted values of this model as \hat{y}_{0i} .

- The difference between models can be codified by the null hypothesis

$$H_0 : \beta_q = \beta_{q+1} = \cdots = \beta_{p-1} = 0.$$

which is contrasted to

$$H_A : \text{At least one } \beta_j \neq 0, j = q, \dots, p - 1$$

Nested Models III

- Denote $SS_{res}(H_0)$, the sum of squared of residuals under H_0 and $SS_{res}(H_A)$ the sum of squared of residuals under H_A
- The fit of the smaller model in general is such that

$$SS_{res}(H_0) \geq SS_{res}(H_A)$$

- If $SS_{res}(H_0) - SS_{res}(H_A)$ is small, then the fit of the smaller model is almost as good as the larger model and so we would prefer the smaller (more parsimonious) model on the grounds of simplicity.
- On the other hand, if the difference is large, then the superior fit of the larger model would be preferred. This suggests that something like:

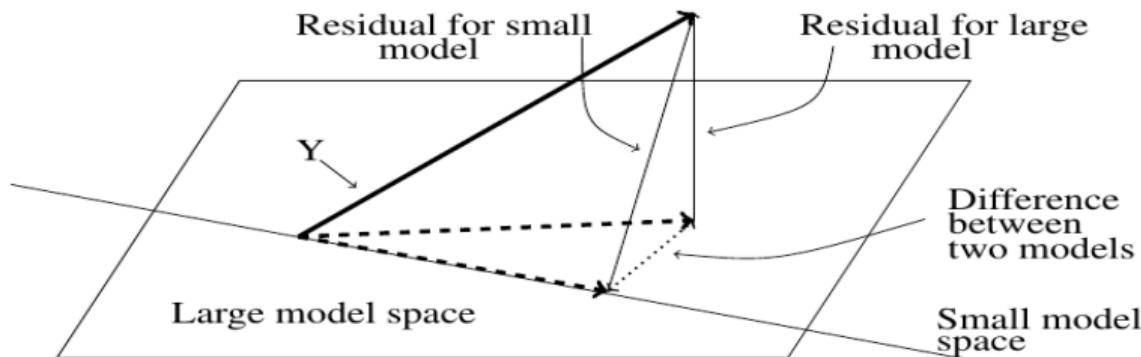
$$\frac{SS_{res}(H_0) - SS_{res}(H_A)}{SS_{res}(H_A)}$$

would be a potentially good test statistic where the denominator is used for scaling purposes.

Nested Models IV

- It turns out that

$$\begin{aligned} SS_{res}(H_0) - SS_{res}(H_A) &= \sum_{i=1}^n (y_i - \hat{y}_{o,i})^2 - \sum_{i=1}^n (y_i - \hat{y}_{A,i})^2 \\ &= \sum_{i=1}^n (\hat{y}_{A,i} - \hat{y}_{o,i})^2 \end{aligned}$$



Nested Models V

- We can then perform this test using an F -test, which is the result of the following **ANOVA table**.

Source	Sum of Squares	Degrees of Freedom	Mean Square	F
diff	$\sum_{i=1}^n (\hat{y}_{A,i} - \hat{y}_{0,i})^2$	$p - q$	$SS_{diff}/(p - q)$	MS_{diff}/MS_{res}
full	$\sum_{i=1}^n (y_i - \hat{y}_{A,i})^2$	$n - p$	$SS_{res}/(n - p)$	
null	$\sum_{i=1}^n (y_i - \hat{y}_{0,i})^2$	$n - q$		

$$\begin{aligned} F &= \frac{(SS_{res}(H_0) - SS_{res}(H_A))/(p - q)}{SS_{res}(H_A)/(n - p)} \\ &= \frac{\sum_{i=1}^n (\hat{y}_{A,i} - \hat{y}_{0,i})^2/(p - q)}{\sum_{i=1}^n (y_i - \hat{y}_{A,i})^2/(n - p)}. \end{aligned}$$

Nested Models VI

- Notice that the row for *diff* compares the sum of the squared differences of the predicted values. The degrees of freedom is the difference in the number of β -parameters estimated in the two models.

Nested Models VII

- For example, the `autompq` dataset has a number of additional variables that we have yet to use.

`names(autompq)`

```
[1] "mpg"   "cyl"   "disp"  "hp"    "wt"    "acc"   "year"
```

- We will consider two different models:

- Full: `mpg ~wt + year + cyl + disp + hp + acc`
- Null: `mpg ~wt + year`

i.e.

$$H_0 : \beta_{\text{cyl}} = \beta_{\text{disp}} = \beta_{\text{hp}} = \beta_{\text{acc}} = 0$$

The alternative is that at least one of the β_j from the null is not 0.

Nested Models VIII

- In R

```
null_mpg.fit <- lm(mpg ~ wt + year, data = autompg)
full_mpg.fit <- lm(mpg ~ wt + year + cyl
                     + disp + hp + acc, data = autompg)
anova(null_mpg.fit, full_mpg.fit)
```

Analysis of Variance Table

Model 1: mpg ~ wt + year

Model 2: mpg ~ wt + year + cyl + disp + hp + acc

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	387	4556.6				
2	383	4530.5	4	26.18	0.5533	0.6967

Nested Models IX

- The F statistic is 0.553, and the p-value is very large, so we fail to reject the null hypothesis at any reasonable α and say that none of cyl, disp, hp, and acc are significant with wt and year already in the model.

Nested Models X

- Verification in R

```
(ssdiff <- sum((fitted(full_mpg.fit) - fitted(null_mpg.fit)) ^ 2)) # SSdiff
[1] 26.17981
sum(resid(full_mpg.fit) ^ 2) # SSR (For Full)
[1] 4530.466
sum(resid(null_mpg.fit) ^ 2) # SSR (For Null)
[1] 4556.646
# Degrees of Freedom: Diff
(dfdfit <- length(coef(full_mpg.fit)) - length(coef(null_mpg.fit)))
[1] 4
# Degrees of Freedom: Full
length(resid(full_mpg.fit)) - length(coef(full_mpg.fit))
[1] 383
# Degrees of Freedom: Null
length(resid(null_mpg.fit)) - length(coef(null_mpg.fit))
[1] 387
# F value
(ssdiff/dfdfit) / (sum(resid(full_mpg.fit) ^ 2)/full_mpg.fit$df.resid)
[1] 0.5533023
```

Quality Criterion I

- We wish to find a model that explain large parts of the variance of the original data: measure this with R^2 :

```
summary(full_mpg.fit)$ r.squared
```

```
[1] 0.8093372
```

```
summary(null_mpg.fit)$ r.squared
```

```
[1] 0.8082355
```

- The added variables are not significant, why does R^2 increase?
- Overfitting: increasing the number of predictors always increases R^2 :

```
useless_covariates <- matrix(rnorm(390*389), ncol = 389)
```

```
summary(lm(autompg$mpg ~ useless_covariates))$ r.squared
```

```
[1] 1
```

Quality Criterion II

- A perfect fit! It does have some issues though

```
summary(lm(autompg$mpg ~ useless_covariates))$coef[1:3,]
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	22.635588	NaN	NaN	NaN
useless_covariates1	-3.288943	NaN	NaN	NaN
useless_covariates2	1.737415	NaN	NaN	NaN

- Criteria for assessing quality of fit such as R^2 and RMSE have a fatal flaw: it is impossible to add a predictor to a model and make R^2 or RMSE worse.
- This suggests that we need a quality criteria that takes into account the size of the model, since our preference is for small models that still fit well.
- Sacrifice a small amount of "goodness-of-fit" to obtain a smaller model.

Quality Criterion III

- We will look at three criteria that do this explicitly: AIC, BIC, and Adjusted R^2 .
- We will also look at one, Cross-Validated RMSE, which implicitly considers the size of the model.
- We will look at frameworks that also prevents overfitting

Information Criteria I

- An information criterion balances the fitness of a model with the number of predictors employed.

$$IC = \text{goodness of fit} + \text{penalty for the complexity}$$

- Complexity of a model is a function of number of parameter $p(\mathcal{M})$
- The goodness of fit can be taken to be the negative log-likelihood: we have chosen the model which minimize its value.
- We use ICs of the form

$$IC = -2 * \text{logLik}(\mathcal{M}_{\mathcal{V}}) + k * p(\mathcal{M})$$

- Choose models which minimize IC: balance good fit with complexity

Information Criteria II

- For a linear regression model, the maximized likelihood reduces to a function of the sum of square of residuals $SS_{res}(\mathcal{M})$ after fitting the model \mathcal{M} since:

$$\begin{aligned}lik(\hat{\beta}, \hat{\sigma}; y, x) &= \sum_{i=1}^n \left\{ -\log 2\pi - \log(\hat{\sigma}) - \frac{1}{2} \left(\frac{y - \hat{y}_i}{\hat{\sigma}} \right)^2 \right\} = \\&= -n \log 2\pi - \frac{n}{2} \log \frac{\sum_{i=1}^n (y - \hat{y}_i)^2}{n} - \frac{1}{2} n = \\&= \text{constant} - \frac{n}{2} \log MSS_{res}(\mathcal{M})\end{aligned}$$

because the maximum likelihood estimator of σ is $\hat{\sigma} = (\sum_{i=1}^n (y - \hat{y}_i)^2)/n$

- (In rare occasions ICs are written as $IC = 2 * \text{logLik}(\mathcal{M}) - k * p(\mathcal{M})$: make sure you know how the software you are using defines them)

Information Criteria III

- For linear models therefore we have:

Akaike's Information Criterion (AIC)

$$AIC(\mathcal{M}) = n \times \log MSS_{res}(\mathcal{M}) + 2p(\mathcal{M})$$

Bayesian Information Criterion (BIC)

$$BIC(\mathcal{M}) = n \times \log MSS_{res}(\mathcal{M}) + \log(n)p(\mathcal{M})$$

- The BIC replaces the 2 in the AIC penalization with $\log(n)$ so it penalizes more complex models more (since $\log(n) > 2$ if $n \geq 8$).

Information Criteria IV

- This is one of the reasons why BIC is preferred by some practitioners for model comparison. Also, because it is consistent in selecting the true model: if enough data is provided, the BIC is guaranteed to select the data-generating model among a list of candidate models.
- The AIC and BIC can be computed in R through the functions AIC and BIC. They take a model as the input.

```
# Two models with different predictors
mod1 <- lm(mpg ~ wt + year, data = autompg)
mod2 <- lm(mpg ~ wt + year+cyl, data = autompg)
# AICs
c(AIC(mod1) , AIC(mod2) )
[1] 2073.468 2074.996
```

Information Criteria V

```
# BICs
c(BIC(mod1) , BIC(mod2) )
[1] 2089.333 2094.826
## adding cyl doesn't pay off
# BIC can also be derived as - see ?AIC
AIC(mod1, k=log(nrow(automp)))
[1] 2089.333
summary(mod2)
```

Information Criteria VI

Call:

```
lm(formula = mpg ~ wt + year + cyl, data = autompg)
```

Residuals:

Min	1Q	Median	3Q	Max
-9.0248	-2.2957	-0.0566	2.0072	14.3550

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-14.1808241	4.0811704	-3.475	0.000569 ***
wt	-0.0063538	0.0004638	-13.700	< 2e-16 ***
year	0.7559116	0.0504040	14.997	< 2e-16 ***
cyl	-0.1601492	0.2341222	-0.684	0.494360

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.434 on 386 degrees of freedom

Multiple R-squared: 0.8085, Adjusted R-squared: 0.807

F-statistic: 543.1 on 3 and 386 DF, p-value: < 2.2e-16

Adjusted R^2 I

- For a least squares model with p variables, the adjusted R^2 statistic is calculated as

$$\text{Adjusted } R^2 = 1 - \frac{SS_{res}/(n - p - 1)}{SS_{tot}/(n - 1)}$$

- ICs: small value indicates a low test error. Adjusted R^2 : high value indicates a model with a small test error.
- Maximizing the adjusted R^2 is equivalent to minimizing $SS_{res}/(n - p - 1)$.
- While SS_{res} always decreases as the number of the variables in the model increases, $SS_{res}/(n - p - 1)$ may increase or decrease, due to the presence of p in the denominator.
- Unlike the R^2 statistic, the adjusted R^2 statistic pays a price for the inclusion of unnecessary variables in the model
- The adjusted R^2 can be computed in R through the functions `summary`

Adjusted R^2 II

```
summary(full_mpg.fit)$adj.r.squared
```

```
[1] 0.8063504
```

```
summary(null_mpg.fit)$adj.r.squared
```

```
[1] 0.8072444
```

- The improvement in the goodness of fit for the more complex model is not large enough to justify the increase in model complexity.

Variable selection

- Find the simplest model to explain the data. Smaller models lead to less variable inference/prediction and can be more explainable than complex models. Use Occam's Razor principle: the smallest model that fits the data is best.
- Simplest model for best fit: use any of the criteria discussed before to choose.
- The most direct approach is called all subsets or best subsets regression: compute the least squares fit for all possible subsets and then choose between them based on some criterion.
- However we often can not examine all possible models, since they are 2^{p-1} of them; for example when $p - 1 = 30$ there are 1073741824 models!
- Instead we need an automated approach that searches through a subset of them. We discuss some commonly used approaches next.

Forward stepwise selection

- ➊ Let \mathcal{M}_0 denote the null model, which contains no predictors.
- ➋ For $k = 1, \dots, p - 2$:
 - Consider all $p - k$ models that augment the predictors in \mathcal{M}_k with one additional predictor.
 - Choose the best among these $p - k$ models, and call it \mathcal{M}_{k+1} . Here best is defined as having smallest SS_{res} .
- ➌ Select a single best model from among $\mathcal{M}_0, \dots, \mathcal{M}_{p-1}$ using cross-validated prediction error, AIC, BIC, or adjusted R^2 .
- Computational advantage over best subset selection is clear.
- It is not guaranteed to find the best possible model out of all 2^{p-1} models containing subsets of the $p - 1$ predictors.

Backward stepwise selection I

- ① Let \mathcal{M}_{p-1} denote the full model, which contains all $p - 1$ predictors.
- ② For $k = p - 1, p - 2, \dots, 1$:
 - Consider all k models that contain all but one of the predictors in \mathcal{M}_k , for a total of $k - 1$ predictors
 - Choose the best among these k models, and call it \mathcal{M}_{k-1} . Here best is defined as having smallest SS_{res} .
- ③ Select a single best model from among $\mathcal{M}_0, \dots, \mathcal{M}_{p-1}$ using cross-validated prediction error, AIC, BIC, or adjusted R^2 .

Backward stepwise selection II

- Like forward stepwise selection, the backward selection approach searches through only $1 + p(p - 1)/2$ models, and so can be applied in settings where p is too large to apply best subset selection
- Like forward stepwise selection, backward stepwise selection is not guaranteed to yield the best model containing a subset of the $p - 1$ predictors.
- Backward selection requires that the number of samples n is larger than the number of variables $p - 1$ (so that the full model can be fit). In contrast, forward stepwise can be used even when $n < p - 1$, and so is the only viable subset method when p is very large.

Forward and Backward selection in R I

- Forward search (trace=0 gives only the final model)

```
null<-lm(mpg ~ 1,data = autompg)
full<-lm(mpg ~ .,data = autompg)
step(null, scope=list(lower=null, upper=full), direction="forward",
k=2, trace=1)
```

Start: AIC=1604.78

mpg ~ 1

	Df	Sum of Sq	RSS	AIC
+ wt	1	16444.5	7317.1	1147.4
+ disp	1	15391.8	8369.8	1199.8
+ hp	1	14387.1	9374.6	1244.0
+ cyl	1	14349.7	9412.0	1245.6
+ year	1	7976.4	15785.3	1447.3
+ acc	1	4319.0	19442.7	1528.5
<none>		23761.7	1604.8	

Step: AIC=1147.41

mpg ~ wt

Forward and Backward selection in R II

	Df	Sum of Sq	RSS	AIC
+ year	1	2760.48	4556.6	964.7
+ hp	1	327.93	6989.2	1131.5
+ acc	1	172.88	7144.2	1140.1
+ disp	1	151.53	7165.6	1141.2
+ cyl	1	114.18	7202.9	1143.3
<none>		7317.1		1147.4

Step: AIC=964.7

mpg ~ wt + year

	Df	Sum of Sq	RSS	AIC
<none>		4556.6	964.70	
+ acc	1	9.4275	4547.2	965.89
+ cyl	1	5.5169	4551.1	966.22
+ hp	1	2.9872	4553.7	966.44
+ disp	1	0.0749	4556.6	966.69

Forward and Backward selection in R III

Call:

```
lm(formula = mpg ~ wt + year, data = autompg)
```

Coefficients:

(Intercept)	wt	year
-14.637642	-0.006635	0.761402

Forward and Backward selection in R IV

- Backward search

```
step(full, scope=list(lower=null, upper=full),
      direction="backward", k=2, trace=1)
```

Start: AIC=970.45

mpg ~ cyl + disp + hp + wt + acc + year

	Df	Sum of Sq	RSS	AIC
- hp	1	0.03	4530.5	968.45
- acc	1	7.29	4537.8	969.08
- cyl	1	13.28	4543.8	969.59
- disp	1	14.01	4544.5	969.65
<none>		4530.5	970.45	
- wt	1	1211.71	5742.2	1060.88
- year	1	2428.31	6958.8	1135.83

Step: AIC=968.45

mpg ~ cyl + disp + wt + acc + year

Df	Sum of Sq	RSS	AIC
----	-----------	-----	-----

Forward and Backward selection in R V

```
- acc   1     13.15 4543.6  967.58  
- cyl   1     13.31 4543.8  967.60  
- disp  1     14.55 4545.0  967.70  
<none>          4530.5  968.45  
- wt    1     1520.47 6051.0 1079.31  
- year  1     2558.16 7088.6 1141.04
```

Step: AIC=967.58

mpg ~ cyl + disp + wt + year

	Df	Sum of Sq	RSS	AIC
- disp	1	7.48	4551.1	966.22
- cyl	1	12.92	4556.6	966.69
<none>		4543.6	967.58	
- wt	1	1569.79	6113.4	1081.32
- year	1	2614.75	7158.4	1142.86

Step: AIC=966.22

mpg ~ cyl + wt + year

Forward and Backward selection in R VI

```
Df Sum of Sq    RSS    AIC
- cyl   1      5.52 4556.6  964.70
<none>           4551.1  966.22
- wt    1  2213.10 6764.2 1118.77
- year  1  2651.82 7202.9 1143.28
```

Step: AIC=964.7

mpg ~ wt + year

```
Df Sum of Sq    RSS    AIC
<none>           4556.6  964.7
- year  1      2760.5 7317.1 1147.4
- wt    1     11228.6 15785.3 1447.3
```

Call:

```
lm(formula = mpg ~ wt + year, data = autompg)
```

Coefficients:

Forward and Backward selection in R VII

	wt	year
(Intercept)	-14.637642	0.761402
wt	-0.006635	

Stepwise Search I

- Stepwise search 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.
- We perform stepwise search using AIC as our criterion.

```
step(null, scope = list(upper=full),
      direction="both", trace=1, k=2)
```

Start: AIC=1604.78

mpg ~ 1

	Df	Sum of Sq	RSS	AIC
+ wt	1	16444.5	7317.1	1147.4
+ disp	1	15391.8	8369.8	1199.8
+ hp	1	14387.1	9374.6	1244.0
+ cyl	1	14349.7	9412.0	1245.6
+ year	1	7976.4	15785.3	1447.3
+ acc	1	4319.0	19442.7	1528.5
<none>		23761.7	1604.8	

Stepwise Search II

Step: AIC=1147.41

mpg ~ wt

	Df	Sum of Sq	RSS	AIC
+ year	1	2760.5	4556.6	964.7
+ hp	1	327.9	6989.2	1131.5
+ acc	1	172.9	7144.2	1140.1
+ disp	1	151.5	7165.6	1141.2
+ cyl	1	114.2	7202.9	1143.3
<none>			7317.1	1147.4
- wt	1	16444.5	23761.7	1604.8

Step: AIC=964.7

mpg ~ wt + year

	Df	Sum of Sq	RSS	AIC
<none>			4556.6	964.70
+ acc	1	9.4	4547.2	965.89

Stepwise Search III

```
+ cyl    1      5.5  4551.1  966.22  
+ hp     1      3.0  4553.7  966.44  
+ disp   1      0.1  4556.6  966.69  
- year   1    2760.5  7317.1 1147.41  
- wt     1   11228.6 15785.3 1447.27
```

Call:

```
lm(formula = mpg ~ wt + year, data = autompg)
```

Coefficients:

(Intercept)	wt	year
-14.637642	-0.006635	0.761402

Exhaustive Search I

- The all possible regressions approach considers all possible subsets of the pool of explanatory variables and finds the model that best fits the data according to some criteria
- These criteria assign scores to each model and allow us to choose the model with the best score. The function `regsubsets` in the package `leaps` can be used for regression subset selection. Thereafter, one can view the ranked models according to different scoring criteria by plotting the results of `regsubsets`
- Before using the function for the first time you will need to install the library using the command

```
install.packages("leaps")
library(leaps)
all.mod <- summary(regsubsets(mpg ~ ., data = autompg))
```

Exhaustive Search II

- We'll now look at the information stored in all.mod

`all.mod$which`

	(Intercept)	cyl	disp	hp	wt	acc	year
1	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
2	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE
3	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE
4	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE
5	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE
6	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE

- Using \$which gives us the best model, according to SS_{res} , for a model of each possible size, in this case ranging from one to six predictors. For example the best model with three predictors ($p = 4$) would use wt, acc and year.

Exhaustive Search III

- The corresponding SS_{res} , BIC, adjusted R^2 are

```
all.mod$rss
```

```
[1] 7317.127 4556.646 4547.218 4543.648 4530.493 4530.466
```

```
all.mod$bic
```

```
[1] -447.4316 -626.1815 -621.0231 -615.3633 -610.5279 -604.5641
```

```
all.mod$adjr2
```

```
[1] 0.6912681 0.8072444 0.8071449 0.8067958 0.8068535 0.8063504
```

To find which model has the highest BIC we can use the `which.max` function.

```
best.BIC<- which.max(all.mod$bic)
```

```
all.mod$which[best.BIC, ]
```

(Intercept)	cyl	disp	hp	wt	acc	year
TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE

Variable selection - some perspective I

- Automatic methods are tempting but might not yield the "best" model.
- Models still need to be assessed - for example checking the linearity and other assumptions (see also later).
- Stepwise variable selection tends to pick models that are smaller than desirable for prediction purposes.
- One variable being out of the model doesn't mean it is not related to Y : it means other variables relate more or explain the same pattern (for the dataset under study).
- We select a model that *predicts well*. This does not mean the model uncovers a *true relationship* between real-world variables.
- We use the data twice: to select the model and then to do inference - that's cheating! We would be overconfident in our results. The *Inference after Selection* problem.

Inference after Selection

- All statistical theory for the regression models is derived assuming the model is fixed and known **before** seeing the data.
- If we use the data to select a model we will be selecting the "best" variables: significance will be, for example, overstated.
- The data is random, and model selection depends on the data, so it becomes random: the statistical formulas we use do not account for this additional randomness.
- (this is a really complicated issue, even plotting the data to "have a look" is using the data twice)
- How can we solve this?
 - If you are only interested in the "best model" you can disregard this issue, but typically we are interested in inference
 - Use more advanced statistical theory (we don't see that here)
 - Data splitting (assuming you have independent observations): choose the model using a data subset and estimate its final form using another the remaining data points.

Validation-based selection I

- Each of the previous three metrics explicitly used p , the number of parameters, in their calculations. Thus, they all explicitly limit the size of models chosen when used to compare models.
- The calculations rely on an assumption of "true" underlying model
- Often times models are needed for out-of-sample prediction: we need to quantify the possible error.
- First up: **cross-validation**.

Validation-based selection II

- Fit any model to the dataset available *without* one observation i .
- Predict y_i for the model in which (x_i, y_i) was not included in the model. Denote this with $\hat{y}_{[i]}$.
- Now $e_{[i]}$ is the residual for the i th observation, when that observation is not used to fit the model: $e_{[i]} = y_i - \hat{y}_{[i]}$
- We define the **leave-one-out cross-validated RMSE** to be

$$\text{RMSE}_{\text{LOOCV}} = \sqrt{\frac{1}{n} \sum_{i=1}^n e_{[i]}^2}.$$

- LOOCV gives a measure of the possible out-of-sample prediction error
- In general, to perform this calculation, we would be required to fit the model n times, once with each possible observation removed.

Validation-based selection III

- However, for leave-one-out cross-validation and linear models, the equation can be rewritten as

$$\text{RMSE}_{\text{LOOCV}} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{e_i}{1 - h_{ii}} \right)^2},$$

where h_{ii} are the **leverages** the diagonal elements of the (projection) matrix

$$H = X(X^\top X)^{-1}X^\top$$

and e_i are the usual residuals.

- This is great, because now we can obtain the $\text{RMSE}_{\text{LOOCV}}$ by fitting only one model!

Validation-based selection IV

- In practice 5 or 10 fold cross-validation are much more popular. For example, in 5-fold cross-validation, the model is fit 5 times, each time leaving out a fifth of the data, then predicting on those values.
- More on cross-validation to in a master course of statistical learning and simply use LOOCV here
- Let's calculate $\text{RMSE}_{\text{LOOCV}}$ for some of the previous models
- We first write a function which calculates the LOOCV RMSE as defined using the shortcut formula for linear models.

```
calc_loocv_rmse = function(model) {  
  sqrt(mean((resid(model) / (1 - hatvalues(model)))) ^ 2))  
}  
  
calc_loocv_rmse(lm(mpg ~ 1, data = autompg)) # no variable model  
[1] 7.825664
```

Validation-based selection V

```
calc_loocv_rmse(lm(mpg ~ wt+year, data = autompg)) # "best" model  
[1] 3.444656  
  
calc_loocv_rmse(lm(mpg ~ ., data = autompg)) # model with all variables  
[1] 3.480746
```

- The model selected by stepwise selection also has the lowest RMSE_{LOOCV}

Categorical Predictors and Interactions⁸

⁸Material in these slides was heavily influenced by David Dalpiaz *Applied Statistics with R!*, the book is under active development.

Introductory example I

- We consider the `mtcars` dataset

```
head(mtcars)
```

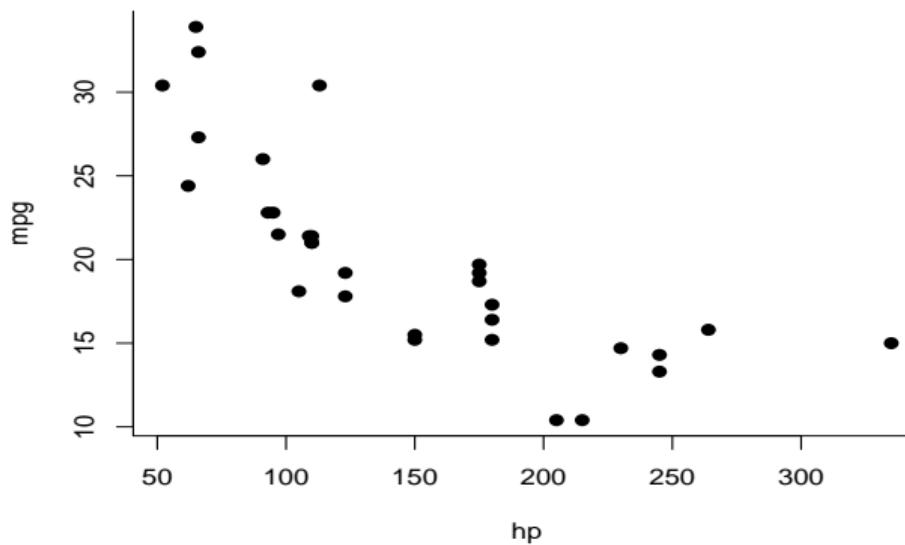
	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

and three of the variables: `mpg`, `hp`, and `am`.

We wish to see whether the car's horsepower (hp , X_1) affect its mean consumption (mpg , Y).

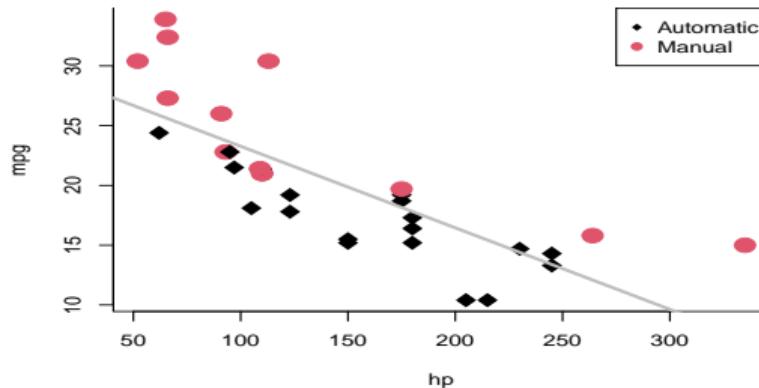
Introductory example II

- We will start by plotting the data:



Introductory example III

- We now fit the simple linear regression model: $Y = \beta_0 + \beta_1 x_1 + \epsilon$
`mpg_hp_slr <- lm(mpg ~ hp, data = mtcars)`



- We notice some systematic deviations for most manual/automatic cars from the mean.

Dummy Variables I

- A new model

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon,$$

where x_1 and Y remain the same, but now

$$x_2 = \begin{cases} 1 & \text{manual transmission} \\ 0 & \text{automatic transmission} \end{cases}$$

- We call x_2 a **dummy variable**

A dummy variable is a numerical variable that is used in a regression analysis to code for a binary categorical variable.

Dummy Variables II

- Since x_2 can only take values 0 and 1, we can effectively write two different models, one for manual and one for automatic transmissions.
 - ➊ For automatic transmissions, that is $x_2 = 0$, we have,

$$Y = \beta_0 + \beta_1 x_1 + \epsilon.$$

- ➋ For manual transmissions, that is $x_2 = 1$, we have,

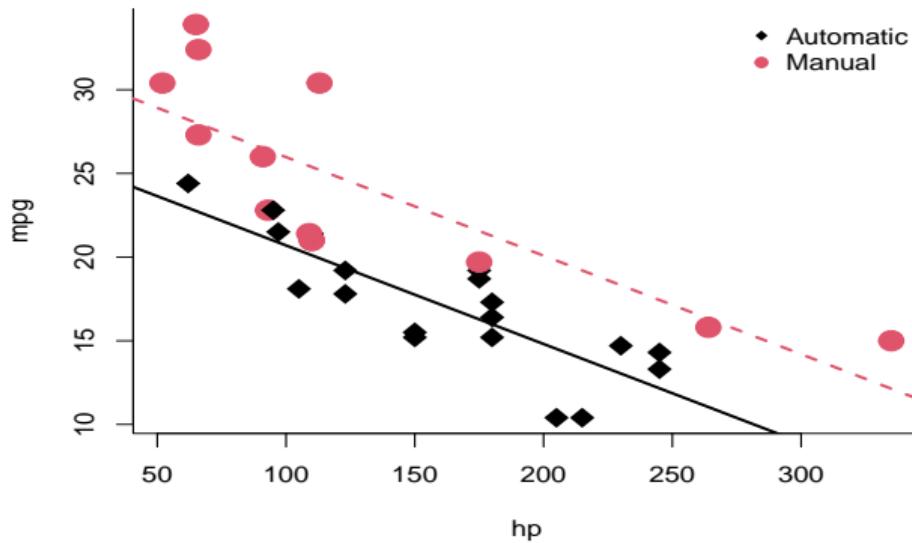
$$Y = (\beta_0 + \beta_2) + \beta_1 x_1 + \epsilon.$$

- Notice that these models share the same slope, β_1 , but have different intercepts, differing by β_2 .
- We fit the model

```
mpg_hp_add <- lm(mpg ~ hp + am, data = mtcars)
```

and we plot the 'two' fitted model

Dummy Variables III



Dummy Variables IV

- Not quite two different models since σ is "shared" by both groups and estimated using all data points.
- We would like to test:

$$H_0 : \beta_2 = 0 \quad \text{vs} \quad H_A : \beta_2 \neq 0.$$

- The test statistic and p-value for the t -test

```
summary(mpg_hp_add)$coefficients["am",]  
Estimate Std. Error t value Pr(>|t|)  
5.27708530818 1.07954057578 4.88826953480 0.00003460318
```

- The F test

```
anova(mpg_hp_slr, mpg_hp_add)
```

Dummy Variables V

Analysis of Variance Table

```
Model 1: mpg ~ hp
Model 2: mpg ~ hp + am
  Res.Df   RSS Df Sum of Sq    F    Pr(>F)
  1     30 447.67
  2     29 245.44  1    202.24 23.895 0.0000346 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Notice that the F test statistic is the t test statistic squared.

