Math Prefresher for Political Scientists

August 2018

Contents

About this Booklet

The Harvard Gov Prefresher is held each year in August. All relevant information is on our website: `https://projects.iq.harvard.edu/prefresher`. The 2018 Prefresher instructors are Shiro Kuriwaki and Yon Soo Park, and the faculty sponsor is Gary King.

This booklet is the "text" for the Prefresher, and it is the product of generations of Prefresher Instructors: Curt Signorino 1996-1997; Ken Scheve 1997-1998; Eric Dickson 1998-2000; Orit Kedar 1999; James Fowler 2000-2001; Kosuke Imai 2001-2002; Jacob Kline 2002; Dan Epstein 2003; Ben Ansell 2003-2004; Ryan Moore 2004-2005; Mike Kellermann 2005-2006; Ellie Powell 2006-2007; Jen Katkin 2007-2008; Patrick Lam 2008-2009; Viridiana Rios 2009-2010; Jennifer Pan 2010-2011; Konstantin Kashin 2011-2012; Soledad Prillaman 2013; Stephen Pettigrew 2013-2014; Anton Strezhnev 2014-2015; Mayya Komisarchik 2015-2016; Connor Jerzak 2016-2017; Shiro Kuriwaki 2017-2018; Yon Soo Park 2018-

We transitioned the booklet into a Rmarkdown (bookdown) document and into a public github repo in 2018. As we update this version, any bug reports or fixes would be greatly appreciated.

Thanks to Juan Dodyk (juandodyk), Hunter Rendleman (hrendleman), and Tyler Simko (tylersimko) for contributing to the booklet for corrections and improvements as students.

Pre-PreFresher Exercises

Before our first meeting on August 20th, please try solving these questions. They are a sample of the very beginning of each math section. We have provided links to the parts of the book you can read if the concepts are new to you.

The goal of this "pre"-prefresher assignment is not to intimidate you but to set common expectations so you can make the most out of the actual Prefresher. Even if you do not understand some or all of these questions after skimming through the linked sections, your effort will pay off and you will be better prepared for the math prefresher. We are also open to adjusting these expectations based on feedback (this class is for *you*), so please do not hesitate to write to the instructors for feedback.

Linear Algebra

Vectors

Define the vectors $u = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$, $v = \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix}$, and the scalar $c = 2$. Calculate the following:

1. $u + v$
2. cv
3. $u \cdot v$

If you are having trouble with these problems, please review Section 1.1 "Working with Vectors" in Chapter 1.

Are the following sets of vectors linearly independent?

1. $u = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$, $v = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$
2. $u = \begin{pmatrix} 1 \\ 2 \\ 5 \end{pmatrix}$, $v = \begin{pmatrix} 3 \\ 7 \\ 9 \end{pmatrix}$
3. $a = \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix}$, $b = \begin{pmatrix} 3 \\ -4 \\ -2 \end{pmatrix}$, $c = \begin{pmatrix} 5 \\ -10 \\ -8 \end{pmatrix}$ (this requires some guesswork)

If you are having trouble with these problems, please review Section 1.2.

Matrices

$$\mathbf{A} = \begin{pmatrix} 7 & 5 & 1 \\ 11 & 9 & 3 \\ 2 & 14 & 21 \\ 4 & 1 & 5 \end{pmatrix}$$

What is the dimensionality of matrix $\mathbf{A}$?

What is the element a_{23} of $\mathbf{A}$?

Given that

$$\mathbf{B} = \begin{pmatrix} 1 & 2 & 8 \\ 3 & 9 & 11 \\ 4 & 7 & 5 \\ 5 & 1 & 9 \end{pmatrix}$$

What is $\mathbf{A} + \mathbf{B}$?

Given that

$$\mathbf{C} = \begin{pmatrix} 1 & 2 & 8 \\ 3 & 9 & 11 \\ 4 & 7 & 5 \end{pmatrix}$$

What is $\mathbf{A} + \mathbf{C}$?

Given that

$$c = 2$$

What is $c\mathbf{A}$?

If you are having trouble with these problems, please review Section 1.3.

Operations

Summation

Simplify the following

1. $\sum\limits_{i=1}^{3} i$

2. $\sum_{k=1}^{3}(3k+2)$

3. $\sum_{i=1}^{4}(3k+i+2)$

Products

1. $\prod_{i=1}^{3} i$

2. $\prod_{k=1}^{3}(3k+2)$

To review this material, please see Section 2.1.

Logs and exponents

Simplify the following

1. 4^2
2. $4^2 2^3$
3. $\log_{10} 100$
4. $\log_2 4$
5. $\log e$, where log is the natural log (also written as ln) – a log with base e, and e is Euler's constant
6. $e^a e^b e^c$, where a, b, c are each constants
7. $\log 0$
8. e^0
9. e^1
10. $\log e^2$

To review this material, please see Section 2.3

Limits

Find the limit of the following.

1. $\lim_{x \to 2}(x-1)$
2. $\lim_{x \to 2} \frac{(x-2)(x-1)}{(x-2)}$
3. $\lim_{x \to 2} \frac{x^2-3x+2}{x-2}$

To review this material please see Section 3.3

Calculus

For each of the following functions $f(x)$, find the derivative $f'(x)$ or $\frac{d}{dx}f(x)$

1. $f(x) = c$
2. $f(x) = x$
3. $f(x) = x^2$
4. $f(x) = x^3$
5. $f(x) = 3x^2 + 2x^{1/3}$
6. $f(x) = (x^3)(2x^4)$

For a review, please see Section 4.1 - 4.2

Optimization

For each of the followng functions $f(x)$, does a maximum and minimum exist in the domain $x \in \mathbf{R}$? If so, for what are those values and for which values of x?

1. $f(x) = x$
2. $f(x) = x^2$
3. $f(x) = -(x-2)^2$

If you are stuck, please try sketching out a picture of each of the functions.

Probability

1. If there are 12 cards, numbered 1 to 12, and 4 cards are chosen, how many distinct possible choices are there? (unordered, without replacement)
2. Let $A = \{1, 3, 5, 7, 8\}$ and $B = \{2, 4, 7, 8, 12, 13\}$. What is $A \cup B$? What is $A \cap B$? If A is a subset of the Sample Space $S = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, what is the complement A^C?
3. If we roll two fair dice, what is the probability that their sum would be 11?
4. If we roll two fair dice, what is the probability that their sum would be 12?

For a review, please see Sections 6.2 - 6.3.

Part I

Math

Chapter 1

Linear Algebra

Topics: • Working with Vectors • Linear Independence • Basics of Matrix Algebra • Square Matrices • Linear Equations • Systems of Linear Equations • Systems of Equations as Matrices • Solving Augmented Matrices and Systems of Equations • Rank • The Inverse of a Matrix • Inverse of Larger Matrices

1.1 Working with Vectors

Vector: A vector in n-space is an ordered list of n numbers. These numbers can be represented as either a row vector or a column vector:

$$\mathbf{v}\begin{pmatrix} v_1 & v_2 & \dots & v_n \end{pmatrix}, \mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}$$

We can also think of a vector as defining a point in n-dimensional space, usually $\mathbf{R}^n$; each element of the vector defines the coordinate of the point in a particular direction.

Vector Addition and Subtraction: If two vectors, $\mathbf{u}$ and $\mathbf{v}$, have the same length (i.e. have the same number of elements), they can be added (subtracted) together:

$$\mathbf{u} + \mathbf{v} = \begin{pmatrix} u_1 + v_1 & u_2 + v_2 & \cdots & u_k + v_n \end{pmatrix}$$

$$\mathbf{u} - \mathbf{v} = \begin{pmatrix} u_1 - v_1 & u_2 - v_2 & \cdots & u_k - v_n \end{pmatrix}$$

Scalar Multiplication: The product of a scalar c (i.e. a constant) and vector $\mathbf{v}$ is:

$$c\mathbf{v} = \begin{pmatrix} cv_1 & cv_2 & \dots & cv_n \end{pmatrix}$$

Vector Inner Product: The inner product (also called the dot product or scalar product) of two vectors $\mathbf{u}$ and $\mathbf{v}$ is again defined if and only if they have the same number of elements

$$\mathbf{u} \cdot \mathbf{v} = u_1 v_1 + u_2 v_2 + \cdots + u_n v_n = \sum_{i=1}^{n} u_i v_i$$

If $\mathbf{u} \cdot \mathbf{v} = 0$, the two vectors are orthogonal (or perpendicular).

Vector Norm: The norm of a vector is a measure of its length. There are many different ways to calculate the norm, but the most common is the Euclidean norm (which corresponds to our usual conception of distance in three-dimensional space):

$$||\mathbf{v}|| = \sqrt{\mathbf{v} \cdot \mathbf{v}} = \sqrt{v_1 v_1 + v_2 v_2 + \cdots + v_n v_n}$$

Example 1.1 (Vector Algebra)**.** Let $a = \begin{pmatrix} 2 & 1 & 2 \end{pmatrix}$, $b = \begin{pmatrix} 3 & 4 & 5 \end{pmatrix}$. Calculate the following:

1. $a - b$
2. $a \cdot b$

Exercise 1.1 (Vector Algebra)**.** Let $u = \begin{pmatrix} 7 & 1 & -5 & 3 \end{pmatrix}$, $v = \begin{pmatrix} 9 & -3 & 2 & 8 \end{pmatrix}$, $w = \begin{pmatrix} 1 & 13 & -7 & 2 & 15 \end{pmatrix}$, and $c = 2$. Calculate the following:

1. $u - v$
2. cw
3. $u \cdot v$
4. $w \cdot v$

1.2 Linear Independence

Linear combinations: The vector $\mathbf{u}$ is a linear combination of the vectors $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_k$ if

$$\mathbf{u} = c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 + \cdots + c_k \mathbf{v}_k$$

For example, $\begin{pmatrix} 9 & 13 & 17 \end{pmatrix}$ is a linear combination of the following three vectors: $\begin{pmatrix} 1 & 2 & 3 \end{pmatrix}$, $\begin{pmatrix} 2 & 3 & 4 \end{pmatrix}$, and $\begin{pmatrix} 3 & 4 & 5 \end{pmatrix}$. This is because $\begin{pmatrix} 9 & 13 & 17 \end{pmatrix} = (2) \begin{pmatrix} 1 & 2 & 3 \end{pmatrix} + (-1) \begin{pmatrix} 2 & 3 & 4 \end{pmatrix} + 3 \begin{pmatrix} 3 & 4 & 5 \end{pmatrix}$

Linear independence: A set of vectors $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_k$ is linearly independent if the only solution to the equation

$$c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 + \cdots + c_k \mathbf{v}_k = 0$$

is $c_1 = c_2 = \cdots = c_k = 0$. If another solution exists, the set of vectors is linearly dependent.

A set S of vectors is linearly dependent if and only if at least one of the vectors in S can be written as a linear combination of the other vectors in S.

Linear independence is only defined for sets of vectors with the same number of elements; any linearly independent set of vectors in n-space contains at most n vectors.

Since $\begin{pmatrix} 9 & 13 & 17 \end{pmatrix}$ is a linear combination of $\begin{pmatrix} 1 & 2 & 3 \end{pmatrix}$, $\begin{pmatrix} 2 & 3 & 4 \end{pmatrix}$, and $\begin{pmatrix} 3 & 4 & 5 \end{pmatrix}$, these 4 vectors constitute a linearly dependent set.

Example 1.2 (Linear Independence)**.** Are the following sets of vectors linearly independent?

1. $\begin{pmatrix} 2 & 3 & 1 \end{pmatrix}$ and $\begin{pmatrix} 4 & 6 & 1 \end{pmatrix}$
2. $\begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$, $\begin{pmatrix} 0 & 5 & 0 \end{pmatrix}$, and $\begin{pmatrix} 10 & 10 & 0 \end{pmatrix}$

Exercise 1.2 (Linear Independence)**.** Are the following sets of vectors linearly independent?

1.
$$\mathbf{v}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \mathbf{v}_2 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \mathbf{v}_3 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

2.
$$\mathbf{v}_1 = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}, \mathbf{v}_2 = \begin{pmatrix} -4 \\ 6 \\ 5 \end{pmatrix}, \mathbf{v}_3 = \begin{pmatrix} -2 \\ 8 \\ 6 \end{pmatrix}$$

1.3 Basics of Matrix Algebra

Matrix: A matrix is an array of real numbers arranged in m rows by n columns. The dimensionality of the matrix is defined as the number of rows by the number of columns, $m \times n$.

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

Note that you can think of vectors as special cases of matrices; a column vector of length k is a $k \times 1$ matrix, while a row vector of the same length is a $1 \times k$ matrix.

It's also useful to think of matrices as being made up of a collection of row or column vectors. For example,

$$\mathbf{A} = \begin{pmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_m \end{pmatrix}$$

Matrix Addition: Let $\mathbf{A}$ and $\mathbf{B}$ be two $m \times n$ matrices.

$$\mathbf{A} + \mathbf{B} = \begin{pmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \cdots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & a_{22} + b_{22} & \cdots & a_{2n} + b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} + b_{m1} & a_{m2} + b_{m2} & \cdots & a_{mn} + b_{mn} \end{pmatrix}$$

Note that matrices $\mathbf{A}$ and $\mathbf{B}$ must have the same dimensionality, in which case they are **conformable for addition**.

Example 1.3.

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}, \qquad \mathbf{B} = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 1 & 2 \end{pmatrix}$$

$\mathbf{A} + \mathbf{B} =$

Scalar Multiplication: Given the scalar s, the scalar multiplication of $s\mathbf{A}$ is

$$s\mathbf{A} = s \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} = \begin{pmatrix} sa_{11} & sa_{12} & \cdots & sa_{1n} \\ sa_{21} & sa_{22} & \cdots & sa_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ sa_{m1} & sa_{m2} & \cdots & sa_{mn} \end{pmatrix}$$

Example 1.4. $s = 2, \qquad \mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$

$s\mathbf{A} =$

Matrix Multiplication: If $\mathbf{A}$ is an $m \times k$ matrix and $\mathbf{B}$ is a $k \times n$ matrix, then their product $\mathbf{C} = \mathbf{AB}$ is the $m \times n$ matrix where

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{ik}b_{kj}$$

Example 1.5. 1. $\begin{pmatrix} a & b \\ c & d \\ e & f \end{pmatrix} \begin{pmatrix} A & B \\ C & D \end{pmatrix} =$

2. $\begin{pmatrix} 1 & 2 & -1 \\ 3 & 1 & 4 \end{pmatrix} \begin{pmatrix} -2 & 5 \\ 4 & -3 \\ 2 & 1 \end{pmatrix} =$

Note that the number of columns of the first matrix must equal the number of rows of the second matrix, in which case they are **conformable for multiplication**. The sizes of the matrices (including the resulting product) must be

$$(m \times k)(k \times n) = (m \times n)$$

Also note that if $\mathbf{AB}$ exists, $\mathbf{BA}$ exists only if $\dim(\mathbf{A}) = m \times n$ and $\dim(\mathbf{B}) = n \times m$.

This does not mean that $\mathbf{AB} = \mathbf{BA}$. $\mathbf{AB} = \mathbf{BA}$ is true only in special circumstances, like when $\mathbf{A}$ or $\mathbf{B}$ is an identity matrix or $\mathbf{A} = \mathbf{B}^{-1}$.

Laws of Matrix Algebra:

1. Associative: $(\mathbf{A} + \mathbf{B}) + \mathbf{C} = \mathbf{A} + (\mathbf{B} + \mathbf{C})$
   $(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$

2. Commutative: $\mathbf{A}+\mathbf{B}=\mathbf{B}+\mathbf{A}$
3. Distributive: $\mathbf{A}(\mathbf{B}+\mathbf{C})=\mathbf{AB}+\mathbf{AC}$
   $(\mathbf{A}+\mathbf{B})\mathbf{C}=\mathbf{AC}+\mathbf{BC}$

Commutative law for multiplication does not hold – the order of multiplication matters:

$$\mathbf{AB}\neq\mathbf{BA}$$

For example,

$$\mathbf{A}=\begin{pmatrix}1 & 2\\ -1 & 3\end{pmatrix},\qquad \mathbf{B}=\begin{pmatrix}2 & 1\\ 0 & 1\end{pmatrix}$$

$$\mathbf{AB}=\begin{pmatrix}2 & 3\\ -2 & 2\end{pmatrix},\qquad \mathbf{BA}=\begin{pmatrix}1 & 7\\ -1 & 3\end{pmatrix}$$

Transpose: The transpose of the $m\times n$ matrix $\mathbf{A}$ is the $n\times m$ matrix $\mathbf{A}^T$ (also written $\mathbf{A}'$) obtained by interchanging the rows and columns of $\mathbf{A}$.

