

UNIVERSITY *of* WASHINGTON

# Data Science UW

# Methods for Data

# Analysis

---

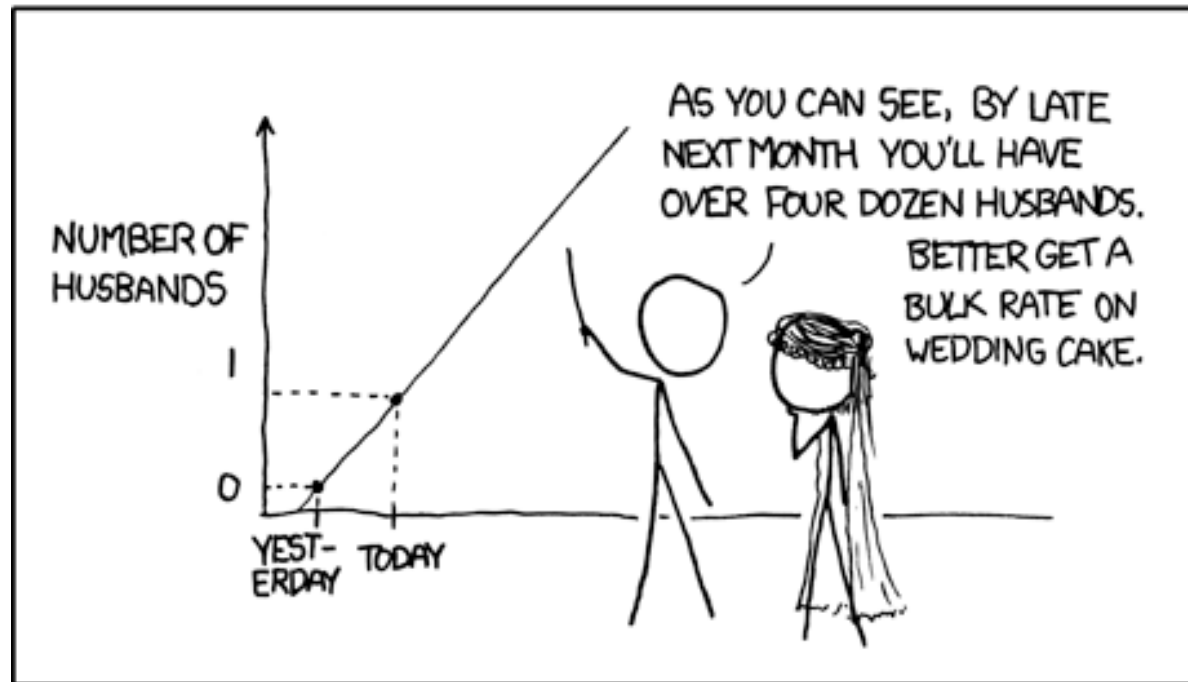
SVD, More Regression and Intro to Time Series

Lecture 6

Nick McClure



## MY HOBBY: EXTRAPOLATING



W

# Topics

---

- > Review
- > Linear Algebra overview
- > Decomposition Methods
- > Lasso Regression
- > Ridge Regression
- > Logistic Regression
- > Binary Classification
- > Time Series



# Review

- > Linear Regression
- > Multiple Regression
- > Introduction to Python / iPython
- > Introduction to Graph Theory
  - Gephi Visualization
  - Degree Distribution Tests



# Linear Algebra

- > Matrix: a rectangular array of values, with dimensions  $n$  by  $m$  ( $n$  rows,  $m$  columns).
- > Vector: a one dimensional array of values ( $n \mid m = 1$ ).
- > Square matrix: a  $n \times n$  matrix.
- > Identity matrix: a square matrix with 1's on the diagonal and 0's elsewhere.
- > R demo.