Interactions I

- We will return to the autompg dataset with a few modifications.

```
dl <- "http://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/auto-mpg.data"
autompg = read.table(dl, quote = "\"", comment.char = "",
                     stringsAsFactors = FALSE)
colnames(autompg) <- c("mpg", "cyl", "disp",
                       "hp", "wt", "acc", "year", "origin", "name")
autompg <- subset(autompg, autompg$hp != "?")
autompg <- subset(autompg, autompg$name != "plymouth reliant")
rownames(autompg) <- paste(autompg$cyl, "cylinder", autompg$year, autompg$name)
autompg <- subset(autompg,
                  select = c("mpg", "cyl", "disp", "hp", "wt", "acc", "year", "origin"))
autompg$domestic <- as.numeric(autompg$origin == 1)
autompg$hp <- as.numeric(autompg$hp)
autompg <- autompg[autompg$cyl != 5,]
autompg <- autompg[autompg$cyl != 3,]
autompg$cyl <- as.factor(autompg$cyl)
```

- We have removed cars with 3 and 5 cylinders
- A new variable domestic indicates whether a car was built in the United States.

Interactions II

- We have made cyl into factor variable
- We will be concerned with three variables: mpg, disp, and domestic.

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon,$$

- Y is mpg, the fuel efficiency in miles per gallon,
- x_1 is disp, the displacement in cubic inches,
- x_2 is domestic is a dummy variable (1 if the car was built in the United States, and 0 otherwise)

Interactions III

```
mpg_disp_add <- lm(mpg ~ disp + domestic , data = autompg)
summary(mpg_disp_add)
```

Call:

```
lm(formula = mpg ~ disp + domestic, data = autompg)
```

Residuals:

Min	1Q	Median	3Q	Max
-10.559	-2.904	-0.576	2.434	18.814

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)		
(Intercept)	35.486645	0.489885	72.439	<2e-16 ***		
disp	-0.057251	0.002907	-19.696	<2e-16 ***		
domestic	-1.300402	0.632372	-2.056	0.0404 *		

Signif. codes:	0 ‘***’	0.001 ‘**’	0.01 ‘*’	0.05 ‘.’	0.1 ‘ ’	1

Residual standard error: 4.506 on 380 degrees of freedom

Multiple R-squared: 0.6721, Adjusted R-squared: 0.6704

F-statistic: 389.5 on 2 and 380 DF, p-value: < 2.2e-16

Interactions IV

- Now consider the following model,

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \epsilon,$$

where x_1 , x_2 , and Y are the same as before, but we have added a new **interaction** term $x_1 x_2$ which multiplies x_1 and x_2 , so we also have an additional β parameter β_3 .

- This model essentially creates two slopes and two intercepts, β_2 being the difference in intercepts and β_3 being the difference in slopes.

- For $x_2 = 0$ we have

$$Y = \beta_0 + \beta_1 x_1 + \epsilon$$

- For $x_2 = 1$ we have

$$Y = (\beta_0 + \beta_2) + (\beta_1 + \beta_3)x_1 + \epsilon$$

Interactions V

- How do we fit this model in R?

- ➊ Create a new variable, then fit a model like any other.

```
autompq$x3 <- autompq$disp * autompq$domestic  
do_not_do_this <- lm(mpg ~ disp + domestic + x3,  
                      data = autompq)
```

You should only do this as a last resort!

- ➋ Use the existing data with an interaction term such as the : operator.

```
mpg_disp_int <- lm(mpg ~ disp + domestic + disp:domestic,  
                     data = autompq)
```

- ➌ An alternative method uses the * operator. This method automatically creates the interaction term, as well as any "lower order terms" which in this case are the first order terms for disp and domestic

```
mpg_disp_int2 <- lm(mpg ~ disp * domestic, data = autompq)
```

Interactions VI

```
summary(mpg_disp_int)
Call:
lm(formula = mpg ~ disp + domestic + disp:domestic, data = autompg)

Residuals:
    Min      1Q  Median      3Q     Max 
-10.8332 -2.8956 -0.8332  2.2828 18.7749 

Coefficients:
            Estimate Std. Error t value    Pr(>|t|)    
(Intercept) 46.05484   1.80582  25.504 < 2e-16 ***
disp        -0.15692   0.01668  -9.407 < 2e-16 ***
domestic    -12.57547  1.95644  -6.428 0.00000000039 ***
disp:domestic  0.10252   0.01692   6.060 0.000000000329 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.308 on 379 degrees of freedom
Multiple R-squared:  0.7011,      Adjusted R-squared:  0.6987 
F-statistic: 296.3 on 3 and 379 DF,  p-value: < 2.2e-16
```

Interactions VII

- In this case, testing for $\beta_3 = 0$ is testing for two lines with parallel slopes versus two lines with possibly different slopes.

$$H_0 : \beta_3 = 0 \quad VS \quad H_0 : \beta_3 \neq 0$$

The `disp:domestic` line in the `summary` output uses a *t*-test to perform the test.

```
summary(mpg_disp_int)$coefficients["disp:domestic",]  
Estimate Std. Error t value Pr(>|t|)  
0.102518371591524 0.016918176197766 6.059658582173837 0.000000003293721
```

Interactions VIII

- We could also use an ANOVA F -test: the additive model (null model) against the interaction model (the alternative model).

```
anova(mpg_disp_add, mpg_disp_int)
```

Analysis of Variance Table

Model 1: mpg ~ disp + domestic

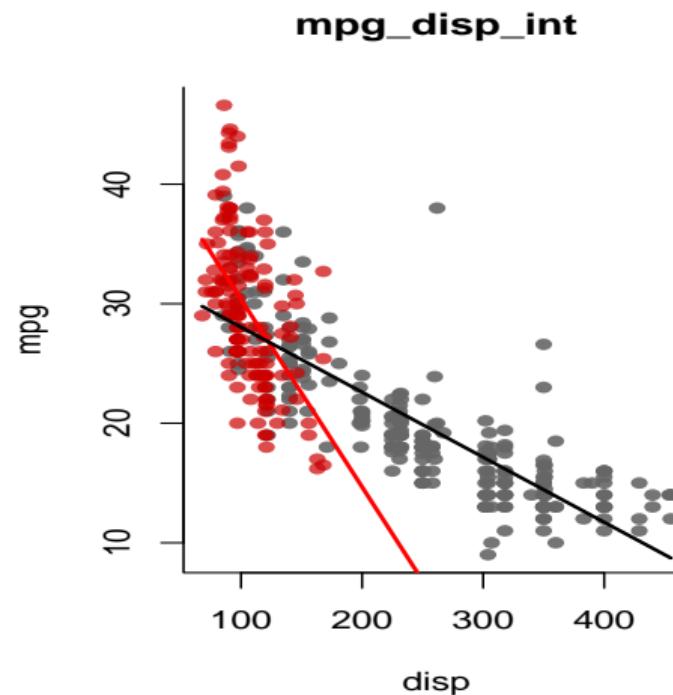
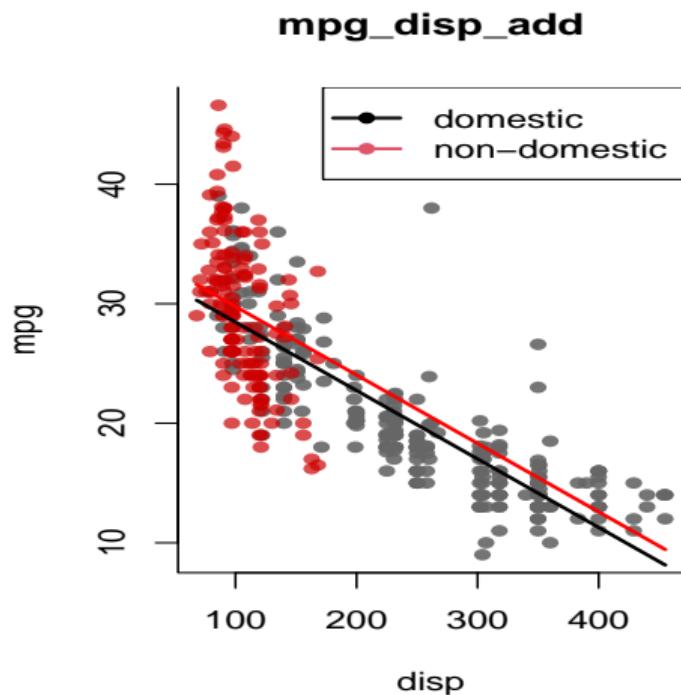
Model 2: mpg ~ disp + domestic + disp:domestic

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	380	7714.0			
2	379	7032.6	1	681.36	36.719 0.000000003294 ***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

- We see this test has the same p-value as the t -test. Also the p-value is extremely low, so between the two, we choose the interaction model.

Interactions IX



Factor Variables I

- Till now: binary categorical variables.
- We will now discuss **factor** variables, which is a special way that R deals with categorical variables.
- The `domestic` variable was not a factor variable.

```
is.factor(autompg$domestic)
```

```
[1] FALSE
```

- The `origin` variable stores `domestic` for domestic cars and `foreign` for foreign cars.

```
autompg$origin[autompg$domestic == 1] = "domestic"
```

```
autompg$origin[autompg$domestic == 0] = "foreign"
```

```
autompg$origin[1:4]
```

```
[1] "domestic" "domestic" "domestic" "domestic"
```

```
is.factor(autompg$origin)
```

```
[1] FALSE
```

Factor Variables II

- We will coerce this origin variable to a factor variable.

```
autompg$origin <- as.factor(autompg$origin)
```

- Factor variables have **levels** which are the possible values (categories) that the variable may take, in this case foreign or domestic.

```
autompg$origin[1:4]
```

```
[1] domestic domestic domestic domestic
```

```
Levels: domestic foreign
```

```
levels(autompg$origin)
```

```
[1] "domestic" "foreign"
```

- factor can also contain ordinal variables - see ?factor to know more

Factor Variables III

- We fit the model

```
mnum<-lm(mpg ~ disp * domestic, data = autompg)
```

- Now let's try to do the same, but using our new factor variable.

```
mfac<-lm(mpg ~ disp * origin, data = autompg)
```

- We have the same fitting

```
summary(mnum)$r.squared
```

```
[1] 0.701081
```

```
summary(mfac)$r.squared
```

```
[1] 0.701081
```

Factor Variables IV

- However it seems that it doesn't produce the same results.

mnum

Call:

```
lm(formula = mpg ~ disp * domestic, data = autompg)
```

Coefficients:

(Intercept)	disp
46.0548	-0.1569
domestic	disp:domestic
-12.5755	0.1025

mfac

Call:

```
lm(formula = mpg ~ disp * origin, data = autompg)
```

Coefficients:

(Intercept)	disp
33.47937	-0.05441
originforeign	disp:originforeign
12.57547	-0.10252

Factor Variables V

- Right away we notice that the intercept is different, as is the coefficient in front of disp. We also notice that the remaining two coefficients are of the same magnitude as their respective counterparts using the domestic variable, but with a different sign.
- What is happening?
- It turns out, that by using a factor variable, R is automatically creating a dummy variable for us. However, it is not the dummy variable that we had originally used ourselves.
- Actually our variable was not a proper dummy variable - as the model allowed us to do this:

```
predict(mnum,newdata = data.frame(disp=197,domestic=0.2))
```

1

16.66497

```
##a useless prediction - what does domestic=0.2 mean???
```

Factor Variables VI

- When we include the factor variable R is fitting the model

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \epsilon,$$

- Y is mpg, the fuel efficiency in miles per gallon,
 - x_1 is disp, the displacement in cubic inches,
 - x_2 is a dummy variable created by R. It uses 1 to represent a foreign car.
- When R created x_2 , the dummy variable, it used domestic cars as the **reference** level, that is the default value of the factor variable. So when the dummy variable is 0, the model represents this reference level, which is domestic (R makes this choice because domestic comes before foreign alphabetically.)

Factor Variables VII

- Using a factor recognizes that the variable can only take two values. So we can predict values for a car that is either domestic or foreign

```
## we can predict this
```

```
predict(mfac, newdata = data.frame(disp=c(195,195),  
                                     origin=c("domestic","foreign")))
```

```
1      2
```

```
22.87030 15.45469
```

but R can not handle the case of an unknown label:

```
predict(mfac,  
        newdata = data.frame(disp=197,origin="unknown"))  
## this gives an error
```

Factors with More Than Two Levels I

- Let's now consider a factor variable with more than two levels. cyl is an example.

```
is.factor(autompg$cyl)
```

```
[1] TRUE
```

```
levels(autompg$cyl)
```

```
[1] "4" "6" "8"
```

- cyl as a numerical variable forces the difference in average mpg between 4 and 6 cylinders to be the same as the difference in average mpg between 6 and 8 cylinders.
- This is a decision commonly made with ordinal variables.
- With a large number of categories, the decision to treat them as numerical variables is appropriate because, otherwise, a large number of dummy variables are then needed to represent these variables.

Factors with More Than Two Levels II

- Let's define three dummy variables related to the cyl factor variable.

$$v_1 = \begin{cases} 1 & \text{4 cylinder} \\ 0 & \text{not 4 cylinder} \end{cases}$$

$$v_2 = \begin{cases} 1 & \text{6 cylinder} \\ 0 & \text{not 6 cylinder} \end{cases}$$

$$v_3 = \begin{cases} 1 & \text{8 cylinder} \\ 0 & \text{not 8 cylinder} \end{cases}$$

Factors with More Than Two Levels III

- We fit an additive model in R, using mpg as the response, and disp and cyl as predictors that uses "three regression lines" to model mpg, one for each of the possible cyl levels.

$$Y = \beta_0 + \beta_1 x + \beta_2 v_2 + \beta_3 v_3 + \epsilon,$$

where

- Y is mpg, the fuel efficiency in miles per gallon,
 - x is disp, the displacement in cubic inches,
 - v_2 and v_3 are the dummy variables define above.
- Notice we use two dummy variables to codify the three level categorical variable.

Factors with More Than Two Levels IV

- In R

```
mpg_disp_add_cyl <- lm(mpg ~ disp + cyl,  
                         data = autompg)
```

mpg_disp_add_cyl

Call:

```
lm(formula = mpg ~ disp + cyl, data = autompg)
```

Coefficients:

(Intercept)	disp	cyl6	cyl8
34.99929	-0.05217	-3.63325	-2.03603

- R doesn't use v_1 because it doesn't need to.
- To create three lines, it only needs two dummy variables since it is using a **reference level**.

Factors with More Than Two Levels V

- R automatically creates an appropriate model matrix:

```
head(model.matrix(mpg_disp_add_cyl),4)
```

	(Intercept)	disp	cyl6	cyl8
--	-------------	------	------	------

[1,]	1	307	0	1
[2,]	1	350	0	1
[3,]	1	318	0	1
[4,]	1	304	0	1

```
colSums(model.matrix(mpg_disp_add_cyl))
```

(Intercept)	disp	cyl6	cyl8
383.0	75214.5	83.0	103.0

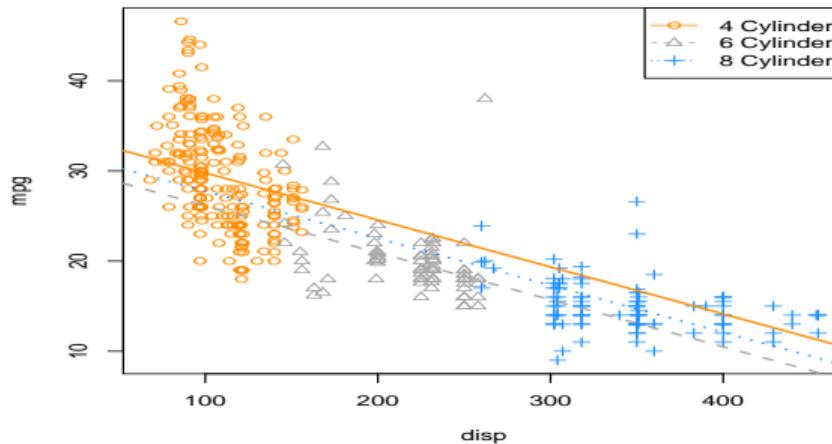
```
table(autompq$cyl) ## correct totals
```

4	6	8
197	83	103

Factors with More Than Two Levels VI

- The three "sub models" are then:
 - 4 Cylinder: $Y = \beta_0 + \beta_1x + \epsilon$
 - 6 Cylinder: $Y = (\beta_0 + \beta_2) + \beta_1x + \epsilon$
 - 8 Cylinder: $Y = (\beta_0 + \beta_3) + \beta_1x + \epsilon$
- Notice that they all have the same slope. However, using two dummy variables, we achieve three intercepts.
- In this case 4 cylinder is the **reference level**, β_0 is specific to 4 cylinders, but β_2 and β_3 are used to represent quantities relative to 4 cylinders.

Factors with More Than Two Levels VII



- The odd result here is that we're estimating that 8 cylinder cars have better fuel efficiency than 6 cylinder cars at any displacement!

Factors with More Than Two Levels VIII

- The dotted blue line is always above the dashed grey line. That doesn't seem right. Maybe for very large displacement engines that could be true, but that seems wrong for medium to low displacement.
- To attempt to fix this, we will try using an interaction model

```
mpg_disp_int_cyl <- lm(mpg ~ disp * cyl, data = autompg)  
mpg_disp_int_cyl
```

Call:

```
lm(formula = mpg ~ disp * cyl, data = autompg)
```

Coefficients:

(Intercept)	disp	cyl6	cyl8	disp:cyl6	disp:cyl8
43.59052	-0.13069	-13.20026	-20.85706	0.08299	0.10817

- R has fit the model

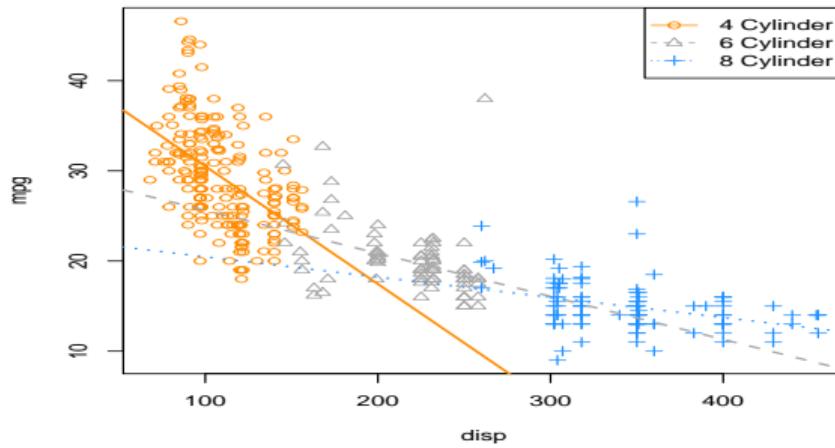
$$Y = \beta_0 + \beta_1 x + \beta_2 v_2 + \beta_3 v_3 + \gamma_2 x v_2 + \gamma_3 x v_3 + \epsilon$$

(We're using γ like a β parameter for simplicity, so that, for example β_2 and γ_2 are both associated with v_2)

Factors with More Than Two Levels IX

- Now, the three "sub models" are:
 - 4 Cylinder: $Y = \beta_0 + \beta_1 x + \epsilon.$
 - 6 Cylinder: $Y = (\beta_0 + \beta_2) + (\beta_1 + \gamma_2)x + \epsilon.$
 - 8 Cylinder: $Y = (\beta_0 + \beta_3) + (\beta_1 + \gamma_3)x + \epsilon.$
- Interpretation of some parameters and coefficients:
 - $(\hat{\beta}_0 + \hat{\beta}_2) = 43.591 - 13.2 = 30.39$ is the estimated average mpg of a 6 cylinder car with 0 disp
 - $(\hat{\beta}_1 + \hat{\gamma}_3) = -0.131 + 0.108 = -0.023$ is the estimated change in average mpg for an increase of one disp, for an 8 cylinder car.
- So, as we have seen before β_2 and β_3 change the intercepts for 6 and 8 cylinder cars relative to the reference level of β_0 for 4 cylinder cars.
- Now, similarly γ_2 and γ_3 change the slopes for 6 and 8 cylinder cars relative to the reference level of β_1 for 4 cylinder cars.

Factors with More Than Two Levels X



Factors with More Than Two Levels XI

- To completely justify the interaction model (i.e., a unique slope for each cyl level) compared to the additive model (single slope), we can perform an F -test

$$H_0 : \gamma_2 = \gamma_3 = 0$$

which represents the parallel regression lines we saw before,

$$Y = \beta_0 + \beta_1 x + \beta_2 v_2 + \beta_3 v_3 + \epsilon.$$

Factors with More Than Two Levels XII

```
anova(mpg_disp_add_cyl, mpg_disp_int_cyl)
Analysis of Variance Table

Model 1: mpg ~ disp + cyl
Model 2: mpg ~ disp * cyl
  Res.Df   RSS Df Sum of Sq    F    Pr(>F)
  1     379 7299.5
  2     377 6551.7  2     747.79 21.515 0.000000001419 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- As expected, we see a very low p-value, and thus reject the null. We prefer the interaction model over the additive model.

Linear models repurposed I

- What if we only use a factor variable in the analysis? See a factor with two levels:

```
mfa_only <- lm(mpg ~ origin, data = autompg)  
summary(mfa_only)$coef
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	19.962963	0.4103362	48.65026	1.518921e-165
originforeign	9.453466	0.6786959	13.92887	6.201240e-36

- The fitted sub model are:
 - domestic cars: $Y = \beta_0 + \epsilon.$
 - foreign cars: $Y = (\beta_0 + \beta_1) + \epsilon.$
- So $E[Y|domestic] = \beta_0$ and $E[Y|foreign] = \beta_0 + \beta_1$ so β_1 represents the difference in means and a test with $H_0 : \beta_1 = 0$ can be re-interpreted as a test for $E[Y|domestic] = E[Y|foreign]$.
- This is a t-test

Linear models repurposed II

- Specifically a t-test in which equal variances are assumed:

```
t.test(mpg ~ origin, data = autompg, var.equal = TRUE)
```

Two Sample t-test

```
data: mpg by origin
t = -13.929, df = 381, p-value < 2.2e-16
alternative hypothesis: true difference in means between group domestic and group foreign is not equal to 0
95 percent confidence interval:
-10.787924 -8.119007
sample estimates:
mean in group domestic mean in group foreign
          19.96296           29.41643
summary(mfa_only)$coef
            Estimate Std. Error   t value    Pr(>|t|)    
(Intercept) 19.962963  0.4103362 48.65026 1.518921e-165
originforeign 9.453466  0.6786959 13.92887 6.201240e-36
```

Linear models repurposed III

- We get the same confidence interval for the difference in the two means:

```
confint(mfa_only)[2,]
```

```
2.5 % 97.5 %
```

```
8.119007 10.787924
```

```
t.test(mpg ~ origin, data = autompg, var.equal = TRUE)$conf.int
```

```
[1] -10.787924 -8.119007
```

```
attr("conf.level")
```

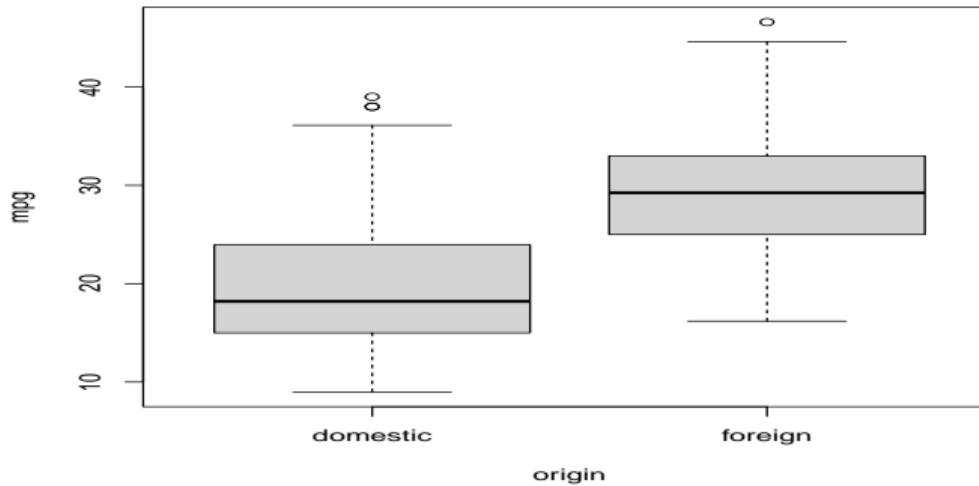
```
[1] 0.95
```

- In this case we have evidence to say the two means are different

Linear models repurposed IV

- Visually:

```
boxplot(mpg ~ origin, data = autompg)
```



Linear models repurposed V

- What happens to factors with more than levels:

```
summary(lm(mpg ~ cyl, data = autompg))$coef
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	29.290863	0.3346673	87.52233	6.037066e-254
cyl6	-9.317369	0.6146859	-15.15793	6.311377e-41
cyl8	-14.327756	0.5711567	-25.08551	1.288581e-82

- There a widely-employed statistical technique called ANOVA which generalizes the T-test.