For example,

$$\mathbf{A}=\begin{pmatrix}4 & -2 & 3\\ 0 & 5 & -1\end{pmatrix},\qquad \mathbf{A}^T=\begin{pmatrix}4 & 0\\ -2 & 5\\ 3 & -1\end{pmatrix}$$

$$\mathbf{B}=\begin{pmatrix}2\\ -1\\ 3\end{pmatrix},\qquad \mathbf{B}^T=\begin{pmatrix}2 & -1 & 3\end{pmatrix}$$

The following rules apply for transposed matrices:

1. $(\mathbf{A}+\mathbf{B})^T=\mathbf{A}^T+\mathbf{B}^T$
2. $(\mathbf{A}^T)^T=\mathbf{A}$
3. $(s\mathbf{A})^T=s\mathbf{A}^T$
4. $(\mathbf{AB})^T=\mathbf{B}^T\mathbf{A}^T$; and by induction $(\mathbf{ABC})^T=\mathbf{C}^T\mathbf{B}^T\mathbf{A}^T$

Example of $(\mathbf{AB})^T=\mathbf{B}^T\mathbf{A}^T$:

$$\mathbf{A}=\begin{pmatrix}1 & 3 & 2\\ 2 & -1 & 3\end{pmatrix},\qquad \mathbf{B}=\begin{pmatrix}0 & 1\\ 2 & 2\\ 3 & -1\end{pmatrix}$$

$$(\mathbf{AB})^T=\left[\begin{pmatrix}1 & 3 & 2\\ 2 & -1 & 3\end{pmatrix}\begin{pmatrix}0 & 1\\ 2 & 2\\ 3 & -1\end{pmatrix}\right]^T=\begin{pmatrix}12 & 7\\ 5 & -3\end{pmatrix}$$

$$\mathbf{B}^T\mathbf{A}^T=\begin{pmatrix}0 & 2 & 3\\ 1 & 2 & -1\end{pmatrix}\begin{pmatrix}1 & 2\\ 3 & -1\\ 2 & 3\end{pmatrix}=\begin{pmatrix}12 & 7\\ 5 & -3\end{pmatrix}$$

Exercise 1.3 (Matrix Multiplication)**.** Let

$$A = \begin{pmatrix} 2 & 0 & -1 & 1 \\ 1 & 2 & 0 & 1 \end{pmatrix}$$

$$B = \begin{pmatrix} 1 & 5 & -7 \\ 1 & 1 & 0 \\ 0 & -1 & 1 \\ 2 & 0 & 0 \end{pmatrix}$$

$$C = \begin{pmatrix} 3 & 2 & -1 \\ 0 & 4 & 6 \end{pmatrix}$$

Calculate the following:

1. $$AB$$

2. $$BA$$

3. $$(BC)^T$$

4. $$BC^T$$

1.4 Systems of Linear Equations

Linear Equation: $a_1x_1 + a_2x_2 + \cdots + a_nx_n = b$

a_i are parameters or coefficients. x_i are variables or unknowns.

Linear because only one variable per term and degree is at most 1.

We are often interested in solving linear systems like

$$\begin{array}{ccccc} x & - & 3y & = & -3 \\ 2x & + & y & = & 8 \end{array}$$

More generally, we might have a system of m equations in n unknowns

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & b_2 \\ \vdots & & & & \vdots & & & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \cdots & + & a_{mn}x_n & = & b_m \end{array}$$

A **solution** to a linear system of m equations in n unknowns is a set of n numbers $x_1, x_2, \cdots, x_n$ that satisfy each of the m equations.

Example: $x = 3$ and $y = 2$ is the solution to the above 2×2 linear system. If you graph the two lines, you will find that they intersect at $(3, 2)$.

Does a linear system have one, no, or multiple solutions? For a system of 2 equations with 2 unknowns (i.e., two lines): _
One solution: The lines intersect at exactly one point.

No solution: The lines are parallel.

Infinite solutions: The lines coincide.

Methods to solve linear systems:

1. Substitution
2. Elimination of variables
3. Matrix methods

Exercise 1.4 (Linear Equations)**.** Provide a system of 2 equations with 2 unknowns that has

1. one solution

2. no solution

3. infinite solutions

1.5 Systems of Equations as Matrices

Matrices provide an easy and efficient way to represent linear systems such as

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & b_2 \\ \vdots & & & & \vdots & & & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \cdots & + & a_{mn}x_n & = & b_m \end{array}$$

as

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

where

The $m \times n$ **coefficient matrix A** is an array of mn real numbers arranged in m rows by n columns:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

The unknown quantities are represented by the vector $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$.

The right hand side of the linear system is represented by the vector $\mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$.

Augmented Matrix: When we append $\mathbf{b}$ to the coefficient matrix $\mathbf{A}$, we get the augmented matrix $\widehat{\mathbf{A}} = [\mathbf{A}|\mathbf{b}]$

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} & | & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & | & b_2 \\ \vdots & & \ddots & \vdots & | & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & | & b_m \end{pmatrix}$$

Exercise 1.5 (Augmented Matrix)**.** Create an augmented matrix that represent the following system of equations:

$$2x_1 - 7x_2 + 9x_3 - 4x_4 = 8$$

$$41x_2 + 9x_3 - 5x_6 = 11$$

$$x_1 - 15x_2 - 11x_5 = 9$$

1.6 Finding Solutions to Augmented Matrices and Systems of Equations

Row Echelon Form: Our goal is to translate our augmented matrix or system of equations into row echelon form. This will provide us with the values of the vector $\mathbf{x}$ which solve the system. We use the row operations to change coefficients in the lower triangle of the augmented matrix to 0. An augmented matrix of the form

$$\begin{pmatrix} \boxed{a'_{11}} & a'_{12} & a'_{13} & \cdots & a'_{1n} & | & b'_1 \\ 0 & \boxed{a'_{22}} & a'_{23} & \cdots & a'_{2n} & | & b'_2 \\ 0 & 0 & \boxed{a'_{33}} & \cdots & a'_{3n} & | & b'_3 \\ 0 & 0 & 0 & \ddots & \vdots & | & \vdots \\ 0 & 0 & 0 & 0 & \boxed{a'_{mn}} & | & b'_m \end{pmatrix}$$

is said to be in row echelon form — each row has more leading zeros than the row preceding it.

Reduced Row Echelon Form: We can go one step further and put the matrix into reduced row echelon form. Reduced row echelon form makes the value of $\mathbf{x}$ which solves the system very obvious. For a system of m equations in m unknowns, with no all-zero rows, the reduced row echelon form would be

$$\begin{pmatrix} \boxed{1} & 0 & 0 & 0 & 0 & | & b_1^* \\ 0 & \boxed{1} & 0 & 0 & 0 & | & b_2^* \\ 0 & 0 & \boxed{1} & 0 & 0 & | & b_3^* \\ 0 & 0 & 0 & \ddots & 0 & | & \vdots \\ 0 & 0 & 0 & 0 & \boxed{1} & | & b_m^* \end{pmatrix}$$

Gaussian and Gauss-Jordan elimination: We can conduct elementary row operations to get our augmented matrix into row echelon or reduced row echelon form. The methods of transforming a matrix or system into row echelon and reduced row echelon form are referred to as Gaussian elimination and Gauss-Jordan elimination, respectively.

Elementary Row Operations: To do Gaussian and Gauss-Jordan elimination, we use three basic operations to transform the augmented matrix into another augmented matrix that represents an equivalent linear system – equivalent in the sense that the same values of x_j solve both the original and transformed matrix/system:

Interchanging Rows: Suppose we have the augmented matrix

$$\widehat{\mathbf{A}} = \begin{pmatrix} a_{11} & a_{12} & | & b_1 \\ a_{21} & a_{22} & | & b_2 \end{pmatrix}$$

If we interchange the two rows, we get the augmented matrix

$$\begin{pmatrix} a_{21} & a_{22} & | & b_2 \\ a_{11} & a_{12} & | & b_1 \end{pmatrix}$$

which represents a linear system equivalent to that represented by matrix $\widehat{\mathbf{A}}$.

Multiplying by a Constant: If we multiply the second row of matrix $\widehat{\mathbf{A}}$ by a constant c, we get the augmented matrix

$$\begin{pmatrix} a_{11} & a_{12} & | & b_1 \\ ca_{21} & ca_{22} & | & cb_2 \end{pmatrix}$$

which represents a linear system equivalent to that represented by matrix $\widehat{\mathbf{A}}$.

Adding (subtracting) Rows: If we add (subtract) the first row of matrix $\widehat{\mathbf{A}}$ to the second, we obtain the augmented matrix

$$\begin{pmatrix} a_{11} & a_{12} & | & b_1 \\ a_{11}+a_{21} & a_{12}+a_{22} & | & b_1+b_2 \end{pmatrix}$$

which represents a linear system equivalent to that represented by matrix $\widehat{\mathbf{A}}$.

Example 1.6. Solve the following system of equations by using elementary row operations:

$$\begin{array}{cccccc} x & - & 3y & = & -3 \\ 2x & + & y & = & 8 \end{array}$$

Exercise 1.6 (Solving Systems of Equations)**.** Put the following system of equations into augmented matrix form. Then, using Gaussian or Gauss-Jordan elimination, solve the system of equations by putting the matrix into row echelon or reduced row echelon form.

$$1. \begin{cases} x+y+2z=2 \\ 3x-2y+z=1 \\ y-z=3 \end{cases}$$

$$2. \begin{cases} 2x+3y-z=-8 \\ x+2y-z=12 \\ -x-4y+z=-6 \end{cases}$$

1.7 Rank — and Whether a System Has One, Infinite, or No Solutions

To determine how many solutions exist, we can use information about (1) the number of equations m, (2) the number of unknowns n, and (3) the **rank** of the matrix representing the linear system.

Rank: The maximum number of linearly independent row or column vectors in the matrix. This is equivalent to the number of nonzero rows of a matrix in row echelon form. For any matrix $\mathbf{A}$, the row rank always equals column rank, and we refer to this number as the rank of $\mathbf{A}$.

For example

$$\begin{pmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{pmatrix}$$

Rank $= 3$

$$\begin{pmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 0 \end{pmatrix}$$

Rank $= 2$

Exercise 1.7 (Rank of Matrices)**.** Find the rank of each matrix below:

(Hint: transform the matrices into row echelon form. Remember that the number of nonzero rows of a matrix in row echelon form is the rank of that matrix)

$$1. \begin{pmatrix} 1 & 1 & 2 \\ 2 & 1 & 3 \\ 1 & 2 & 3 \end{pmatrix}$$

$$2. \begin{pmatrix} 1 & 3 & 3 & -3 & 3 \\ 1 & 3 & 1 & 1 & 3 \\ 1 & 3 & 2 & -1 & -2 \\ 1 & 3 & 0 & 3 & -2 \end{pmatrix}$$

Answer to Exercise 1.7:

1. rank is 2
2. rank is 3

1.8 The Inverse of a Matrix

Identity Matrix: The $n \times n$ identity matrix $\mathbf{I}_n$ is the matrix whose diagonal elements are 1 and all off-diagonal elements are 0. Examples:

$$\mathbf{I}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad \mathbf{I}_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Inverse Matrix: An $n \times n$ matrix $\mathbf{A}$ is **nonsingular** or **invertible** if there exists an $n \times n$ matrix $\mathbf{A}^{-1}$ such that

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}_n$$

where $\mathbf{A}^{-1}$ is the inverse of $\mathbf{A}$. If there is no such $\mathbf{A}^{-1}$, then $\mathbf{A}$ is singular or not invertible.

Example: Let

$$\mathbf{A} = \begin{pmatrix} 2 & 3 \\ 2 & 2 \end{pmatrix}, \qquad \mathbf{B} = \begin{pmatrix} -1 & \frac{3}{2} \\ 1 & -1 \end{pmatrix}$$

Since

$$\mathbf{AB} = \mathbf{BA} = \mathbf{I}_n$$

we conclude that $\mathbf{B}$ is the inverse, $\mathbf{A}^{-1}$, of $\mathbf{A}$ and that $\mathbf{A}$ is nonsingular.

Properties of the Inverse:

- If the inverse exists, it is unique.
- If $\mathbf{A}$ is nonsingular, then $\mathbf{A}^{-1}$ is nonsingular.
- $(\mathbf{A}^{-1})^{-1} = \mathbf{A}$
- If $\mathbf{A}$ and $\mathbf{B}$ are nonsingular, then $\mathbf{AB}$ is nonsingular
- $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$
- If $\mathbf{A}$ is nonsingular, then $(\mathbf{A}^T)^{-1} = (\mathbf{A}^{-1})^T$

Procedure to Find $\mathbf{A}^{-1}$: We know that if $\mathbf{B}$ is the inverse of $\mathbf{A}$, then

$$\mathbf{AB} = \mathbf{BA} = \mathbf{I}_n$$

Looking only at the first and last parts of this

$$\mathbf{AB} = \mathbf{I}_n$$

Solving for $\mathbf{B}$ is equivalent to solving for n linear systems, where each column of $\mathbf{B}$ is solved for the corresponding column in $\mathbf{I}_n$. We can solve the systems simultaneously by augmenting

$\mathbf{A}$ with $\mathbf{I}_n$ and performing Gauss-Jordan elimination on $\mathbf{A}$. If Gauss-Jordan elimination on $[\mathbf{A}|\mathbf{I}_n]$ results in $[\mathbf{I}_n|\mathbf{B}]$, then $\mathbf{B}$ is the inverse of $\mathbf{A}$. Otherwise, $\mathbf{A}$ is singular.

To summarize: To calculate the inverse of $\mathbf{A}$

1. Form the augmented matrix $[\mathbf{A}|\mathbf{I}_n]$
2. Using elementary row operations, transform the augmented matrix to reduced row echelon form.
3. The result of step 2 is an augmented matrix $[\mathbf{C}|\mathbf{B}]$.
   a. If $\mathbf{C} = \mathbf{I}_n$, then $\mathbf{B} = \mathbf{A}^{-1}$.
   b. If $\mathbf{C} \neq \mathbf{I}_n$, then $\mathbf{C}$ has a row of zeros. This means $\mathbf{A}$ is singular and $\mathbf{A}^{-1}$ does not exist.

Example 1.7. Find the inverse of the following matricies:

1. $\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 3 \\ 5 & 5 & 1 \end{pmatrix}$

Exercise 1.8 (Finding the inverse of matrices)**.** Find the inverse of the following matrix:

1. $\mathbf{A} = \begin{pmatrix} 1 & 0 & 4 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

1.9 Linear Systems and Inverses

Let's return to the matrix representation of a linear system

$$\mathbf{Ax} = \mathbf{b}$$

If $\mathbf{A}$ is an $n \times n$ matrix,then $\mathbf{Ax} = \mathbf{b}$ is a system of n equations in n unknowns. Suppose $\mathbf{A}$ is nonsingular. Then $\mathbf{A^{-1}}$ exists. To solve this system, we can multiply each side by $\mathbf{A^{-1}}$ and reduce it as follows:

$$\begin{aligned} \mathbf{A^{-1}}(\mathbf{Ax}) &= \mathbf{A^{-1}b} \\ (\mathbf{A^{-1}A})\mathbf{x} &= \mathbf{A^{-1}b} \\ \mathbf{I_n x} &= \mathbf{A^{-1}b} \\ \mathbf{x} &= \mathbf{A^{-1}b} \end{aligned}$$

Hence, given $\mathbf{A}$ and $\mathbf{b}$ and given that $\mathbf{A}$ is nonsingular, then $\mathbf{x} = \mathbf{A^{-1}b}$ is a unique solution to this system.

Exercise 1.9 (Solve linear system using inverses)**.** Use the inverse matrix to solve the following linear system:

$$\begin{aligned} -3x + 4y &= 5 \\ 2x - y &= -10 \end{aligned}$$

Hint: the linear system above can be written in the matrix form

$\mathbf{Az} = \mathbf{b}$

given

$$\mathbf{A} = \begin{pmatrix} -3 & 4 \\ 2 & -1 \end{pmatrix},$$

$$\mathbf{z} = \begin{pmatrix} x \\ y \end{pmatrix},$$

and

$$\mathbf{b} = \begin{pmatrix} 5 \\ -10 \end{pmatrix}$$

1.10 Determinants

Singularity: Determinants can be used to determine whether a square matrix is nonsingular.