# Linear Algebra

## > Algebraic Properties of Matrices:

- Add/subtract matrices: Must be of the same dimensions
- Multiplication of matrices:
  - > Inner dimensions must match.

$$\begin{bmatrix} \boxed{a} & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \times \begin{bmatrix} \boxed{j} & k & l \\ m & n & o \\ p & q & r \end{bmatrix} = \begin{bmatrix} \boxed{aj + bm + cp} & ak + bn + cq & al + bo + cr \\ dj + em + fp & dk + en + fq & dl + eo + fr \\ gj + hm + ip & gk + hn + iq & gl + ho + ir \end{bmatrix}$$

$$- [n \times m] * [m \times p] = [n \times p]$$



- Note that matrix multiplication is not commutative



# Linear Algebra

- > Transpose (given an element in position i,j, the transpose has the same element in position j,i.)
- > Inverse:
  - Just like the multiplicative inverse of n is 1/n, matrices also have multiplicative inverses:

$$A_{n \times n} \cdot A_{n \times n}^{-1} = I_{n \times n}$$

$$A_{n \times n}^{-1} \cdot A_{n \times n} = I_{n \times n}$$

$$\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \cdot \begin{bmatrix} -2 & \frac{3}{2} \\ 1 & -\frac{1}{2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



# Linear Algebra

> For a 2x2 matrix:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ c & d \end{bmatrix}$$

- > What if  $(ad-bc) = 0$ ? That means that  $ad=bc$  or  $a/c = b/d$ .
- > If  $a/c = b/d$ , then one of the columns is a multiple of the other!
- > These columns are dependent on each other.
  - If these were columns in our numerical data frame, then one column would be a multiple of the other.
  - Examples: Using Fahrenheit and Celsius as separate predictors.





# Linear Algebra

- > Eigenvalues: Given a  $n \times n$  matrix,  $A$ ,  $\lambda$  is an eigenvalue if there exists a vector  $X$  such that:

$$AX = \lambda X$$

- > Finding the eigenvectors of  $A$  involves lots of computation.
- > If  $A$  rotates and shifts a vector  $X$ , then we can think of eigenvalues as a geometric hinge on which the 'A' operation acts.
- > Eigenvalues have corresponding eigenvectors.
- > This may seem insignificant at the moment, but eigenvalues and eigenvectors play an important role in manipulating our data.



# Linear Algebra

- > Matrix Decompositions allow us to write a matrix,  $A$ , in many different forms.
- > The one that is the most used, is Singular Value Decomposition (SVD).
- > The SVD is a way to express a transformation from one  $n \times n$  space (the space  $A$  lies in) to another  $n \times n$  space by writing  $A$  as a product of three matrices.
- > This transformation is a type of right angle (orthogonal) transformation as we will see later.
- > These three matrices, say,  $X, Y, Z$ , ( $A = XYZ$ ), have very specific properties that we can use to our advantage when describing a data set.
- > R-demo.



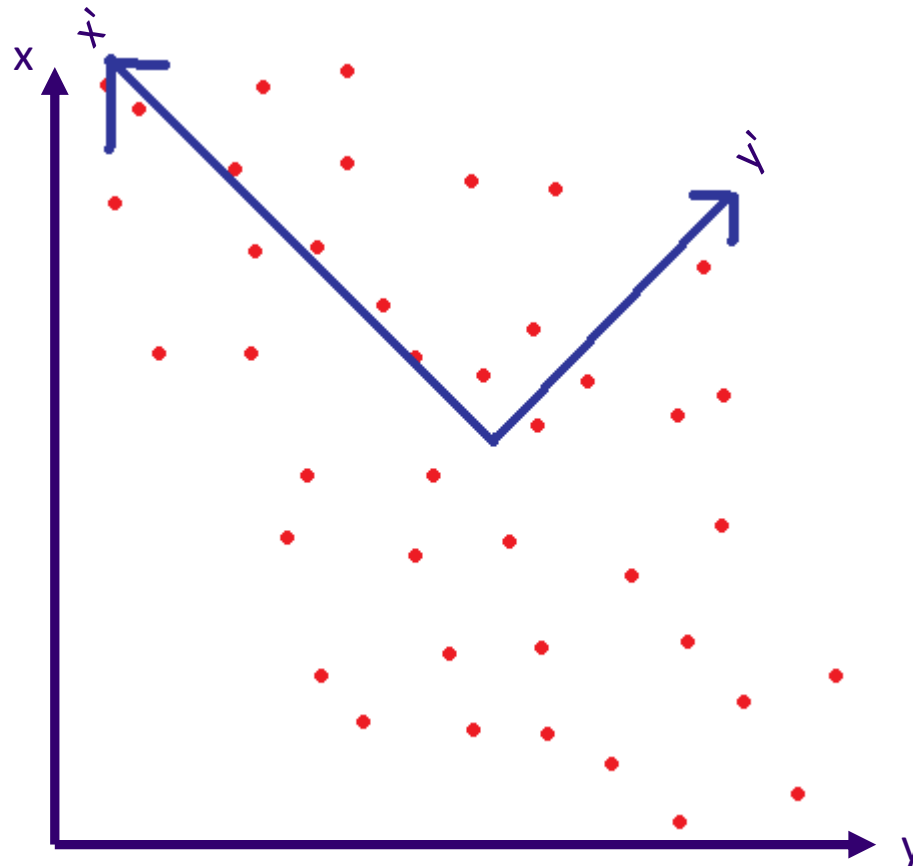
# Deriving Independent Features from Dependence