```
summary(aov(mpg ~ cyl, data = autompg))
```

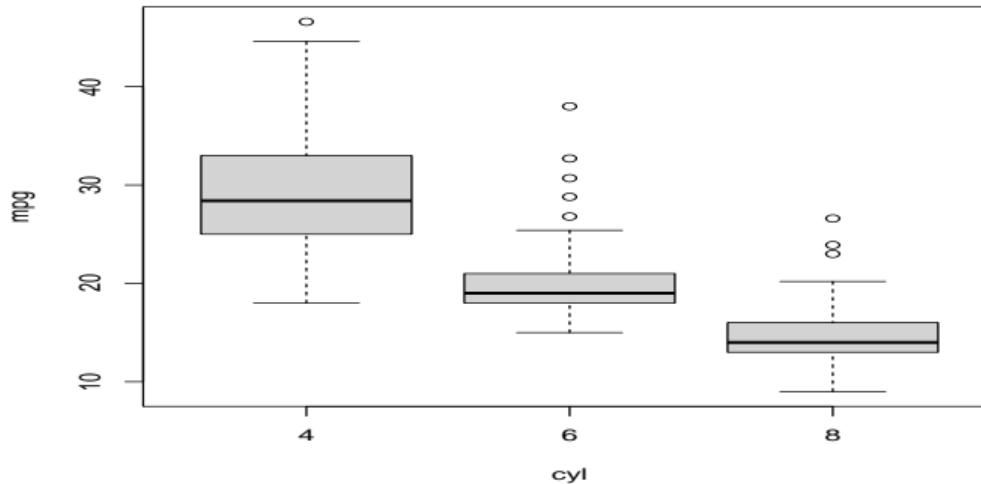
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
cyl	2	15142	7571	343.1	<2e-16 ***
Residuals	380	8384	22		

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Linear models repurposed VI

- Visually:

```
boxplot(mpg ~ cyl, data = autompg)
```



Linear models repurposed VII

- Notice that equal variances are assumed - this might be not an obvious assumption in some situations.

Model checking⁹

⁹Material in these slides was heavily influenced by David Dalpiaz *Applied Statistics with R!*, the book is under active development.

Model assumptions I

- The least square estimate is “optimal” if the relationship between Y and (X_1, \dots, X_p) is approximately linear.
- We have discussed methods to test for significance and for estimating the variability of estimates/predictions which is based on the assumption of iid normal errors.
- This is typically expressed as:

$$Y_i \sim N(\beta_0 + \beta_1 x_{1,i} + \dots + \beta_p x_{p,i}, \sigma) \text{ for each } i,$$

which can be rewritten as

$$\epsilon_i = (Y_i - (\beta_0 + \beta_1 x_{1,i} + \dots + \beta_p x_{p,i})) \sim N(0, \sigma) \text{ for each } i.$$

- If the assumptions are not valid we can not rely on the theory to do inference.

Model assumptions II

- Often, the assumptions of linear regression, are stated as:
Linearity: the response can be written as a linear combination of the predictors. (With noise about this true linear relationship.)
Independence: the errors are independent.
Normality: the distribution of the errors should follow a normal distribution.
Equal Variance: the error variance is the same at any set of predictor values.
- There are a number of statistical tests and graphical approaches to verify the validity of these assumptions.
- Notice that other things can go wrong and make the model not valid: statistical modeling is a craft more than a science.

Residuals-based displays I

- We define the residuals r_i (often indicated also as e_i) which are a sample estimate of the errors ϵ_i :

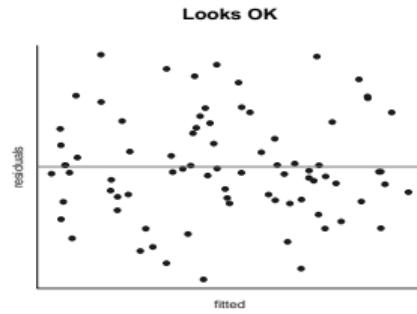
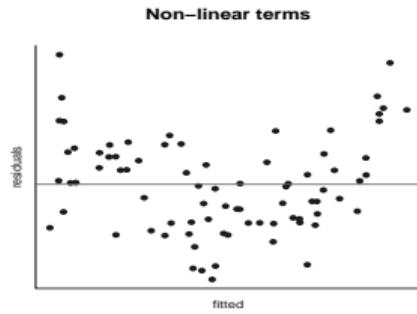
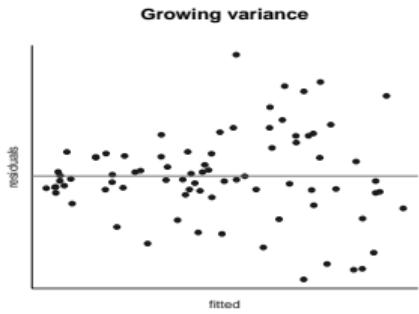
$$r_i = (y_i - \hat{y}_i)$$

- By definition: $\sum r_i/n = 0$ and $\hat{\sigma}^2 = \sum r_i^2/(n - p)$.
- We use the residuals to verify whether assumptions are met (like for simple linear models, but now it's harder to separate out the effect of each variable on the quality of the fit)
- We use residual plots to check both the linearity and constant variance assumptions.
- We use the qqplot of residuals to verify the normality assumption.
- We can use residuals to verify the independence assumption - but we should also control for this when designing the data collection.

Residuals-based displays II

- If the model is well specified, the sample of (r_1, \dots, r_n) should be iid normally distributed with a constant variance.
- For each model we fit to a dataset we have a different vector of residuals r_i .
- First residual plot: plot r_i against \hat{y}_i . There should be no discernible pattern in the data (no systematic under- or over-estimation) and constant variance.
- If the variance of the data is constant there should be a scatter around the mean (which is 0)
- If the linear terms capture the entire relationship between X and Y , r_i there should be no systematic under-or over-estimation for some values of X .
- Looking at the plots can give some indication on the possible solution to the problem.

Residuals-based displays III



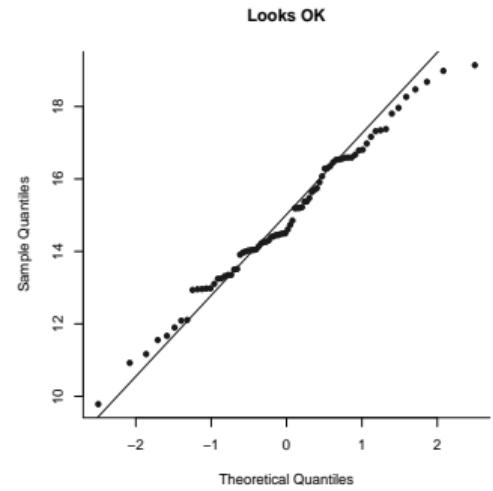
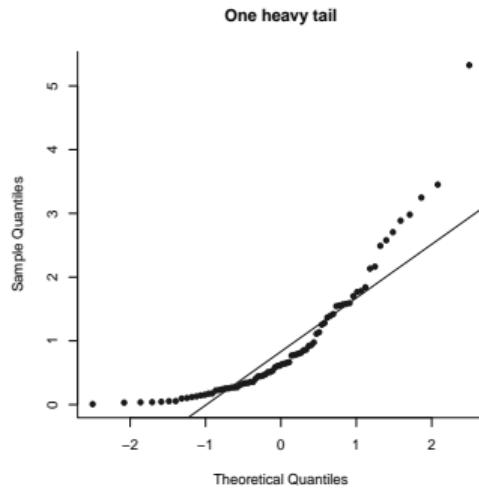
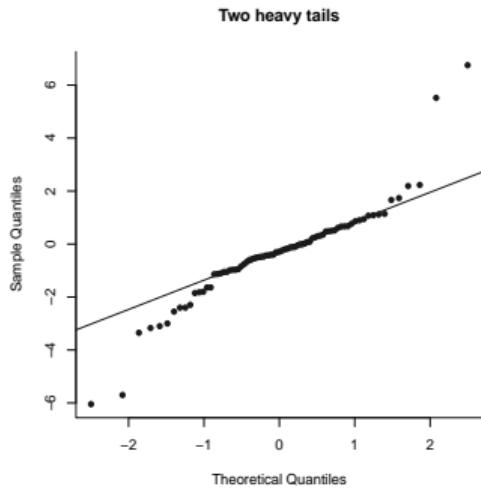
QQplots I

- The residuals should be normally distributed.
- If that was true the empirical cdf/pdf should look like the cdf/pdf of a normal. We compare them.
- Option 1: plot a histogram of residuals.
- Option 2 (a better option): compare empirical and theoretical quantiles via a qqplot.
- Can help identify long tails (excessive variance)
- Sometimes for qqplot and summary statistics it is easier to use the standardized residuals (see `rstandard`)

$$\frac{y_i - \hat{y}_i}{\hat{\sigma}}$$

which should follow a standard normal. This means for example that 95% of the residuals should have values within (-2,2).

QQplots II



QQplots III

- `qqplot` and `qqline` in R
- See also the `qqPlot` function in the `car` library
- Sort the residuals from the smallest $r_{(1)}$ to the largest $r_{(n)}$
- Assign to the observation in position i the empirical cdf value, for example $(i - 0.5)/n$.
- Compare the value of $r_{(i)}$ to the theoretical value of a normal sample of size n .
- Useful to identify specific points with particularly high residuals.
- Useful to identify deviations from normality

Model checking

- If model assumptions are not valid the inference might be dubious.
- But nothing is set in stone: models which deviate from the assumptions can be still be OK.
- Nevertheless if keeping a non-significance variable in the model improves the model-checks it might be a good reason to keep the variable.
- The same hold for transformations discussed below.
- Model building is an iterative process: check the data, fit a model or two, check residuals and iterate.
- Some subjective decisions on what is a “good fit”
- In R `plot(lm.object)` gives some useful plots for model-checking (more than what we discuss).
- There can be other problematic cases which we do not discuss - it is a wide field of research.

Transformations¹⁰

¹⁰Material in these slides was heavily influenced by David Dalpiaz *Applied Statistics with R!*, the book is under active development.

Transformations

All happy families are alike; each unhappy family is unhappy in its own way. (Leo Tolstoy)

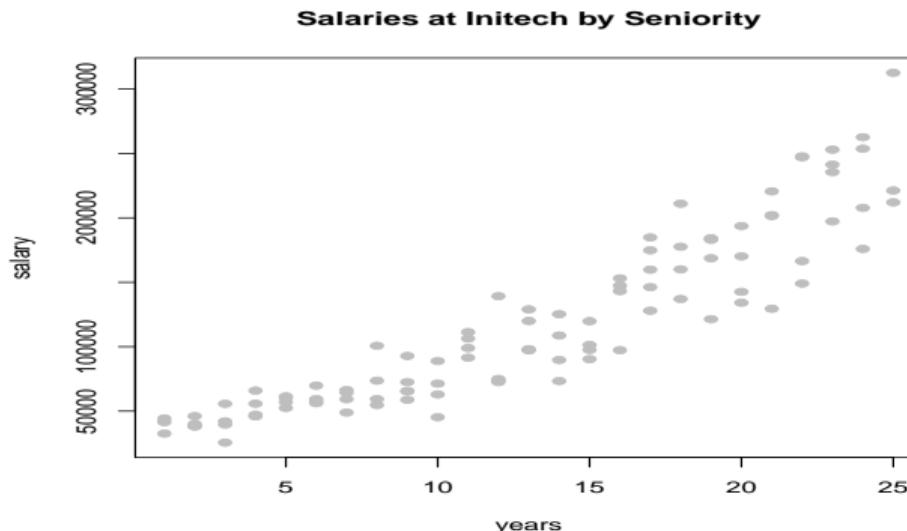
Models can be problematic in many different ways...

- Model checks can highlight issues with linearity and homoschedasticity assumptions (among others)
- Assumption is: **linear** relationship between X and Y
- Sometimes the relationship between X and Y is strong but not linear: one possible approach is to transform X or Y to make the relationship linear
- Assumption is: **constant variance**
- Transform Y to make the variance more constant

Response Transformation I

- Some fictional salary data from the fictional company Initech. We will try to model salary as a function of years of experience.

```
initech <- read.csv("initech.csv")
```



Response Transformation II

- We first fit a simple linear model.

```
initech_fit <- lm(salary ~ years, data = initech)
summary(initech_fit)
```

Call:

```
lm(formula = salary ~ years, data = initech)
```

Residuals:

Min	1Q	Median	3Q	Max
-57225	-18104	241	15589	91332

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5302	5750	0.922	0.359
years	8637	389	22.200	<2e-16

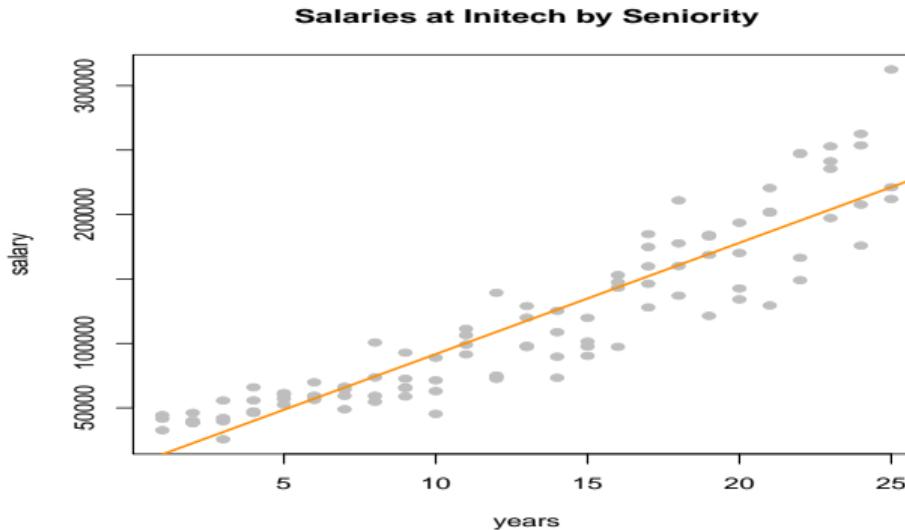
Residual standard error: 27360 on 98 degrees of freedom

Multiple R-squared: 0.8341, Adjusted R-squared: 0.8324

F-statistic: 492.8 on 1 and 98 DF, p-value: < 2.2e-16

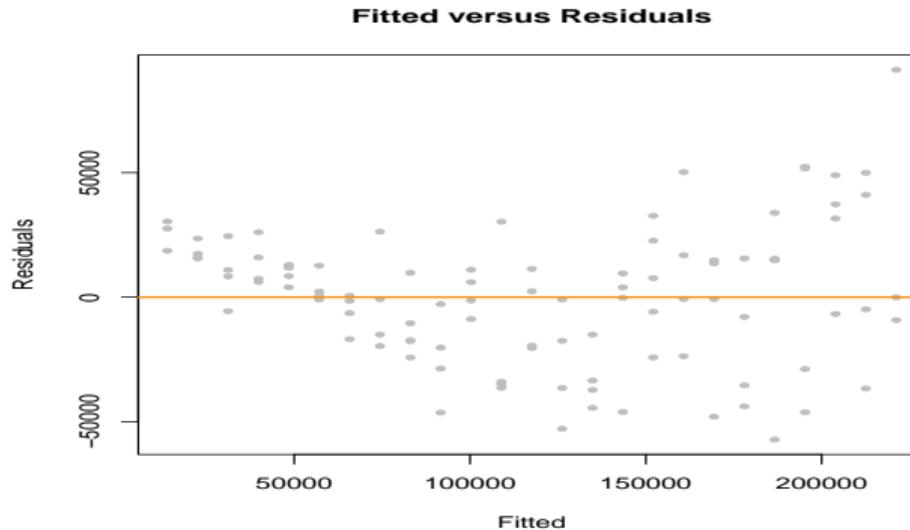
Response Transformation III

- Does it meet the model assumptions?



Looks like a decent fit.

Response Transformation IV



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

Variance stabilizing transformations I

- Under usual assumptions

$$\text{Var}[Y|X = x] = \sigma^2$$

which is a constant value for any value of x .

- However, here we see that the variance is a function of the mean,

$$\text{Var}[Y | X = x] = h(\text{E}[Y | X = x]).$$

In this case, h is some increasing function.

Variance stabilizing transformations II

- In order to correct for this, we would like to find some function of Y , $g(Y)$ (**variance stabilizing transformation**) such that,

$$\text{Var}[g(Y) | X = x] = c$$

where c is a constant that does not depend on the mean, $E[Y | X = x]$.

- A common variance stabilizing transformation when we see increasing variance in a fitted versus residuals plot is $\log(Y)$.
- Also, 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.

$$\log(Y_i) = \beta_0 + \beta_1 x_i + \epsilon_i.$$

Variance stabilizing transformations III

- In the original scale of the data we have

$$Y_i = \exp(\beta_0 + \beta_1 x_i) \cdot \exp(\epsilon_i)$$

which has the errors entering the model in a multiplicative fashion.

Variance stabilizing transformations IV

- Fitting this model in R requires only a minor modification to our formula specification.

```
initech_fit_log <- lm(log(salary) ~ years, data = initech)
```



Variance stabilizing transformations V

- On the original scale

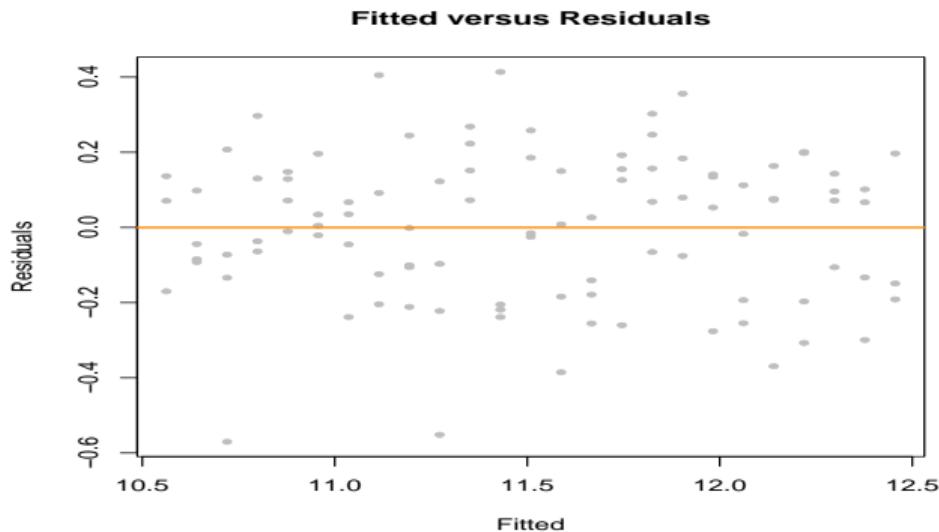
```
plot(salary ~ years, data = initech, col = "grey", pch = 20, cex = 1.5,  
     main = "Salaries at Initech, By Seniority")  
curve(exp(initech_fit_log$coef[1] + initech_fit_log$coef[2] * x),  
      from = 0, to = 30, add = TRUE, col = "darkorange", lwd = 2)
```

Variance stabilizing transformations VI



Variance stabilizing transformations VII

- We check the residuals



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

Variance stabilizing transformations VIII

```
sqrt(mean(resid(initech_fit) ^ 2))  
[1] 27080.16  
  
sqrt(mean(resid(initech_fit_log) ^ 2))  
[1] 0.1934907
```

But wait, that isn't fair, this difference is simply due to the different scales being used.

```
sqrt(mean((initech$salary - fitted(initech_fit)) ^ 2))  
[1] 27080.16  
  
sqrt(mean((initech$salary - exp(fitted(initech_fit_log)))) ^ 2))  
[1] 24280.36
```

- The transformed response is a **linear** combination of the predictors,

$$\log(\hat{y}(x)) = \hat{\beta}_0 + \hat{\beta}_1 x = 10.484 + 0.079x.$$

Variance stabilizing transformations IX

- If we re-scale the data from a log scale back to the original scale of the data, we now have

$$\hat{y}(x) = \exp(\hat{\beta}_0) \exp(\hat{\beta}_1 x) = \exp(10.484) \exp(0.079 x).$$

- Comment: average salary increases $\exp(0.079) = 1.0822$ times for one additional year of experience.
- Comparing the RMSE using the original and transformed response, we also see that the log transformed model simply fits better, with a smaller average squared error.
- Issue: from probability we know that $E[g(X)] \neq g(E[X])$: we are modeling a different variable, back-transforming needs to be done with care. [Notice that instead the $\text{median}(\log(X)) = \log(\text{median}(X))$.]

Box-Cox transform I

Box-Cox transform

$$y_\lambda = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \lambda \neq 0 \\ \log(y) & \lambda = 0 \end{cases}$$

- Transformation is defined for positive data only.
- Possible remedy: add a constant to the data.
- Choose: $\lambda < 1$ for positively skewed data, $\lambda > 1$ for negatively skewed data.
- The λ parameter is chosen by numerically maximizing the log-likelihood,

$$L(\lambda) = -\frac{n}{2} \log(SS_{\text{err}}(\lambda)/n) + (\lambda - 1) \sum_{i=1}^n \log(y_i).$$

where $SS_{\text{err}}(\lambda) = \sum_{i=1}^n (y_{\lambda,i} - \hat{y}_{\lambda,i})^2$

Box-Cox transform II

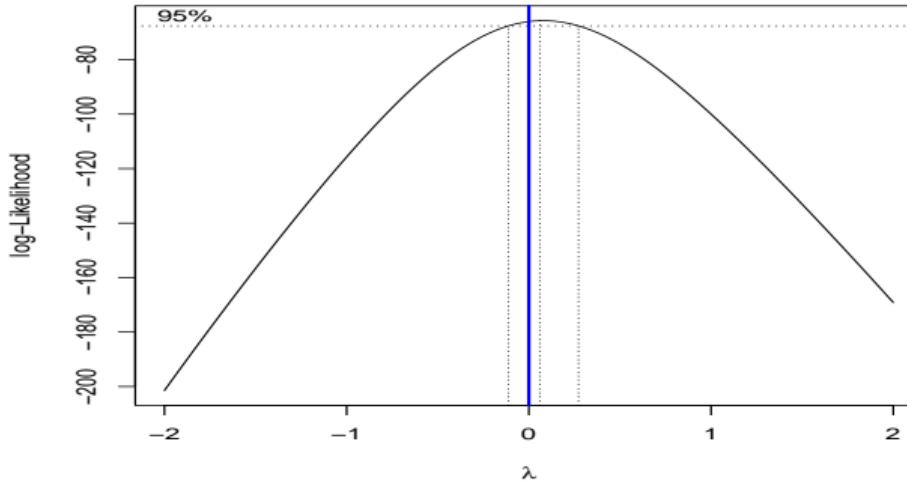
- Procedure is implemented in the package MASS
`library(MASS)`
- We then use the `boxcox` function to find the best transformation of the form considered by the Box-Cox method.
- A $100(1 - \alpha)\%$ confidence interval for λ is,

$$\left\{ \lambda : L(\lambda) > L(\hat{\lambda}) - \frac{1}{2}\chi^2_{1,\alpha} \right\}$$

which R will plot for us to help quickly select an appropriate λ value.

Box-Cox transform III

```
boxcox(initech_fit, plotit = TRUE)
```



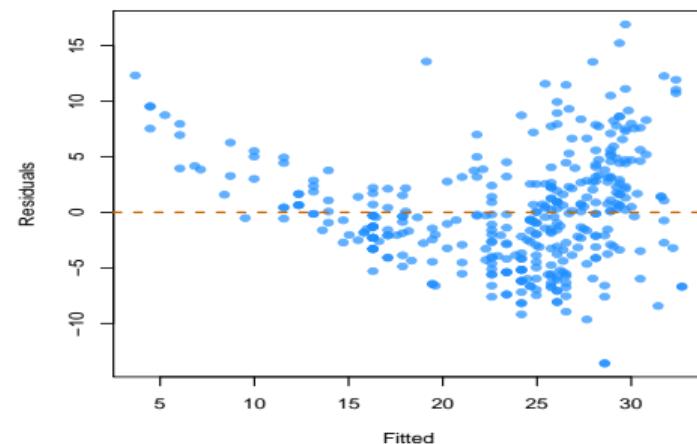
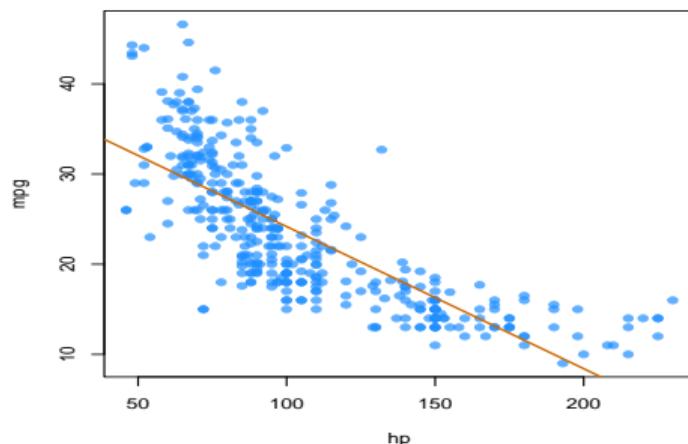
- We often choose a "nice" value from within the confidence interval, instead of the value of λ , $\hat{\lambda} = 0.061$, that truly maximizes the likelihood. In this case we choose $\lambda = 0$.

Transformations of predictor variables. I

- The linear model is termed **linear** not because the regression curve is a plane, but because **the effects of the parameters are linear**.
- The following models can be transformed into simple linear models:
 - $Y = \beta_0 + \beta_1 x^2 + \varepsilon$
 - $Y = \beta_0 + \beta_1 \log(x) + \varepsilon$
 - $Y = \beta_0 + \beta_1(x^3 - \log(|x|) + 2^x) + \varepsilon$
- Rather than working with the sample $(x_1, y_1), \dots, (x_n, y_n)$, we consider the transformed sample $(\tilde{x}_1, y_1), \dots, (\tilde{x}_n, y_n)$ with
 - $\tilde{x}_i = x_i^2, i = 1, \dots, n$.
 - $\tilde{x}_i = \log(x_i), i = 1, \dots, n$.
 - $\tilde{x}_i = x_i^3 - \log(|x_i|) + 2^{x_i}, i = 1, \dots, n$

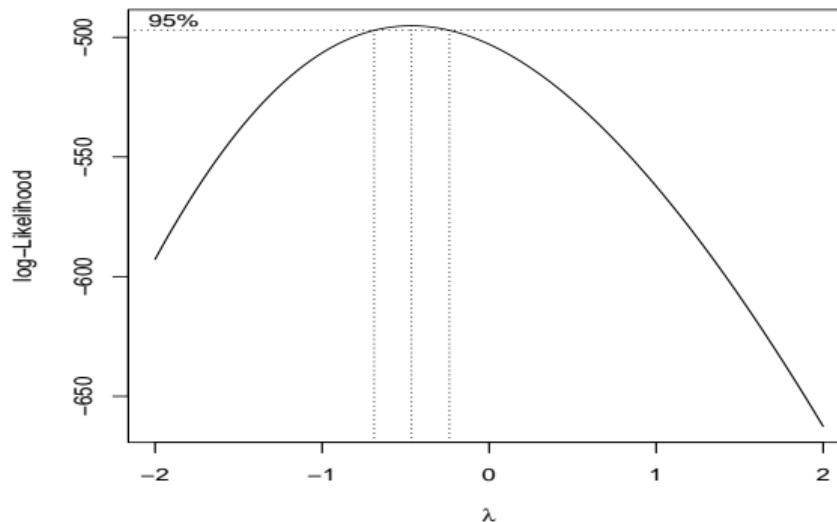
Transformations of predictor variables. II

- Sometimes these transformations can help with violation of model assumptions and other times they can be used to simply fit a more flexible model.
- We will attempt to model `mpg` as a function of `hp`.
- We first attempt a SLR, but we see a rather obvious pattern in the fitted versus residuals plot, which includes increasing variance.

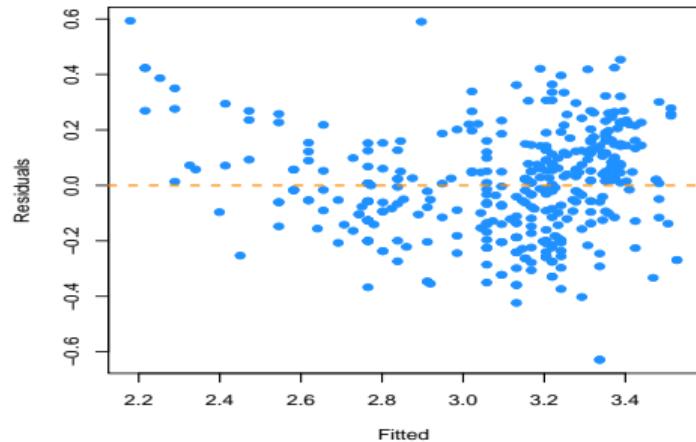
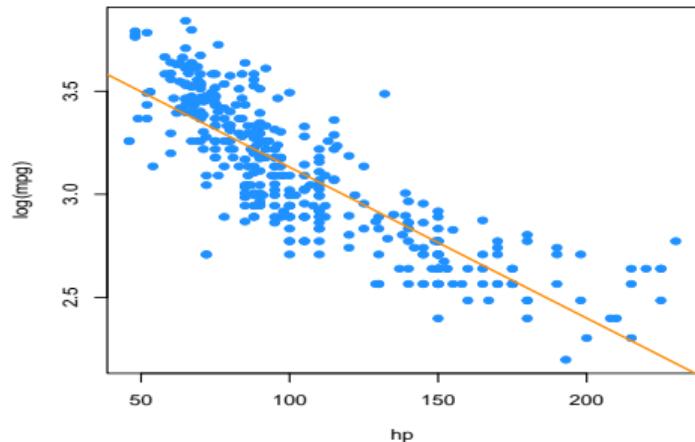


Transformations of predictor variables. III

- We attempt a log transform of the response (rough approximation)

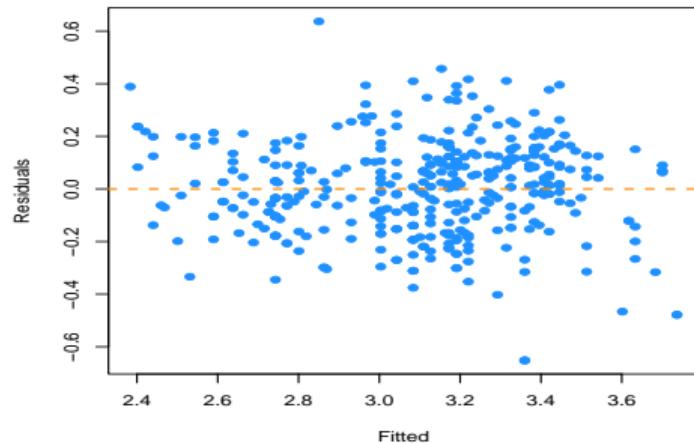
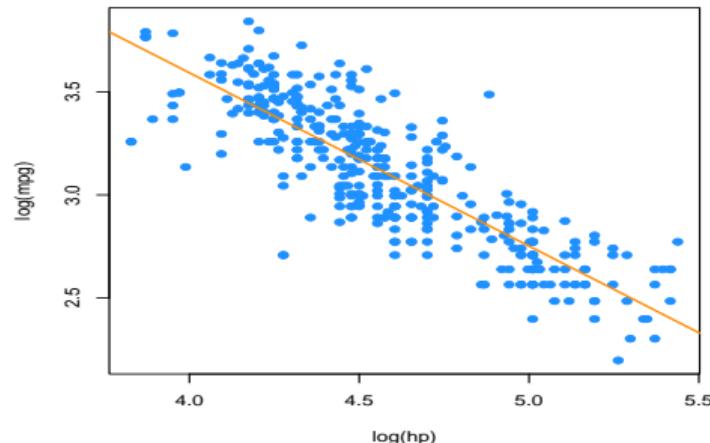


Transformations of predictor variables. IV



Transformations of predictor variables. V

- After performing the log transform of the response, we still have some of the same issues with the fitted versus response. We try also log transforming the predictor.

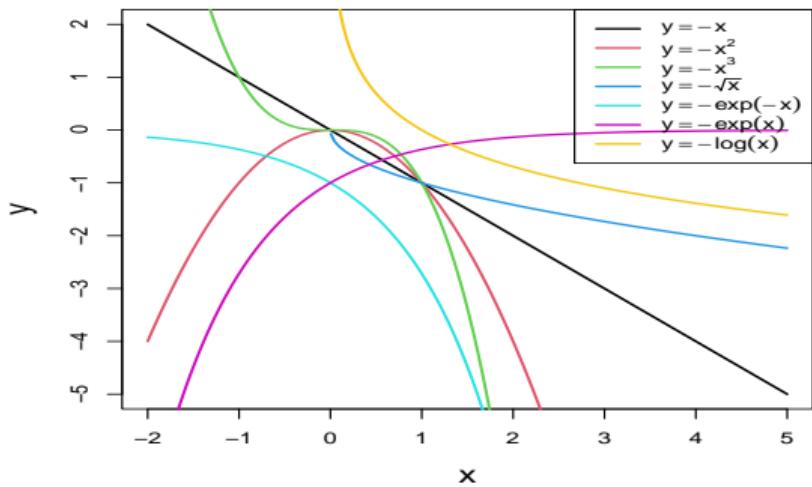
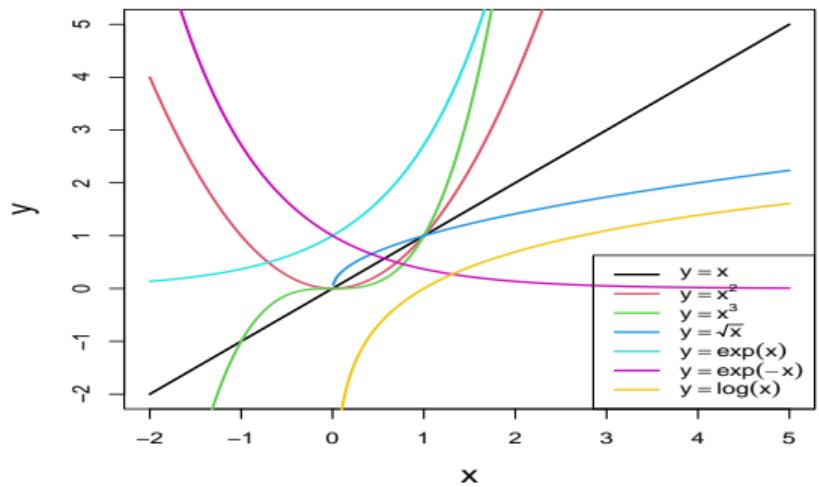


Here, our fitted versus residuals plot looks good.

Tip

- If you apply a nonlinear transformation, namely f , and fit the linear model $Y = \beta_0 + \beta_1 f(x) + \varepsilon$, then there is no point in fit also the model resulting from the negative transformation $-f$. The model with $-f$ is exactly the same as the one with f but with the sign of β_1 flipped!
- As a rule of thumb, use the next figure with the transformations to compare it with the data pattern, then choose the most similar curve, and finally apply the corresponding function with **positive sign**.

Tip



Polynomials I

- A common "transformation" of a predictor variable is the polynomial transformation. Polynomials are very useful as they allow for more flexible models, but do not change the units of the variables.
- Consider the **nonlinear** transformation, namely f ,

$$Y = f(x) + \varepsilon$$

We make no global assumptions about the function f but assume that locally it can be well approximated with a member of a simple class of parametric function, e.g. a constant or straight line.

Polynomials II

- Taylor's theorem says that any continuous function can be approximated with polynomial.

Taylor's theorem

Suppose f is a real function on $[a, b]$, $f^{(K-1)}$ is continuous on $[a, b]$, $f^{(K)}(x)$ is bounded for $x \in (a, b)$ then for any distinct points $x_0 < x_1$ in $[a, b]$ there exist a point \tilde{x} between $x_0 < \tilde{x} < x_1$ such that

$$f(x_1) = f(x_0) + \sum_{k=1}^{K-1} \frac{f^{(k)}(x_0)}{k!} (x_1 - x_0)^k + \frac{f^{(K)}(\tilde{x})}{K!} (x_1 - x_0)^K.$$

Notice: if we view $f(x_0) + \sum_{k=1}^{K-1} \frac{f^{(k)}(x_0)}{k!} (x_1 - x_0)^k$ as function of x_1 , it's a polynomial in the family of polynomials

$$\mathcal{P}_{K+1} = \{f(x) = a_0 + a_1 x + \cdots + a_K x^K, (a_0, \dots, a_K)' \in \mathbb{R}^{K+1}\}.$$

Polynomials III

- We could fit a polynomial of an arbitrary order K ,

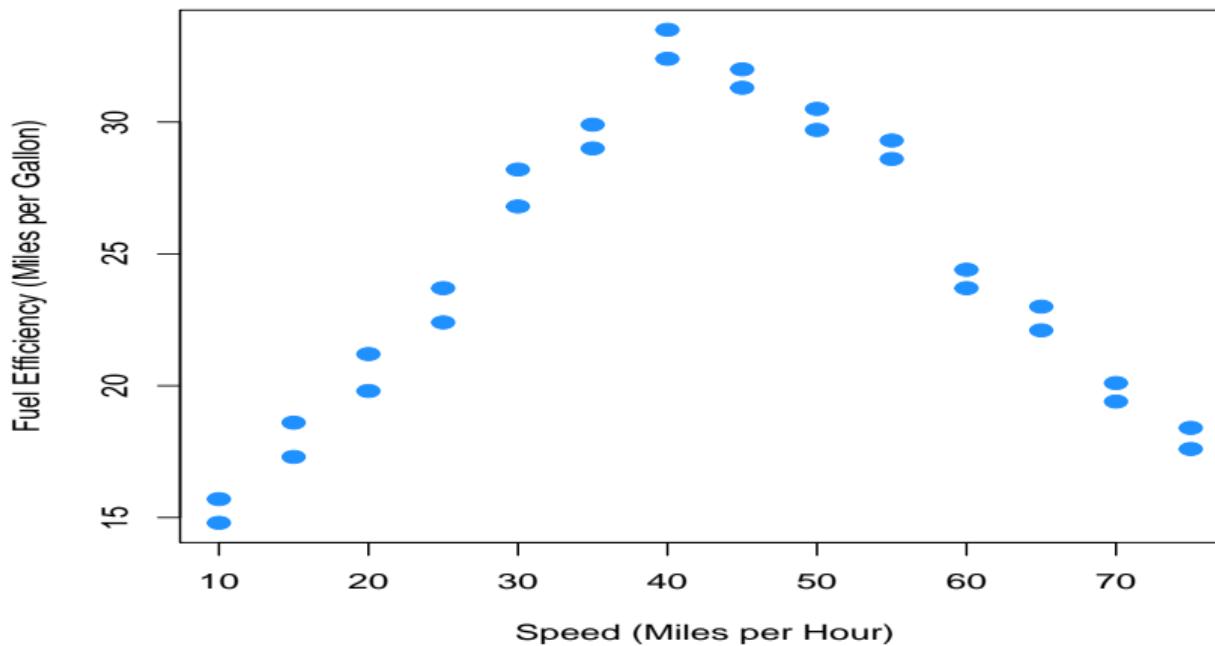
$$Y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_K x_i^K + \epsilon_i$$

and we can think of the polynomial model as the Taylor series expansion of the unknown function.

Polynomials: example I

- Suppose you work for an automobile manufacturer which makes a large luxury sedan. You would like to know how the car performs from a fuel efficiency standpoint when it is driven at various speeds.
- Instead of testing the car at every conceivable speed (which would be impossible) you create an experiment where the car is driven at speeds of interest in increments of 5 miles per hour (**Response surface designs**)
- Our goal then, is to fit a model to this data in order to be able to predict fuel efficiency when driving at certain speeds.

Polynomials: example II



Polynomials: example III

We first fit a simple linear regression to this data.