A square matrix is nonsingular if and only if its determinant is not zero.

Determinant of a 1×1 matrix, $\mathbf{A}$, equals a_{11}

Determinant of a 2×2 matrix, $\mathbf{A}$, $\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$:

$$\begin{aligned} \det(\mathbf{A}) &= |\mathbf{A}| \\ &= a_{11}|a_{22}| - a_{12}|a_{21}| \\ &= a_{11}a_{22} - a_{12}a_{21} \end{aligned}$$

We can extend the second to last equation above to get the definition of the determinant of a 3×3 matrix:

$$\begin{aligned} \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} &= a_{11}\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12}\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13}\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \\ &= a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31}) \end{aligned}$$

Let's extend this now to any $n \times n$ matrix. Let's define $\mathbf{A}_{ij}$ as the $(n-1) \times (n-1)$ submatrix of $\mathbf{A}$ obtained by deleting row i and column j. Let the (i, j)th **minor** of $\mathbf{A}$ be the determinant of $\mathbf{A}_{ij}$:

$$M_{ij} = |\mathbf{A}_{ij}|$$

Then for any $n \times n$ matrix $\mathbf{A}$

$$|\mathbf{A}| = a_{11}M_{11} - a_{12}M_{12} + \cdots + (-1)^{n+1}a_{1n}M_{1n}$$

For example, in figuring out whether the following matrix has an inverse?

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 3 \\ 5 & 5 & 1 \end{pmatrix}$$

1. Calculate its determinant.

$$\begin{aligned} &= 1(2-15) - 1(0-15) + 1(0-10) \\ &= -13 + 15 - 10 \\ &= -8 \end{aligned}$$

2. Since $|\mathbf{A}| \neq 0$, we conclude that $\mathbf{A}$ has an inverse.

Exercise 1.10 (Determinants and Inverses)**.** Determine whether the following matrices are nonsingular:

1. $\begin{pmatrix} 1 & 0 & 1 \\ 2 & 1 & 2 \\ 1 & 0 & -1 \end{pmatrix}$

2. $\begin{pmatrix} 2 & 1 & 2 \\ 1 & 0 & 1 \\ 4 & 1 & 4 \end{pmatrix}$

1.11 Getting Inverse of a Matrix using its Determinant

Thus far, we have a number of algorithms to

1. Find the solution of a linear system,
2. Find the inverse of a matrix

but these remain just that — algorithms. At this point, we have no way of telling how the solutions x_j change as the parameters a_{ij} and b_i change, except by changing the values and "rerunning" the algorithms.

With determinants, we can provide an explicit formula for the inverse and therefore provide an explicit formula for the solution of an $n \times n$ linear system.

Hence, we can examine how changes in the parameters and b_i affect the solutions x_j.

Determinant Formula for the Inverse of a 2×2:

The determinant of a 2×2 matrix $\mathbf{A}$ $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is defined as:

$$\frac{1}{\det(\mathbf{A})} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

For example, Let's calculate the inverse of matrix A from Exercise 1.9 using the determinant formula.

Recall,

$$A = \begin{pmatrix} -3 & 4 \\ 2 & -1 \end{pmatrix}$$

$$\det(\mathbf{A}) = (-3)(-1) - (4)(2) = 3 - 8 = -5$$

$$\frac{1}{\det(\mathbf{A})} \begin{pmatrix} -1 & -4 \\ -2 & -3 \end{pmatrix}$$

$$\frac{1}{-5} \begin{pmatrix} -1 & -4 \\ -2 & -3 \end{pmatrix}$$

$$\begin{pmatrix} \frac{1}{5} & \frac{4}{5} \\ \frac{2}{5} & \frac{3}{5} \end{pmatrix}$$

Exercise 1.11 (Calculate Inverse using Determinant Formula)**.** Caculate the inverse of A

$$A = \begin{pmatrix} 3 & 5 \\ -7 & 2 \end{pmatrix}$$

Answers to Examples and Exercises

Answer to Example 1.1:

1. $\begin{pmatrix} -1 & -3 & -3 \end{pmatrix}$
2. $6 + 4 + 10 = 20$

Answer to Exercise 1.1:

1. $\begin{pmatrix} -2 & 4 & -7 & -5 \end{pmatrix}$
2. $\begin{pmatrix} 2 & 26 & -14 & 4 & 30 \end{pmatrix}$
3. 63 -3 -10 + 24 = 74
4. undefined

Answer to Example 1.2:

1. yes
2. no

Answer to Exercise 1.2:

1. yes
2. no $(-v_1 - v_2 + v_3 = 0)$

Answer to Example 1.3:

$$\mathbf{A}+\mathbf{B} = \begin{pmatrix} 2 & 4 & 4 \\ 6 & 6 & 8 \end{pmatrix}$$

Answer to Example 1.4:

$$s\mathbf{A} = \begin{pmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \end{pmatrix}$$

Answer to Example 1.5:

1. $\begin{pmatrix} aA+bC & aB+bD \\ cA+dC & cB+dD \\ eA+fC & eB+fD \end{pmatrix}$
2. $\begin{pmatrix} 1(-2)+2(4)-1(2) & 1(5)+2(-3)-1(1) \\ 3(-2)+1(4)+4(2) & 3(5)+1(-3)+4(1) \end{pmatrix} = \begin{pmatrix} 4 & -2 \\ 6 & 16 \end{pmatrix}$

Answer to Exercise 1.3:

1. $AB = \begin{pmatrix} 4 & 11 & -15 \\ 5 & 7 & -7 \end{pmatrix}$
2. $BA =$ undefined
3. $(BC)^T =$ undefined
4. $BC^T = \begin{pmatrix} 1 & 5 & -7 \\ 1 & 1 & 0 \\ 0 & -1 & 1 \\ 2 & 0 & 0 \end{pmatrix} \begin{pmatrix} 3 & 0 \\ 2 & 4 \\ -1 & 6 \end{pmatrix} = \begin{pmatrix} 20 & -22 \\ 5 & 4 \\ -3 & 2 \\ 6 & 0 \end{pmatrix}$

Answer to Exercise 1.4:

There are many answers to this. Some possible simple ones are as follows:

1. One solution:
$$\begin{array}{rcrcl} -x & + & y & = & 0 \\ x & + & y & = & 2 \end{array}$$
2. No solution:
$$\begin{array}{rcrcl} -x & + & y & = & 0 \\ x & - & y & = & 2 \end{array}$$
3. Infinite solutions:
$$\begin{array}{rcrcl} -x & + & y & = & 0 \\ 2x & - & 2y & = & 0 \end{array}$$

Answer to Exercise 1.5:

$$\begin{pmatrix} 2 & -7 & 9 & -4 & 0 & 0| & 8 \\ 0 & 41 & 9 & 0 & 0 & 5| & 11 \\ 1 & -15 & 0 & 0 & -11 & 0| & 9 \end{pmatrix}$$

Answer to Example 1.6:

$$\begin{array}{rcrcr} x & - & 3y & = & -3 \\ 2x & + & y & = & 8 \end{array}$$

$$\begin{array}{rcrcr} x & - & 3y & = & -3 \\ & & 7y & = & 14 \end{array}$$

$$\begin{array}{rcrcr} x & - & 3y & = & -3 \\ & & y & = & 2 \end{array}$$

$$\begin{array}{rcr} x & = & 3 \\ y & = & 2 \end{array}$$

Answer to Exercise 1.6:

1. x = 2, y = 2, z = -1
2. x = -17, y = -3, z = -35

Answer to Exercise 1.7:

1. rank is 2
2. rank is 3

Answer to Example 1.7:

$$\left(\begin{array}{ccc|ccc} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 2 & 3 & 0 & 1 & 0 \\ 5 & 5 & 1 & 0 & 0 & 1 \end{array}\right)$$

$$\left(\begin{array}{ccc|ccc} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 2 & 3 & 0 & 1 & 0 \\ 0 & 0 & -4 & -5 & 0 & 1 \end{array}\right)$$

$$\left(\begin{array}{ccc|ccc} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 2 & 3 & 0 & 1 & 0 \\ 0 & 0 & 1 & 5/4 & 0 & -1/4 \end{array}\right)$$

$$\left(\begin{array}{ccc|ccc} 1 & 1 & 0 & -1/4 & 0 & 1/4 \\ 0 & 2 & 0 & -15/4 & 1 & 3/4 \\ 0 & 0 & 1 & 5/4 & 0 & -1/4 \end{array}\right)$$

$$\left(\begin{array}{ccc|ccc} 1 & 1 & 0 & -1/4 & 0 & 1/4 \\ 0 & 1 & 0 & -15/8 & 1/2 & 3/8 \\ 0 & 0 & 1 & 5/4 & 0 & -1/4 \end{array}\right)$$

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 13/8 & -1/2 & -1/8 \\ 0 & 1 & 0 & -15/8 & 1/2 & 3/8 \\ 0 & 0 & 1 & 5/4 & 0 & -1/4 \end{array}\right)$$

$$\mathbf{A}^{-1} = \begin{pmatrix} 13/8 & -1/2 & -1/8 \\ -15/8 & 1/2 & 3/8 \\ 5/4 & 0 & -1/4 \end{pmatrix}$$

Answer to Exercise 1.8:

1. $\mathbf{A}^{-1} = \begin{pmatrix} 1 & 0 & -4 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Answer to Exercise 1.9:

$$\mathbf{z} = \mathbf{A^{-1}b} = \begin{pmatrix} 1/5 & 4/5 \\ 2/5 & 3/5 \end{pmatrix} \begin{pmatrix} 5 \\ -10 \end{pmatrix} = \begin{pmatrix} -7 \\ -4 \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}$$

Answer to Exercise 1.10:

1. nonsingular
2. singular

Answer to Exercise 1.11:

$$\begin{pmatrix} \frac{2}{41} & \frac{-5}{41} \\ \frac{7}{41} & \frac{3}{41} \end{pmatrix}$$

Chapter 2

Functions and Operations

Topics Dimensionality; Interval Notation for $\mathbf{R}^1$; Neighborhoods: Intervals, Disks, and Balls; Introduction to Functions; Domain and Range; Some General Types of Functions; log, ln, and exp; Other Useful Functions; Graphing Functions; Solving for Variables; Finding Roots; Limit of a Function; Continuity; Sets, Sets, and More Sets.

2.1 Summation Operators $\sum$ and $\prod$

Addition (+), Subtraction (-), multiplication and division are basic operations of arithmetic – combining numbers. In statistics and calculus, we want to add a *sequence* of numbers that can be expressed as a pattern without needing to write down all its components. For example, how would we express the sum of all numbers from 1 to 100 without writing a hundred numbers?

For this we use the summation operator $\sum$ and the product operator $\prod$.

Summation:

$$\sum_{i=1}^{100} x_i = x_1 + x_2 + x_3 + \cdots + x_{100}$$

The bottom of the $\sum$ symbol indicates an index (here, i), and its start value 1. At the top is where the index ends. The notion of "addition" is part of the $\sum$ symbol. The content to the right of the summation is the meat of what we add. While you can pick your favorite index, start, and end values, the content must also have the index.

- $\sum\limits_{i=1}^{n} cx_i = c \sum\limits_{i=1}^{n} x_i$
- $\sum\limits_{i=1}^{n} (x_i + y_i) = \sum\limits_{i=1}^{n} x_i + \sum\limits_{i=1}^{n} y_i$
- $\sum\limits_{i=1}^{n} c = nc$

Product:

$$\prod_{i=1}^{n} x_i = x_1 x_2 x_3 \cdots x_n$$

Properties:

- $\prod\limits_{i=1}^{n} cx_i = c^n \prod\limits_{i=1}^{n} x_i$
- $\prod\limits_{i=k}^{n} cx_k = c^{n-k} \prod\limits_{i=k}^{n} x_i$
- $\prod\limits_{i=1}^{n} (x_i + y_i) =$ a total mess
- $\prod\limits_{i=1}^{n} c = c^n$

Other Useful Functions

Factorials!:

$$x! = x \cdot (x-1) \cdot (x-2) \cdots (1)$$

Modulo: Tells you the remainder when you divide the first number by the second.

- $17 \mod 3 = 2$
- $100 \ \% \ 30 = 10$

Example 2.1 (Operators). 1. $\sum\limits_{i=1}^{5} i =$

2. $\prod\limits_{i=1}^{5} i =$

3. $14 \mod 4 =$

4. $4! =$

Exercise 2.1 (Operators). Let $x_1 = 4, x_2 = 3, x_3 = 7, x_4 = 11, x_5 = 2$

1. $\sum\limits_{i=1}^{3} (7)x_i$

2. $\sum\limits_{i=1}^{5} 2$

3. $\prod\limits_{i=3}^{5} (2)x_i$

2.2 Introduction to Functions

A **function** (in $\mathbf{R}^1$) is a mapping, or transformation, that relates members of one set to members of another set. For instance, if you have two sets: set A and set B, a function from A to B maps every value a in set A such that $f(a) \in B$. Functions can be "many-to-one", where many values or combinations of values from set A produce a single output in set B, or they can be "one-to-one", where each value in set A corresponds to a single value in set B. A function by definition has a single function value for each element of its domain. This means, there cannot be "one-to-many" mapping.

Dimensionality: $\mathbf{R}^1$ is the set of all real numbers extending from $-\infty$ to $+\infty$ — i.e., the real number line. $\mathbf{R}^n$ is an n-dimensional space, where each of the n axes extends from $-\infty$ to $+\infty$.

- $\mathbf{R}^1$ is a one dimensional line.
- $\mathbf{R}^2$ is a two dimensional plane.
- $\mathbf{R}^3$ is a three dimensional space.

Points in $\mathbf{R}^n$ are ordered n-tuples (just means an combination of n elements where order matters), where each element of the n-tuple represents the coordinate along that dimension.

For example:

- $\mathbf{R}^1$: (3)
- $\mathbf{R}^2$: (-15, 5)
- $\mathbf{R}^3$: (86, 4, 0)

Examples of mapping notation:

Function of one variable: $f : \mathbf{R}^1 \to \mathbf{R}^1$

- $f(x) = x + 1$. For each x in $\mathbf{R}^1$, $f(x)$ assigns the number $x + 1$.

Function of two variables: $f : \mathbf{R}^2 \to \mathbf{R}^1$.

- $f(x, y) = x^2 + y^2$. For each ordered pair (x, y) in $\mathbf{R}^2$, $f(x, y)$ assigns the number $x^2 + y^2$.

We often use variable x as input and another y as output, e.g. $y = x + 1$

Example 2.2 (Functions)**.** For each of the following, state whether they are one-to-one or many-to-one functions.

1. For $x \in [0, \infty]$, $f : x \to x^2$ (this could also be written as $f(x) = x^2$).
2. For $x \in [-\infty, \infty]$, $f : x \to x^2$.

Exercise 2.2 (Functions)**.** For each of the following, state whether they are one-to-one or many-to-one functions.

1. For $x \in [-3, \infty]$, $f : x \to x^2$.
2. For $x \in [0, \infty]$, $f : x \to \sqrt{x}$

Some functions are defined only on proper subsets of $\mathbf{R}^n$.

- **Domain**: the set of numbers in X at which $f(x)$ is defined.
- **Range**: elements of Y assigned by $f(x)$ to elements of X, or

$$f(X) = \{y : y = f(x), x \in X\}$$

  Most often used when talking about a function $f : \mathbf{R}^1 \to \mathbf{R}^1$.
- **Image**: same as range, but more often used when talking about a function $f : \mathbf{R}^n \to \mathbf{R}^1$.

Some General Types of Functions

Monomials: $f(x) = ax^k$

a is the coefficient. k is the degree.

Examples: $y = x^2$, $y = -\frac{1}{2}x^3$

Polynomials: sum of monomials.

Examples: $y = -\frac{1}{2}x^3 + x^2$, $y = 3x + 5$

The degree of a polynomial is the highest degree of its monomial terms. Also, it's often a good idea to write polynomials with terms in decreasing degree.

Exponential Functions: Example: $y = 2^x$

2.3 log and exp

Relationship of logarithmic and exponential functions:

$$y = \log_a(x) \iff a^y = x$$

The log function can be thought of as an inverse for exponential functions. a is referred to as the "base" of the logarithm.