- > With larger data sets, we've seen that no matter the quality, we can find a explanatory feature.
- > If we consider our data as a matrix, we know that having dependent columns is a problem.
- > Solutions:
  - Remove columns that do not contain enough 'information'.
    - > Too much missing data.
    - > Low Variance.
  - Remove columns that are correlated
  - Maybe we can transform our axes such that our data is more independent?



# Possible Axis Transformations

- > If two variables are correlated, we can transform both of the axes to directions in which they are not correlated.
- > These new axes are called the Principal Components.



**W**

# SVD

- > This transformation is called the Singular Value Decomposition, or SVD.
- > It holds true for as many features (dimensions) as we wish to choose, up to the number of original dimensions.
- > Each of the new axes is some function of all the old axes.
- > The SVD assures us that:
  - The first axis explains the most variation, the second axis the most variation after the first, and so on.
  - All axes are right-angled to each other (orthogonal).
- > Usually, we keep less than the original amount of axes, so that we can reduce the amount of dimensions we have to keep track of.



# SVD

- > Know that instead of our original system:

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots$$

- > We now have the system:

$$y_i = \beta_0 + \beta_1 f_1(x_1, x_2, \dots) + \beta_2 f_2(x_1, x_2, \dots) + \dots$$

- > The  $f$  functions are called our principle components.
- > The  $f$  function outputs are guaranteed to be independent of each other.
- > We can no longer interpret our linear model coefficients!



# SVD

- > SVD returns the same amount of components as our number of features.
- > Since these are *all* orthogonal, the first few will explain much more variance than the last few axes. How do we decide how many to keep?
- > We look at the magnitude of the associated eigenvalues for each principal component.
- > R-demo.



# SVD

- > This seems like an awful lot of work for little improvement and loss of interpretability.
- > But note that we lost the dependence in the data set!
- > There are other applications as well...