```
econ <- read.csv("data/fuel-econ.csv")
fit1 <- lm(mpg ~ mph, data = econ)
summary(fit1)
```

Call:

```
lm(formula = mpg ~ mph, data = econ)
```

Residuals:

Min	1Q	Median	3Q	Max
-8.337	-4.895	-1.007	4.914	9.191

Coefficients:

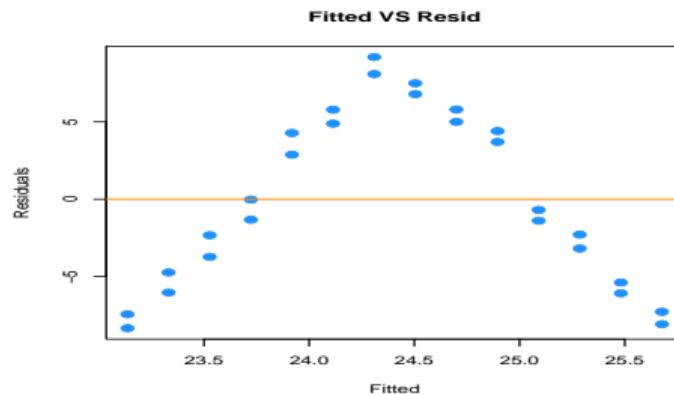
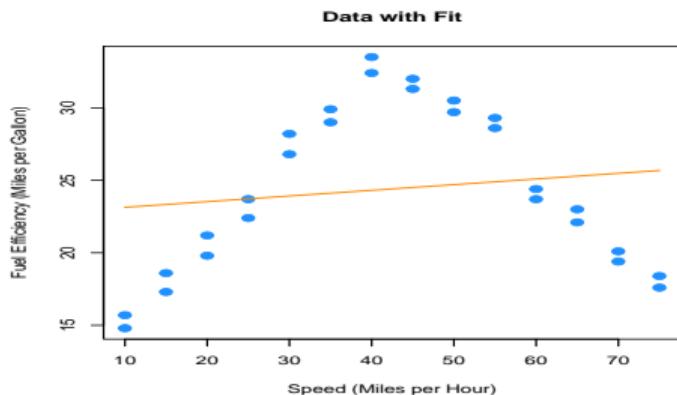
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	22.74637	2.49877	9.103	1.45e-09
mph	0.03908	0.05312	0.736	0.469

Residual standard error: 5.666 on 26 degrees of freedom

Multiple R-squared: 0.02039, Adjusted R-squared: -0.01729

F-statistic: 0.5411 on 1 and 26 DF, p-value: 0.4686

Polynomials: example IV



- Pretty clearly we can do better. Yes fuel efficiency does increase as speed increases, but only up to a certain point.
- We will now add polynomial terms until we fit a suitable fit.
- To add the second order term we need to use the `I()` function in the model specification.
- `I()` is the identity function, which tells R “leave this alone”.

Polynomials: example V

```
fit2 <- lm(mpg ~ mph + I(mph ^ 2), data = econ)
summary(fit2)
```

Call:

```
lm(formula = mpg ~ mph + I(mph^2), data = econ)
```

Residuals:

Min	1Q	Median	3Q	Max
-2.8411	-0.9694	0.0017	1.0181	3.3900

Coefficients:

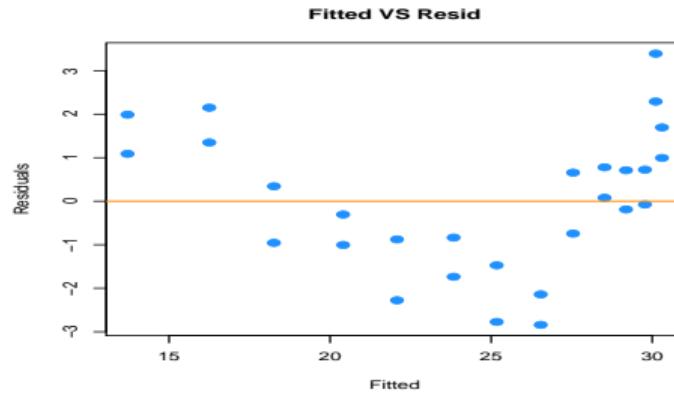
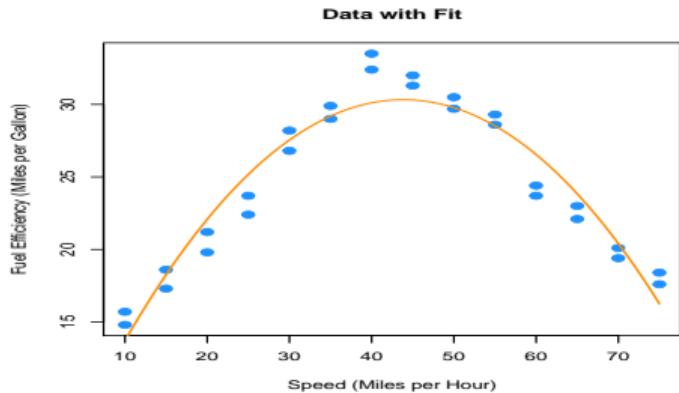
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.4444505	1.4241091	1.716	0.0984
mph	1.2716937	0.0757321	16.792	3.99e-15
I(mph^2)	-0.0145014	0.0008719	-16.633	4.97e-15

Residual standard error: 1.663 on 25 degrees of freedom

Multiple R-squared: 0.9188, Adjusted R-squared: 0.9123

F-statistic: 141.5 on 2 and 25 DF, p-value: 2.338e-14

Polynomials: example VI



- While this model clearly fits much better, and the second order term is significant, we still see a pattern in the fitted versus residuals plot which suggests higher order terms will help. Also, we would expect the curve to flatten as speed increases or decreases, not go sharply downward as we see here.

Polynomials: example VII

```
fit3 <- lm(mpg ~ mph + I(mph ^ 2) + I(mph ^ 3), data = econ)
summary(fit3)
```

Call:

```
lm(formula = mpg ~ mph + I(mph^2) + I(mph^3), data = econ)
```

Residuals:

Min	1Q	Median	3Q	Max
-2.8112	-0.9677	0.0264	1.0345	3.3827

Coefficients:

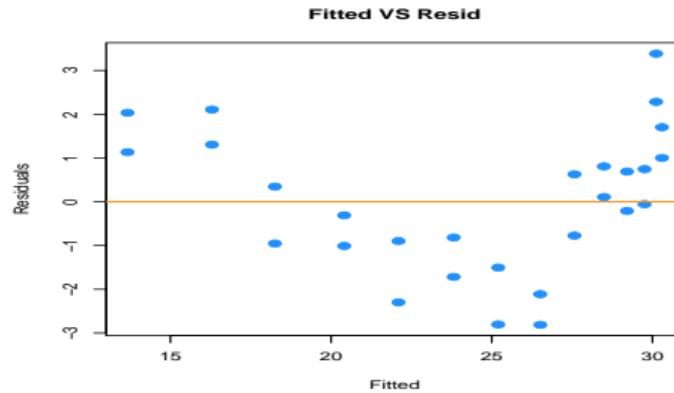
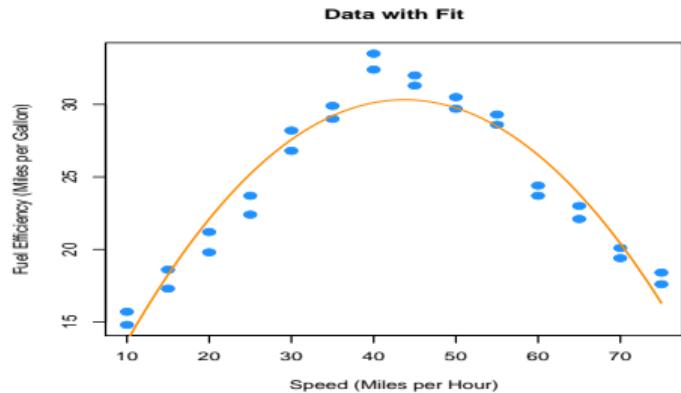
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.258e+00	2.768e+00	0.816	0.4227
mph	1.291e+00	2.529e-01	5.103	3.2e-05
I(mph^2)	-1.502e-02	6.604e-03	-2.274	0.0322
I(mph^3)	4.066e-06	5.132e-05	0.079	0.9375

Residual standard error: 1.697 on 24 degrees of freedom

Multiple R-squared: 0.9188, Adjusted R-squared: 0.9087

F-statistic: 90.56 on 3 and 24 DF, p-value: 3.17e-13

Polynomials: example VIII



- Adding the third order term doesn't seem to help at all. The fitted curve hardly changes. This makes sense, since what we would like is for the curve to flatten at the extremes.
- For this we will need an **even** degree polynomial term.

Polynomials: example IX

Call:

```
lm(formula = mpg ~ mph + I(mph^2) + I(mph^3) + I(mph^4), data = econ)
```

Residuals:

Min	1Q	Median	3Q	Max
-1.57410	-0.60308	0.04236	0.74481	1.93038

Coefficients:

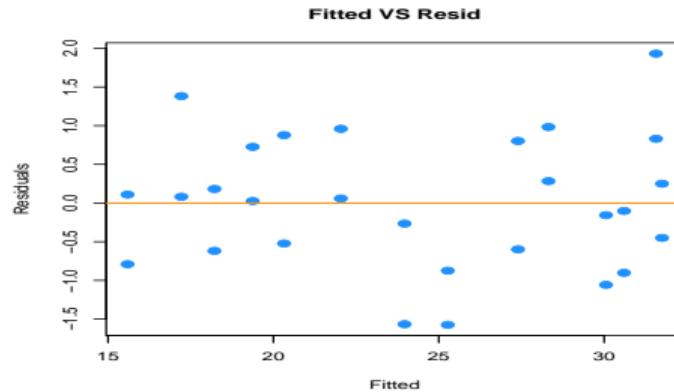
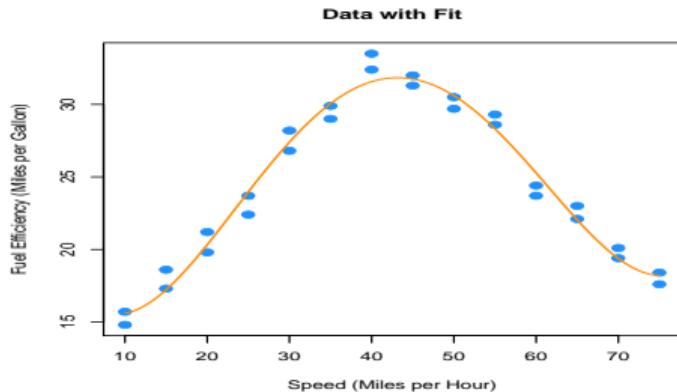
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.146e+01	2.965e+00	7.238	2.28e-07
mph	-1.468e+00	3.913e-01	-3.751	0.00104
I(mph^2)	1.081e-01	1.673e-02	6.463	1.35e-06
I(mph^3)	-2.130e-03	2.844e-04	-7.488	1.31e-07
I(mph^4)	1.255e-05	1.665e-06	7.539	1.17e-07

Residual standard error: 0.9307 on 23 degrees of freedom

Multiple R-squared: 0.9766, Adjusted R-squared: 0.9726

F-statistic: 240.2 on 4 and 23 DF, p-value: < 2.2e-16

Polynomials: example X



- The fourth order term is significant with the other terms in the model. Also we are starting to see what we expected for low and high speed. However, there still seems to be a bit of a pattern in the residuals, so we will again try more higher order terms.
- We will add the fifth and sixth together, since adding the fifth will be similar to adding the third.

Polynomials: example XI

Call:

```
lm(formula = mpg ~ mph + I(mph^2) + I(mph^3) + I(mph^4) + I(mph^5) +
  I(mph^6), data = econ)
```

Residuals:

Min	1Q	Median	3Q	Max
-1.1129	-0.5717	-0.1707	0.5026	1.5288

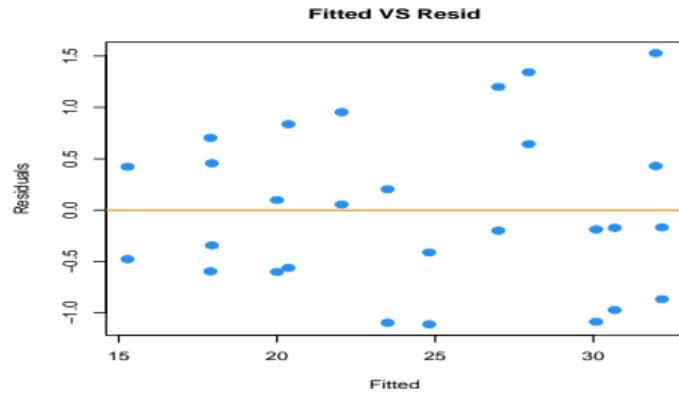
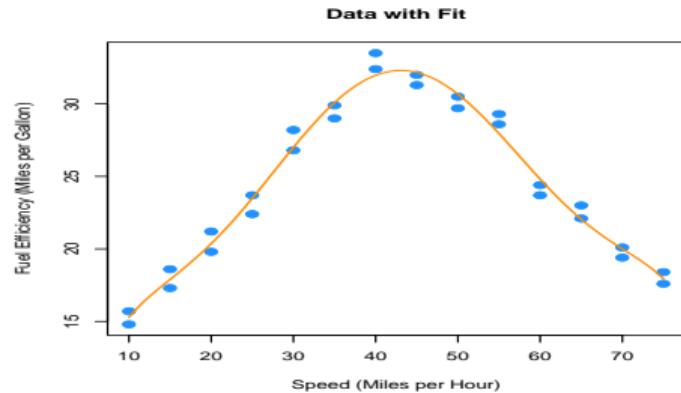
Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-4.206e+00	1.204e+01	-0.349	0.7304
mph	4.203e+00	2.553e+00	1.646	0.1146
I(mph^2)	-3.521e-01	2.012e-01	-1.750	0.0947
I(mph^3)	1.579e-02	7.691e-03	2.053	0.0527
I(mph^4)	-3.473e-04	1.529e-04	-2.271	0.0338
I(mph^5)	3.585e-06	1.518e-06	2.362	0.0279
I(mph^6)	-1.402e-08	5.941e-09	-2.360	0.0280

Residual standard error: 0.8657 on 21 degrees of freedom

Polynomials: example XII

Multiple R-squared: 0.9815, Adjusted R-squared: 0.9762
F-statistic: 186 on 6 and 21 DF, p-value: < 2.2e-16



Again the sixth order term is significant with the other terms in the model and here we see less pattern in the residuals plot.

Polynomials: example XIII

- Let's now test for which of the previous two models we prefer. We will test

$$H_0 : \beta_5 = \beta_6 = 0.$$

Analysis of Variance Table

Model 1: mpg ~ mph + I(mph^2) + I(mph^3) + I(mph^4)

Model 2: mpg ~ mph + I(mph^2) + I(mph^3) + I(mph^4) + I(mph^5) + I(mph^6)

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	23	19.922			
2	21	15.739	2	4.1828	2.7905 0.0842

- This test does not reject the null hypothesis at a level of significance of $\alpha = 0.05$, however the p-value is still rather small, and the fitted versus residuals plot is much better for the model with the sixth order term. This makes the sixth order model a good choice.

Polynomials: example XIV

- We could repeat this process one more time.

Analysis of Variance Table

```
Model 1: mpg ~ mph + I(mph^2) + I(mph^3) + I(mph^4) + I(mph^5) + I(mph^6)
Model 2: mpg ~ mph + I(mph^2) + I(mph^3) + I(mph^4) + I(mph^5) + I(mph^6) +
           I(mph^7) + I(mph^8)
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	21	15.739			
2	19	15.506	2	0.2324	0.1424

The eighth order term is not significant with the other terms in the model and the F-test does not reject.

- There is a quicker way to specify a model with many higher order terms. The method produces the same fitted values ...

```
fit6_alt <- lm(mpg ~ poly(mph, 6), data = econ)
all.equal(fitted(fit6), fitted(fit6_alt))
[1] TRUE
```

Polynomials: example XV

... but the estimated coefficients are different.

```
coef(fit6)
(Intercept) mph I(mph^2) I(mph^3) I(mph^4) I(mph^5) I(mph^6)
-4.206224e+00 4.203382e+00 -3.521452e-01 1.579340e-02 -3.472665e-04 3.585201e-06 -1.401995e-08

coef(fit6_alt)
(Intercept) poly(mph, 6)1 poly(mph, 6)2 poly(mph, 6)3 poly(mph, 6)4 poly(mph, 6)5 poly(mph, 6)6
24.40714286 4.16769628 -27.66685755 0.13446747 7.01671480 0.09288754 -2.04307796
```

- This is because `poly()` uses **orthogonal polynomials**.

Polynomials: example XVI

- To use `poly()` to obtain the same results as using `I()` repeatedly, we would need to set `raw = TRUE`.

```
fit6_alt2 <- lm(mpg ~ poly(mph, 6, raw = TRUE), data = econ)
coef(fit6_alt2)

(Intercept) poly(mph, 6, raw = TRUE)1 poly(mph, 6, raw = TRUE)2 poly(mph, 6, raw = TRUE)3
-4.206224e+00 4.203382e+00 -3.521452e-01 1.579340e+00
poly(mph, 6, raw = TRUE)4 poly(mph, 6, raw = TRUE)5 poly(mph, 6, raw = TRUE)6
-3.472665e-04 3.585201e-06 -1.401995e-08
```

Melting artic I

- Melting Arctic sea ice is monitored and used as an indicator for the impacts of climate change.
- The summer ice in the Arctic Ocean reflects sunlight. As the ice melts, the much darker sea water absorbs sunlight. This feedback mechanism is understood as an important driver of climate change throughout geologic history.
- Other effects : changes in ocean currents and atmospheric weather patterns as well as the possibility of releasing further greenhouse gases by accelerating the melting of Arctic permafrost on land and on the East Siberian Arctic Shelf.

Melting artic II

- September is the month when the ice stops melting each summer and reaches its minimum extent.

Melting artic III

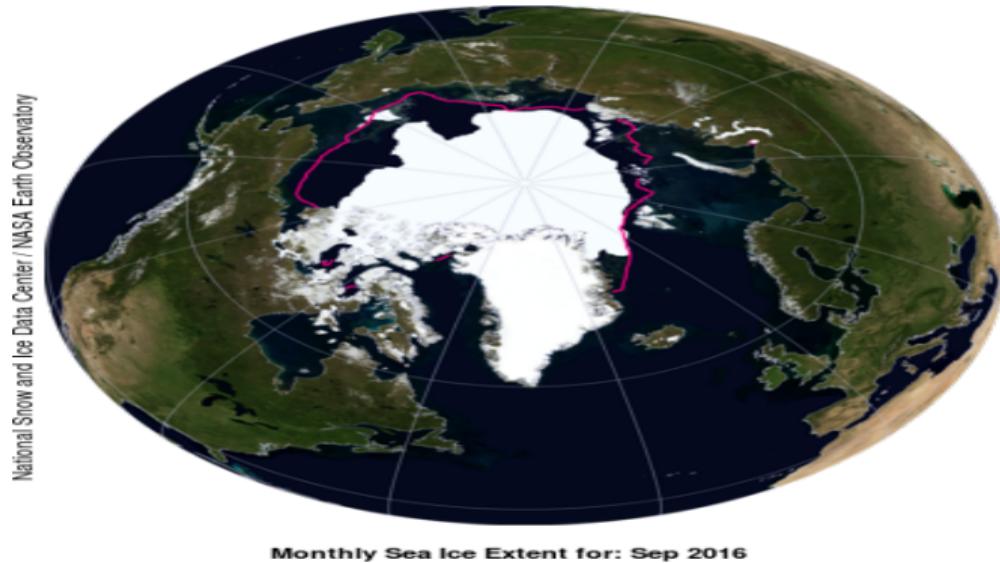
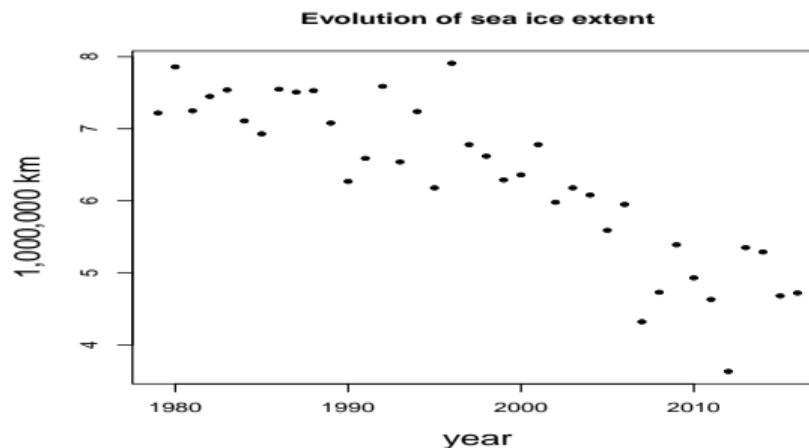


Figure: Source: National Snow and Ice Data Center

Melting artic IV

- The data we will analyze is a time series of September Arctic sea ice extent from 1979 until 2012.



A (simple) possible model I

- Scientific question: "*Is the September Arctic sea ice extent decreasing with time ?*"

$$\text{Extent} = f(\text{Time})$$

- Simple model: $f(\cdot)$ is **linear**

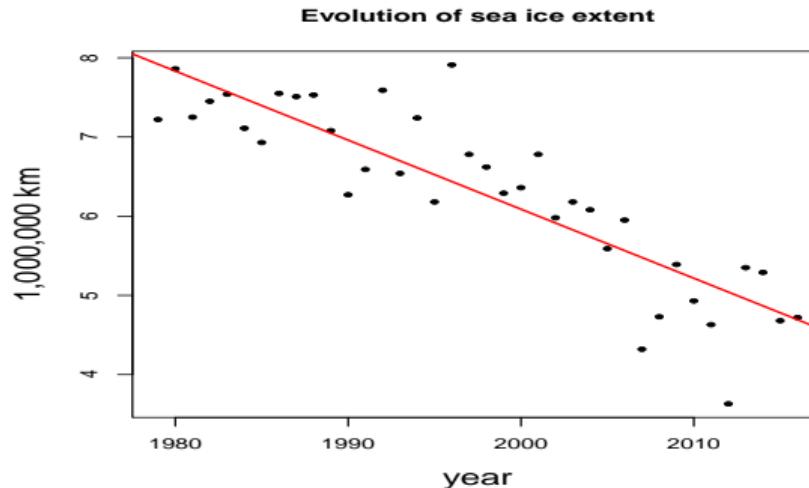
$$\text{Extent} = \beta_0 + \beta_1 \text{Time}, \quad \beta_1 < 0$$

- Linear regression model: n observations y_1, y_2, \dots, y_n from

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad (i = 1, \dots, n)$$

A (simple) possible model II

```
plot(year,extent,ylab="1,000,000 km",
     main="Evolution of sea ice extent",pch=20,cex.lab=1.5)
fit<-lm(extent~year)
abline(fit,lwd=2,col="red")
```

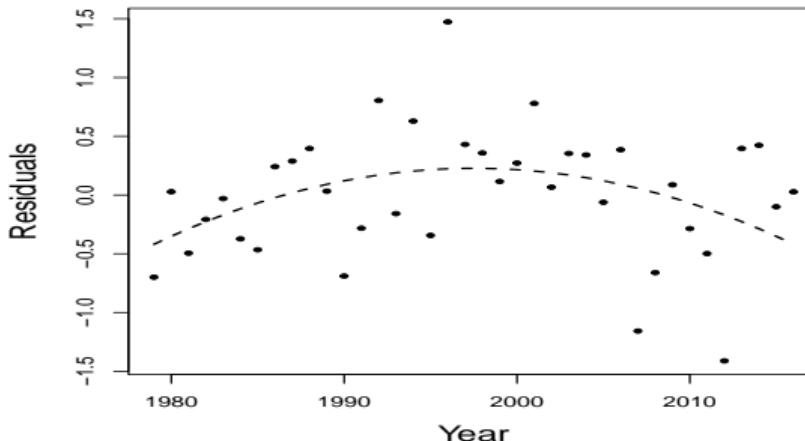


A (simple) possible model III

- Fit is not too-bad, except for a few points...

A (simple) possible model IV

- The plot of residuals versus the time that reinforces this point.



- The residuals from the linear regression tend to be negative in early and late years, and positive in the middle years.