Common Bases: The two most common logarithms are base 10 and base e.

1. Base 10: $y = \log_{10}(x) \iff 10^y = x$. The base 10 logarithm is often simply written as "log(x)" with no base denoted.
2. Base e: $y = \log_e(x) \iff e^y = x$. The base e logarithm is referred to as the "natural" logarithm and is written as "ln(x)“.

Properties of exponential functions:

- $a^x a^y = a^{x+y}$
- $a^{-x} = 1/a^x$
- $a^x/a^y = a^{x-y}$
- $(a^x)^y = a^{xy}$
- $a^0 = 1$

Properties of logarithmic functions (any base):

Generally, when statisticians or social scientists write $\log(x)$ they mean $\log_e(x)$. In other words: $\log_e(x) \equiv \ln(x) \equiv \log(x)$

$$\log_a(a^x) = x$$

and

$$a^{\log_a(x)} = x$$

- $\log(xy) = \log(x) + \log(y)$
- $\log(x^y) = y\log(x)$
- $\log(1/x) = \log(x^{-1}) = -\log(x)$
- $\log(x/y) = \log(x \cdot y^{-1}) = \log(x) + \log(y^{-1}) = \log(x) - \log(y)$
- $\log(1) = \log(e^0) = 0$

Change of Base Formula: Use the change of base formula to switch bases as necessary:

$$\log_b(x) = \frac{\log_a(x)}{\log_a(b)}$$

Example:

$$\log_{10}(x) = \frac{\ln(x)}{\ln(10)}$$

You can use logs to go between sum and product notation. This will be particularly important when you're learning maximum likelihood estimation.

$$\begin{aligned}
\log\left(\prod_{i=1}^{n} x_i\right) &= \log(x_1 \cdot x_2 \cdot x_3 \cdots \cdot x_n) \\
&= \log(x_1) + \log(x_2) + \log(x_3) + \cdots + \log(x_n) \\
&= \sum_{i=1}^{n} \log(x_i)
\end{aligned}$$

Therefore, you can see that the log of a product is equal to the sum of the logs. We can write this more generally by adding in a constant, c:

$$\begin{aligned}
\log\left(\prod_{i=1}^{n} cx_i\right) &= \log(cx_1 \cdot cx_2 \cdots cx_n) \\
&= \log(c^n \cdot x_1 \cdot x_2 \cdots x_n) \\
&= \log(c^n) + \log(x_1) + \log(x_2) + \cdots + \log(x_n) \\
&= n\log(c) + \sum_{i=1}^{n} \log(x_i)
\end{aligned}$$

Example 2.3 (Logarithmic Functions)**.** Evaluate each of the following logarithms

1. $\log_4(16)$
2. $\log_2(16)$

Simplify the following logarithm. By “simplify”, we actually really mean - use as many of the logarithmic properties as you can.

3. $\log_4(x^3y^5)$

Exercise 2.3 (Logarithmic Functions)**.** Evaluate each of the following logarithms

1. $\log_{\frac{3}{2}}(\frac{27}{8})$

Simplify each of the following logarithms. By “simplify”, we actually really mean - use as many of the logarithmic properties as you can.

2. $\log(\frac{x^9y^5}{z^3})$
3. $\ln\sqrt{xy}$

2.4 Graphing Functions

What can a graph tell you about a function?

- Is the function increasing or decreasing? Over what part of the domain?
- How “fast” does it increase or decrease?
- Are there global or local maxima and minima? Where?
- Are there inflection points?
- Is the function continuous?
- Is the function differentiable?
- Does the function tend to some limit?
- Other questions related to the substance of the problem at hand.

2.5 Solving for Variables and Finding Roots

Sometimes we’re given a function $y = f(x)$ and we want to find how x varies as a function of y. Use algebra to move x to the left hand side (LHS) of the equation and so that the right hand side (RHS) is only a function of y.

Example 2.4 (Solving for Variables)**.** Solve for x:

1. $y = 3x + 2$
2. $y = e^x$

Solving for variables is especially important when we want to find the **roots** of an equation: those values of variables that cause an equation to equal zero. Especially important in finding equilibria and in doing maximum likelihood estimation.

Procedure: Given $y = f(x)$, set $f(x) = 0$. Solve for x.

Multiple Roots:

$$f(x) = x^2 - 9 \quad \Longrightarrow \quad 0 = x^2 - 9 \quad \Longrightarrow \quad 9 = x^2 \quad \Longrightarrow \quad \pm\sqrt{9} = \sqrt{x^2} \quad \Longrightarrow \quad \pm 3 = x$$

Quadratic Formula: For quadratic equations $ax^2 + bx + c = 0$, use the quadratic formula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Exercise 2.4 (Finding Roots)**.** Solve for x:

1. $f(x) = 3x + 2 = 0$
2. $f(x) = x^2 + 3x - 4 = 0$
3. $f(x) = e^{-x} - 10 = 0$

2.6 Sets

Interior Point: The point $\mathbf{x}$ is an interior point of the set S if $\mathbf{x}$ is in S and if there is some ϵ-ball around $\mathbf{x}$ that contains only points in S. The **interior** of S is the collection of all interior points in S. The interior can also be defined as the union of all open sets in S.

- If the set S is circular, the interior points are everything inside of the circle, but not on the circle's rim.
- Example: The interior of the set $\{(x, y) : x^2 + y^2 \leq 4\}$ is $\{(x, y) : x^2 + y^2 < 4\}$.

Boundary Point: The point $\mathbf{x}$ is a boundary point of the set S if every ϵ-ball around $\mathbf{x}$ contains both points that are in S and points that are outside S. The **boundary** is the collection of all boundary points.

- If the set S is circular, the boundary points are everything on the circle's rim.
- Example: The boundary of $\{(x, y) : x^2 + y^2 \leq 4\}$ is $\{(x, y) : x^2 + y^2 = 4\}$.

Open: A set S is open if for each point $\mathbf{x}$ in S, there exists an open ϵ-ball around $\mathbf{x}$ completely contained in S.

- If the set S is circular and open, the points contained within the set get infinitely close to the circle's rim, but do not touch it.
- Example: $\{(x, y) : x^2 + y^2 < 4\}$

Closed: A set S is closed if it contains all of its boundary points.

- Alternatively: A set is closed if its complement is open.
- If the set S is circular and closed, the set contains all points within the rim as well as the rim itself.
- Example: $\{(x, y) : x^2 + y^2 \leq 4\}$
- Note: a set may be neither open nor closed. Example: $\{(x, y) : 2 < x^2 + y^2 \leq 4\}$

Complement: The complement of set S is everything outside of S.

- If the set S is circular, the complement of S is everything outside of the circle.
- Example: The complement of $\{(x,y) : x^2 + y^2 \leq 4\}$ is $\{(x,y) : x^2 + y^2 > 4\}$.

Empty: The empty (or null) set is a unique set that has no elements, denoted by $\{\}$ or $\emptyset$.

- The empty set is an example of a set that is open and closed, or a "clopen" set.
- Examples: The set of squares with 5 sides; the set of countries south of the South Pole.

Answers to Examples and Exercises

Answer to Example 2.1:

1. $1 + 2 + 3 + 4 + 5 = 15$
2. $1 * 2 * 3 * 4 * 5 = 120$
3. 2
4. $4 * 3 * 2 * 1 = 24$

Answer to Exercise 2.1:

1. $7(4 + 3 + 7) = 98$
2. $2 + 2 + 2 + 2 + 2 = 10$
3. $2^3(7)(11)(2) = 1232$

Answer to Example 2.2:

1. one-to-one
2. many-to-one

Answer to Exercise 2.2:

1. many-to-one
2. one-to-one

Answer to Example 2.3:

1. 2
2. 4
3. $3\log_4(x) + 5\log_4(y)$

Answer to Exercise 2.3:

1. 3
2. $9\log(x) + 5\log(y) - 3\log(z)$
3. $\frac{1}{2}(\ln x + \ln y)$

Answer to Example 2.4:

1. $y = 3x + 2 \quad \Longrightarrow \quad -3x = 2 - y \quad \Longrightarrow \quad 3x = y - 2 \quad \Longrightarrow \quad x = \frac{1}{3}(y-2)$
2. $x = \ln y$

Answer to Exercise 2.4:

1. $\frac{-2}{3}$
2. x = {1, -4}
3. x = - $\ln 10$

Chapter 3

Limits

Solving limits, i.e. finding out the value of functions as its input moves closer to some value, is important for the social scientist's mathematical toolkit for two related tasks. The first is for the study of calculus, which will be in turn useful to show where certain functions are maximized or minimized. The second is for the study of statistical inference, which is the study of inferring things about things you cannot see by using things you can see.

Example: The Central Limit Theorem

Perhaps the most important theorem in statistics is the Central Limit Theorem,

Theorem 3.1 (Central Limit Theorem (i.i.d. case))**.** *For any series of independent and identically distributed random variables $X_1, X_2, \cdots$, we know the distribution of its sum even if we do not know the distribution of X. The distribution of the sum is a Normal distribution.*

$$\frac{\bar{X}_n - \mu}{\sigma/\sqrt{n}} \xrightarrow{d} \textit{Normal}(0, 1),$$

where μ is the mean of X and σ is the standard deviation of X. The arrow is read as "converges in distribution to". Normal$(0, 1)$ indicates a Normal Distribution with mean 0 and variance 1.

That is, the limit of the distribution of the lefthand side is the distribution of the righthand side.

The sign of a limit is the arrow "→". Although we have not yet covered probability (in Section 6) so we have not described what distributions and random variables are, it is worth foreshadowing the Central Limit Theorem. The Central Limit Theorem is powerful because it gives us a *guarantee* of what would happen if $n \to \infty$, which in this case means we collected more data.

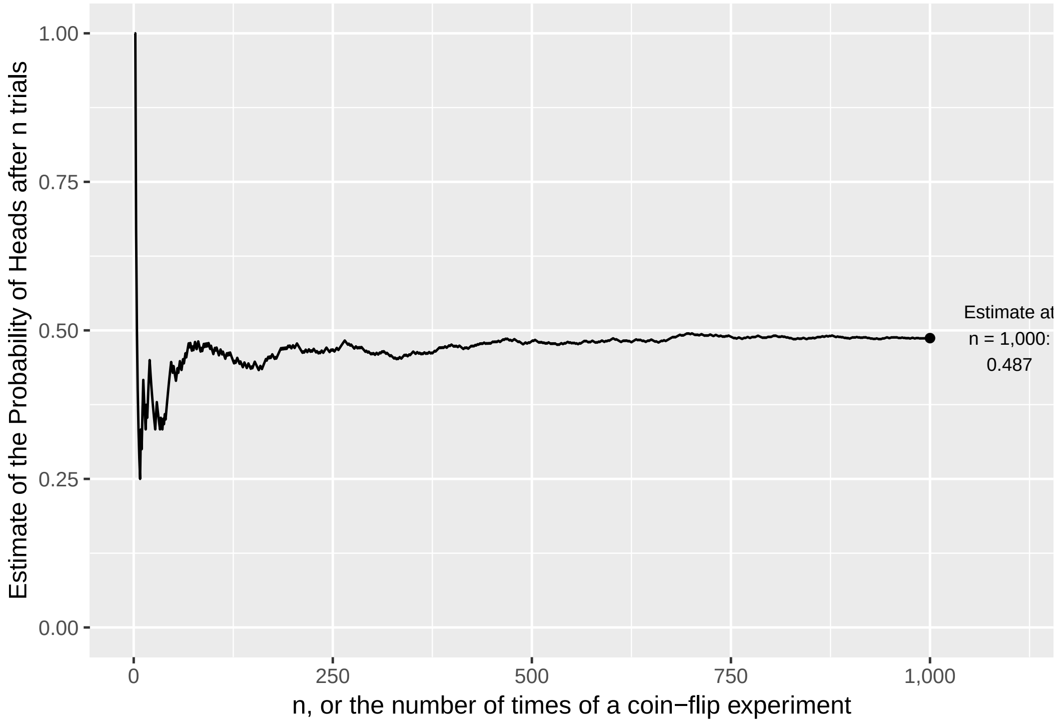


Figure 3.1: As the number of coin tosses goes to infinity, the average probabiity of heads converges to 0.5

Example: The Law of Large Numbers

A finding that perhaps rivals the Central Limit Theorem is the Law of Large Numbers:

Theorem 3.2 ((Weak) Law of Large Numbers)**.** *For any draw of identically distributed independent variables with mean μ, the sample average after n draws, $\bar{X}_n$, converges in probability to the true mean as $n \to \infty$:*

$$\lim_{n\to\infty} P(|\bar{X}_n - \mu| > \varepsilon) = 0$$

A shorthand of which is $\bar{X}_n \xrightarrow{p} \mu$, where the arrow is read as "converges in probability to".

Intuitively, the more data, the more accurate is your guess. For example, the Figure 3.1 shows how the sample average from many coin tosses converges to the true value : 0.5.

3.1 Sequences

We need a couple of steps until we get to limit theorems in probability. First we will introduce a "sequence", then we will think about the limit of a sequence, then we will think about the limit of a *function*.

A **sequence**

$$\{x_n\} = \{x_1, x_2, x_3, \dots, x_n\}$$

is an ordered set of real numbers, where x_1 is the first term in the sequence and y_n is the nth term. Generally, a sequence is infinite, that is it extends to $n = \infty$. We can also write the sequence as

$$\{x_n\}_{n=1}^{\infty}$$

where the subscript and superscript are read together as "from 1 to infinity."

Example 3.1 (Sequences)**.** How does these sequence behave?

1. $\{A_n\} = \left\{2 - \frac{1}{n^2}\right\}$
2. $\{B_n\} = \left\{\frac{n^2+1}{n}\right\}$
3. $\{C_n\} = \left\{(-1)^n \left(1 - \frac{1}{n}\right)\right\}$

We find the sequence by simply "plugging in" the integers into each n. The important thing is to get a sense of how these numbers are going to change. Example 1's numbers seem to come closer and closer to 2, but will it ever surpass 2? Example 2's numbers are also increasing each time, but will it hit a limit? What is the pattern in Example 3? Graphing helps you make this point more clearly. See the sequence of $n = 1, ...20$ for each of the three examples in Figure 3.2.

3.2 The Limit of a Sequence

The notion of "converging to a limit" is the behavior of the points in Example 3.1. In some sense, that's the counterfactual we want to know. What happens as $n \to \infty$?

1. Sequences like 1 above that converge to a limit.
2. Sequences like 2 above that increase without bound.
3. Sequences like 3 above that neither converge nor increase without bound — alternating over the number line.

Definition 3.1. The sequence $\{y_n\}$ has the limit L, which we write as

$$\lim_{n \to \infty} y_n = L$$

, if for any $\epsilon > 0$ there is an integer N (which depends on ϵ) with the property that $|y_n - L| < \epsilon$ for each $n > N$. $\{y_n\}$ is said to converge to L. If the above does not hold, then $\{y_n\}$ diverges.

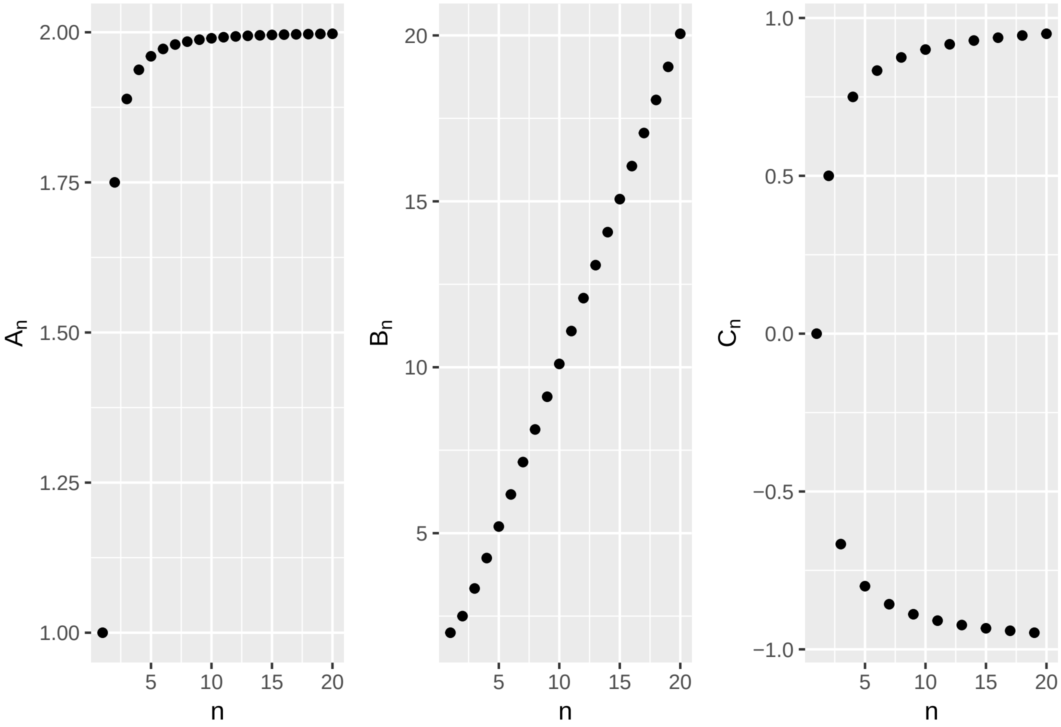


Figure 3.2: Behavior of Some Sequences

We can also express the behavior of a sequence as bounded or not:

1. Bounded: if $|y_n| \leq K$ for all n
2. Monotonically Increasing: $y_{n+1} > y_n$ for all n
3. Monotonically Decreasing: $y_{n+1} < y_n$ for all n

A limit is *unique*: If $\{y_n\}$ converges, then the limit L is unique.