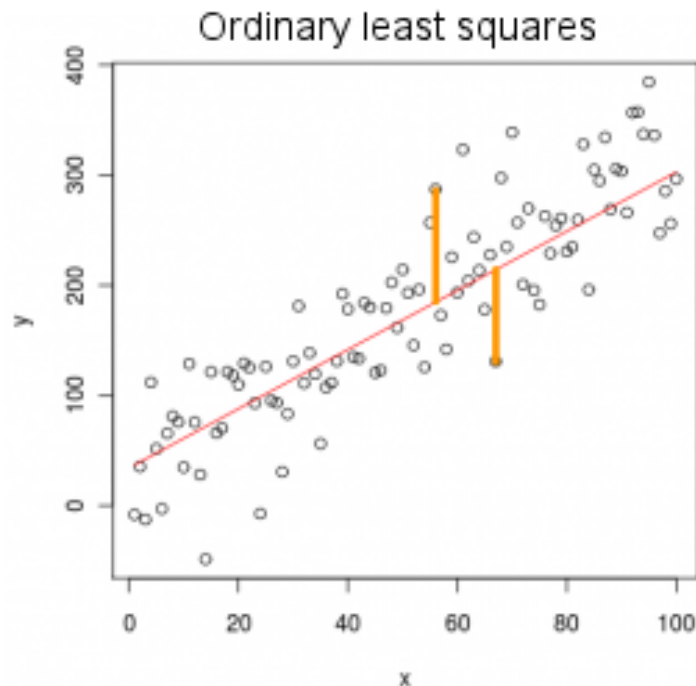




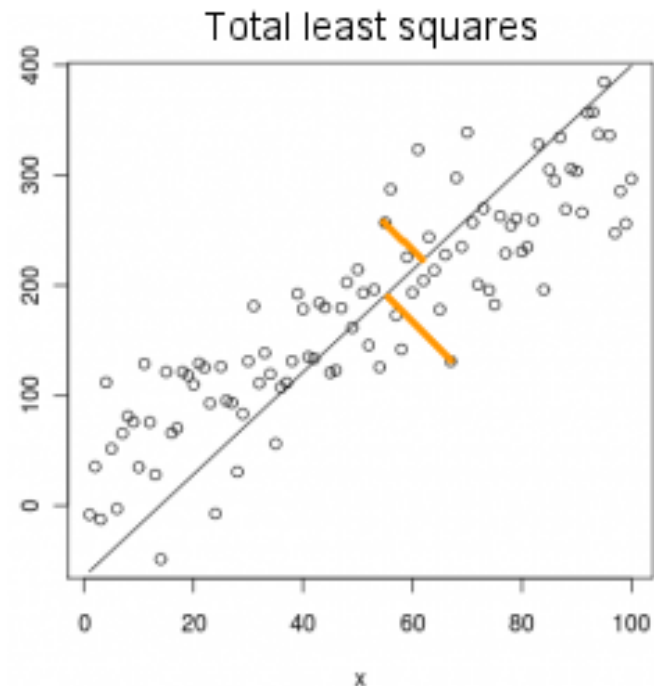
# SVD, as a type of regression

- > Also, looking at the first principal component, we can consider SVD as a new type of regression, which is called total least squares. (Also called Deming regression or PCA Regression)

Regressing y on x



SVD Primary Principal Component



- > R demo



# SVD, as a type of regression

- > When to use total least squares:
  - If we want to control for error in  $x$  as well as  $y$ .
  - We are minimizing the distance from the point to the line as opposed to the distance between the  $y$ -values.
  - R-squared doesn't really apply here, at least in the way we have defined it.



# SVD, as a way to compress information

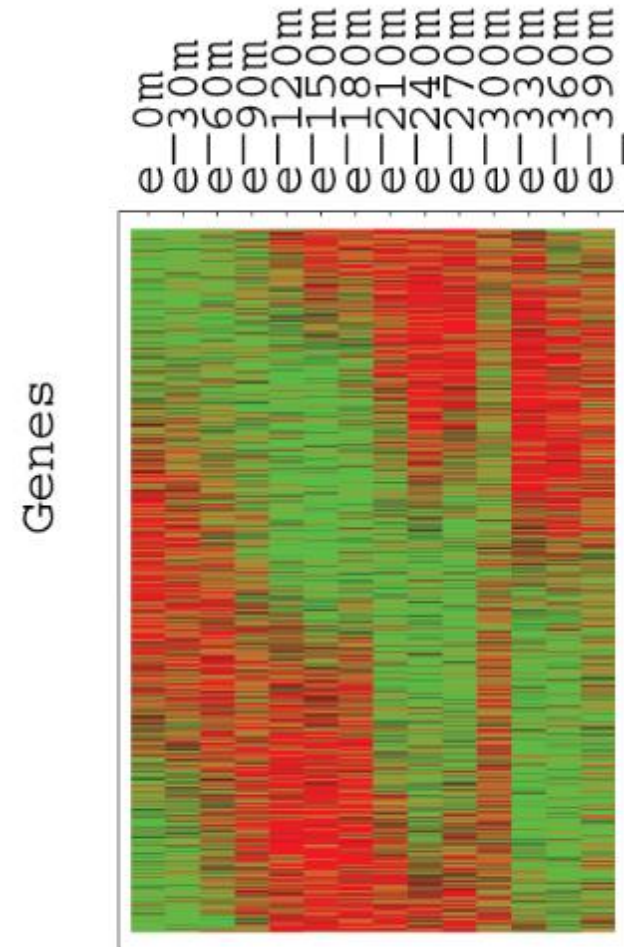
- > We can group together similar points via SVD and store them as multiples of principal components.