A (simple) possible model V

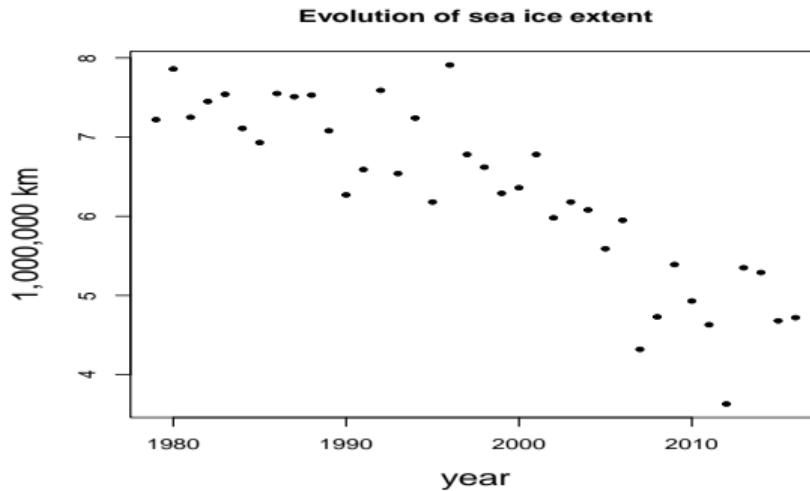
- Our previous model postulates ($x_i = t_i$)

$$\mathbb{E}[Y_i|t_i] = \beta_0 + \beta_1 t_i$$

- Look back at the data: the decrease is larger in the later years

```
plot(year,extent,ylab="1,000,000 km",
     main="Evolution of sea ice extent",pch=20,cex.lab=1.5)
```

A (simple) possible model VI



A (simple) possible model VII

- The next table 1 shows the estimated slope from analogous regressions using data from 1979 until the end of September for each of the last sixteen years.
- Notice that the slope steepens

starting.year	final.year	slope
1979	2001	-0.043
1979	2002	-0.049
1979	2003	-0.051
1979	2004	-0.053
1979	2005	-0.058
1979	2006	-0.059
1979	2007	-0.070
1979	2008	-0.077
1979	2009	-0.078
1979	2010	-0.080
1979	2011	-0.084
1979	2012	-0.091
1979	2013	-0.089
1979	2014	-0.087
1979	2015	-0.087
1979	2016	-0.087

A (simple) possible model VIII

- Suppose that β_1 is changing with respect to time i.e. $\beta_1(t)$
- A simple model that allows for the slope to change at a constant rate.

$$\begin{aligned}\mathbb{E}[Y_i|t_i] &= \beta_0 + \beta_1 t_i + \beta_2 t_i^2 \\ &= \beta_0 + \underbrace{\beta_1 \left[1 + \frac{\beta_2}{\beta_1} t_i \right]}_{\beta_1(t_i)} t_i \\ &= \beta_0 + \beta_1(t_i) t_i\end{aligned}$$

- A model with a quadratic term corresponds to a model in which the slope is allowed to change as a function of the predictor
- In linear model the first derivative is constant, with quadratic terms the first derivative varies.
- Similar concepts apply to higher order polynomials

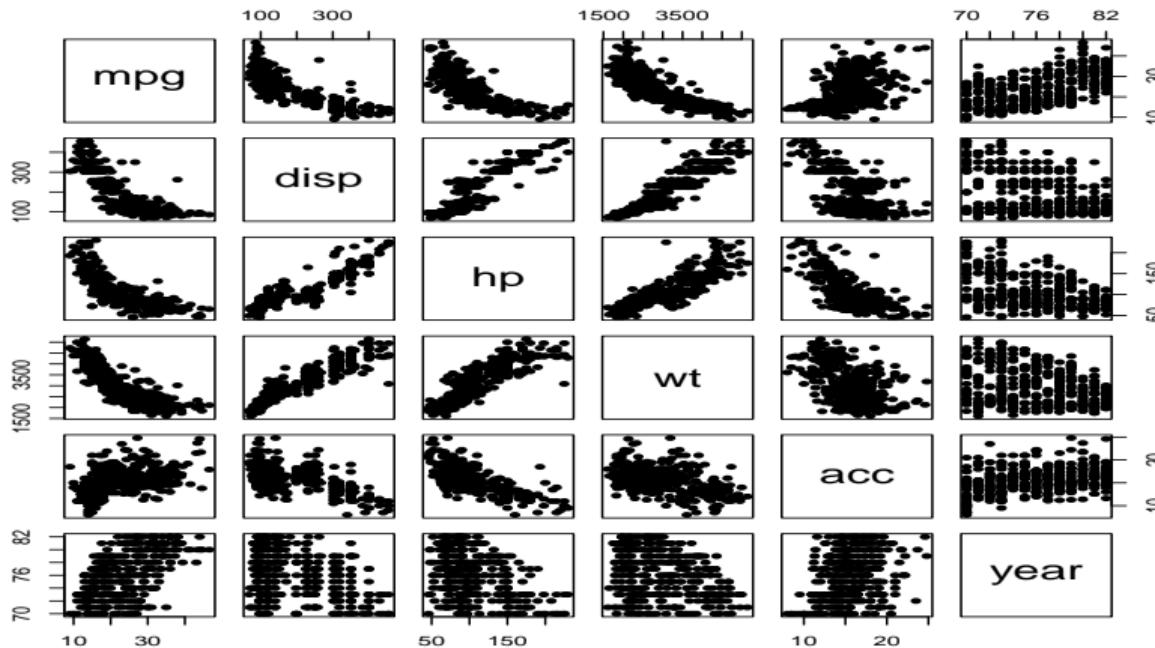
Multicollinearity ¹¹

¹¹Material in these slides was heavily influenced by Cosma Shalizi's notes.

Why collinearity is a problem I

Let's go back to the autompg dataset: many factors affect a car's efficiency:

Why collinearity is a problem II



Why collinearity is a problem III

We could add all variables into the model:

```
fit_big <- lm(mpg ~ ., data = autompg); summary(fit_big)
```

Call:

```
lm(formula = mpg ~ ., data = autompg)
```

Residuals:

Min	1Q	Median	3Q	Max
-8.3378	-2.3682	-0.1095	1.9052	14.2452

Coefficients:

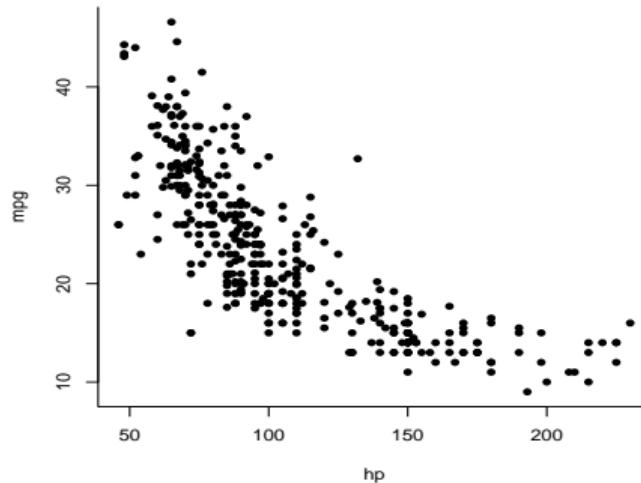
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1.394e+01	4.606e+00	-3.026	0.00265
disp	-1.937e-03	5.544e-03	-0.349	0.72695
hp	6.274e-03	1.352e-02	0.464	0.64278
wt	-6.708e-03	6.619e-04	-10.134	< 2e-16
acc	2.796e-02	1.009e-01	0.277	0.78182
year	7.463e-01	5.189e-02	14.384	< 2e-16

Why collinearity is a problem IV

Residual standard error: 3.353 on 377 degrees of freedom

Multiple R-squared: 0.8198, Adjusted R-squared: 0.8174

F-statistic: 343 on 5 and 377 DF, p-value: < 2.2e-16



Model is significant! But....

β_{hp} is 0.00627416 and not significant.

Why collinearity is a problem V

What is happening?

```
signif(cor(autompg), 4)
```

	mpg	disp	hp	wt	acc	year
mpg	1.0000	-0.8176	-0.7800	-0.8424	0.4224	0.5786
disp	-0.8176	1.0000	0.9027	0.9352	-0.5632	-0.3720
hp	-0.7800	0.9027	1.0000	0.8687	-0.6947	-0.4154
wt	-0.8424	0.9352	0.8687	1.0000	-0.4337	-0.3127
acc	0.4224	-0.5632	-0.6947	-0.4337	1.0000	0.2922
year	0.5786	-0.3720	-0.4154	-0.3127	0.2922	1.0000

Strong relationships to mpg but lots of correlation between variables

Why collinearity is a problem VI

- The estimated coefficients in a multiple linear regression is

$$\hat{\beta} = (X^\top X)^{-1} X^\top y$$

provide that $X^\top X$ is invertible.

- Similarly, the variance of the estimates,

$$\text{Var} [\hat{\beta}] = \sigma^2 (X^\top X)^{-1}$$

will blow up when $X^\top X$ is singular.

- If that matrix isn't exactly singular, but is close to being non-invertible, the estimation will become unstable and the variances will become huge.
- The estimation becomes unreliable and very variable
- (Might not be a big problem if all we care about is prediction)

Why collinearity is a problem VII

- It looks like we have p different predictor variables (**Important**: I've included as predictor variable the column defining the constant term!), but really some of them are linear combinations of the others, so they don't add any information.
- The real number of distinct variables is $q < p$, the column rank of X .
- If the exact linear relationship holds among more than two variables, we talk about **multicollinearity**.
- **Collinearity** can refer either to the general situation of a linear dependence among the predictors, or, by contrast to multicollinearity, a linear relationship among just two of the predictors.

Dealing with collinearity by deleting predictor variables

- Since not all of the p predictor variables are actually contributing information, a natural way of dealing with collinearity is to drop some predictor variables from the model.
- If you want to do this, you should think very carefully about *which* predictor variable to delete.
- Ideally you would use knowledge on the measuring processes and about the process under study

Diagnosing collinearity among pairs of predictor variables I

- Easy: we make the pairs plot of all the predictor variables, and we see if any of them fall on a straight line, or close to one.
- If the number of predictor variables *is* huge, look at the correlation matrix, and worry about any entry off the diagonal which is (nearly) ± 1 .
- But a multicollinear relationship involving three or more predictor variables might be totally invisible on a pairs plot.
- We use some checks and metrics which can help identify issues

Diagnosing collinearity among pairs of predictor variables II

- Red flag 1: Large changes in estimated coefficients when one other predictor variable is included or removed
- Red flag 2: non-significant results in individual tests on β_j for variables X_j which appear to be important when taken individually
- Red flag 3: estimated value of β_j with opposite sign from what we see in scatterplot of (X_j, Y) or that we expect from theoretical considerations
- Red flag 4: large sample correlations between X_i s and X_j s.
- Is $(X^\top X)$ (almost) singular: check if any eigenvalues are ≈ 0 ?
- If any eigenvalue is exactly 0 some of the β_j can not be estimated
- A more formal quantity that we can compute is the Variance Inflation Factor (VIF)

Variance inflation factors (VIF) I

- If the predictors were uncorrelated, the variance of $\hat{\beta}_i$ would be

$$\text{Var} [\hat{\beta}_i] = \frac{\sigma^2}{ns_{X_{i+1}}^2}$$

- With correlated predictors we have

$$\text{Var} [\hat{\beta}_i] = \sigma^2 (X^\top X)^{-1}_{i+1,i+1}$$

- The ratio

$$\text{VIF}_i = (X^\top X)^{-1}_{i+1,i+1} * ns_{X_i}^2$$

is the **variance inflation factor** for the i^{th} coefficient,

- The average of the variance inflation factors across all predictors is often noted $\overline{\text{VIF}}$, or just VIF.

Variance inflation factors (VIF) II

- It is possible to show that variance inflation factor for X_i can be found by regressing X_i on all of the other X_j , computing the R^2 of this regression say R_i^2 , and setting

$$\text{VIF}_i = \frac{1}{1 - R_i^2}$$

- Consequence: if $\text{VIF}_i \geq 1$, i.e. the predictors are correlated with each other, the standard errors of the coefficient estimates will be bigger than if the predictors were uncorrelated.
- The variance inflation factor increases as X_i becomes more correlated with some linear combination of the other predictors ($\text{VIF}_i \geq 10 \Leftrightarrow R_i^2 \geq 0.9$)
- Folklore says that $\text{VIF}_i > 10$ indicates “serious” multicollinearity for the predictor.
- Folklore also says that a $\text{VIF} \gg 1$ indicates “serious” multicollinearity.
- These are rules of thumb, not hard boundaries.

Variance inflation factors (VIF) III

For the autompg model we have:

```
car::vif(fit_big)
      disp          hp          wt          acc          year
11.492407  9.357184 10.894270  2.631376  1.240917
eigen(cor(model.matrix(fit_big)[,-1]))$values
[1] 3.43467889 0.80685624 0.62848320 0.07994727 0.05003440
```

We try to drop disp and wt:

```
car::vif(lm(mpg ~ hp + acc + year, data = autompg))
```

```
      hp          acc          year
2.136251  1.932649 1.208566
```

```
eigen(cor(model.matrix(lm(mpg ~ hp + acc + year, data = autompg))[, -1]))$values
[1] 1.9573993 0.7515672 0.2910334
```

Variance inflation factors (VIF) IV

```
summary(lm(mpg ~ hp + acc + year, data = autompg))
```

Call:

```
lm(formula = mpg ~ hp + acc + year, data = autompg)
```

Residuals:

Min	1Q	Median	3Q	Max
-11.7092	-2.8513	-0.6334	2.1798	15.5216

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.006697	5.636870	0.179	0.858
hp	-0.164426	0.008104	-20.289	< 2e-16
acc	-0.662701	0.108492	-6.108	2.5e-09
year	0.657771	0.064254	10.237	< 2e-16

Residual standard error: 4.208 on 379 degrees of freedom

Multiple R-squared: 0.7148, Adjusted R-squared: 0.7125

F-statistic: 316.6 on 3 and 379 DF, p-value: < 2.2e-16

β_{hp} now is negative and significant.

Is multicollinearity a problem? I

Multicollinearity is another instance of the model correctness vs. usefulness.

- A model with multicollinearity might be perfectly valid in the sense of respecting the assumptions of the model. It does not matter whether the predictors are related or not. At least for the verification of the assumptions.
- But the model will be useless if the multicollinearity is high, since it can inflate the variability of the estimation without any kind of bound.

Some more advanced methods do exist to deal with multicollinearity, for example *ridge regression* or *Principle Components Regression*: we do not discuss these in the course.

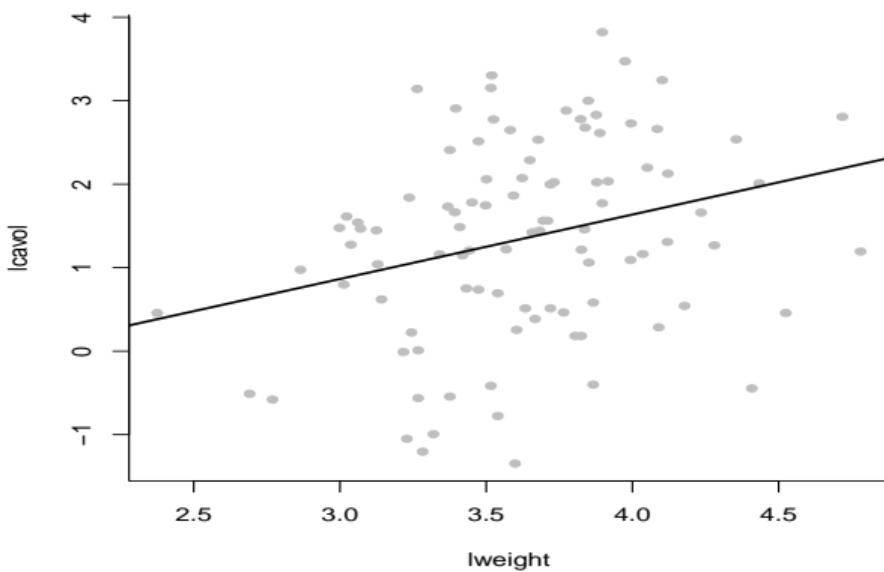
Influential points¹²

¹²Material is based on the Lecture 20 of Cosma Shalizi's course

Influential points I

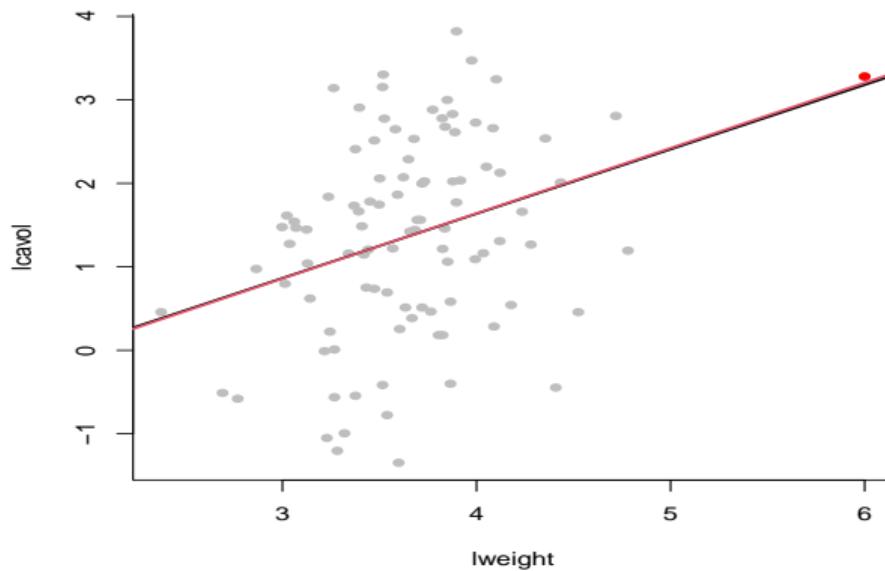
Consider the prostate dataset we have seen in Lab 06: we focus now on the relationship between lcavol and lweight

Influential points II



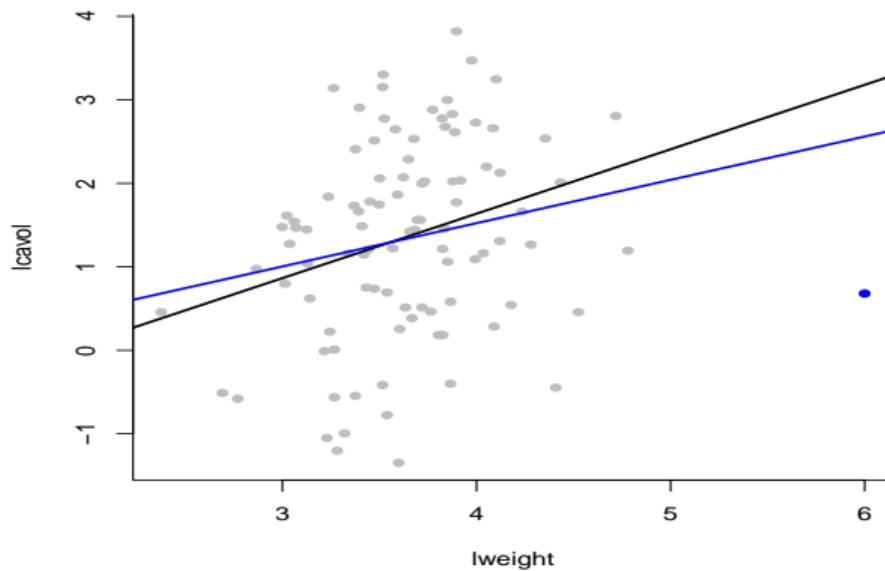
Influential points III

Imagine now we had some additional observations:



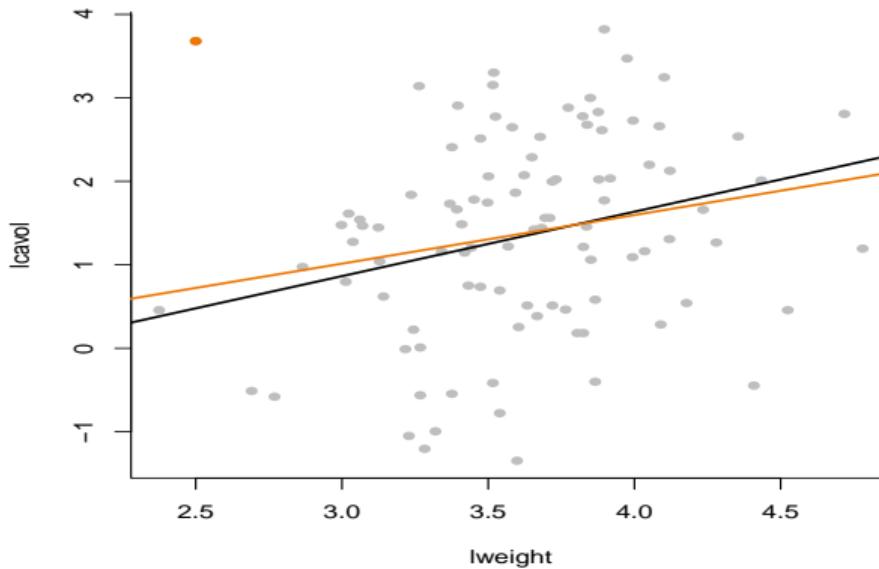
Influential points IV

Imagine now we had some additional observations:



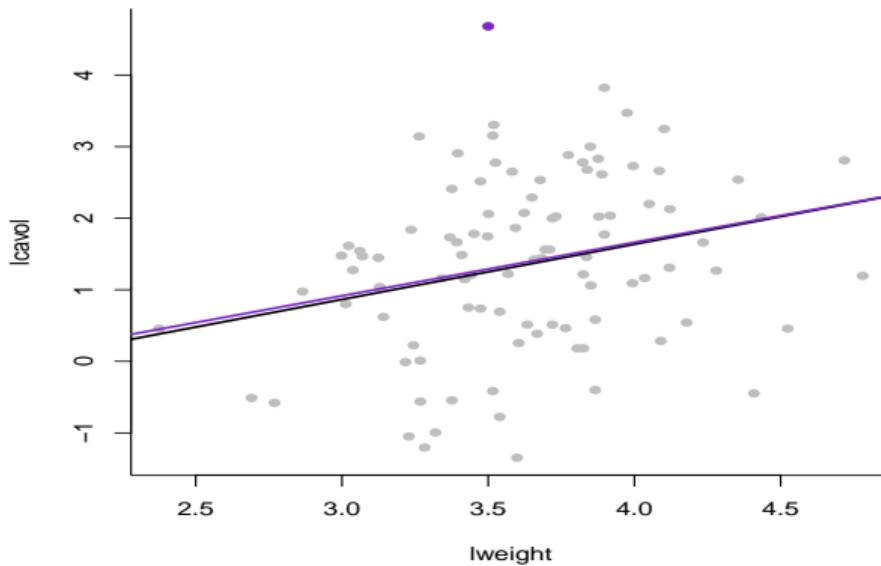
Influential points V

Imagine now we had some additional observations:



Influential points VI

Imagine now we had some additional observations:



Influential points VII

- Some points can be particularly **influential** in the estimation of (β_0, β_1) .
- These are points which break the pattern of the main relationship between X and Y .
- Their values might be not pattern-breaking when considering the x_i value or y_i value, but the combination of the (x_i, y_i) value might be at odds with the rest of the data.
- On the other hand points which have extreme values of X and Y but which fall in line with the rest of the data do not greatly affect the estimation of (β_0, β_1) .
- We want to have a way to quantify if any individual point is unduly affecting our estimation
- Let's see what happens in the simple linear regression case

Influential points VIII

- In SLR we know that

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

- Let us turn this around. The fitted value at $X = \bar{x}$ is

$$\hat{\beta}_0 + \hat{\beta}_1 \bar{x} = \bar{y}$$

- Suppose we had a data point, say the i^{th} point, where $X = \bar{x}$. Then the actual value of y_i almost wouldn't matter for the fitted value there — the regression line *has* to go through \bar{y} at \bar{x} , never mind whether y_i there is close to \bar{y} or far away.
- If $x_i = \bar{x}$, we say that y_i has little *leverage* over \hat{m}_i , or little *influence* on \hat{m}_i .
- It has *some* influence, because y_i is part of what we average to get \bar{y} , but that's not a lot of influence.

Influential points IX

- Moreover, with simple linear regression, we know that

$$\hat{\beta}_1 = \frac{c_{XY}}{s_X^2}$$

- How does y_i show up in this? It's

$$\hat{\beta}_1 = \frac{n^{-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{s_X^2}$$

- Notice that when $x_i = \bar{x}$, y_i doesn't actually matter at all to the slope.
- If x_i is far from \bar{x} , then $y_i - \bar{y}$ will contribute to the slope, and its contribution will get bigger (whether positive or negative) as $x_i - \bar{x}$ grows.
- y_i will also make a big contribution to the slope when $y_i - \bar{y}$ is big (unless, again, $x_i = \bar{x}$).

Influential points X

- Let's write a general formula for the predicted value, at an arbitrary point $X = x$.

$$\begin{aligned}\hat{m}(x) &= \hat{\beta}_0 + \hat{\beta}_1 x \\ &= \bar{y} - \hat{\beta}_1 \bar{x} + \hat{\beta}_1 x \\ &= \bar{y} + \hat{\beta}_1(x - \bar{x}) \\ &= \bar{y} + \frac{1}{n} \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{s_X^2} (x - \bar{x})\end{aligned}$$

So, in words:

- The predicted value is always a weighted average of all the y_i .
- As x_i moves away from \bar{x} , y_i gets more weight (possibly a large negative weight). When $x_i = \bar{x}$, y_i only matters because it contributes to the global mean \bar{y} .
- The weights on all data points increase in magnitude when the point x where we're trying to predict is far from \bar{x} . If $x = \bar{x}$, only \bar{y} matters.

Influential points XI

- All of this is still true of the fitted values at the original data points:
 - If x_i is at \bar{x} , y_i only matters for the fit because it contributes to \bar{y} .
 - As x_i moves away from \bar{x} , in either direction, it makes a bigger contribution to *all* the fitted values.
- Why is this happening? We get the coefficient estimates by minimizing the mean squared error, and the MSE treats all data points equally:

$$\frac{1}{n} \sum_{i=1}^n (y_i - \hat{m}(x_i))^2$$

- But we're not just using any old function $\hat{m}(x)$; we're using a linear function.
- This has only two parameters, so we can't change the predicted value to match each data point — altering the parameters to bring $\hat{m}(x_i)$ closer to y_i might actually increase the error elsewhere.

Influential points XII

- By minimizing the over-all MSE with a linear function, we get two constraints,

$$\bar{y} = \hat{\beta}_0 + \hat{\beta}_1 \bar{x} \quad \text{and} \quad \sum_i e_i(x_i - \bar{x}) = 0$$

- The first of these makes the regression line insensitive to y_i values when x_i is close to \bar{x} .
- The second makes the regression line *very* sensitive to residuals when $(x_i - \bar{x})$ is big. When $(x_i - \bar{x})$ is large, a big residual is harder to balance out than if $(x_i - \bar{x})$ were smaller.
- So, let's sum this up:
 - Least squares estimation tries to bring all the predicted values closer to y_i , but it can't match each data point at once, because the fitted values are all functions of the same coefficients.
 - If x_i is close to \bar{x} , y_i makes little difference to the coefficients or fitted values — they're pinned down by needing to go through the mean of the data.
 - As x_i moves away from \bar{x} , $y_i - \bar{y}$ makes a bigger and bigger impact on both the coefficients and on the fitted values.

Influential points XIII

- Points which don't fall on the same regression line as the others, might throw off our estimate
- This is going to be a concern when x_i is far from \bar{x} , or when the combination of $x_i - \bar{x}$ and $y_i - \bar{y}$ makes that point have a disproportionate impact on the estimates.
- We also worry about points with large residuals, particularly when they correspond to values with large $(x_1 - \bar{x})$ value: the model tries very hard to fit some individual points
- All of this also holds for multiple regression models where things become more complicated because of the high dimensionality (harder to know when x_i is far from \bar{x}) and we need to deal with matrices

Influential points XIV

- In MLR we have seen that

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

from which we get our fitted values as

$$\hat{\mathbf{m}} = \mathbf{x} \hat{\beta} = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} = \mathbf{H} \mathbf{y}$$

with the hat matrix $\mathbf{H} \equiv \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$.

- This leads to a very natural sense in which one observation might be more or less influential than another:

$$\frac{\partial \hat{\beta}_k}{\partial y_i} = \left((\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \right)_{ki} \quad \text{and} \quad \frac{\partial \hat{m}_k}{\partial y_i} = H_{ki}$$

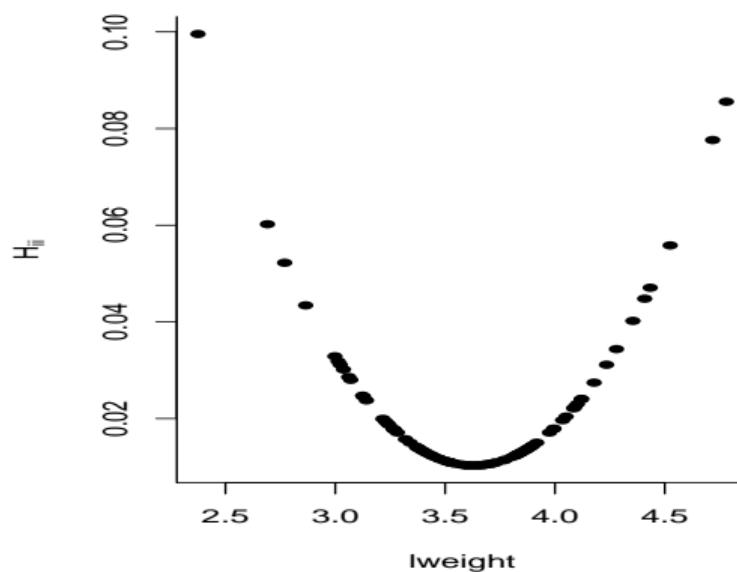
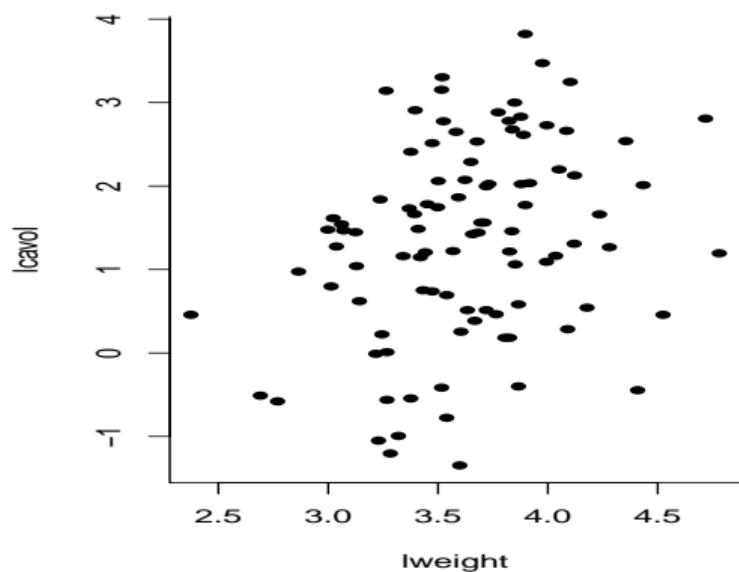
Influential points XV

- If y_i were different, it would change the estimates for all the coefficients and for all the fitted values.
- The rate at which the k^{th} coefficient or fitted value changes is given by the ki^{th} entry in these matrices
- Notice that these matrices are completely defined by the design matrix \mathbf{X} .
- H_{ii} is the influence of y_i on its own fitted value; it tells us how much of \hat{m}_i is just y_i .
- This quantity is called the **leverage** - sometimes written as h_i .
- The leverage of the i^{th} data point doesn't depend on y_i , only on the design matrix.
- It can be shown the sum of the i leverages is equal to p , the number of regression coefficients:

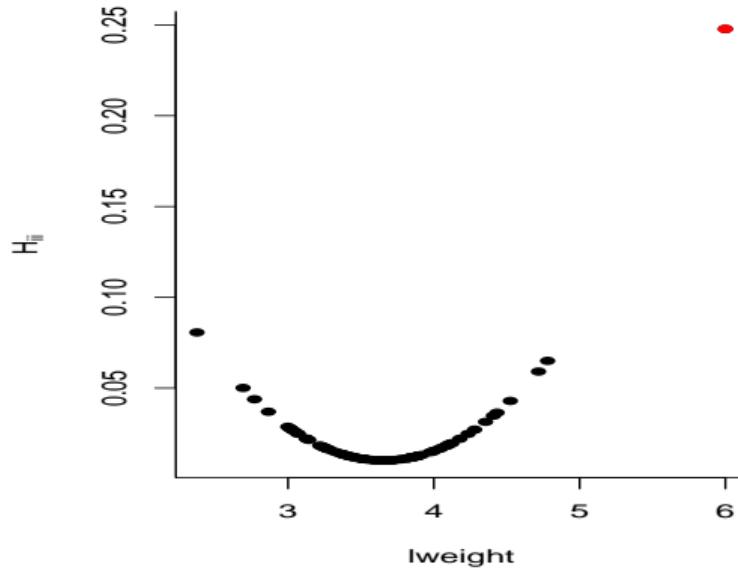
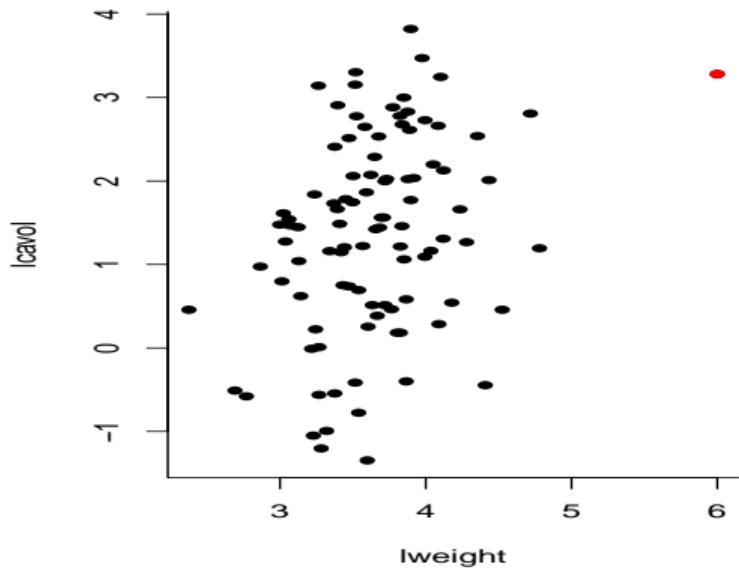
$$\text{tr}\{\mathbf{H}\} = p = \sum_{i=1}^n H_{ii}$$

- The regression function will be pulled towards points with higher leverage.
- Notice that the values of the leverage only depend on \mathbf{X} .