If a sequence converges, then the sum of such sequences also converges. Let $\lim\limits_{n\to\infty} y_n = y$ and $\lim\limits_{n\to\infty} z_n = z$. Then

1. $\lim\limits_{n\to\infty} [ky_n + \ell z_n] = ky + \ell z$
2. $\lim\limits_{n\to\infty} y_n z_n = yz$
3. $\lim\limits_{n\to\infty} \frac{y_n}{z_n} = \frac{y}{z}$, provided $z \neq 0$

This looks reasonable enough. The harder question, obviously is when the parts of the fraction *don't* converge. If $\lim_{n\to\infty} y_n = \infty$ and $\lim_{n\to\infty} z_n = \infty$, What is $\lim_{n\to\infty} y_n - z_n$? What is $\lim_{n\to\infty} \frac{y_n}{z_n}$?

It is nice for a sequence to converge in limit. We want to know if complex-looking sequences converge or not. The name of the game here is to break that complex sequence up into sums of simple fractions where n only appears in the denominator: $\frac{1}{n}$, $\frac{1}{n^2}$, and so on. Each of these will converge to 0, because the denominator gets larger and larger. Then, because of the properties above, we can then find the final sequence.

Example 3.2 (Simplifying a Fraction into Sums)**.** Find the limit of

$$\lim_{n\to\infty} \frac{n+3}{n},$$

Solution. At first glance, $n+3$ and n both grow to ∞, so it looks like we need to divide infinity by infinity. However, we can express this fraction as a sum, then the limits apply separately:

$$\lim_{n\to\infty} \frac{n+3}{n} = \lim_{n\to\infty} \left(1 + \frac{3}{n}\right) = \underbrace{\lim_{n\to\infty} 1}_{1} + \underbrace{\lim_{n\to\infty} \left(\frac{3}{n}\right)}_{0}$$

so, the limit is actually 1.

After some practice, the key to intuition is whether one part of the fraction grows "faster" than another. If the denominator grows faster to infinity than the numerator, then the fraction will converge to 0, even if the numerator will also increase to infinity. In a sense, limits show how not all infinities are the same.

Exercise 3.1. Find the following limits of sequences, then explain in English the intuition for why that is the case.

1. $\lim\limits_{n\to\infty} \frac{2n}{n^2+1}$
2. $\lim\limits_{n\to\infty} (n^3 - 100n^2)$

3.3 Limits of a Function

We've now covered functions and just covered limits of sequences, so now is the time to combine the two.

A function f is a compact representation of some behavior we care about. Like for sequences, we often want to know if $f(x)$ approaches some number L as its independent variable x moves to some number c (which is usually 0 or $\pm\infty$). If it does, we say that the limit of $f(x)$, as x approaches c, is L: $\lim\limits_{x\to c} f(x) = L$. Unlike a sequence, x is a continuous number, and we can move in decreasing order as well as increasing.

For a limit L to exist, the function $f(x)$ must approach L from both the left (increasing) and the right (decreasing).

Definition 3.2 (Limit of a function)**.** Let $f(x)$ be defined at each point in some open interval containing the point c. Then L equals $\lim\limits_{x\to c} f(x)$ if for any (small positive) number ϵ, there exists a corresponding number $\delta > 0$ such that if $0 < |x-c| < \delta$, then $|f(x) - L| < \epsilon$.

A neat, if subtle result is that $f(x)$ does not necessarily have to be defined at c for $\lim\limits_{x\to c}$ to exist.

Properties: Let f and g be functions with $\lim\limits_{x\to c} f(x) = k$ and $\lim\limits_{x\to c} g(x) = \ell$.

1. $\lim\limits_{x\to c}[f(x) + g(x)] = \lim\limits_{x\to c} f(x) + \lim\limits_{x\to c} g(x)$
2. $\lim\limits_{x\to c} kf(x) = k \lim\limits_{x\to c} f(x)$
3. $\lim\limits_{x\to c} f(x)g(x) = \left[\lim\limits_{x\to c} f(x)\right] \cdot \left[\lim\limits_{x\to c} g(x)\right]$
4. $\lim\limits_{x\to c} \frac{f(x)}{g(x)} = \frac{\lim\limits_{x\to c} f(x)}{\lim\limits_{x\to c} g(x)}$, provided $\lim\limits_{x\to c} g(x) \neq 0$.

Simple limits of functions can be solved as we did limits of sequences. Just be careful which part of the function is changing.

Example 3.3 (Limits of Functions)**.** Find the limit of the following functions.

1. $\lim_{x\to c} k$
2. $\lim_{x\to c} x$
3. $\lim_{x\to 2}(2x-3)$
4. $\lim_{x\to c} x^n$

Limits can get more complex in roughly two ways. First, the functions may become large polynomials with many moving pieces. Second,the functions may become discontinuous.

The function can be thought of as a more general or "smooth" version of sequences. For example,

Exercise 3.2 (Limits of a Fraction of Functions)**.** Find the limit of

$$\lim_{x\to\infty} \frac{(x^4 + 3x99)(2x^5)}{(18x^7 + 9x^6 3x^2 1)(x + 1)}$$

Now, the functions will become a bit more complex:

Exercise 3.3. Solve the following limits of functions

1. $\lim\limits_{x\to 0} |x|$
2. $\lim\limits_{x\to 0} \left(1 + \frac{1}{x^2}\right)$

So there are a few more alternatives about what a limit of a function could be:

1. Right-hand limit: The value approached by $f(x)$ when you move from right to left.
2. Left-hand limit: The value approached by $f(x)$ when you move from left to right.
3. Infinity: The value approached by $f(x)$ as x grows infinitely large. Sometimes this may be a number; sometimes it might be ∞ or $-\infty$.
4. Negative infinity: The value approached by $f(x)$ as x grows infinitely negative. Sometimes this may be a number; sometimes it might be ∞ or $-\infty$.

The distinction between left and right becomes important when the function is not determined for some values of x. What are those cases in the examples below?

3.4 Continuity

To repeat a finding from the limits of functions: $f(x)$ does not necessarily have to be defined at c for $\lim\limits_{x\to c}$ to exist. Functions that have breaks in their lines are called discontinuous. Functions that have no breaks are called continuous. Continuity is a concept that is more fundamental to, but related to that of "differentiability", which we will cover next in calculus.

Definition 3.3 (Continuity)**.** Suppose that the domain of the function f includes an open interval containing the point c. Then f is continuous at c if $\lim\limits_{x\to c} f(x)$ exists and if $\lim\limits_{x\to c} f(x) = f(c)$. Further, f is continuous on an open interval (a, b) if it is continuous at each point in the interval.

To prove that a function is continuous for all points is beyond this practical introduction to math, but the general intuition can be grasped by graphing.

Example 3.4 (Continuous and Discontinuous Functions)**.** For each function, determine if it is continuous or discontinuous.

1. $f(x) = \sqrt{x}$
2. $f(x) = e^x$
3. $f(x) = 1 + \frac{1}{x^2}$
4. $f(x) = \text{floor}(x)$.

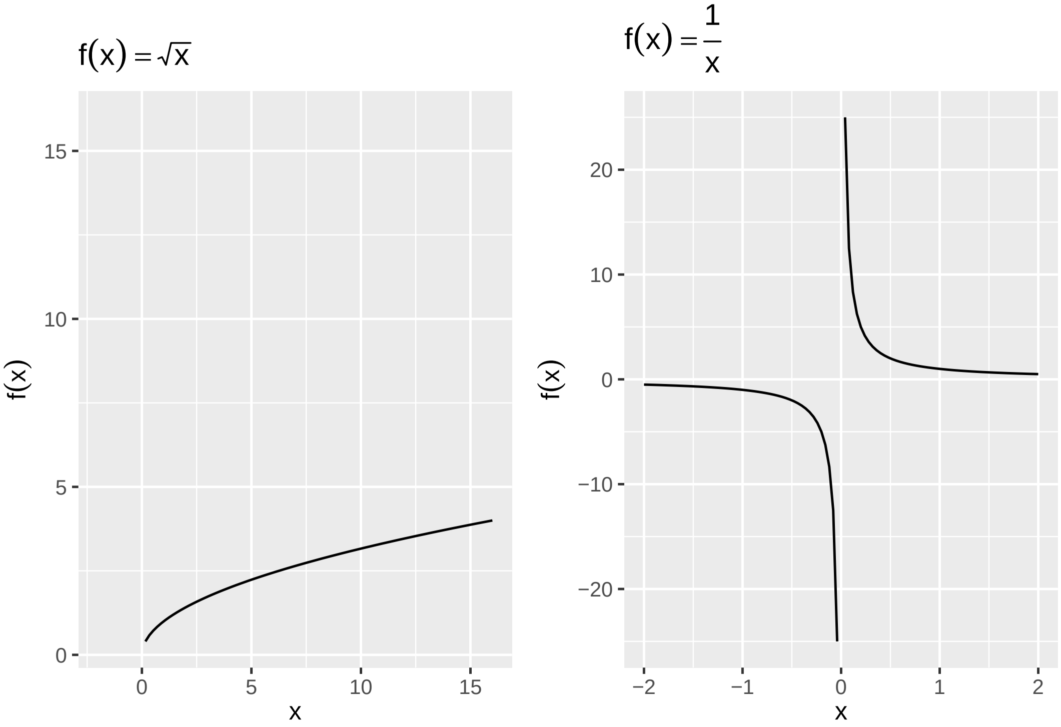


Figure 3.3: Functions which are not defined in some areas

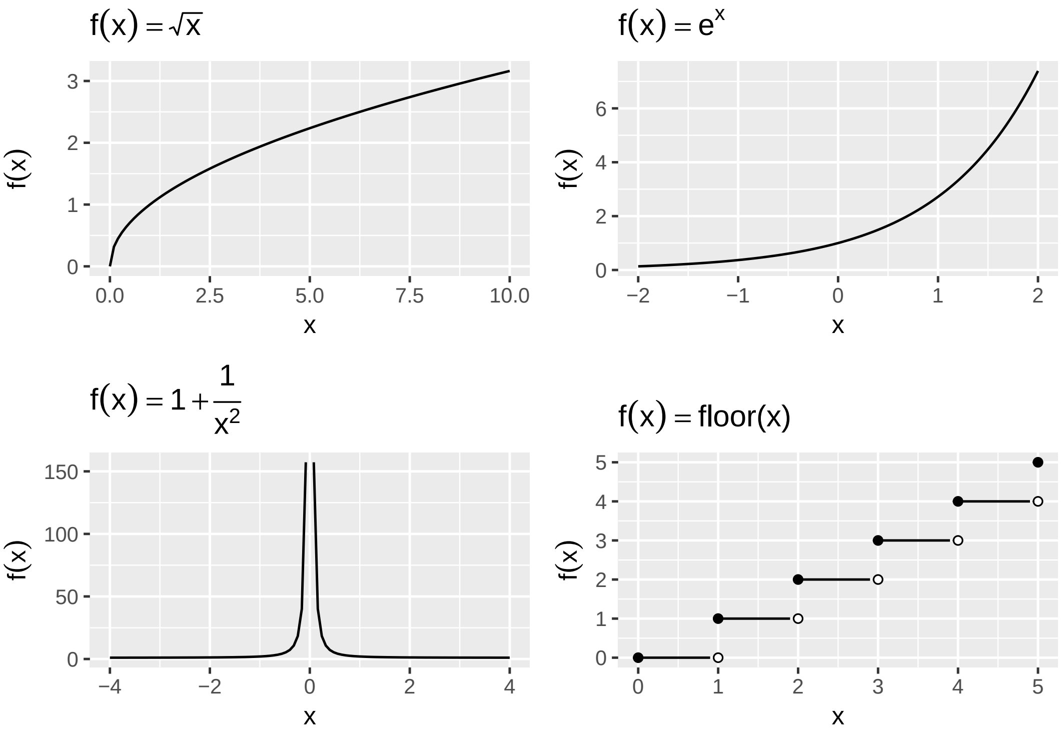


Figure 3.4: Continuous and Discontinuous Functions

The floor is the smaller of the two integers bounding a number. So floor($x = 2.999$) $= 2$, floor($x = 2.0001$) $= 2$, and floor($x = 2$) $= 2$.

Solution. In Figure 3.4, we can see that the first two functions are continuous, and the next two are discontinuous. $f(x) = 1 + \frac{1}{x^2}$ is discontinuous at $x = 0$, and $f(x) = \text{floor}(x)$ is discontinuous at each whole number.

Some properties of continuous functions:

1. If f and g are continuous at point c, then $f + g$, $f - g$, $f \cdot g$, $|f|$, and αf are continuous at point c also. f/g is continuous, provided $g(c) \neq 0$.
2. Boundedness: If f is continuous on the closed bounded interval $[a, b]$, then there is a number K such that $|f(x)| \leq K$ for each x in $[a, b]$.
3. Max/Min: If f is continuous on the closed bounded interval $[a, b]$, then f has a maximum and a minimum on $[a, b]$. They may be located at the end points.

Exercise 3.4 (Limit when Denominator converges to 0)**.** Let

$$f(x) = \frac{x^2 + 2x}{x}.$$

1. Graph the function. Is it defined everywhere?
2. What is the functions limit at $x \to 0$?

Answers to Examples

Example 3.1

Solution. 1. $\{A_n\} = \left\{2 - \frac{1}{n^2}\right\} = \left\{1, \frac{7}{4}, \frac{17}{9}, \frac{31}{16}, \frac{49}{25}, \ldots\right\} = 2$

2. $\{B_n\} = \left\{\frac{n^2+1}{n}\right\} = \left\{2, \frac{5}{2}, \frac{10}{3}, \frac{17}{4} ..., \right\}$

3. $\{C_n\} = \left\{(-1)^n \left(1 - \frac{1}{n}\right)\right\} = \left\{0, \frac{1}{2}, -\frac{2}{3}, \frac{3}{4}, -\frac{4}{5}\right\}$

Exercise 3.1

Example 3.3

Solution. 1. k

2. c

3. $\lim_{x\to 2}(2x-3) = 2\lim\limits_{x\to 2} x - 3\lim\limits_{x\to 2} 1 = 1$

4. $\lim_{x\to c} x^n = \lim\limits_{x\to c} x \cdots [\lim\limits_{x\to c} x] = c \cdots c = c^n$

Exercise 3.2

Solution. Although this function seems large, the thing our eyes should focus on is where the highest order polynomial remains. That will grow the fastest, so if the highest order term is on the denominator, the fraction will converge to 0, if it is on the numerator it will converge to negative infinity. Previewing the multiplication by hand, we can see that the $-x^9$ on the numerator will be the largest power. So the answer will be $-\infty$. We can also confirm this by writing out fractions:

$$\begin{aligned}
&\lim_{x\to\infty} \frac{\left(1+\frac{3}{x^3}-\frac{99}{4x^4}\right)\left(-\frac{2}{x^5}+1\right)}{\left(1+\frac{9}{18x}-\frac{3}{18x^5}-\frac{1}{18x^7}\right)\left(1+\frac{1}{x}\right)} \\
&\times \frac{x^4}{1} \times -\frac{x^5}{1} \times \frac{1}{18x^7} \times \frac{1}{x} \\
=&1 \times \lim_{-x\to\infty} \frac{x}{18}
\end{aligned}$$

Exercise 3.4

Solution. See Figure 3.5.