- > R-demo.



# SVD, as a way to cluster data

- > We can group together similar points via which SVD component is closest to representing original point.
- One of the most common uses is clustering individuals or genes as it pertains to RNA expression.
  - In the microarray to the right, red represents absence of expression and green represents over expression.
  - Each row is a gene (thousands of them) and each column is a sample (or patient).
- 



# Ridge Regression

- > Ridge regression is a way to limit the amount of independent variables in the regression.
- > Our regular least squares criterion minimizes the least squares of the error plus a regularization term that is a product of a constant and the sum of squared coefficients :

$$\min \sum (y - y_i)^2 + \alpha \sum \beta^2$$

- > Essentially this is preventing the partial slope terms from getting too large.



# Lasso Regression

- > Lasso regression is another way to limit the amount of independent variables in the regression.
- > Our regular least squares criterion minimizes the least squares of the error:

$$\min \sum (y - y_i)^2$$

- > Lasso regression minimizes the same with the addition of a 'regularization' term:

$$\min \sum (y - y_j)^2 \quad \text{Such that} \quad \sum |\beta_i| < \lambda$$

- > Here,  $y$  is the predicted for  $j$  points. There are  $i$  terms with beta coefficients. Lambda is a fixed value that limits the betas.

**W**

# Using Linear Regression to Predict Limited Dependent Variables

- > Let's say we wanted to predict if someone evacuated their home during hurricane Katrina.
- > R demo.



# Logistic Regression

- > The purpose of logistic regression is to use linear regression to predict a limited dependent variable.
- > Usually our dependent variable has 2 outcomes (1 or 0) or occurrence.
- > Examples:
  - Bank gives a yes (1) or no (0) outcome to loan applications.
  - Success/Failures of clinical trials.
  - Morbidity outcomes.
  - Marketing outcomes (will a user click on an add).
- > Logistic predictions will result in a probability of success.





# Logistic Regression

- > Logistic regression is also called the 'logit' model:
- > Original model:

$$y_i = \beta_0 + \beta_1 x_1 + \varepsilon_0$$

- > Logit model:

$$\ln \left[ \frac{p_i}{1 - p_i} \right] = \beta_0 + \beta_1 x_1 + \varepsilon_0$$



Log-odds-ratio

- > So estimated probabilities follow: (solving for p)

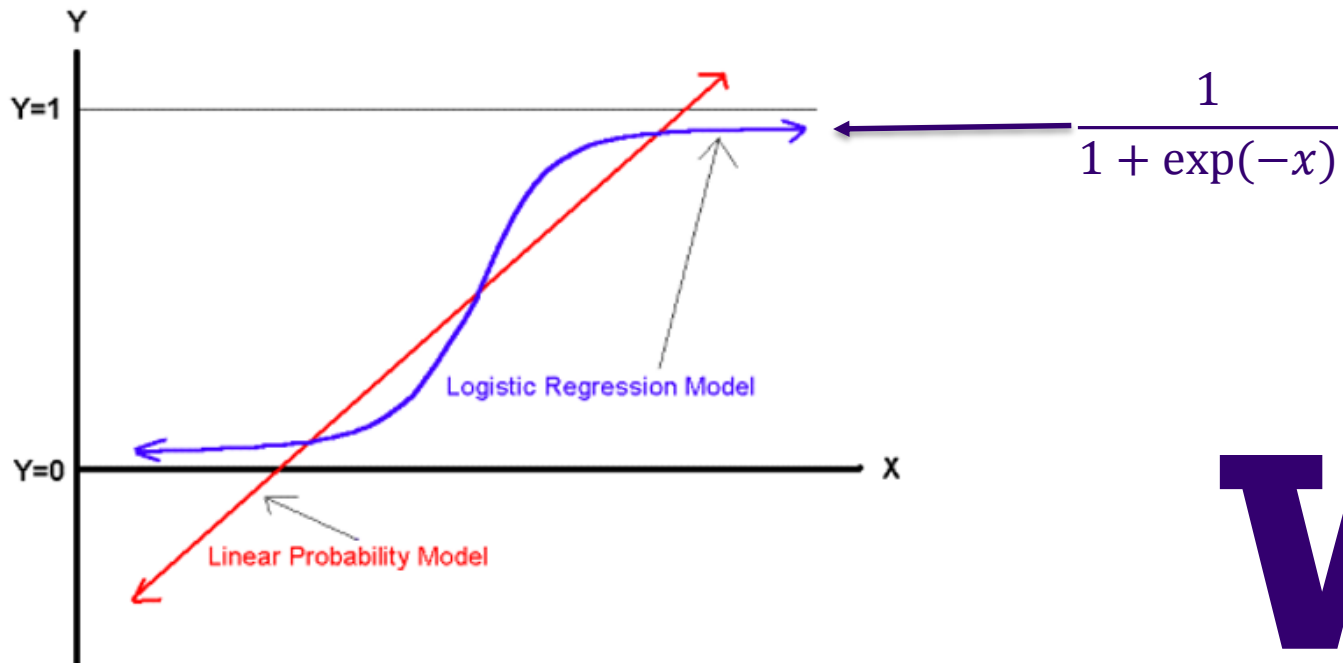
$$p_i = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 x_1))}$$