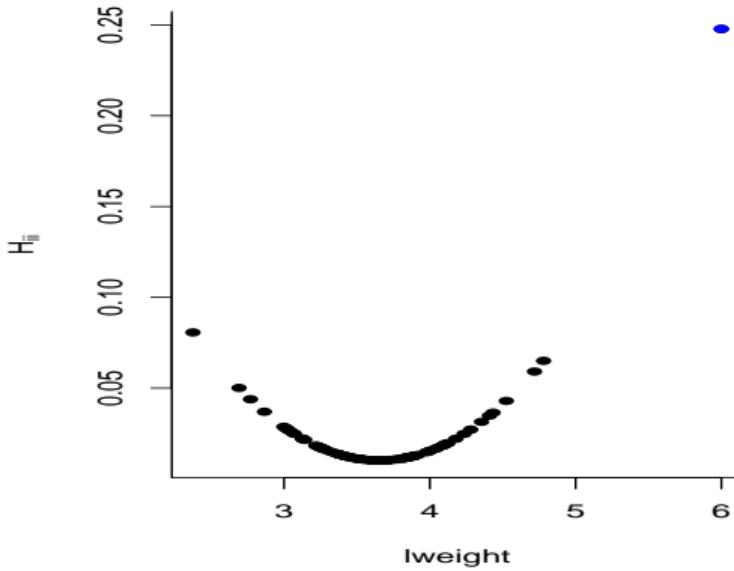
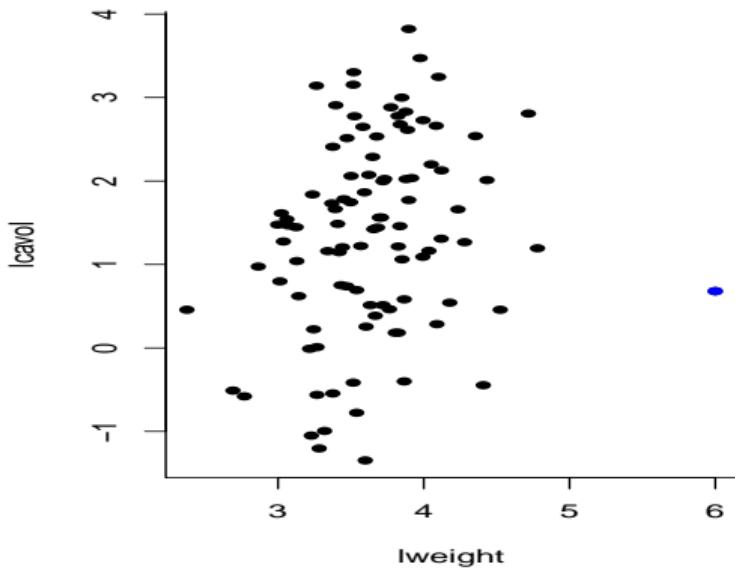
Leverage I



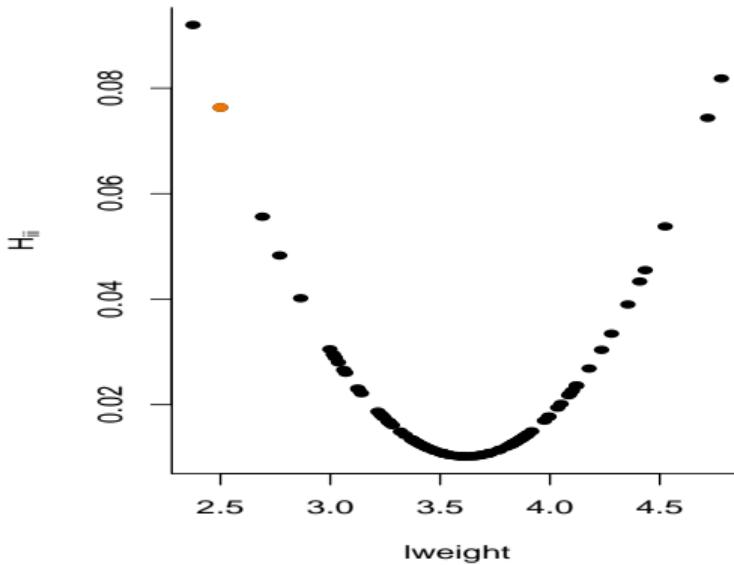
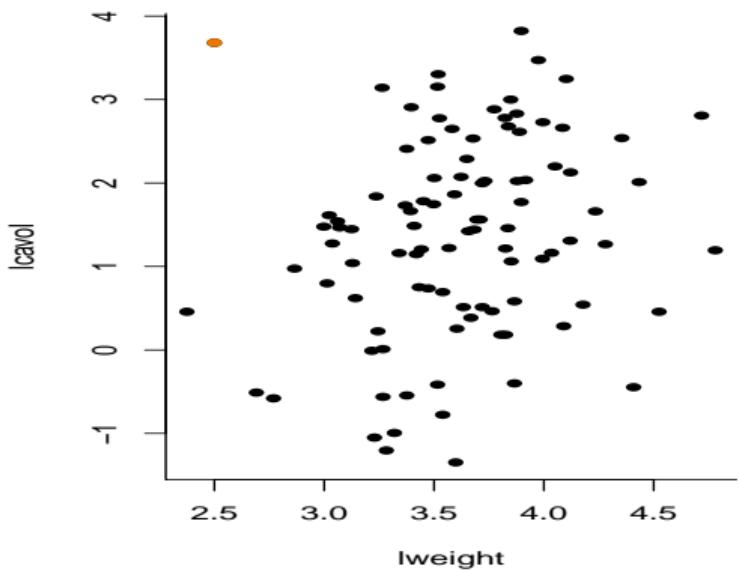
Leverage II



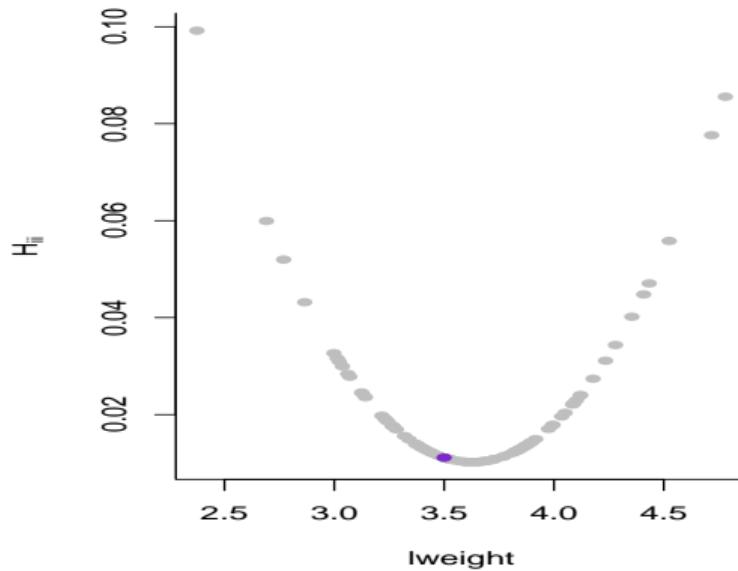
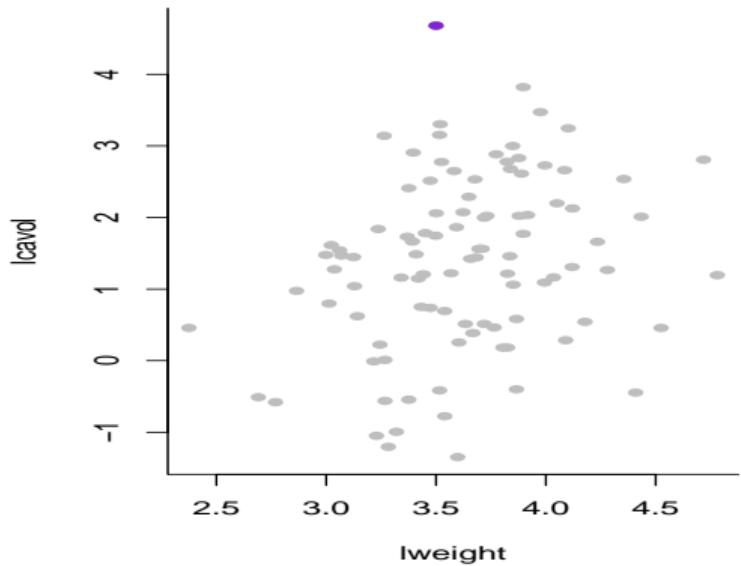
Leverage III



Leverage IV



Leverage V



Standardized and Studentized residuals I

Let's consider the model residuals:

$$\mathbf{e} = \mathbf{y} - \hat{\mathbf{m}} = (\mathbf{I} - \mathbf{H})\mathbf{y} = (\mathbf{I} - \mathbf{H})\boldsymbol{\epsilon}$$

We then have that:

$$\mathbb{E}[\mathbf{e}] = \mathbf{0} \text{ and } \text{Var}[\mathbf{e}] = \sigma^2(\mathbf{I} - \mathbf{H})(\mathbf{I} - \mathbf{H})^T = \sigma^2(\mathbf{I} - \mathbf{H})$$

The variance of the residual for the i^{th} data point depends on i through the hat matrix:

$$\text{Var}[e_i] = \sigma^2(\mathbf{I} - \mathbf{H})_{ii} = \sigma^2(1 - H_{ii})$$

So the bigger the leverage of i , the smaller the variance of the residual there (the model tries very hard to fit these points).

We define the **standardized** or (internally) **studentized residuals**

$$r_i \equiv \frac{e_i}{\hat{\sigma}\sqrt{1 - H_{ii}}}$$

Standardized and Studentized residuals II

Why “studentized”? Because we’re dividing by an estimate of the standard error, just like in “Student’s” t -test for differences in means

Standardized residuals are sometimes preferred in residual plots, as they have been standardized to have equal variance.

Indeed the plots we get when applying `plot` to an `lm` object in R are based on Standardized residuals, which can be obtained using `rstandard`

Standardized and Studentized residuals III

In defining the Leave-one-out cross validation (LOOCV) we considered what would be our estimate of y_i if y_i was not included in the dataset.

We denoted this value with $\hat{y}_{[i]}$ while $e_{[i]}$ as the residual for the i th observation, when that observation is not used to fit the model:

$$e_{[i]} = y_i - \hat{y}_{[i]}$$

Leaving out the data point i would give us an MSE of $\hat{\sigma}_{[i]}^2$. A little work says that

$$t_i \equiv \frac{e_{[i]}}{\hat{\sigma}_{[i]} \sqrt{1 + \mathbf{x}_i^T (\mathbf{x}_{[i]}^T \mathbf{x}_{[i]}^{-1}) \mathbf{x}_i}} \sim t_{n-p}$$

These are called the **cross-validated**, or **jackknife**, or **externally studentized**, residuals.

Standardized and Studentized residuals IV

Fortunately, we can compute this without having to actually re-run the regression:

$$\begin{aligned} t_i &= \frac{e_{[i]}}{\hat{\sigma}_{[i]} \sqrt{1 + \mathbf{x}_i^T (\mathbf{x}_{[i]}^T \mathbf{x}_{[i]})^{-1} \mathbf{x}_i}} \\ &= \frac{e_i}{\hat{\sigma}_{[i]} \sqrt{1 - H_{ii}}} \\ &= r_i \sqrt{\frac{n - p}{n - p - r_i^2}} \end{aligned}$$

Cook's Distance I

Omitting point i will generally change all of the fitted values, not just the fitted value at that point: is the change big for a certain point i ?

Compare $\hat{\mathbf{m}}$ and $\hat{\mathbf{m}}_{[i]}$. Specifically we use a squared difference:

$$\|\hat{\mathbf{m}} - \hat{\mathbf{m}}_{[i]}\|^2 = (\hat{\mathbf{m}} - \hat{\mathbf{m}}_{[i]})^T (\hat{\mathbf{m}} - \hat{\mathbf{m}}_{[i]})$$

To make this more comparable across data sets, it's conventional to divide this by $p \hat{\sigma}^2$.

This is called the **Cook's distance** or **Cook's statistic** for point i :

$$D_i = \frac{(\hat{\mathbf{m}} - \hat{\mathbf{m}}_{[i]})^T (\hat{\mathbf{m}} - \hat{\mathbf{m}}_{[i]})}{(p) \hat{\sigma}^2}$$

Cook's Distance II

As usual, there is a simplified formula, which evades having to re-fit the regression:

$$D_i = \frac{1}{p} e_i^2 \frac{H_{ii}}{(1 - H_{ii})^2}$$

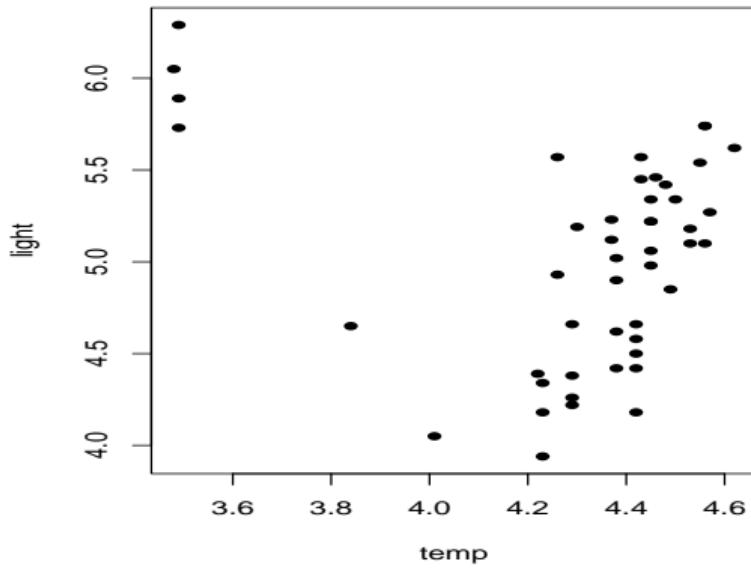
The total influence of a point over all the fitted values grows with both its leverage (H_{ii}) and the size of its residual when it is included (e_i^2).

CYG OB1 stars I

The data `stars` contains information on the log of the surface temperature and the log of the light intensity of 47 stars in the star cluster CYG OB1, which is in the direction of Cygnus.

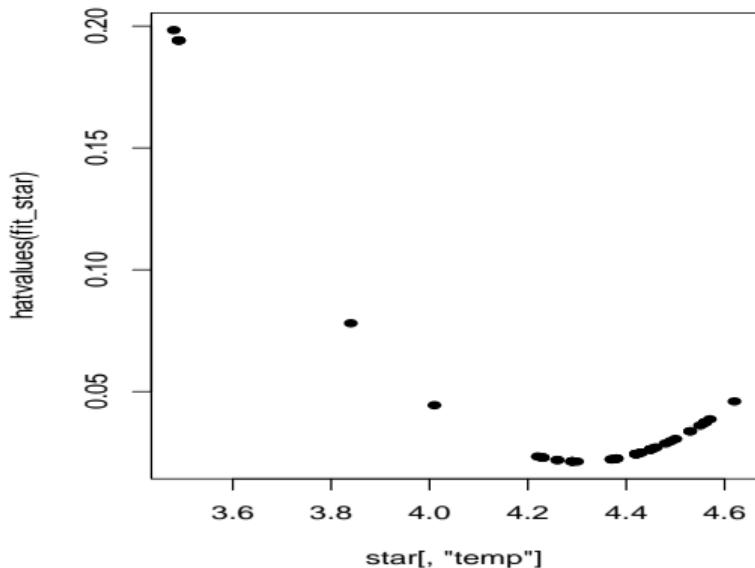
```
data(star, package = "faraway")
plot(star)
fit_star <- lm(light ~ temp, data = star)
```

CYG OB1 stars II



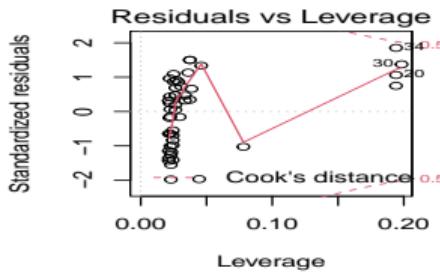
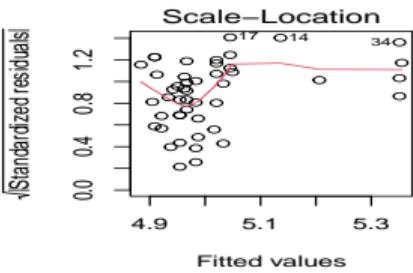
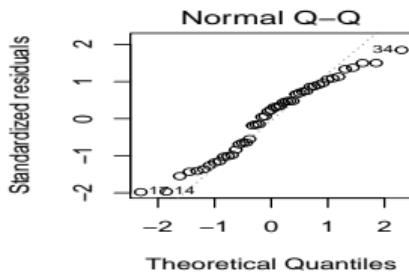
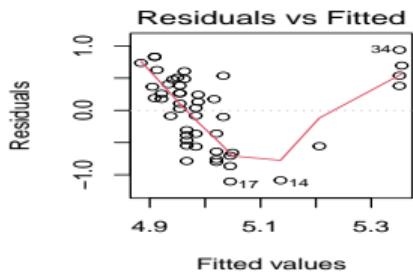
CYG OB1 stars III

```
plot(star[, "temp"], hatvalues(fit_star), pch = 16)
```



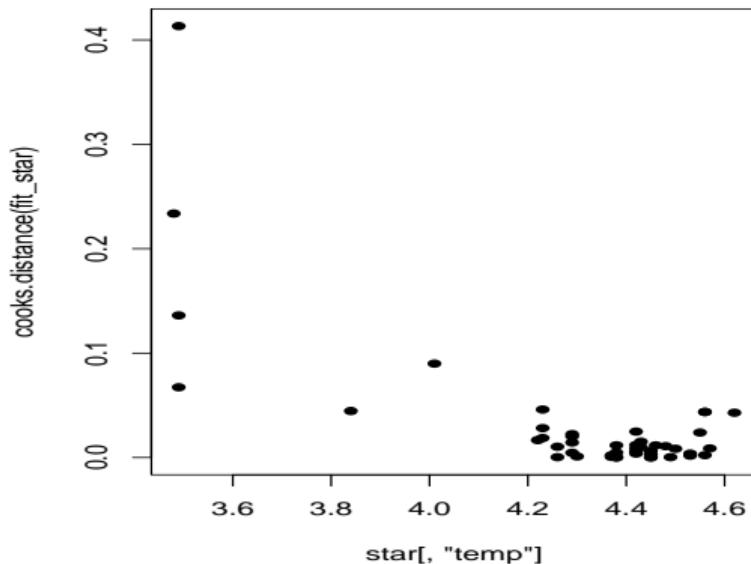
CYG OB1 stars IV

```
par(mfrow=c(2,2)); plot(fit_star)
```



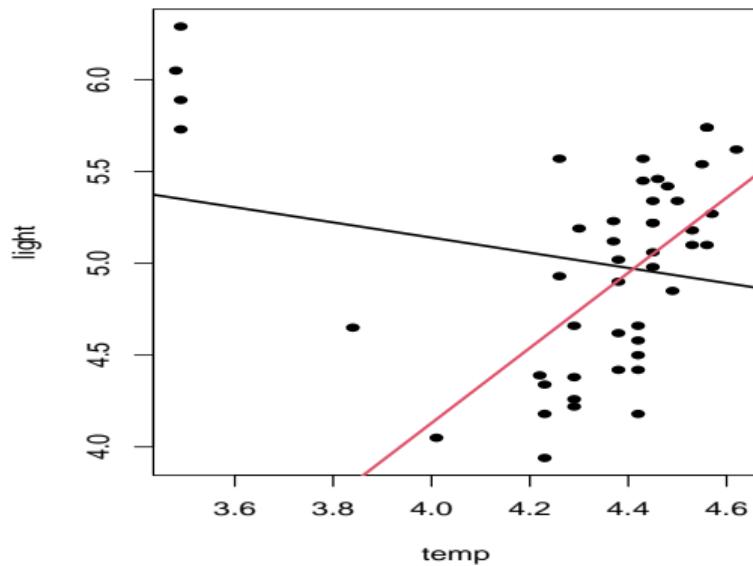
CYG OB1 stars V

```
plot(star[, "temp"], cooks.distance(fit_star), pch = 16)
```



CYG OB1 stars VI

Fitted lines with and without giant stars:



What to do about influential points I

- Automatic detection of influential points can be useful to find points which are problematic
- Sometimes these methods help us identify data recording issues or problematic measurement and we can discard the points
- But sometimes they are the symptom of something more complicated going on in the process: need to question what is the origin of these outlying points

Generalized linear models

The Generalized Linear Model (GLM) I

- In the Gaussian linear regression model

$$(Y | X_1 = x_1, \dots, X_{p-1} = x_{p-1}) \sim \mathcal{N}(\beta_0 + \beta_1 x_1 + \dots + \beta_{p-1} x_{p-1}, \sigma^2)$$

- Under this assumption

$$\mathbb{E}[Y | X_1 = x_1, \dots, X_{p-1} = x_{p-1}] = \beta_0 + \beta_1 x_1 + \dots + \beta_{p-1} x_{p-1}$$

- Two modifications of this situation

- We'll allow for other distributions different from the normal
- Instead of the conditional mean being a linear combination of the predictors, it can be some function of a linear combination of the predictors.

The Generalized Linear Model (GLM) II

- A **generalized linear model** has two parts:
 - a distribution of the response conditioned on the predictors. (This distribution needs to be from the **exponential family** of distributions.)
 - a **link** function, $g()$, that defines how the **linear combination** of the $p - 1$ predictors,

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_{p-1} x_{p-1}$$

is related to the mean of the response conditioned on the predictors, i.e.

$$g(\mathbb{E}[Y|X_1 = x_1, \dots, X_{p-1} = x_{p-1}]) = \beta_0 + \beta_1 x_1 + \dots + \beta_{p-1} x_{p-1}$$

Binary response and logistic regression I

- Categorical variables with two classes such as yes/no, cat/dog, sick/healthy, etc. can be coded in a binary variable, Y , using 0 and 1.

$$Y = \begin{cases} 1 & \text{yes} \\ 0 & \text{no} \end{cases}$$

- With a binary (Bernoulli) response, we'll mostly focus on the case when $Y = 1$, since we can obtain probabilities of $Y = 0$ with:

$$\Pr[Y = 0] = 1 - \Pr[Y = 1] = 1 - p$$

- Moreover

$$\mathbb{E}[Y] = \Pr[Y = 1] = p$$

Binary response and logistic regression II

Probability odd

$$\text{odd} = \frac{\Pr[Y = 1]}{\Pr[Y = 0]} = \frac{p}{1 - p}$$

- Interpretation: the odd is the probability for a positive event ($Y = 1$) divided by the probability of a negative event ($Y = 0$).
- When the odd is 1, the two events have equal probability. Odds greater than 1 favor a positive event. Odds smaller than 1 favor a negative event.
- The log odd is the **logit** transform applied to p .

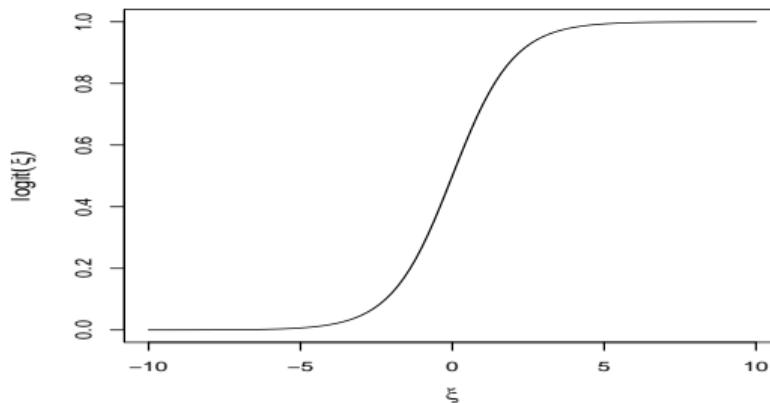
$$\text{logit}(\xi) = \log\left(\frac{\xi}{1 - \xi}\right)$$

- Since $p \in (0, 1)$, $\text{logit}(p) \in (-\infty, \infty)$.

Binary response and logistic regression III

- The inverse logit, also known as the **logistic** or **sigmoid** function is

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



Note that for $\xi \in (-\infty, \infty)$, the logistic takes values between 0 and 1.

Binary response and logistic regression IV

- We have a binary response Y : this is the variable we wish to model as function of a vector of $p - 1$ predictors X_1, \dots, X_{p-1}
- We denote

$$\begin{aligned} p(x_1, \dots, x_{p-1}) &= \Pr[Y = 1 \mid X_1 = x_1, \dots, X_{p-1} = x_{p-1}] \\ &= \mathbb{E}[Y \mid X_1 = x_1, \dots, X_{p-1} = x_{p-1}] \end{aligned}$$

Binary response and logistic regression V

- A **logistic regression** model for Y is a model such that the logarithm of the odd

$$\frac{p(x_1, \dots, x_{p-1})}{1 - p(x_1, \dots, x_{p-1})} = \frac{\Pr[Y = 1 | X_1 = x_1, \dots, X_{p-1} = x_{p-1}]}{\Pr[Y = 0 | X_1 = x_1, \dots, X_{p-1} = x_{p-1}]}$$

is a linear combination of the predictors

$$\log \left(\frac{p(x_1, \dots, x_{p-1})}{1 - p(x_1, \dots, x_{p-1})} \right) = \beta_0 + \beta_1 x_1 + \dots + \beta_{p-1} x_{p-1}$$

i.e.

$$p(x_1, \dots, x_{p-1}) = \frac{e^{\beta_0 + \beta_1 x_1 + \dots + \beta_{p-1} x_{p-1}}}{1 + e^{\beta_0 + \beta_1 x_1 + \dots + \beta_{p-1} x_{p-1}}}$$

Binary response and logistic regression VI

- Then

$$\mathbb{E}[Y|X_1 = x_1, \dots, X_{p-1} = x_{p-1}] = \frac{e^{\beta_0 + \beta_1 x_1 + \dots + \beta_{p-1} x_{(p-1)}}}{1 + e^{\beta_0 + \beta_1 x_1 + \dots + \beta_{p-1} x_{(p-1)}}}$$

- Note that the **link** function g such that

$$g(\mathbb{E}[Y|X_1 = x_1, \dots, X_{p-1} = x_{p-1}]) = \beta_0 + \beta_1 x_1 + \dots + \beta_{p-1} x_{p-1}$$

is the logit transformation $\text{logit}(\xi) = \log\left(\frac{\xi}{1 - \xi}\right)$

Poisson response and log-linear regression I

- Suppose we have count responses Y that we wish to model in terms of a vector of predictors x_1, \dots, x_{p-1} .
- Assume that Y is Poisson with mean $\mathbb{E}[Y] = \mu > 0$, then:

$$\Pr(Y = y) = \frac{\mu^y}{y!} \exp(-\mu)$$

- Model μ in terms of the predictors:

$$(Y | X_1 = x_1, \dots, X_{p-1} = x_{p-1}) \sim \text{Poisson}(\mu(x_1, \dots, x_{p-1})),$$

we need to specify the function $\mu(\cdot)$

- We use a linear combination of the x_i to form the linear predictor.

Poisson response and log-linear regression II

- Since we require that $\mu(\cdot) \geq 0$, we can ensure this by that is:

$$\mu(x_1, \dots, x_{p-1}) = \exp\{\beta_0 + \beta_1 x_1 + \dots + \beta_{p-1} x_{p-1}\}$$

i.e.

$$\log(\mu(x_1, \dots, x_{p-1})) = \beta_0 + \beta_1 x_1 + \dots + \beta_{p-1} x_{(p-1)}$$

- The link function g

$$g(\mathbb{E}[Y|X_1 = x_1, \dots, X_{p-1} = x_{p-1}]) = \beta_0 + \beta_1 x_1 + \dots + \beta_{p-1} x_{p-1}$$

is the logarithm and the model is termed **log-linear** model

- This corresponds to the canonical link: it is a natural way to construct the model

Generalizing I

- For the Poisson and Bernoulli case we have defined
 $g(E[Y]) = \beta_0 + \beta_1 x_1 + \cdots + \beta_{p-1} x_{p-1}$ taking different functions $g(\cdot)$
- We then have

$$Y \sim \text{Bern}(g^{-1}(\beta_0 + \beta_1 x_1 + \cdots + \beta_{p-1} x_{p-1}))$$

and

$$Y \sim \text{Pois}(g^{-1}(\beta_0 + \beta_1 x_1 + \cdots + \beta_{p-1} x_{p-1}))$$

- Poisson and Bernoulli are 1-parameter distribution in which the variance is related to the mean
- In the Normal distribution we also have a nuisance parameter σ^2
- The Normal, Bernoulli and Poisson all belong to the Exponential Family of distribution: GLM generalizes the linear model to all distributions which belong to the Exponential Family

Generalizing II

- GLM is a broad class of models. We can use many different functions $g()$: for each such function, we have a different GLM.

Exponential family I

- GLM assumes a specified parametric class for the conditional distribution of Y given x_1, \dots, x_{p-1} : the exponential family.
- This is formally defined as a parametric family whose probability density/mass function has the form

$$f(y; \theta, \phi) = \exp \left\{ \frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi) \right\}$$

where $a(\cdot)$, $b(\cdot)$, and $c(\cdot, \cdot)$ are specific functions.