Divide each part by x, and we get $x + \frac{2}{x}$ on the numerator, 1 on the denominator. So, without worrying about a function being not defined, we can say $\lim_{x\to 0} f(x) = 0$.

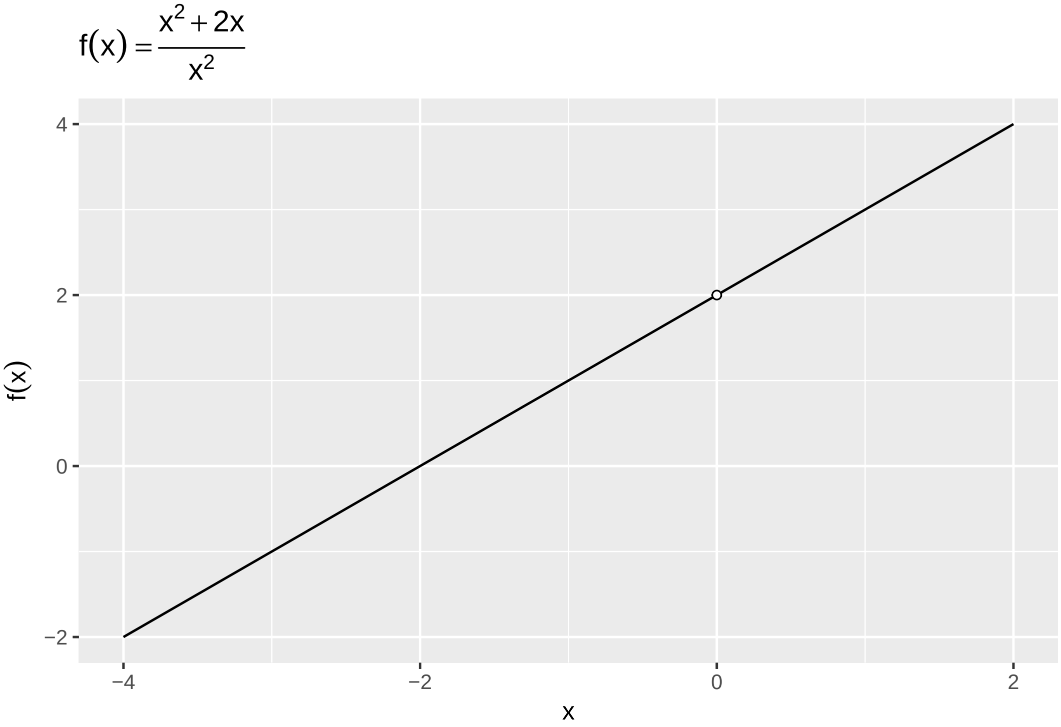


Figure 3.5: A function undedefined at x = 0

Chapter 4

Calculus

Calculus is a fundamental part of any type of statistics exercise. Although you may not be taking derivatives and integral in your daily work as an analyst, calculus undergirds many concepts we use: maximization, expectation, and cumulative probability.

Example: The Mean is a Type of Integral

The average of a quantity is a type of weighted mean, where the potential values are weighted by their likelihood, loosely speaking. The integral is actually a general way to describe this weighted average when there are conceptually an infinite number of potential values.

If X is a continuous random variable, its expected value $E(X)$ – the center of mass – is given by

$$E(X) = \int_{-\infty}^{\infty} x f(x) dx$$

where $f(x)$ is the probability density function of X.

This is a continuous version of the case where X is discrete, in which case

$$E(X) = \sum_{j=1}^{\infty} x_j P(X = x_j)$$

even more concretely, if the potential values of X are finite, then we can write out the expected value as a weighted mean, where the weights is the probability that the value occurs.

$$E(X) = \sum_{x} \quad \left(\underbrace{x}_{\text{value}} \cdot \underbrace{P(X = x)}_{\text{weight, or PMF}} \right)$$

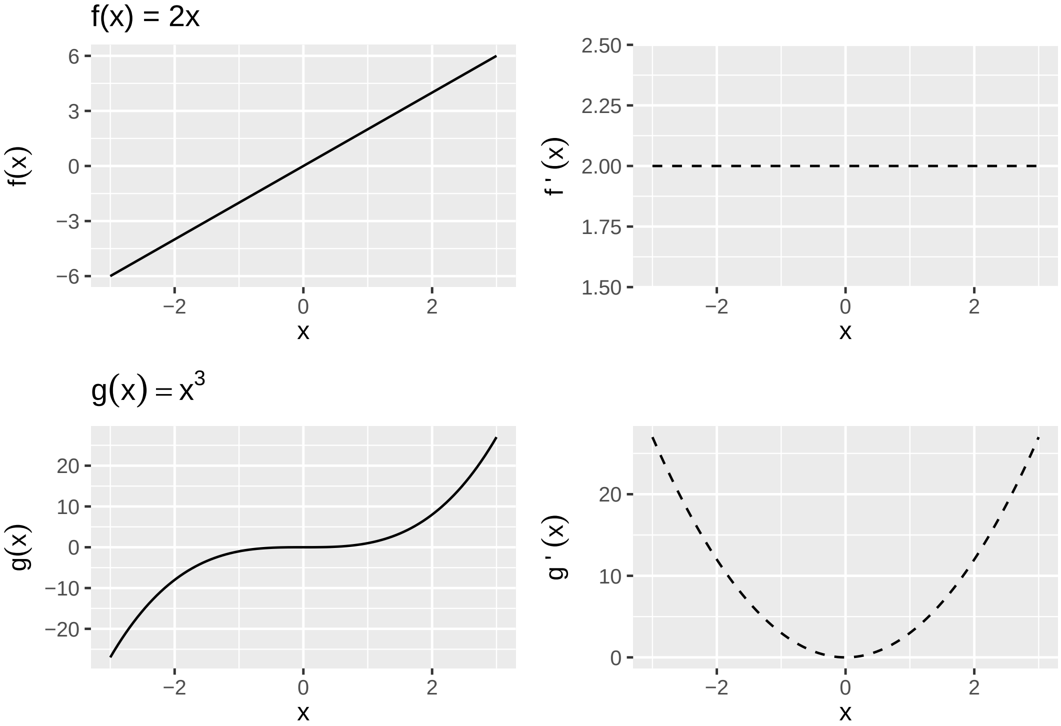


Figure 4.1: The Derivative as a Slope

4.1 Derivatives

The derivative of f at x is its rate of change at x: how much $f(x)$ changes with a change in x. The rate of change is a fraction — rise over run — but because not all lines are straight and the rise over run formula will give us different values depending on the range we examine, we need to take a limit (Section 3).

Definition 4.1 (Derivative)**.** Let f be a function whose domain includes an open interval containing the point x. The derivative of f at x is given by

$$\frac{d}{dx}f(x) = \lim_{h\to 0}\frac{f(x+h)-f(x)}{(x+h)-x} = \lim_{h\to 0}\frac{f(x+h)-f(x)}{h}$$

There are a two main ways to denote a derivate:

- Leibniz Notation: $\frac{d}{dx}(f(x))$
- Prime or Lagrange Notation: $f'(x)$

If $f(x)$ is a straight line, the derivative is the slope. For a curve, the slope changes by the values of x, so the derivative is the slope of the line tangent to the curve at x. See, For example, Figure 4.1.

If $f'(x)$ exists at a point x_0, then f is said to be **differentiable** at x_0. That also implies that $f(x)$ is continuous at x_0.

Properties of derivatives

Suppose that f and g are differentiable at x and that α is a constant. Then the functions $f \pm g$, αf, fg, and f/g (provided $g(x) \neq 0$) are also differentiable at x. Additionally,

Constant rule:

$$[kf(x)]' = kf'(x)$$

Sum rule:

$$[f(x) \pm g(x)]' = f'(x) \pm g'(x)$$

With a bit more algebra, we can apply the definition of derivatives to get a formula for of the derivative of a product and a derivative of a quotient.

Product rule:

$$[f(x)g(x)]' = f'(x)g(x) + f(x)g'(x)$$

Quotient rule:

$$[f(x)/g(x)]' = \frac{f'(x)g(x) - f(x)g'(x)}{[g(x)]^2}, g(x) \neq 0$$

Finally, one way to think of the power of derivatives is that it takes a function a notch down in complexity. The power rule applies to any higher-order function:

Power rule:

$$\left[x^k\right]' = kx^{k-1}$$

The power rule is proved by induction.

These "rules" become apparent by applying the definition of the derivative above to each of the things to be "derived", but these come up so frequently that it is best to repeat until it is muscle memory.

Exercise 4.1 (Derivative of Polynomials)**.** For each of the following functions, find the first-order derivative $f'(x)$.

1. $f(x) = c$
2. $f(x) = x$
3. $f(x) = x^2$
4. $f(x) = x^3$
5. $f(x) = \frac{1}{x^2}$
6. $f(x) = (x^3)(2x^4)$
7. $f(x) = x^4 - x^3 + x^2 - x + 1$
8. $f(x) = (x^2 + 1)(x^3 - 1)$
9. $f(x) = 3x^2 + 2x^{1/3}$
10. $f(x) = \frac{x^2+1}{x^2-1}$

4.2 Higher-Order Derivatives (Derivatives of Derivatives of Derivatives)

The first derivative is applying the definition of derivatives on the function, and it can be expressed as

$$f'(x), \quad y', \quad \frac{d}{dx}f(x), \quad \frac{dy}{dx}$$

We can keep applying the differentiation process to functions that are themselves derivatives. The derivative of $f'(x)$ with respect to x, would then be

$$f''(x) = \lim_{h\to 0} \frac{f'(x+h) - f'(x)}{h}$$

and we can therefore call it the **Second derivative:**

$$f''(x), \quad y'', \quad \frac{d^2}{dx^2}f(x), \quad \frac{d^2y}{dx^2}$$

Similarly, the derivative of $f''(x)$ would be called the third derivative and is denoted $f'''(x)$. And by extension, the **nth derivative** is expressed as $\frac{d^n}{dx^n}f(x)$, $\frac{d^ny}{dx^n}$.

Example 4.1 (Succession of Derivatives)**.**

$$\begin{aligned} f(x) &= x^3 \\ f'(x) &= 3x^2 \\ f''(x) &= 6x \\ f'''(x) &= 6 \\ f''''(x) &= 0 \end{aligned}$$

Earlier, in Section 4.1, we said that if a function differentiable at a given point, then it must be continuous. Further, if $f'(x)$ is itself continuous, then $f(x)$ is called continuously differentiable. All of this matters because many of our findings about optimization (Section 5) rely on differentiation, and so we want our function to be differentiable in as many layers. A function that is continuously differentiable infinitly is called "smooth". Some examples: $f(x) = x^2$, $f(x) = e^x$.

4.3 Composite Functions and the Chain Rule

As useful as the above rules are, many functions you'll see won't fit neatly in each case immediately. Instead, they will be functions of functions. For example, the difference between $x^2 + 1^2$ and $(x^2 + 1)^2$ may look trivial, but the sum rule can be easily applied to the former, while it's actually not obvious what do with the latter.

Composite functions are formed by substituting one function into another and are denoted by

$$(f \circ g)(x) = f[g(x)].$$

To form $f[g(x)]$, the range of g must be contained (at least in part) within the domain of f. The domain of $f \circ g$ consists of all the points in the domain of g for which $g(x)$ is in the domain of f.

Example 4.2. Let $f(x) = \log x$ for $0 < x < \infty$ and $g(x) = x^2$ for $-\infty < x < \infty$.

Then

$$(f \circ g)(x) = \log x^2, -\infty < x < \infty - \{0\}$$

Also

$$(g \circ f)(x) = [\log x]^2, 0 < x < \infty$$

Notice that $f \circ g$ and $g \circ f$ are not the same functions.

With the notation of composite functions in place, now we can introduce a helpful additional rule that will deal with a derivative of composite functions as a chain of concentric derivatives.

Chain Rule:

Let $y = (f \circ g)(x) = f[g(x)]$. The derivative of y with respect to x is

$$\frac{d}{dx}\{f[g(x)]\} = f'[g(x)]g'(x)$$

We can read this as: "the derivative of the composite function y is the derivative of f evaluated at $g(x)$, times the derivative of g."

The chain rule can be thought of as the derivative of the "outside" times the derivative of the "inside", remembering that the derivative of the outside function is evaluated at the value of the inside function.

- The chain rule can also be written as

  $$\frac{dy}{dx} = \frac{dy}{dg(x)} \frac{dg(x)}{dx}$$

  This expression does not imply that the $dg(x)$'s cancel out, as in fractions. They are part of the derivative notation and you can't separate them out or cancel them.)

Example 4.3 (Composite Exponent)**.** Find $f'(x)$ for $f(x) = (3x^2 + 5x - 7)^6$.

The direct use of a chain rule is when the exponent of is itself a function, so the power rule could not have applied generaly:

Generalized Power Rule:

If $f(x) = [g(x)]^p$ for any rational number p,

$$f'(x) = p[g(x)]^{p-1}g'(x)$$

4.4 Derivatives of natural logs and the exponent

Natural logs and exponents (they are inverses of each other; see Section 2.3) crop up everywhere in statistics. Their derivative is a special case from the above, but quite elegant.

Theorem 4.1. *The functions e^x and the natural logarithm $\log(x)$ are continuous and differentiable in their domains, and their first derivate is*

$$(e^x)' = e^x$$

$$\log(x)' = \frac{1}{x}$$

Also, when these are composite functions, it follows by the generalized power rule that

$$\left(e^{g(x)}\right)' = e^{g(x)} \cdot g'(x)$$

$$(\log g(x))' = \frac{g'(x)}{g(x)}, \quad \textit{if} \;\; g(x) > 0$$

We will relegate the proofs to small excerpts.

Derivatives of natural exponential function (e)

To repeat the main rule in Theorem 4.1, the intuition is that

1. Derivative of e^x is itself: $\frac{d}{dx}e^x = e^x$ (See Figure 4.2)
2. Same thing if there were a constant in front: $\frac{d}{dx}\alpha e^x = \alpha e^x$
3. Same thing no matter how many derivatives there are in front: $\frac{d^n}{dx^n}\alpha e^x = \alpha e^x$
4. Chain Rule: When the exponent is a function of x, remember to take derivative of that function and add to product. $\frac{d}{dx}e^{g(x)} = e^{g(x)}g'(x)$

Example 4.4 (Derivative of exponents)**.** Find the derivative for the following.