# Logistic Regression

$$p_i = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 x_1))}$$

- > As  $(\beta_0 + \beta_1 x_1)$  gets really big,  $p$  approaches 1.
- > As  $(\beta_0 + \beta_1 x_1)$  gets really small,  $p$  approaches 0.



W

# Logistic Regression

- > Differences between linear and logistic regression.
- > Predictions
  - Linear regression outcomes are unbounded.
  - Logistic regression outcomes are bounded between 0 and 1.

$$p_i = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 x_1))}$$

- > Error distribution
  - Linear regression errors are normally distributed.
  - Logistic regression errors are Bernoulli distributed.
- > R demo



# Time Series Modeling

- > Time series measurements are represented by observations over time:

$$Y = (Y_1, Y_2, Y_3, \dots, Y_T)$$

- > Stochastic Process is a process that evolves over time.
- > Regular statistical analysis is concerned with estimations of repeated samples.
- > With time series, we usually cannot measure repeatedly and have to observe over time how something changes.
  - E.g.: Mortality Rate, Temperature, ...
- > A stochastic process is said to be 'stationary' if there is no trend in the data.
  - This is a nice assumption because it implies the correlation of a process is fixed over time. I.e. any two points should have the same relationship.



# Time Series Modeling

- > Fields dedicated to time series:
  - Dynamical Systems
    - > Population growth, radioactive decay, chemical reactions,...
  - Signal Processing
    - > De-noising, deblurring...
  - Operations Research
    - > Industrial processes, quality control,...
  - Financial Analysis/Mathematics
    - > Financial Assets, Economic Indices,...
- > Time Series Analysis Objectives
  - Estimate True values in the presence of noise or trend
  - Forecast future values



# Time Series Modeling

- > Representations
  - Continuous Functions: Solutions to ODEs, PDEs...
- > Random processes
  - A collection of random variables that depend on the previous observation.
- > Sum of Periodic functions
  - Daily Trend + Weekly Trend + Seasonal Trend + ...
- > Arbitrary function of historical values
  - Moving Average, weighted averages, ...



# Moving Average

- > Given data in the form  $(t_i, x_i)$  where  $t_i < t_{i+1}$
- > Past value smoothing
  - Based on a set of past values.

$$x_{p.v.} = \sum_{i=T-p}^T w_i x_i \quad \sum w_i < 1$$

- > Moving window smoothing
  - Based on a surrounding window from current point

$$x_{wind} = \sum_{i=T-p/2}^{T+p/2} w_i x_i \quad \sum w_i < 1$$

- > Moving averages suffer from lag and smoothing extremes (amplitude error).
- > Moving averages are very sensitive to the window lengths
- > R-demo

**W**

# Assignment

---

## > Complete Homework 6:

- Perform Lasso-Logistic Regression on a subset of Microarray data.
  - > Description, dataset and homework hint on Moodle.
- Data comes from:
  - > <http://www.ncbi.nlm.nih.gov/pubmed/21532620>
- You should submit:
  - > A R-script.
- Read Introduction to Data Science, Chapter 16.
- Read two articles about p-values and reproducible research.