- The parameter θ is called the **canonical parameter** and represents the location, while ϕ is called the **dispersion** parameter and represents the scale.
- Some nice properties

$$\mu = \mathbb{E}[Y] = b'(\theta), \quad \text{Var}[Y] = b''(\theta)a(\phi)$$

Exponential family II

- The **canonical link function** g is the one that transforms $\mathbb{E}[Y] = b'(\theta)$ into the **canonical parameter** θ . this is happens if

$$\theta = g(\mathbb{E}[Y])$$

or if

$$\theta = b'^{-1}(\mathbb{E}[Y])$$

- The Variance of the distribution is also written as a function of θ (and ϕ)
- The $b''(\theta)$ function is called the *variance function*: specifies how the variance depends on the location parameter
- Some well-known examples of distributions belonging to the EF are given below.

Exponential family III

- Gaussian distribution:

$$\begin{aligned}f(y; \mu, \sigma^2) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{1}{2} \left(\frac{y-\mu}{\sigma} \right)^2 \right\} \\&= \exp \left\{ -\frac{1}{2} \left(\frac{y-\mu}{\sigma} \right)^2 - \frac{1}{2} \log(\sqrt{2\pi\sigma^2}) \right\} \\&= \exp \left\{ \frac{y\mu - 0.5\mu^2}{\sigma^2} - \frac{1}{2} \left(\frac{y^2}{\sigma^2} + \log(\sqrt{2\pi\sigma^2}) \right) \right\} \\&= \exp \left\{ \frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi) \right\}\end{aligned}$$

where $\theta = \mu$, $\phi = \sigma^2$, $a(\phi) = \phi$, $b(\theta) = 0.5\theta^2$, and $c(y, \phi) = -\frac{1}{2} \left(\frac{y^2}{\phi} + \log(\sqrt{2\pi\phi}) \right)$

$$\mathbb{E}[Y] = b'(\theta) = \theta = \mu, \quad \text{Var}[Y] = b''(\theta)a(\phi) = \phi = \sigma^2$$

Exponential family IV

canonical link function $g(\xi) = \xi$, the identity function

- Gamma distribution:

$$f(y; \lambda, \nu) = \frac{1}{\Gamma(\nu)} \lambda^\nu y^{\nu-1} e^{-\lambda y}$$

and we have $E[Y] = \nu/\lambda$ and $Var[Y] = \nu/\lambda^2$. We can reparameterize the distribution taking $\mu = \nu/\lambda$, i.e. $\lambda = \nu/\mu$:

$$f(y; \mu, \nu) = \frac{1}{\Gamma(\nu)} \left(\frac{\nu}{\mu} \right)^\nu y^{\nu-1} e^{-\frac{\nu}{\mu} y}$$

so that

$$E[Y] = \mu \quad \text{and} \quad Var[Y] = \frac{\mu^2}{\nu}$$

Exponential family V

$$\begin{aligned}f(y; \mu, \nu) &= \frac{1}{\Gamma(\nu)} \left(\frac{\nu}{\mu} \right)^\nu y^{\nu-1} e^{-\nu y / \mu} \\&= \exp \{ -\log(\Gamma(\nu)) + \nu \log(\nu/\mu) + (\nu - 1) \log(y) - \nu y / \mu \} \\&= \exp \left\{ \left(-\frac{1}{\mu} y - \log(\mu) \right) \nu \right. \\&\quad \left. - \log(\Gamma(\nu)) + \nu \log(\nu) + (\nu - 1) \log(y) \right\} \\&= \exp \left\{ \frac{\theta y - b(\theta)}{a(\phi)} + c(y, \phi) \right\}\end{aligned}$$

where $\theta = -1/\mu$, $\phi = \nu$, $a(\phi) = 1/\nu$, $b(\theta) = \log(-1/\theta) = -\log(-\theta)$, and
 $c(y, \phi) = -\log(\Gamma(\nu)) + \nu \log(\nu) + (\nu - 1) \log(y)$

$$\mathbb{E}[Y] = b'(\theta) = -1/\theta = \mu, \quad \text{Var}[Y] = b''(\theta)a(\phi) = \frac{1}{\theta^2} \frac{1}{\nu}$$

Exponential family VI

canonical link function $g(\xi) = -1/\xi$

- Bernoulli distribution:

$$\begin{aligned} f(y; p) &= p^y(1-p)^{1-y} = \exp \left\{ y \log \frac{p}{1-p} + \log(1-p) \right\} \\ &= \exp \{ y\theta - \log(1 + \exp(\theta)) \} \\ &= \exp \left\{ \frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi) \right\} \end{aligned}$$

with $\theta = \log \left(\frac{p}{1-p} \right)$, $a(\phi) = 1$, $b(\theta) = \log(1 + \exp(\theta))$, $c(y, \phi) = 0$

$$\mathbb{E}[Y] = b'(\theta) = \frac{\exp(\theta)}{1 + \exp(\theta)} = p,$$

$$\text{Var}[Y] = b''(\theta)a(\phi) = \frac{\exp(\theta)}{[1 + \exp(\theta)]^2} = p(1 - p)$$

Exponential family VII

canonical link function $g(\xi) = \log\left(\frac{\xi}{1-\xi}\right)$, the logit function

- Poisson distribution:

$$\begin{aligned}f(y; \lambda) &= \exp\{-\lambda\} \frac{\lambda^y}{y!} \\&= \exp\{y \log \lambda - \lambda - \log(y!)\} \\&= \exp\{y\theta - \exp(\theta) - \log(y!)\} \\&= \exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi)\right\}\end{aligned}$$

with $\theta = \log(\lambda)$, $a(\phi) = 1$, $b(\theta) = \exp(\theta)$, and $c(y, \phi) = -\log(y!)$

$$\mathbb{E}[Y] = b'(\theta) = \exp(\theta) = \lambda,$$

$$\text{Var}[Y] = b''(\theta)a(\phi) = \exp(\theta) = \lambda$$

canonical link function $g(\xi) = \log(\xi)$, the logarithmic function

Estimation of a GLM I

- In the sequel we denote $Y \sim EF(\theta, \phi, a, b, c)$ if Y has a distribution that belongs to an exponential family
- With n observations, we write the model indexed with i to note that it is being applied to each observation.

$$Y_i \sim EF(\theta_i, \phi, a, b, c)$$

- We define a **link** between the mean and expected value of the distribution:

$$\eta_i = g(\mu_i) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_{p-1} x_{i(p-1)}$$

- Often we choose the **canonical link**: $\eta = \theta = g(\mu)$
- The parameters, β , of a GLM can be estimated using maximum likelihood.

Estimation of a GLM II

- The log-likelihood for a single observation

$$\left\{ \frac{y_i \theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi) \right\}$$

then the log-likelihood is

$$\ell(\beta) = \sum_{i=1}^n \left\{ \frac{y_i \theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi) \right\}$$

- Unfortunately, unlike ordinary linear regression, there is no analytical solution for this maximization problem.
- Instead, it will need to be solved using an iteratively reweighted least squares algorithm (IRLS).

Estimation of a GLM III

- IRLS outline: we wish to regress $g(y)$ onto X accounting for weights proportional to $\text{var}(g(y))$.
- $g(y)$ could be numerically unstable - we linearize it.
- Using Taylor expansion for $g(y)$ around μ we define:

$$g(y) \approx g(\mu) + (y - \mu)g'(\mu) = \eta + (y - \mu)\frac{d\eta}{d\mu} = z$$

and

$$\widehat{\text{var}}(z) = \left(\frac{d\eta}{d\mu}\right)^2 V(y)|_{\hat{\mu}} = \frac{1}{v}$$

- Use z as pseudo-data to estimate β using a weighted regression

Estimation of a GLM IV

- The IRLS algorithm:

- ➊ Initialize $\hat{\eta}_0$ (with some $\hat{\beta}_0$) and $\hat{\mu}_0$
- ➋ Create $z_0 = \hat{\eta}_0 + (y - \hat{\mu}_0) \frac{d\eta}{d\mu} \Big|_{\hat{\eta}_0}$
- ➌ Form the weights

$$v_0^{-1} = \left(\frac{d\eta}{d\mu} \right)^2 \Big|_{\hat{\eta}_0} V(y) \Big|_{\hat{\mu}_0}$$

- ➍ Estimate $\hat{\beta}_1$ to get $\hat{\eta}_1$
 - ➎ Repeat steps 3-4-5 till convergence (on $\hat{\eta}$)
- Typically quite fast, but sometimes it can fail
 - No need to program it ourselves: in R we can use `glm`

R's `glm` function

- The `glm` function fit a GLM model using the maximum likelihood method
- For ordinary usage, the call is the same as for `lm`, except for one extra argument, `family`.
- The default value of `family` is `gaussian`
- In the Poisson regression case the call looks like
glm(y~x, family = poisson)
- In the logistic regression case, for example, the call looks like
glm(y~x, family = binomial)
- For each family we can specify a link function

GLM: the bernoulli Model I

- A study of the Pima tribe of Native Americans, involving factors associated with diabetes.
There is data on 332 women:

```
data("Pima.te", package = "MASS")
```

- Let's predict diabetes from the other variables:

```
logitout <- glm(type ~ . , data = Pima.te,  
family = binomial)  
summary(logitout)
```

GLM: the bernoulli Model II

Call:

```
glm(formula = type ~ ., family = binomial, data = Pima.te)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.9647	-0.6582	-0.3608	0.6158	2.4646

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-9.514019	1.229278	-7.740	9.98e-15 ***
npreg	0.140944	0.059652	2.363	0.01814 *
glu	0.037481	0.005558	6.743	1.55e-11 ***
bp	-0.008675	0.012589	-0.689	0.49076
skin	0.013167	0.020025	0.658	0.51084
bmi	0.078951	0.028432	2.777	0.00549 **
ped	1.110131	0.446921	2.484	0.01299 *
age	0.018055	0.018359	0.983	0.32537

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

GLM: the bernoulli Model III

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 420.30 on 331 degrees of freedom

Residual deviance: 285.79 on 324 degrees of freedom

AIC: 301.79

Number of Fisher Scoring iterations: 5

- *type* is a binary variable. See `?binomial` to see how to handle binomial responses (e.g. number of exams passed out of n exams tried in an academic year).

GLM: the bernoulli Model IV

- As a default R uses the canonical link:

```
binomial()$link
```

```
[1] "logit"
```

- The link can be changed - see ?family

- Specifying the family implies the specification of the variance function:

```
binomial()$variance
```

```
function (mu)
```

```
mu * (1 - mu)
```

```
<bytecode: 0x55e516e99948>
```

```
<environment: 0x55e515001b30>
```

- Specifying the family and the link implies a number of relationships:

```
names(binomial())
```

```
[1] "family"      "link"        "linkfun"     "linkinv"     "variance"
```

```
[6] "dev.resids"  "aic"        "mu.eta"    "initialize"  "validmu"
```

```
[11] "valideta"   "simulate"
```

Fitting issues for the logistic regression I

- We should note that, if there exists some β^* such that

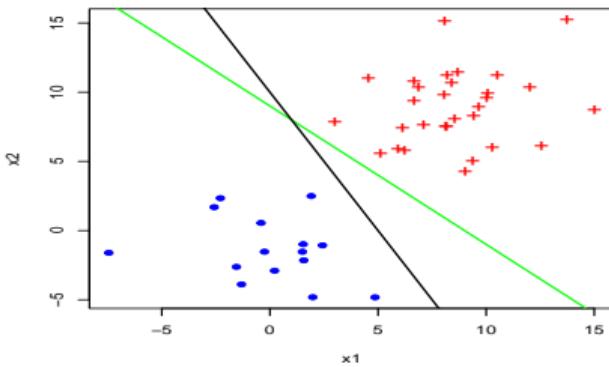
$$\beta_0 + \beta_1 x_{i1} + \cdots + \beta_{p-1} x_{i(p-1)} > 0 \implies y_i = 1$$

and

$$\beta_0 + \beta_1 x_{i1} + \cdots + \beta_{p-1} x_{i(p-1)} < 0 \implies y_i = 0$$

for all observations, then the MLE is not unique.

Fitting issues for the logistic regression II



- Such data is said to be separable.
- This, and similar numeric issues related to estimated probabilities near 0 or 1, will return a warning in R.

Fitting issues for the logistic regression III

Warning messages:

```
1: glm.fit: algorithm did not converge  
2: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

- When this happens, the model is still "fit," but there are consequences, namely, the estimated coefficients are highly suspect.
- This is an issue when then trying to interpret the model.
- However it could be useful for creating a classifier

Inference for model parameters I

- R prints a summary with some information on the regression parameter estimates
- The assumptions on which a generalized linear model is constructed allow us to specify what is the asymptotic distribution of the random vector $\hat{\beta}$ through the theory of MLE
- We assume that the randomness of Y comes only from $(Y|X_1 = x_1, \dots, X_{p-1} = x_{p-1})$ and not from the predictors.
- We rewrite the relationship between the expected value of μ and the linear predictor in matrix form:

$$\eta = g(\mu) = X\beta$$

- β is a vector of parameter that we wish to estimate: we have seen we estimate them using an algorithm that maximizes the likelihood
- so the $\hat{\beta}$ parameters enjoy optimal properties of MLEs

Inference for model parameters II

- There is an important difference between the inference results for the Gaussian linear model and for `glm`:
 - In Gaussian linear model the inference is exact. This is due to the nice properties of the normal, least squares estimation, and linearity. As a consequence, the distributions of the coefficients are perfectly known assuming that the assumptions hold.
 - In generalized linear models the inference is asymptotic. This means that the distributions of the coefficients are unknown except for large sample sizes n , for which we have approximations. The reason is the more complexity of the model in terms of non-linearity. This is the usual situation for the majority of regression models.

Inference for model parameters III

- We can show that

$$\hat{\beta} \stackrel{\text{approx}}{\sim} \mathcal{N}(\beta, \mathcal{I}(\beta)^{-1})$$

where $\stackrel{\text{approx}}{\sim}$ must be understood as asymptotically distributed as when $n \rightarrow \infty$ and

$$\mathcal{I}(\beta) = \mathbb{E} \left[-\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^\top} \right]$$

is the Fisher information matrix.

- The "larger" (large eigenvalues) the matrix is, the more precise the estimation of β is, because that results in smaller variances in.

Inference for model parameters IV

- It turns out that

$$\mathcal{I}(\beta) = X^\top V X$$

where

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1(p-1)} \\ 1 & x_{21} & x_{22} & \cdots & x_{2(p-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{n(p-1)} \end{bmatrix}$$

and $V = \text{diag}(V_1, \dots, V_n)$ with $V_i = \frac{1}{\text{Var}[Y_i]} \left(\frac{d\mu}{d\eta} \right)^2$.

- The V_i elements are the inverse of the weights used in the IRLS
- Notice that in the gaussian linear regression (with identity link) V_i are a constant for all i

Inference for model parameters V

- V_i values for noticeable distributions:
 - for logistic regression

$$V_i = \frac{\exp(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_1 x_{i(p-1)})}{[1 + \exp(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_1 x_{i(p-1)})]^2}$$

- for Poisson regression

$$V_i = \exp\{\beta_0 + \beta_1 x_{i1} + \cdots + \beta_1 x_{i(p-1)}\}$$

- The inverse of the Fisher information matrix is can be estimated by plugging in $\hat{\beta}$ into $\mathcal{I}(\beta)^{-1}$, i.e. by $\mathcal{I}(\hat{\beta})^{-1}$
- Quality of the estimation:
 - Bias. The estimates are **asymptotically** unbiased.
 - Variance. It depends on:
 - Sample size n . Hidden inside $X^\top V^{-1} X$. As n grows, the precision of the estimators increases.

Inference for model parameters VI

- Weighted predictor sparsity $(X^\top V^{-1} X)^{-1}$. The more sparse the predictor is (small eigenvalues of $(X^\top V^{-1} X)^{-1}$), the more precise $\hat{\beta}$ is.
- Note that: the precision of $\hat{\beta}$ is affected by the value of β , which is hidden inside V . This contrasts sharply with the linear model, where the precision of the least squares estimator was not affected by the value of the unknown coefficients. The reason is partially due to the heteroskedasticity of logistic regression and Poisson regression, which implies a dependence of the variance of Y in the predictors, hence in β .

Confidence intervals for the coefficients

- Similar to linear regression, the problem is that V is unknown in practice because it depends on β .
- Plugging-in the estimate $\hat{\beta}$ to β in V results in \hat{V} .
- We can use \hat{V} to get

$$Z_j = \frac{\hat{\beta}_j - \beta_j}{\text{SE}(\hat{\beta}_j)} \stackrel{\text{approx}}{\sim} \mathcal{N}(0, 1)$$

where $\text{SE}(\hat{\beta}_j)^2 = v_j$ with v_j is the j -th element of the diagonal of $(X^\top V X)^{-1}$

- Thanks to normal approximation, we can have the $100(1 - \alpha)\%$ CI for the coefficient β_j

$$\hat{\beta}_j \pm z_{1-\alpha/2} \text{SE}(\hat{\beta}_j)$$

where $z_{1-\alpha/2}$ is the $1 - \alpha/2$ -upper quantile of the $\mathcal{N}(0, 1)$.

Testing with GLMs: Wald test I

- System of hypothesis

$$H_0 : \beta_j = 0 \quad \text{vs} \quad H_A : \beta_j \neq 0$$

- The distribution of the test statistic

$$W = \frac{\hat{\beta}_j}{\text{SE}[\hat{\beta}_j]}$$

is no longer T , but for a large number of observations ($n \rightarrow \infty$)

$$W \stackrel{\text{approx}}{\sim} \mathcal{N}(0, 1)$$

- The t -test for ordinary linear regression, assuming the assumptions were correct, had an exact distribution for any sample size.

Testing with GLMs: Wald test II

- R will obtain the standard error for us. 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.

GLM: the Poisson Model I

- This problem refers to data from a study of nesting horseshoe crabs (J. Brockmann, Ethology 1996);
- Available from the GLMsData package
- Each female horseshoe crab in the study had a male crab attached to her in her nest. The study investigated factors that affect whether the female crab had any other males, called satellites, residing near her.

GLM: the Poisson Model II



- Explanatory variables that are thought to affect this included the female crab's color (C), spine condition (S), weight (Wt), and carapace width (W). The response outcome for each female crab is her number of satellites (Sa).
- There are 173 females in this study and the data can be derived from the library GLMsData

GLM: the Poisson Model III

```
# install.packages("GLMsData")
data(hcrabs, package = "GLMsData")
### here do some exploration of the data
```

- We fit the intercept only model. This model implies the expected number of satellites per each crab is the same

```
model0<-glm(Sat~1, family=poisson(link=log),data=hcrabs)
summary(model0)
```

Call:

```
glm(formula = Sat ~ 1, family = poisson(link = log), data = hcrabs)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.4162	-2.4162	-0.5707	1.1045	4.9942

Coefficients:

Estimate	Std. Error	z value	Pr(> z)
----------	------------	---------	----------

GLM: the Poisson Model IV

```
(Intercept) 1.0713      0.0445    24.07   <2e-16 ***  
---  
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
(Dispersion parameter for poisson family taken to be 1)
```

```
Null deviance: 632.79 on 172 degrees of freedom  
Residual deviance: 632.79 on 172 degrees of freedom  
AIC: 990.09
```

```
Number of Fisher Scoring iterations: 5
```

GLM: the Poisson Model V

- Then we fit a model including the carapace width and weight as predictors

```
model1<-glm(Sat~1+Width+Wt,family=poisson(link=log),  
             data=hcrabs)  
  
summary(model1)
```

Call:

```
glm(formula = Sat ~ 1 + Width + Wt, family = poisson(link = log),  
     data = hcrabs)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.9309	-1.9702	-0.5478	0.9700	4.9904

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.2952111	0.8988960	-1.441	0.14962
Width	0.0460765	0.0467497	0.986	0.32433
Wt	0.0004470	0.0001586	2.818	0.00483 **

GLM: the Poisson Model VI

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
(Dispersion parameter for poisson family taken to be 1)
```

```
Null deviance: 632.79  on 172  degrees of freedom  
Residual deviance: 559.90  on 170  degrees of freedom  
AIC: 921.2
```

```
Number of Fisher Scoring iterations: 6
```

- Natural question: does adding the predictors "improve" the model?
- Notice that we have fitted two nested models

Testing with GLMs: likelihood-ratio test I

- Consider the following full model,

$$g(\mathbb{E}[Y|X_1 = x_1, \dots, X_{p-1} = x_{p-1}]) = \beta_0 + \beta_1 x_1 + \dots + \beta_{p-1} x_{p-1}$$

and we denote the MLE of these β -parameters as $\hat{\beta}_{\text{Full}}$

- Consider a null model,

$$g(\mathbb{E}[Y|X_1 = x_1, \dots, X_{p-1} = x_{p-1}]) = \beta_0 + \beta_1 x_1 + \dots + \beta_{q-1} x_{q-1}$$

where $q < p$. We denote the MLE of these β -parameters as $\hat{\beta}_{\text{Null}}$

- The difference between these two models can be codified by the null hypothesis of a test.

$$H_0 : \beta_q = \beta_{q+1} = \dots = \beta_{p-1} = 0.$$

Testing with GLMs: likelihood-ratio test II

- We define a test statistic, LR ,

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

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

$$LR \stackrel{\text{approx}}{\sim} \chi^2_{p-q}$$

- The test, which we will call the **Likelihood-Ratio Test (LRT)**, will be the analogue to the ANOVA F -test for linear regression.
- To perform the LRT, we'll actually again use the `anova` function in R.
- The LRT is a rather general test, however, here we have presented a specific application to GLMs.

Testing with GLMs: likelihood-ratio test III

- We use the LRT test to compare the two models for the horseshoe crabs

```
logLik(model0); logLik(model1)
'log Lik.' -494.0447 (df=1)
'log Lik.' -457.5991 (df=3)
(tstat <- as.numeric(2*(logLik(model1) - logLik(model0))))
[1] 72.89106
diff_df <- length(model1$coefficients) - length(model0$coefficients)
# pvalue
pchisq(tstat, df = diff_df, lower.tail = FALSE)
[1] 1.485621e-16
```

- Or we can use directly the anova function in R:

```
anova(model0, model1, test = "LRT")
```

Testing with GLMs: likelihood-ratio test IV

Analysis of Deviance Table

Model 1: Sat ~ 1

Model 2: Sat ~ 1 + Width + Wt

	Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
1	172	632.79			
2	170	559.90	2	72.891	< 2.2e-16 ***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

- Notice that we need to specify the test we wish to perform
- The table produced by anova is referred to as the *analysis of deviance* table
- The *deviance* plays in GLM the same role played by the sum of squares in gaussian linear regression

Deviance I

- The deviance for a model is defined as:

$$\begin{aligned} D &= 2[I(\hat{\beta}_{\max}) - I(\hat{\beta})]\phi \\ &= \frac{\phi}{a(\phi)} [y(\hat{\theta}^{\text{sat}} - \hat{\theta}) - b(\hat{\theta}^{\text{sat}}) + b(\hat{\theta})] \end{aligned}$$

where $I(\hat{\beta}_{\max})$ is the maximized likelihood under the saturated model and $\hat{\theta}^{\text{sat}}$ is the estimated value of θ in the saturated model

- The saturated model is the model in which we have a different estimated value θ_i for each observation
- The likelihood under the saturated model is the highest possible likelihood for the data

Deviance II

Saturated model

- $g(\mathbb{E}[Y_i|X_1 = x_{i1}, \dots, X_{p-1} = x_{i,p-1}]) = \theta_i, (i = 1, \dots, n)$
- The log-likelihood for a single observation is

$$\log f(y_i; \theta_i, \phi) = \frac{y_i \theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi)$$

- Taking the derivative with respect to θ_i

$$\begin{aligned}\frac{\partial}{\partial \theta_i} \log f(y_i; \theta_i, \phi) &= \frac{y_i - b'(\theta_i)}{a(\phi)} \\ &= \frac{y_i - \mathbb{E}[Y_i|X_1 = x_{i1}, \dots, X_{p-1} = x_{i,p-1}]}{a(\phi)}\end{aligned}$$

- MLE $\frac{\partial}{\partial \theta_i} \log f(y_i; \theta_i, \phi) = 0, \Rightarrow y_i = b'(\hat{\theta}_i), \Rightarrow \hat{\theta}_i^{\text{sat}} = b'^{-1}(y_i)$
- Prediction $\hat{y}_i^{\text{sat}} = y_i$

Goodness of fit I

- LRT only holds for nested models
- How to compare non-nested models?
- We need general purpose measures of goodness of fit
- We use AIC (and BIC):

$$AIC(\theta(\mathcal{M}), p(\mathcal{M})) = -\logLik(\theta(\mathcal{M})) + 2 * p(\mathcal{M})$$

```
AIC(model1)
```

```
[1] 921.1983
```

```
2*(-as.numeric(logLik(model1)) +
  length(model1$coefficients))
```

```
[1] 921.1983
```

- Same interpretation as for linear models

SAheart example I

- To illustrate the use of logistic regression, we will use the SAheart dataset from the book *The Elements of Statistical Learning*.

```
f1 <- "http://www-stat.stanford.edu/~tibs/ElemStatLearn/datasets/SAheart.  
SAheart <- read.table(f1,  
                      sep=",",head=T,row.names=1)
```

	sbp	tobacco	ldl	adiposity	famhist	typea	obesity	alcohol	age	chd
1	160	12.00	5.73	23.11	Present	49	25.30	97.20	52	1
2	144	0.01	4.41	28.61	Absent	55	28.87	2.06	63	1
3	118	0.08	3.48	32.28	Present	52	29.14	3.81	46	0
4	170	7.50	6.41	38.03	Present	51	31.99	24.26	58	1
5	134	13.60	3.50	27.78	Present	60	25.99	57.34	49	1
6	132	6.20	6.47	36.21	Present	62	30.77	14.14	45	0

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

SAheart example II

- Variables
 - chd whether or not coronary heart disease is present in an individual. (numeric 0 / 1) variable.
 - 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.

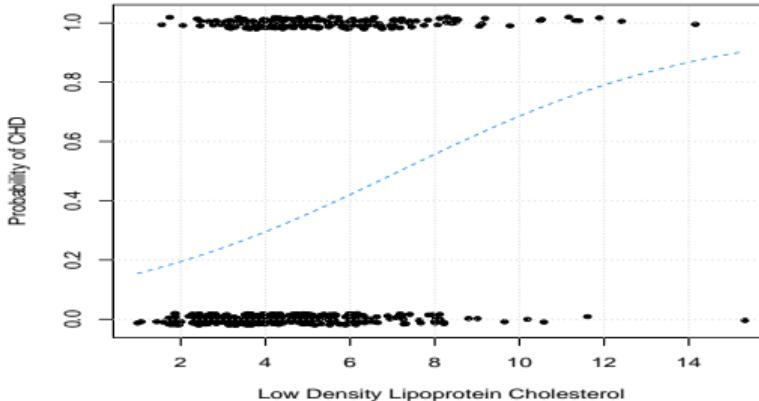
SAheart example III

- First we model the probability of coronary heart disease based on low density lipoprotein cholesterol.

$$\log \left(\frac{\Pr[\text{chd} = 1]}{1 - \Pr[\text{chd} = 1]} \right) = \beta_0 + \beta_{\text{ldl}} x_{\text{ldl}}$$

```
chd_mod_ldl <- glm(chd ~ ldl, data = SAheart, family = binomial)
plot(jitter(chd, factor = 0.1) ~ ldl, data = SAheart, pch = 20,
     ylab = "Probability of CHD", xlab = "Low Density Lipoprotein Cholesterol")
grid()
curve(predict(chd_mod_ldl, data.frame(ldl = x), type = "response"),
      add = TRUE, col = "dodgerblue", lty = 2)
```

SAheart example IV



As we would expect, this plot indicates that as ldl increases, so does the probability of chd.

SAheart example V

- To perform the test

$$H_0 : \beta_{\text{ldl}} = 0 \quad VS \quad H_A : \beta_{\text{ldl}} \neq 0$$

we use the `summary` function

```
coef(summary(chd_mod_ldl))
```

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.9686681	0.27307908	-7.209150	5.630207e-13
ldl	0.2746613	0.05163983	5.318787	1.044615e-07

and we have a low p-value, so we reject the null hypothesis.

- The `ldl` variable appears to be a significant predictor.
- Confidence intervals at 95%

```
confint.default(chd_mod_ldl, level=0.95)
```

SAheart example VI

	2.5 %	97.5 %
(Intercept)	-2.5038933	-1.4334430
ldl	0.1734491	0.3758735

SAheart example VII

- To fit an additive model using all available predictors, we use:

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

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

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

- The LR test statistic,

```
-2 * as.numeric(logLik(chd_mod_ldl) - logLik(chd_mod_additive))  
[1] 92.13879
```

- We can also utilize the anova function, by specifying test = "LRT"

```
anova(chd_mod_ldl, chd_mod_additive, test = "LRT")
```

SAheart example VIII

Analysis of Deviance Table

```
Model 1: chd ~ ldl
Model 2: chd ~ sbp + tobacco + ldl + adiposity + famhist + typea + obesity +
         alcohol + age
Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       460     564.28
2       452     472.14  8    92.139 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The small p-value suggests that we prefer the larger model.

- To select a subset of predictors, we can use a stepwise procedure as we did with ordinary linear regression.
- Recall that AIC and BIC were defined in terms of likelihoods.

SAheart example IX

- Example using AIC with a backwards selection procedure.

```
chd_mod_selected <- step(chd_mod_additive, trace = 1, k = 2)  
Start: AIC=492.14  
chd ~ sbp + tobacco + ldl + adiposity + famhist + typea + obesity +  
    alcohol + age
```

	Df	Deviance	AIC
- alcohol	1	472.14	490.14
- adiposity	1	472.55	490.55
- sbp	1	473.44	491.44
<none>		472.14	492.14
- obesity	1	474.23	492.23
- ldl	1	481.07	499.07
- tobacco	1	481.67	499.67
- typea	1	483.05	501.05
- age	1	486.53	504.53
- famhist	1	488.89	506.89

Step: AIC=490.14

SAheart example X

```
chd ~ sbp + tobacco + ldl + adiposity + famhist + typea + obesity +
    age
```

	Df	Deviance	AIC
- adiposity	1	472.55	488.55
- sbp	1	473.47	489.47
<none>		472.14	490.14
- obesity	1	474.24	490.24
- ldl	1	481.15	497.15
- tobacco	1	482.06	498.06
- typea	1	483.06	499.06
- age	1	486.64	502.64
- famhist	1	488.99	504.99

Step: AIC=488.55

```
chd ~ sbp + tobacco + ldl + famhist + typea + obesity + age
```

	Df	Deviance	AIC
- sbp	1	473.98	487.98

SAheart example XI

```
<none>      472.55 488.55
- obesity   1  474.65 488.65
- tobacco   1  482.54 496.54
- ldl       1  482.95 496.95
- typea     1  483.19 497.19
- famhist   1  489.38 503.38
- age       1  495.48 509.48
```

Step: AIC=487.98

chd ~ tobacco + ldl + famhist + typea + obesity + age

	Df	Deviance	AIC
- obesity	1	475.69	487.69
<none>		473.98	487.98
- tobacco	1	484.18	496.18
- typea	1	484.30	496.30
- ldl	1	484.53	496.53
- famhist	1	490.58	502.58
- age	1	502.11	514.11

SAheart example XII

Step: AIC=487.69

chd ~ tobacco + ldl + famhist + typea + age

	Df	Deviance	AIC
<none>		475.69	487.69
- ldl	1	484.71	494.71
- typea	1	485.44	495.44
- tobacco	1	486.03	496.03
- famhist	1	492.09	502.09
- age	1	502.38	512.38

coef(chd_mod_selected)

(Intercept)	tobacco	ldl	famhistPresent	typea
-6.44644451	0.08037533	0.16199164	0.90817526	0.03711521
age				
0.05046038				

SAheart example XIII

- We could again compare this model to the additive models.

$$H_0 : \beta_{\text{sbp}} = \beta_{\text{adiposity}} = \beta_{\text{obesity}} = \beta_{\text{alcohol}} = 0$$

```
anova(chd_mod_selected, chd_mod_additive, test = "LRT")
```

Analysis of Deviance Table

Model 1: chd ~ tobacco + ldl + famhist + typea + age

Model 2: chd ~ sbp + tobacco + ldl + adiposity + famhist + typea + obesity + alcohol + age

Resid.	Df	Resid.	Dev Df	Deviance	Pr(>Chi)
1		456		475.69	
2	1	452	1	472.14	3.5455 0.471

- Here it seems that we would prefer the selected model.