1. $f(x) = e^{-3x}$
2. $f(x) = e^{x^2}$
3. $f(x) = (x-1)e^x$

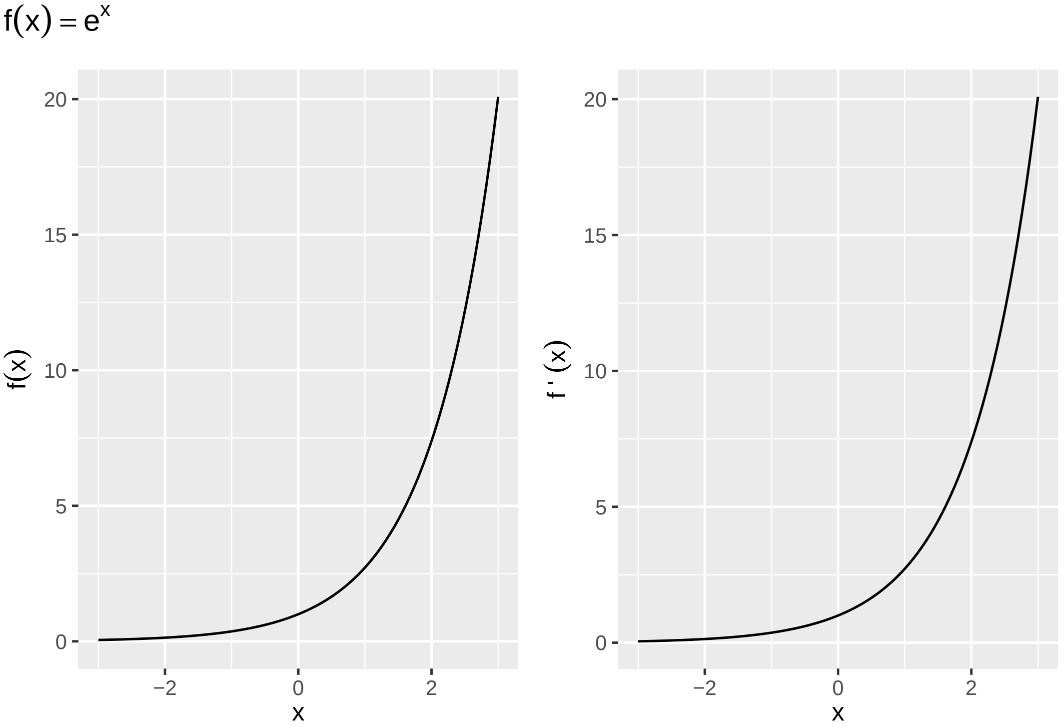


Figure 4.2: Derivative of the Exponential Function

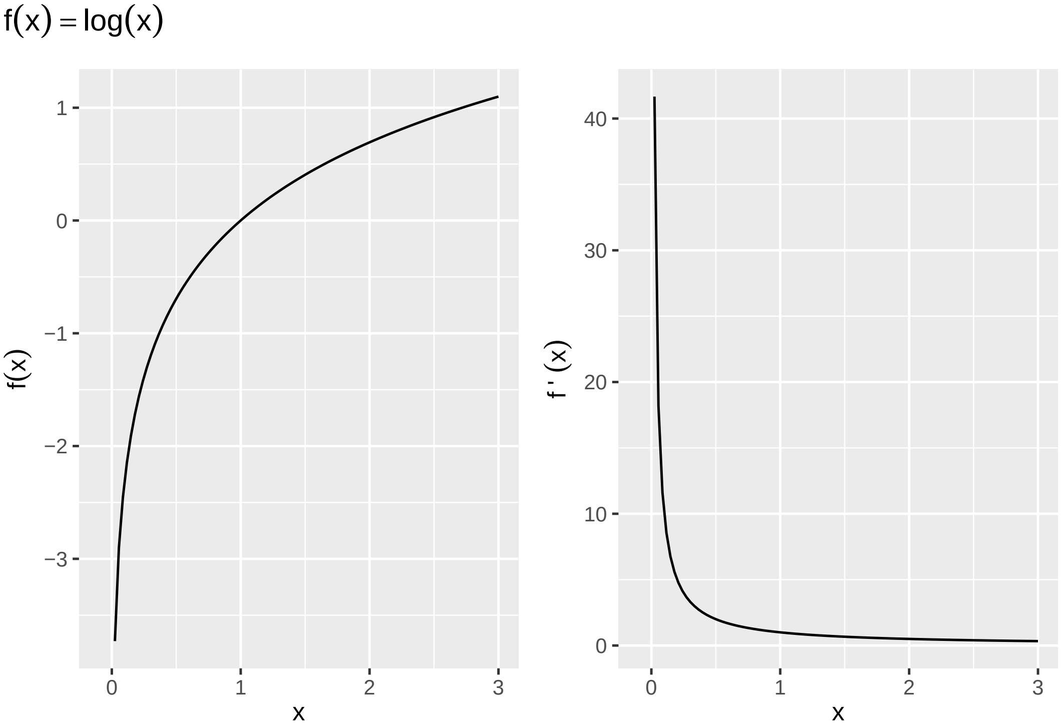


Figure 4.3: Derivative of the Natural Log

Derivatives of log

The natural log is the mirror image of the natural exponent and has mirroring properties, again, to repeat the theorem,

1. log prime x is one over x: $\frac{d}{dx}\log x = \frac{1}{x}$ (Figure 4.3)
2. Exponents become multiplicative constants: $\frac{d}{dx}\log x^k = \frac{d}{dx}k\log x = \frac{k}{x}$
3. Chain rule again: $\frac{d}{dx}\log u(x) = \frac{u'(x)}{u(x)}$
4. For any positive base b, $\frac{d}{dx}b^x = (\log b)(b^x)$.

Example 4.5 (Derivative of logs)**.** Find dy/dx for the following.

1. $f(x) = \log(x^2 + 9)$
2. $f(x) = \log(\log x)$
3. $f(x) = (\log x)^2$
4. $f(x) = \log e^x$

Outline of Proof

We actually show the derivative of the log first, and then the derivative of the exponential naturally follows.

The general derivative of the log at any base a is solvable by the definition of derivatives.

$$(\log_a x)' = \lim_{h\to 0} \frac{1}{h} \log_a \left(1 + \frac{h}{x}\right)$$

Re-express $g = \frac{h}{x}$ and get

$$\begin{aligned}(\log_a x)' &= \frac{1}{x} \lim_{g\to 0} \log_a (1+g)^{\frac{1}{g}} \\ &= \frac{1}{x} \log_a e\end{aligned}$$

By definition of e. As a special case, when $a = e$, then $(\log x)' = \frac{1}{x}$.

Now let's think about the inverse, taking the derivative of $y = a^x$.

$$\begin{aligned} y &= a^x \\ \Rightarrow \log y &= x \log a \\ \Rightarrow \frac{y'}{y} &= \log a \\ \Rightarrow y' = y \log a \end{aligned}$$

Then in the special case where $a = e$,

$$(e^x)' = (e^x)$$

4.5 Partial Derivatives

What happens when there's more than variable that is changing?

> If you can do ordinary derivatives, you can do partial derivatives: just hold all the other input variables constant except for the one you're differentiang with respect to. (Joe Blitzstein's Math Notes)

Suppose we have a function f now of two (or more) variables and we want to determine the rate of change relative to one of the variables. To do so, we would find its partial derivative, which is defined similar to the derivative of a function of one variable.

Partial Derivative: Let f be a function of the variables $(x_1, \ldots, x_n)$. The partial derivative of f with respect to x_i is

$$\frac{\partial f}{\partial x_i}(x_1,\dots,x_n) = \lim_{h\to 0} \frac{f(x_1,\dots,x_i+h,\dots,x_n) - f(x_1,\dots,x_i,\dots,x_n)}{h}$$

Only the ith variable changes — the others are treated as constants.

We can take higher-order partial derivatives, like we did with functions of a single variable, except now the higher-order partials can be with respect to multiple variables.

Example 4.6 (More than one type of partial)**.** Notice that you can take partials with regard to different variables.

Suppose $f(x,y) = x^2 + y^2$. Then

$$\begin{aligned} \frac{\partial f}{\partial x}(x,y) &= \\ \frac{\partial f}{\partial y}(x,y) &= \\ \frac{\partial^2 f}{\partial x^2}(x,y) &= \\ \frac{\partial^2 f}{\partial x \partial y}(x,y) &= \end{aligned}$$

Exercise 4.2. Let $f(x,y) = x^3y^4 + e^x - \log y$. What are the following partial derivaitves?

$$\begin{aligned} \frac{\partial f}{\partial x}(x,y) &= \\ \frac{\partial f}{\partial y}(x,y) &= \\ \frac{\partial^2 f}{\partial x^2}(x,y) &= \\ \frac{\partial^2 f}{\partial x \partial y}(x,y) &= \end{aligned}$$

4.6 Taylor Series Approximation

A common form of appromximation used in statistics involves derivatives. A Taylor series is a way to represent common functions as infinite series (a sum of infinite elements) of the function's derivatives at some point a.

For example, Taylor series are very helpful in representing nonlinear (read: difficult) functions as linear (read: manageable) functions. One can thus **approximate** functions by using lower-order, finite series known as **Taylor polynomials**. If $a = 0$, the series is called a Maclaurin series.

Specifically, a Taylor series of a real or complex function $f(x)$ that is infinitely differentiable in the neighborhood of point a is:

$$\begin{aligned} f(x) &= f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots \\ &= \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n \end{aligned}$$

Taylor Approximation: We can often approximate the curvature of a function $f(x)$ at point a using a 2nd order Taylor polynomial around point a:

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + R_2$$

R_2 is the remainder (R for remainder, 2 for the fact that we took two derivatives) and often treated as negligible, giving us:

$$f(x) \approx f(a) + f'(a)(x-a) + \frac{f''(a)}{2}(x-a)^2$$

The more derivatives that are added, the smaller the remainder R and the more accurate the approximation. Proofs involving limits guarantee that the remainder converges to 0 as the order of derivation increases.

4.7 The Indefinite Integration

So far, we've been interested in finding the derivative $f = F'$ of a function F. However, sometimes we're interested in exactly the reverse: finding the function F for which f is its derivative. We refer to F as the antiderivative of f.

Definition 4.2 (Antiderivative)**.** The antiverivative of a function $f(x)$ is a differentiable function F whose derivative is f.

$$F' = f.$$

Another way to describe is thorugh the inverse formula. Let DF be the derivative of F. And let $DF(x)$ be the derivative of F evaluated at x. Then the antiderivative is denoted by D^{-1} (i.e., the inverse derivative). If $DF = f$, then $F = D^{-1}f$.

This definition bolsters the main takeaway about integrals and derivatives: They are inverses of each other.

Exercise 4.3 (Antiderivative)**.** Find the antiderivative of the following:

1. $f(x) = \frac{1}{x^2}$
2. $f(x) = 3e^{3x}$

We know from derivatives how to manipulate F to get f. But how do you express the procedure to manipulate f to get F? For that, we need a new symbol, which we will call indefinite integration.

Definition 4.3 (Indefinite Integral)**.** The indefinite integral of $f(x)$ is written

$$\int f(x)dx$$

and is equal to the antiderivative of f.

Example 4.7. Draw the function $f(x)$ and its indefinite integral, $\int f(x)dx$

$$f(x) = (x^2 - 4)$$

Solution. The Indefinite Integral of the function $f(x) = (x^2 - 4)$ can, for example, be $F(x) = \frac{1}{3}x^3 - 4x$. But it can also be $F(x) = \frac{1}{3}x^3 - 4x + 1$, because the constant 1 disappears when taking the derivative.

Some of these functions are plotted in the right panel of Figure 4.4 as dotted lines.

Notice from these examples that while there is only a single derivative for any function, there are multiple antiderivatives: one for any arbitrary constant c. c just shifts the curve up or down on the y-axis. If more information is present about the antiderivative — e.g., that it passes through a particular point — then we can solve for a specific value of c.

Common Rules of Integration

Some common rules of integrals follow by virtue of being the inverse of a derivative.

1. Constants are allowed to slip out: $\int af(x)dx = a\int f(x)dx$
2. Integration of the sum is sum of integrations: $\int [f(x)+g(x)]dx = \int f(x)dx + \int g(x)dx$
3. Reverse Power-rule: $\int x^n dx = \frac{1}{n+1}x^{n+1} + c$
4. Exponents are still exponents: $\int e^x dx = e^x + c$
5. Recall the derivative of $\log(x)$ is one over x, and so: $\int \frac{1}{x}dx = \log x + c$
6. Reverse chain-rule: $\int e^{f(x)}f'(x)dx = e^{f(x)} + c$
7. More generally: $\int [f(x)]^n f'(x)dx = \frac{1}{n+1}[f(x)]^{n+1} + c$
8. Remember the derivative of a log of a function: $\int \frac{f'(x)}{f(x)}dx = \log f(x) + c$

Example 4.8 (Common Integration)**.** Simplify the following indefinite integrals:

- $\int 3x^2 dx =$
- $\int (2x+1)dx =$
- $\int e^x e^{e^x} dx =$

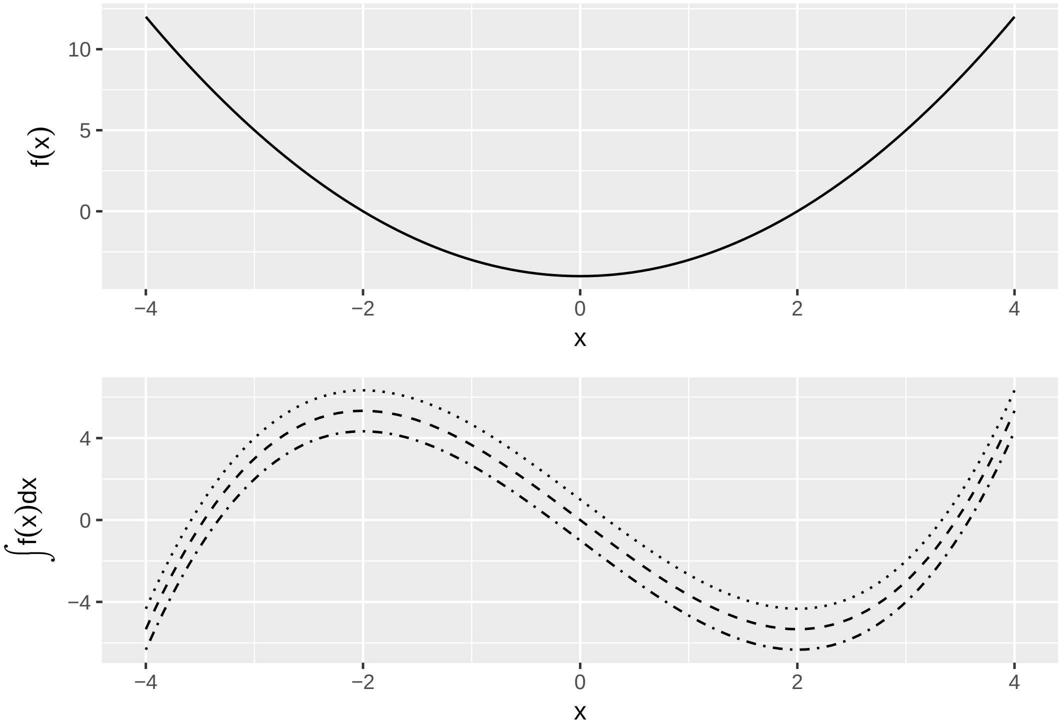


Figure 4.4: The Many Indefinite Integrals of a Function

4.8 The Definite Integral: The Area under the Curve

If there is a indefinite integral, there *must* be a definite integral. Indeed there is, but the notion of definite integrals comes from a different objective: finding the are a under a function. We will find, perhaps remarkably, that the formula we find to get the sum turns out to be expressible by the anti-derivative.

Suppose we want to determine the area $A(R)$ of a region R defined by a curve $f(x)$ and some interval $a \leq x \leq b$.

One way to calculate the area would be to divide the interval $a \leq x \leq b$ into n subintervals of length Δx and then approximate the region with a series of rectangles, where the base of each rectangle is Δx and the height is $f(x)$ at the midpoint of that interval. $A(R)$ would then be approximated by the area of the union of the rectangles, which is given by

$$S(f, \Delta x) = \sum_{i=1}^{n} f(x_i)\Delta x$$

and is called a **Riemann sum**.

As we decrease the size of the subintervals Δx, making the rectangles "thinner," we would expect our approximation of the area of the region to become closer to the true area. This allows us to express the area as a limit of a series:

$$A(R) = \lim_{\Delta x \to 0} \sum_{i=1}^{n} f(x_i)\Delta x$$

This is how we define the "Definite" Integral:

Definition 4.4 (The Definite Integral (Riemann))**.** If for a given function f the Riemann sum approaches a limit as $\Delta x \to 0$, then that limit is called the Riemann integral of f from a to b. We express this with the $\int$, symbol, and write

$$\int\limits_a^b f(x)dx = \lim_{\Delta x \to 0} \sum_{i=1}^{n} f(x_i)\Delta x$$

The most straightforward of a definite integral is the definite integral. That is, we read

$$\int\limits_a^b f(x)dx$$

as the definite integral of f from a to b and we defined as the area under the "curve" $f(x)$ from point $x = a$ to $x = b$.

The fundamental theorem of calculus shows us that this sum is, in fact, the antiderivative.

Theorem 4.2 (First Fundamental Theorem of Calculus)**.** *Let the function f be bounded on $[a, b]$ and continuous on (a, b). Then, suggestively, use the symbol $F(x)$ to denote the definite integral from a to x:*

$$F(x) = \int\limits_a^x f(t)dt, \quad a \leq x \leq b$$

Then $F(x)$ has a derivative at each point in (a, b) and

$$F'(x) = f(x), \quad a < x < b$$

That is, the definite integral function of f is *the one of the antiderivatives of some f.*

This is again a long way of saying that that differentiation is the inverse of integration. But now, we've covered definite integrals.

The second theorem gives us a simple way of computing a definite integral as a function of indefinite integrals.