Prediction I

- Recall that

$$g(\mathbb{E}[Y|X_1 = x_1, \dots, X_{p-1} = x_{p-1}]) = \beta_0 + \beta_1 x_1 + \dots + \beta_{p-1} x_{p-1}$$

- Predicted observation for the current model:

$$\hat{Y} = g^{-1}(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \dots + \hat{\beta}_{p-1} x_{p-1})$$

- Logistic regression

$$\hat{Y} = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \dots + \hat{\beta}_{p-1} x_{(p-1)}}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \dots + \hat{\beta}_{p-1} x_{(p-1)}}}$$

- Poisson regression

$$\hat{y} = \exp\{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \dots + \hat{\beta}_{p-1} x_{p-1}\}$$

Prediction II

- In R we can obtain a prediction for both $\hat{\eta}$ and $\hat{\mu} = g^{-1}(\hat{\eta})$:

```
## back to the hcrabs example
head(predict(model1, type = "link"), 4)
  1          2          3          4
1.3720122 0.4343135 0.9308087 0.7861229
exp(head(predict(model1, type = "link"), 4))
  1          2          3          4
3.943277 1.543903 2.536560 2.194870
head(predict(model1, type = "response"), 4)
  1          2          3          4
3.943277 1.543903 2.536560 2.194870
```

Prediction III

- Since $\hat{\beta} \stackrel{\text{approx}}{\sim} \mathcal{N}(\beta, \mathcal{I}(\beta)^{-1})$ it is possible to construct (approximate) confidence intervals for η .
- Since $\hat{\eta} = X\hat{\beta}$:

$$\hat{V}(\hat{\eta}) = X\hat{V}(\hat{\beta})X^\top$$

- Confidence interval for η_i (confidence level: $1-\alpha$):

$$\left(\hat{\eta}_i + z_{\alpha/2} \left(\hat{V}(\hat{\eta}) \right)_{ii}, \hat{\eta}_i + z_{1-\alpha/2} \left(\hat{V}(\hat{\eta}) \right)_{ii} \right)$$

- We can then back-transform the lower and upper bound of the confidence interval for the linear predictor to obtain (approximate) confidence intervals for μ_i , since $\hat{\mu}_i = g^{-1}(\hat{\eta}_i)$:

$$\left(g^{-1} \left(\hat{\eta}_i + z_{\alpha/2} \left(\hat{V}(\hat{\eta}) \right)_{ii} \right), g^{-1} \left(\hat{\eta}_i + z_{1-\alpha/2} \left(\hat{V}(\hat{\eta}) \right)_{ii} \right) \right)$$

Prediction IV

- In R:

```
lpred <- predict(model1, type = "link", se.fit=TRUE)
cbind(lpred$fit[1:4] + qnorm(.025)*lpred$se.fit[1:4],
      lpred$fit[1:4] + qnorm(.975)*lpred$se.fit[1:4])
 [,1]      [,2]
 1 1.2716445 1.4723799
 2 0.2372926 0.6313344
 3 0.8315315 1.0300860
 4 0.6628823 0.9093635

exp(cbind(lpred$fit[1:4] + qnorm(.025)*lpred$se.fit[1:4],
          lpred$fit[1:4] + qnorm(.975)*lpred$se.fit[1:4]))
 [,1]      [,2]
 1 3.566713 4.359598
 2 1.267812 1.880118
 3 2.296834 2.801307
 4 1.940377 2.482742
```

GLM residuals I

- Residuals represent the difference between the data and the model and are essential to explore the adequacy of the model.
- In the Gaussian case, the residuals are $r_i = y_i - \hat{y}_i$.
- These are called **response residuals** for GLMs, but since the variance of the response is not constant for most GLMs, some modification is necessary.
- The **Pearson residuals** are comparable to the standardized residuals used in linear models and is defined as:

$$r_i^P = \frac{y_i - \hat{y}_i}{\sqrt{\hat{V}_i}}$$

GLM residuals II

- The deviance residuals r_i^D

$$D = \sum_{i=1}^n (r_i^D)^2 = \sum_{i=1}^n d_i$$

and

$$r_i^D = \text{sign}(y_i - \hat{y}_i) \sqrt{d_i}$$

- For example in the Poisson regression:

$$r_i^D = \text{sign}(y_i - \hat{y}_i) \sqrt{2(y_i \log y_i / \hat{y}_i - y_i + \hat{y}_i)}$$

- (notice that if $y_i = 0$ $y_i \log(y_i / \hat{y}_i)$ is replaced by 0, i.e. its value in the limit for $y \rightarrow 0$)

GLM residuals III

- (Crab data) We can obtain the deviance residuals as:

```
residuals(model1)[1:5]
```

1	2	3	4	5
1.7903699	-1.7572153	3.1414336	-2.0951707	0.6101256

These are the default choice of residuals.

- The Pearson residuals are:

```
residuals(model1, "pearson")[1:5]
```

1	2	3	4	5
2.0428979	-1.2425388	4.0582724	-1.4815094	0.6455594

GLM residuals IV

- If we use

```
model1$residuals[1:5]
```

1

2

3

4

5

1.0287693 -1.0000000 2.5481128 -1.0000000 0.3790497

we obtain the working residuals, i.e.

$$\hat{\eta}_i + (y_i - \hat{\mu}_i) \frac{d\eta_i}{d\mu_i}$$

- For linear models, the plot of residuals against fitted values is probably the single most valuable graphic for model diagnostics.
- Which residuals? Which scale (linear predictor or original scale?)

GLM residuals V

- For GLMs, we must decide on the appropriate scale for the fitted values. Usually, it is better to plot the fitted linear predictors .

$$\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_{p-1} x_{i,p-1}$$

rather than the predicted responses \hat{y}_i .

- Scatter plot of Deviance residuals against x_j used to check whether any systematic relationship is present: if so include x_j in the model.
- Plot working residuals against linear predictor. Plot should be linear - if not the link function might be not correctly specified.

Example: Galapagos data I

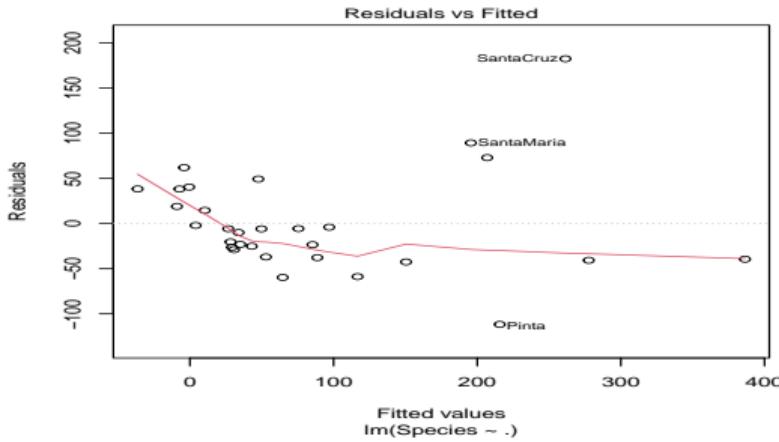
- For 30 Galápagos Islands, we have a count of the number of plant species found on each island and the number that are endemic to that island. We also have five geographic variables for each island.
- We model the number of species using normal linear regression:

```
data(gala, package="faraway")
gala <- gala[,-2]
```

- We throw out the Endemics variable (which falls in the second column of the dataframe) since we won't be using it in this analysis. We fit a linear regression and look at the residual vs. fitted plot:

```
mod1 <- lm(Species ~ . , gala)
plot(mod1, 1)
```

Example: Galapagos data II



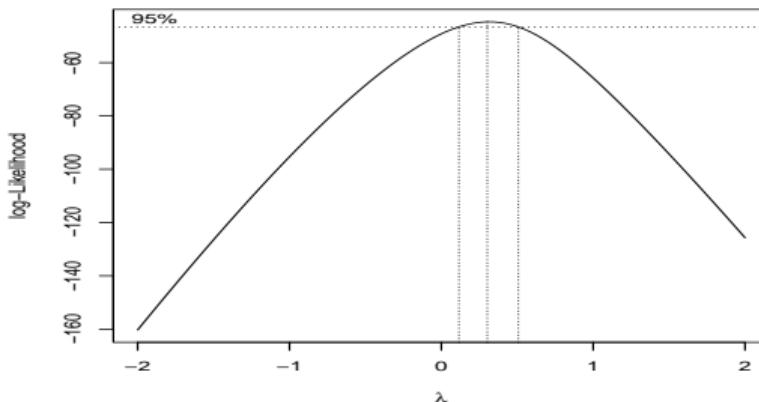
We see clear evidence of nonconstant variance

- the Box-Cox method reveals that a square-root transformation is a sensible transformation

```
library(MASS)
```

```
boxcox(mod1, plotit = TRUE)
```

Example: Galapagos data III



```
modt <- lm(sqrt(Species) ~ . , gala)
summary(modt)
```

Example: Galapagos data IV

Call:

```
lm(formula = sqrt(Species) ~ ., data = gala)
```

Residuals:

Min	1Q	Median	3Q	Max
-4.5572	-1.4969	-0.3031	1.3527	5.2110

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	3.3919243	0.8712678	3.893	0.000690 ***
Area	-0.0019718	0.0010199	-1.933	0.065080 .
Elevation	0.0164784	0.0024410	6.751	5.55e-07 ***
Nearest	0.0249326	0.0479495	0.520	0.607844
Scruz	-0.0134826	0.0097980	-1.376	0.181509
Adjacent	-0.0033669	0.0008051	-4.182	0.000333 ***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

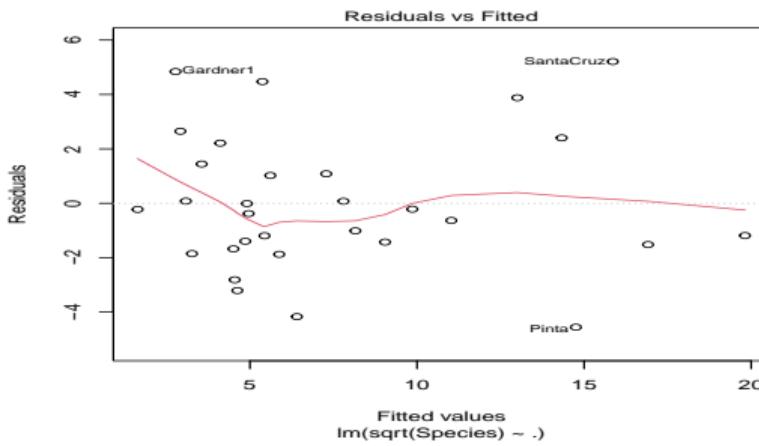
Residual standard error: 2.774 on 24 degrees of freedom

Example: Galapagos data V

Multiple R-squared: 0.7827, Adjusted R-squared: 0.7374

F-statistic: 17.29 on 5 and 24 DF, p-value: 2.874e-07

`plot(modt, 1)`



- We achieved this fit at the cost of transforming the response. This makes interpretation more difficult.

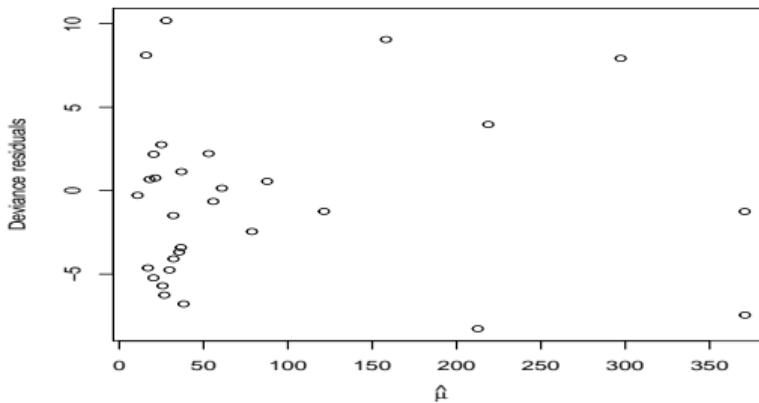
Example: Galapagos data VI

- Furthermore, some of the response values are quite small (single digits) which makes us question the validity of the normal approximation.
- This model may be adequate, but perhaps we can do better.

Example: Galapagos data VII

- Poisson regression

```
modp <- glm(Species ~ ., family=poisson, gala)
plot(residuals(modp) ~ predict(modp, type="response"),
     xlab=expression(hat(mu)),
     ylab="Deviance residuals")
```

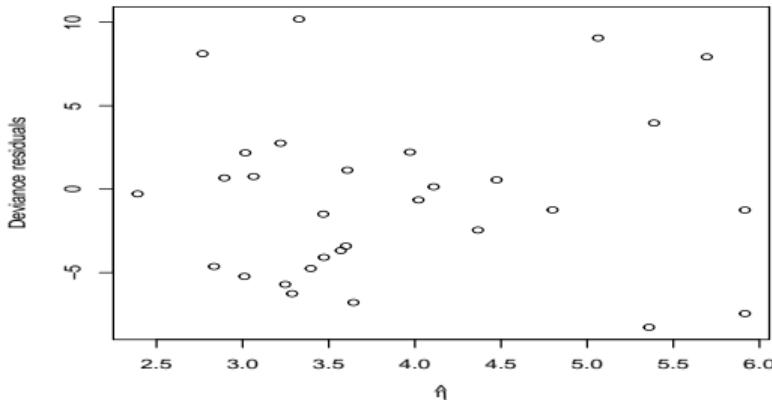


Example: Galapagos data VIII

- There are just a few islands with a large predicted number of species while most predicted response values are small. This makes it difficult to see the relationship between the residuals and the fitted values because most of the points are compressed on the left of the display.

```
plot(residuals(modp) ~ predict(modp, type="link"),
      xlab=expression(hat(eta)), ylab="Deviance residuals")
```

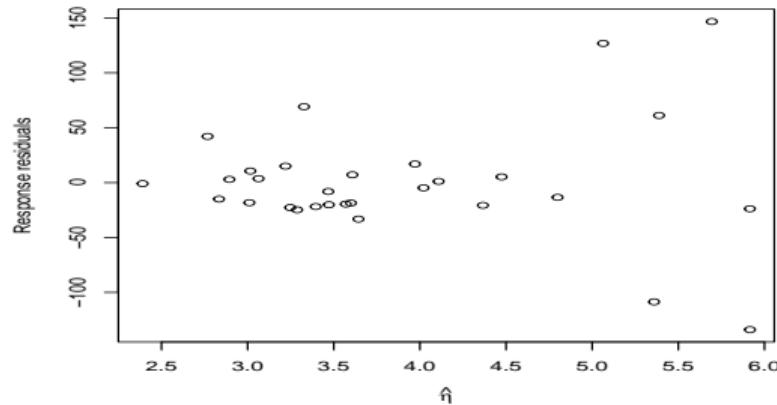
Example: Galapagos data IX



- If we use response residuals

```
plot(residuals(modp,type="response") ~ predict(modp,type="link"),
      xlab=expression(hat(eta)), ylab="Response residuals")
```

Example: Galapagos data X



We see a pattern of increasing variation consistent with the Poisson.

Classification¹³

¹³Material in these slides was heavily influenced by Chapter 17 of David Dal Piaz's book [book](#).

Classification I

- Making predictions in the context of the logistic regression.
- Based on the values of the predictors, an observation is classified as $Y = 1$ or as $Y = 0$.
- Suppose that we know

$$p(x_1, \dots, x_{p-1}) = \Pr[Y = 1 \mid X_1 = x_1, \dots, X_{p-1} = x_{p-1}]$$

and

$$1 - p(x_1, \dots, x_{p-1}) = \Pr[Y = 0 \mid X_1 = x_1, \dots, X_{p-1} = x_{p-1}].$$

- Rule: we classify an observation to the class (0 or 1) with the larger probability. In general, this result is called

the Bayes Classifier

$$C^B(x_1, \dots, x_{p-1}) = \operatorname{argmax}_k P[Y = k \mid X_1 = x_1, \dots, X_{p-1} = x_{p-1}].$$

Classification II

- For a binary response, that is,

$$C^B(x_1, \dots, x_{p-1}) = \begin{cases} 1 & p(x_1, \dots, x_{p-1}) > 0.5 \\ 0 & p(x_1, \dots, x_{p-1}) \leq 0.5 \end{cases}$$

- 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 know the necessary probabilities to directly use the Bayes classifier. Instead we'll have to use estimated probabilities.

$$\hat{C}^B(x_1, \dots, x_{p-1}) = \operatorname{argmax}_k \widehat{\Pr}[Y = k \mid X_1 = x_1, \dots, X_{p-1} = x_{p-1}].$$

Classification III

- In the case of a binary response since

$$\hat{p}(x_1, \dots, x_{p-1}) = 1 - \hat{p}(x_1, \dots, x_{p-1})$$

this becomes

$$\hat{C}(x_1, \dots, x_{p-1}) = \begin{cases} 1 & \hat{p}(x_1, \dots, x_{p-1}) > 0.5 \\ 0 & \hat{p}(x_1, \dots, x_{p-1}) \leq 0.5 \end{cases}$$

- To use logistic regression for classification, we first use logistic regression to obtain estimated probabilities, $\hat{p}(x_1, \dots, x_{p-1})$, then use these in conjunction with the above classification rule.
- Logistic regression is just one of many ways that these probabilities could be estimated.

spam Example I

- The spam dataset

```
spam <- read.table("https://web.stanford.edu/~hastie/ElemStatLearn/datasets/spam.data",
                    header=FALSE)

names(spam) <- c("make", "address", "all", "num3d", "our", "over", "remove",
                 "internet", "order", "mail", "receive", "will", "people", "report",
                 "addresses", "free", "business", "email", "you", "credit", "your",
                 "font", "num000", "money", "hp", "hpl", "george", "num650", "lab",
                 "labs", "telnet", "num857", "data", "num415", "num85", "technology",
                 "num1999", "parts", "pm", "direct", "cs", "meeting", "original",
                 "project", "re", "edu", "table", "conference", "charSemicolon",
                 "charRoundbracket", "charSquarebracket", "charExclamation", "charDollar",
                 "charHash", "capitalAve", "capitalLong", "capitalTotal", "type")
```

- Created in the late 1990s at Hewlett-Packard Labs, it contains 4601 emails, of which 1813 are considered spam. The remaining are not spam.

spam Example II

- The response variable, type, is a **factor** with levels that label each email as spam or email.
- nonspam will be the reference level, $Y = 0$,

```
is.factor(spam$type)
```

```
[1] TRUE
```

```
levels(spam$type)
```

```
[1] "email" "spam"
```

- test-train splitting of the data

```
set.seed(42)
```

```
spam_idx <- sample(nrow(spam), 2000)
```

```
spam_trn <- spam[spam_idx, ]
```

```
spam_tst <- spam[-spam_idx, ]
```

spam Example III

- We fit four logistic regressions, each more complex than the previous. Note that we're suppressing two warnings.

```
fit_caps <- glm(type ~ capitalTotal,  
                 data <- spam_trn, family = binomial)  
fit_selected <- glm(type ~ edu + money + capitalTotal + charDollar,  
                      data = spam_trn, family = binomial)  
fit_additive <- glm(type ~ .,  
                      data = spam_trn, family = binomial)  
fit_over <- glm(type ~ capitalTotal*(.),  
                  data = spam_trn, family = binomial, maxit = 120)
```

- When we receive this warning, we should be highly suspicious of the parameter estimates.
`message("Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred")`
- However, the model can still be used to create a classifier, and we will evaluate that classifier on its own merits.
- We also, "suppressed" the warning:
`message("Warning: glm.fit: algorithm did not converge")`

spam Example IV

- In reality, we didn't actually suppress it, but instead changed 'maxit' to '120', when fitting the model `fit_over` to allow the iteratively reweighted least squares algorithm to converge when fitting the model.

Evaluating Classifiers I

- The metric we'll be most interested in for evaluating the overall performance of a classifier is the **misclassification rate**.

$$\text{Misclass}(\hat{C}, \text{Data}) = \frac{1}{n} \sum_{i=1}^n I(y_i \neq \hat{C}(x_{i,1}, \dots, x_{i,p-1}))$$

$$I(y_i \neq \hat{C}(x_{i,1}, \dots, x_{i,p-1})) = \begin{cases} 0 & y_i = \hat{C}(x_{i,1}, \dots, x_{i,p-1}) \\ 1 & y_i \neq \hat{C}(x_{i,1}, \dots, x_{i,p-1}) \end{cases}$$

- When using this metric on the training data, it will have the same issues as RSS did for ordinary linear regression, that is, it will only go down.

Evaluating Classifiers II

```
# training misclassification rate
mean(ifelse(predict(fit_caps) > 0, "spam", "email") != spam_trn$type)
[1] 0.3515

mean(ifelse(predict(fit_selected) > 0, "spam", "email") != spam_trn$type)
[1] 0.2015

mean(ifelse(predict(fit_additive) > 0, "spam", "email") != spam_trn$type)
[1] 0.0735

mean(ifelse(predict(fit_over) > 0, "spam", "email") != spam_trn$type)
[1] 0.0525
```

- Because of this, training data isn't useful for evaluating, as it would suggest that we should always use the largest possible model, when in reality, that model is likely overfitting.
- Recall, a model that is too complex will overfit. A model that is too simple will underfit. (We're looking for something in the middle.)
- To overcome this, we'll use cross-validation but this time we'll cross-validate the misclassification rate.

Evaluating Classifiers III

- We'll use the `cv.glm` function from the `boot` library.
- Arguments
 - the data (in this case training),
 - a model fit via `glm`, K the number of folds.

See `?cv.glm` for details.

- Essentially we'll repeat the following process 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 misclassification rates. By only needing to refit the model 5 times, instead of n times, we will save a lot of computation time.

Evaluating Classifiers IV

```
library(boot)
set.seed(1)
cv.glm(spam_trn, fit_caps, K = 5)$delta[1]
[1] 0.2184288
cv.glm(spam_trn, fit_selected, K = 5)$delta[1]
[1] 0.1519643
cv.glm(spam_trn, fit_additive, K = 5)$delta[1]
[1] 0.06775837
cv.glm(spam_trn, fit_over, K = 5)$delta[1]
[1] 0.102255
```

- Based on these results, `fit_caps` and `fit_selected` are underfitting relative to `fit_additive`. Similarly, `fit_over` is overfitting relative to `fit_additive`. Thus, based on these results, we prefer the classifier created based on the logistic regression fit and stored in `fit_additive`.
- Going forward, to evaluate and report on the efficacy of this classifier, we'll use the test dataset.

Evaluating Classifiers V

- To quickly summarize how well this classifier works, we'll create a confusion matrix.

		Actual	
		False (0)	True (1)
Predicted	False (0)	True Negative (TN)	False Negative (FN)
	True (1)	False Positive (FP)	True Positive (TP)

It further breaks down the classification errors into false positives and false negatives. We write an R function to create the confusion matrix:

```
make_conf_mat <- function(predicted, actual) {  
  table(predicted = predicted, actual = actual)  
}
```

Evaluating Classifiers VI

- Let's explicitly store the predicted values of our classifier on the test dataset.

```
spam_tst_pred <- ifelse(  
    predict(fit_additive, spam_tst) > 0,  
    "spam", "email")  
  
spam_tst_pred <- ifelse(  
    predict(fit_additive, spam_tst, type = "response") > 0.5,  
    "spam", "email")
```

- The previous two lines of code produce the same predictions, since

$$\beta_0 + \beta_1 x_1 + \dots + \beta_{p-1} x_{p-1} = 0 \iff p(x_1, \dots, x_{p-1}) = 0.5$$

Evaluating Classifiers VII

- We create a confusion matrix.

```
conf_mat_50 <- make_conf_mat(predicted = spam_tst$pred,  
                                actual = spam_tst$type); conf_mat_50
```

actual

predicted	email	spam
email	1526	111
spam	82	882

- We have a *Prevalence* (total number of cases) of:

$$\text{Prev} = \frac{P}{\text{Total Obs}} = \frac{\text{TP} + \text{FN}}{\text{Total Obs}}$$

```
table(spam_tst$type) / nrow(spam_tst)
```

Evaluating Classifiers VIII

```
email      spam  
0.6182238 0.3817762
```

- First, note that for a classifier to be reasonable, it needs to outperform the obvious classifier of simply classifying all observations to the majority class.
- In this case, classifying everything as non-spam for a test misclassification rate of 0.382
- Next, we can see that using the classifier created from `fit_additive`, only a total of $111 + 82 = 193$ from the total of 2601 email in the test set are misclassified. Overall, the accuracy in the test set is

```
mean(spam_tst$pred == spam_tst$type)  
[1] 0.9257978
```

In other words, the test misclassification is

```
mean(spam_tst$pred != spam_tst$type)  
[1] 0.07420223
```

This seems like a decent classifier...

Evaluating Classifiers IX

- However, are all errors created equal? In this case, absolutely not.
- 82 non-spam emails marked as spam (false positives) are a problem.
- On the other hand, 111 spam email that would make it to an inbox (false negatives) are easily dealt with.
- Beside misclassification rate (or accuracy), we'll define two additional metrics: sensitivity and specificity.
- (Many!) other metrics that can be considered.

Sensitivity and Specificity I

Sensitivity

- Sensitivity is essentially the true positive rate. So when sensitivity is high, the number of false negatives is low.

$$\text{Sens} = \text{True Positive Rate} = \frac{\text{TP}}{\text{P}} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Here we have an R function to calculate the sensitivity based on the confusion matrix.

- R code

```
get_sens <- function(conf_mat) {  
  conf_mat[2, 2] / sum(conf_mat[, 2])  
}
```

Sensitivity and Specificity II

Specificity

- Specificity is essentially the true negative rate. So when specificity is high, the number of false positives is low.

$$\text{Spec} = \text{True Negative Rate} = \frac{\text{TN}}{\text{N}} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

- R code

```
get_spec <- function(conf_mat) {  
  conf_mat[1, 1] / sum(conf_mat[, 1])  
}
```

Sensitivity and Specificity III

- We calculate both based on the confusion matrix we had created for our classifier.

```
get_sens(conf_mat_50) # true positive rate
```

```
[1] 0.8882175
```

```
get_spec(conf_mat_50) # true negative rate
```

```
[1] 0.949005
```

- We had created this classifier using a probability of 0.5 as a "cutoff"
- We can modify this cutoff and improve sensitivity or specificity
- The price to pay? The overall accuracy (misclassification rate)

$$\hat{C}(x_1, \dots, x_{p-1}) = \begin{cases} 1 & \hat{p}(x_1, \dots, x_{p-1}) > c \\ 0 & \hat{p}(x_1, \dots, x_{p-1}) \leq c \end{cases}$$

- Additionally, if we change the cutoff to improve sensitivity, we'll decrease specificity, and vice versa.

Sensitivity and Specificity IV

- First let's see what happens when we lower the cutoff from 0.5 to 0.1 to create a new classifier, and thus new predictions.

```
spam_tst_pred_10 = ifelse(predict(fit_additive, spam_tst,  
                                type = "response") > 0.1, "spam", "email")
```

- This is essentially *decreasing* the threshold for an email to be labeled as spam: *more* emails will be labeled as spam (as seen in the confusion matrix)

```
conf_mat_10<-make_conf_mat(predicted = spam_tst_pred_10,  
                             actual = spam_tst$type); conf_mat_10  
  
actual
```

predicted	email	spam
email	1191	19
spam	417	974

Sensitivity and Specificity V

Unfortunately, while this does greatly reduce false negatives, false positives have increased spectacularly. We see this reflected in the sensitivity and specificity.

```
get_sens(conf_mat_10) # true positive rate  
[1] 0.9808661  
  
get_spec(conf_mat_10) # true negative rate  
[1] 0.7406716  
  
# we had  
c(get_sens(conf_mat_50), get_spec(conf_mat_50))  
[1] 0.8882175 0.9490050
```

Sensitivity and Specificity VI

- This classifier, using 0.1 instead of 0.5 has a higher sensitivity, but a much lower specificity. Clearly, we should have moved the cutoff in the other direction. Let's try 0.9.

```
spam_tst_pred_90 <- ifelse(predict(fit_additive,spam_tst,  
                                type = "response") > 0.9, "spam", "email")
```

This is essentially *increasing* the threshold for an email to be labeled as spam, so far fewer emails will be labeled as spam. Again, we see that in the confusion matrix.

```
conf_mat_90 <- make_conf_mat(predicted = spam_tst_pred_90,  
                               actual = spam_tst$type); conf_mat_90  
  
actual
```

predicted	email	spam
email	1589	373
spam	19	620

Sensitivity and Specificity VII

- This is the result we're looking for. We have far fewer false positives. While sensitivity is greatly reduced, specificity has gone up.

```
get_sens(conf_mat_90) # true positive rate
```

```
[1] 0.6243706
```

```
get_spec(conf_mat_90) # true negative rate
```

```
[1] 0.9881841
```

```
# we had
```

```
c(get_sens(conf_mat_50), get_spec(conf_mat_50))
```

```
[1] 0.8882175 0.9490050
```

```
c(get_sens(conf_mat_10), get_spec(conf_mat_10))
```

```
[1] 0.9808661 0.7406716
```

Table of content I

- 1 Introduction and examples
- 2 Reminders from Basic Probability
- 3 Optimal Prediction
- 4 Empirical covariance and correlation
- 5 Simple linear regression model I
 - The lm function
 - Bias and Variance of Parameter Estimates
 - Prediction
 - The sum of squares
 - Residuals
- 6 The Gaussian-noise simple linear regression model
- 7 Multiple Linear Regression
 - Sampling distribution
- 8 Model selection

Table of content II

- Variable selection

9 Categorical predictors

- Factors with More Than Two Levels

10 Model checking

11 Transformations

12 Colinearity

13 Influence

14 GLM

15 Classification