Theorem 4.3 (Second Fundamental Theorem of Calculus)**.** *Let the function f be bounded on $[a, b]$ and continuous on (a, b). Let F be any function that is continuous on $[a, b]$ such that $F'(x) = f(x)$ on (a, b). Then*

$$\int\limits_a^b f(x)dx = F(b) - F(a)$$

So the procedure to calculate a simple definite integral $\int\limits_a^b f(x)dx$ is then

1. Find the indefinite integral $F(x)$.
2. Evaluate $F(b) - F(a)$.

Example 4.9 (Definite Integral of a monomial)**.** Solve $\int\limits_1^3 3x^2dx$.

Let $f(x) = 3x^2$.

Exercise 4.4. What is the value of $\int\limits_{-2}^2 e^x e^{e^x} dx$?

Common Rules for Definite Integrals

The area-interpretation of the definite integral provides some rules for simplification.

1. There is no area below a point:

$$\int_a^a f(x)dx = 0$$

2. Reversing the limits changes the sign of the integral:

$$\int_a^b f(x)dx = -\int_b^a f(x)dx$$

3. Sums can be separated into their own integrals:

$$\int_a^b [\alpha f(x) + \beta g(x)]dx = \alpha \int_a^b f(x)dx + \beta \int_a^b g(x)dx$$

4. Areas can be combined as long as limits are linked:

$$\int_a^b f(x)dx + \int_b^c f(x)dx = \int_a^c f(x)dx$$

Exercise 4.5 (Definite integral shortcuts)**.** Simplify the following definite intergrals.

1. $\int_1^1 3x^2 dx =$
2. $\int_0^4 (2x+1)dx =$
3. $\int_{-2}^0 e^x e^{e^x} dx + \int_0^2 e^x e^{e^x} dx =$

4.9 Integration by Substitution

From the second fundamental theorem of calculus, we now that a quick way to get a definite integral is to first find the indefinite integral, and then just plug in the bounds.

Sometimes the integrand (the thing that we are trying to take an integral of) doesn't appear integrable using common rules and antiderivatives. A method one might try is **integration by substitution**, which is related to the Chain Rule.

Suppose we want to find the indefinite integral

$$\int g(x)dx$$

but $g(x)$ is complex and none of the formulas we have seen so far seem to apply immediately. The trick is to come up with a *new* function $u(x)$ such that

$$g(x) = f[u(x)]u'(x).$$

Why does an introduction of yet another function end of simplifying things? Let's refer to the antiderivative of f as F. Then the chain rule tells us that

$$\frac{d}{dx}F[u(x)] = f[u(x)]u'(x)$$

. So, $F[u(x)]$ is the antiderivative of g. We can then write

$$\int g(x)dx = \int f[u(x)]u'(x)dx = \int \frac{d}{dx}F[u(x)]dx = F[u(x)] + c$$

To summarize, the procedure to determine the indefinite integral $\int g(x)dx$ by the method of substitution:

1. Identify some part of $g(x)$ that might be simplified by substituting in a single variable u (which will then be a function of x).
2. Determine if $g(x)dx$ can be reformulated in terms of u and du.
3. Solve the indefinite integral.
4. Substitute back in for x

Substitution can also be used to calculate a definite integral. Using the same procedure as above,

$$\int\limits_a^b g(x)dx = \int\limits_c^d f(u)du = F(d) - F(c)$$

where $c = u(a)$ and $d = u(b)$.

Example 4.10 (Integration by Substitution I)**.** Solve the indefinite integral

$$\int x^2\sqrt{x+1}dx.$$

For the above problem, we could have also used the substitution $u = \sqrt{x+1}$. Then $x = u^2-1$ and $dx = 2udu$. Substituting these in, we get

$$\int x^2\sqrt{x+1}dx = \int (u^2-1)^2 u2udu$$

which when expanded is again a polynomial and gives the same result as above.

Another case in which integration by substitution is is useful is with a fraction.

Example 4.11 (Integration by Substitutiton II)**.** Simplify

$$\int\limits_0^1 \frac{5e^{2x}}{(1+e^{2x})^{1/3}}dx.$$

4.10 Integration by Parts

Another useful integration technique is **integration by parts**, which is related to the Product Rule of differentiation. The product rule states that

$$\frac{d}{dx}(uv) = u\frac{dv}{dx} + v\frac{du}{dx}$$

Integrating this and rearranging, we get

$$\int u\frac{dv}{dx}dx = uv - \int v\frac{du}{dx}dx$$

or

$$\int u(x)v'(x)dx = u(x)v(x) - \int v(x)u'(x)dx$$

More easily remembered with the mnemonic "Ultraviolet Voodoo":

$$\int udv = uv - \int vdu$$

where $du = u'(x)dx$ and $dv = v'(x)dx$.

For definite integrals, this is simply

$$\int\limits_a^b u\frac{dv}{dx}dx = uv|_a^b - \int\limits_a^b v\frac{du}{dx}dx$$

Our goal here is to find expressions for u and dv that, when substituted into the above equation, yield an expression that's more easily evaluated.

Example 4.12 (Integration by Parts)**.** Simplify the following integrals. These seemingly obscure forms of integrals come up often when integrating distributions.

$$\int xe^{ax}dx$$

Solution. Let $u = x$ and $\frac{dv}{dx} = e^{ax}$. Then $du = dx$ and $v = (1/a)e^{ax}$. Substituting this into the integration by parts formula, we obtain

$$\begin{aligned}\int xe^{ax}dx &= uv - \int vdu \\ &= x\left(\frac{1}{a}e^{ax}\right) - \int \frac{1}{a}e^{ax}dx \\ &= \frac{1}{a}xe^{ax} - \frac{1}{a^2}e^{ax} + c\end{aligned}$$

Exercise 4.6 (Integration by Parts II)**.** 1. Integrate

$$\int x^n e^{ax} dx$$

2. Integrate

$$\int x^3 e^{-x^2} dx$$

Answers to Examples and Exercises

Exercise 4.1

Solution. 1. $f'(x) = 0$

2. $f'(x) = 1$
3. $f'(x) = 2x^3$
4. $f\prime(x) = 3x^2$
5. $f\prime(x) = -2x^{-3}$
6. $f\prime(x) = 14x^6$
7. $f\prime(x) = 4x^3 - 3x^2 + 2x - 1$
8. $f\prime(x) = 5x^4 + 3x^2 - 2x$
9. $f\prime(x) = 6x + \frac{2}{3}x^{\frac{-2}{3}}$
10. $f\prime(x) = \frac{-4x}{x^4 - 2x^2 + 1}$

Example 4.3

Solution. For convenience, define $f(z) = z^6$ and $z = g(x) = 3x^2 + 5x - 7$. Then, $y = f[g(x)]$ and

$$\begin{aligned}\frac{d}{dx}y &= f'(z)g'(x) \\ &= 6(3x^2 + 5x - 7)^5(6x + 5)\end{aligned}$$

Example 4.4

Solution. 1. Let $u(x) = -3x$. Then $u'(x) = -3$ and $f'(x) = -3e^{-3x}$.

2. Let $u(x) = x^2$. Then $u'(x) = 2x$ and $f'(x) = 2xe^{x^2}$.

Example 4.5

Solution. 1. Let $u(x) = x^2 + 9$. Then $u'(x) = 2x$ and

$$\frac{dy}{dx} = \frac{u'(x)}{u(x)} = \frac{2x}{(x^2 + 9)}$$

2. Let $u(x) = \log x$. Then $u'(x) = 1/x$ and $\frac{dy}{dx} = \frac{1}{(x \log x)}$.
3. Use the generalized power rule.

$$\frac{dy}{dx} = \frac{(2 \log x)}{x}$$

4. We know that $\log e^x = x$ and that $dx/dx = 1$, but we can double check. Let $u(x) = e^x$. Then $u'(x) = e^x$ and $\frac{dy}{dx} = \frac{u'(x)}{u(x)} = \frac{e^x}{e^x} = 1$.

Example 4.9
Solution. What is $F(x)$? From the power rule, recognize $\frac{d}{dx}x^3 = 3x^2$ so

$$\begin{aligned}
F(x) &= x^3 \\
\int\limits_1^3 f(x)dx &= F(x=3) - F(x-1) \\
&= 3^3 - 1^3 \\
&= 26
\end{aligned}$$

Example 4.10
Solution. The problem here is the $\sqrt{x+1}$ term. However, if the integrand had $\sqrt{x}$ times some polynomial, then we'd be in business. Let's try $u = x+1$. Then $x = u-1$ and $dx = du$. Substituting these into the above equation, we get

$$\begin{aligned}
\int x^2\sqrt{x+1}dx &= \int (u-1)^2\sqrt{u}du \\
&= \int (u^2 - 2u + 1)u^{1/2}du \\
&= \int (u^{5/2} - 2u^{3/2} + u^{1/2})du
\end{aligned}$$

We can easily integrate this, since it is just a polynomial. Doing so and substituting $u = x+1$ back in, we get

$$\int x^2\sqrt{x+1}dx = 2(x+1)^{3/2}\left[\frac{1}{7}(x+1)^2 - \frac{2}{5}(x+1) + \frac{1}{3}\right] + c$$

Example 4.11
Solution. When an expression is raised to a power, it is often helpful to use this expression as the basis for a substitution. So, let $u = 1 + e^{2x}$. Then $du = 2e^{2x}dx$ and we can set $5e^{2x}dx = 5du/2$. Additionally, $u = 2$ when $x = 0$ and $u = 1 + e^2$ when $x = 1$. Substituting all of this in, we get

$$\begin{aligned}
\int\limits_0^1 \frac{5e^{2x}}{(1+e^{2x})^{1/3}}dx &= \frac{5}{2}\int\limits_2^{1+e^2} \frac{du}{u^{1/3}} \\
&= \frac{5}{2}\int\limits_2^{1+e^2} u^{-1/3}du \\
&= \left.\frac{15}{4}u^{2/3}\right|_2^{1+e^2} \\
&= 9.53
\end{aligned}$$

Exercise 4.6

1.

$$\int x^n e^{ax} dx$$

Solution. As in the first problem, let

$$u = x^n, dv = e^{ax} dx$$

Then $du = nx^{n-1}dx$ and $v = (1/a)e^{ax}$.

Substituting these into the integration by parts formula gives

$$\begin{aligned} \int x^n e^{ax} dx &= uv - \int v du \\ &= x^n \left(\frac{1}{a} e^{ax} \right) - \int \frac{1}{a} e^{ax} n x^{n-1} dx \\ &= \frac{1}{a} x^n e^{ax} - \frac{n}{a} \int x^{n-1} e^{ax} dx \end{aligned}$$

Notice that we now have an integral similar to the previous one, but with x^{n-1} instead of x^n.

For a given n, we would repeat the integration by parts procedure until the integrand was directly integratable — e.g., when the integral became $\int e^{ax} dx$.

2.

$$\int x^3 e^{-x^2} dx$$

Solution. We could, as before, choose $u = x^3$ and $dv = e^{-x^2} dx$. But we can't then find v — i.e., integrating $e^{-x^2} dx$ isn't possible. Instead, notice that

$$\frac{d}{dx} e^{-x^2} = -2xe^{-x^2},$$

which can be factored out of the original integrand

$$\int x^3 e^{-x^2} dx = \int x^2 (xe^{-x^2}) dx.$$

We can then let $u = x^2$ and $dv = xe^{-x^2} dx$. The$du = 2xdx$ and $v = -\frac{1}{2} e^{-x^2}$. Substituting these in, we have

$$\begin{aligned} \int x^3 e^{-x^2} dx &= uv - \int v du \\ &= x^2 \left(-\frac{1}{2} e^{-x^2} \right) - \int \left(-\frac{1}{2} e^{-x^2} \right) 2x dx \\ &= -\frac{1}{2} x^2 e^{-x^2} + \int x e^{-x^2} dx \\ &= -\frac{1}{2} x^2 e^{-x^2} - \frac{1}{2} e^{-x^2} + c \end{aligned}$$

Chapter 5

Optimization

To optimize, we use derivatives and calculus. Optimization is to find the maximum or minimum of a functon, and to find what value of an input gives that extremum. This has obvious uses in engineering. Many tools in the statistical toolkit use optimization. One of the most common ways of estimating a model is through "Maximum Likelihood Estimation", done via optimizing a function (the likelihood).

Optimization also comes up in Economics, Formal Theory, and Political Economy all the time. A go-to model of human behavior is that they optimize a certain utility function. Humans are not pure utility maximizers, of course, but nuanced models of optimization – for example, adding constraints and adding uncertainty – will prove to be quite useful.

Example: Meltzer-Richard

A standard backdrop in comprative political economy, the Meltzer-Richard (1981) model states that redistribution of wealth should be higher in societies where the median income is much smaller than the average income. Why is that the case? Here is a simplified example that is not the exact model by Meltzer and Richard[1], but adapted from Persson and Tabellini[2]

We will set the following things about our model human and model democracy.

- Individuals are indexed by i, and the total population is normalized to unity ("1") without loss of generality.
- $U(\cdot)$, u for "utility", is a function that is concave and increasing, and expresses the utility gained from public goods. This tells us that its first derivative is *positive*, and its second derivative is **negative**.
- y_i is the income of person i
- W_i, w for "welfare", is the welfare of person i

[1] Allan H. Meltzer and Scott F. Richard. "A Rational Theory of the Size of Government". *Journal of Political Economy* 89:5 (1981), p. 914-927

[2] Adapted from Torsten Persson and Guido Tabellini, *Political Economics: Explaining Economic Policy.* MIT Press.

- c_i, c for "consumption", is the consumption utility of person i

Also, the government is democratically elected and sets the following redistribution output:

- τ, t for "tax", is a flat tax rate between 0 and 1 that is applied to everyone's income.
- g, "g" for "goods", is the amount of public goods that the government provides.

Suppose an individual's welfare is given by:

$$W_i = c_i + U(g)$$

The consumption good is the person's post-tax income.

$$c_i = (1 - \tau)y_i$$

Income varies by person (In the next section we will cover probability, by then we will know that we can express this by saying that y is a random variable with the cumulative distribution function F, i.e. $y \sim F$.). Every distribution has a mean and an median.

- $E(y)$ is the average income of the society.
- med(y) is the **median income** of the society.

What will happen in this economy? What will the tax rate be set too? How much public goods will be provided?

We've skipped ahead of some formal theory results of demoracy, but hopefully these are conceptually intuitive. First, if a democracy is competitive, there is no slack in the government's goods, and all tax revenue becomes a public good. So we can go ahead and set the constraint:

$$g = \sum_i \tau y_i P(y_i) = \tau E(y)$$

We can do this trick because of the "normalizes to unity" setting, but this is a general property of the average.

Now given this constraint we can re-write an individual's welfare as

$$\begin{aligned} W_i &= \left(1 - \frac{g}{E(y)}\right) y_i + U(g) \\ &= (E(y) - g)\, \frac{1}{E(y)} y_i + U(g) \\ &= (E(y) - g)\, \frac{y_i}{E(y)} + U(g) \end{aligned}$$

When is the individual's welfare maximized, **as a function of the public good**?

$$\frac{d}{dg}W_i = -\frac{y_i}{E(y)} + \frac{d}{dg}U(g)$$

$\frac{d}{dg}W_i = 0$ when $\frac{d}{dg}U(g) = \frac{y_i}{E(y)}$, and so after expressing the derivative as $U_g = \frac{d}{dg}U(g)$ for simplicity,

$$g_i^\star = U_g{}^{-1}\left(\frac{y_i}{E(y)}\right)$$

Now recall that because we assumed concavity, U_g is a negative sloping function whose value is positive. It can be shown that the inverse of such a function is also decreasing. Thus an individual's preferred level of government is determined by a single continuum, the person's income divided by the average income, and the function is **decreasing** in y_i. This is consistent with our intuition that richer people prefer less redistribution.

That was the amount for any given person. The government has to set one value of g, however. So what will that be? Now we will use another result, the median voter theorem. This says that under certain general electoral conditions (single-peaked preferences, two parties, majority rule), the policy winner will be that preferred by the median person in the population. Because the only thing that determines a person's preferred level of government is $y_i/E(y)$, we can presume that the median voter, whose income is $\text{med}(y)$ will prevail in their preferred choice of government. Therefore, we wil see

$$g^\star = U_g{}^{-1}\left(\frac{\text{med}(y)}{E(y)}\right)$$

What does this say about the level of redistribution we observe in an economy? The higher the average income is than the median income, which often (but not always) means *more* inequality, there should be *more* redistribution.

5.1 Maxima and Minima

The first derivative, $f'(x)$, quantifies the slope of a function. Therefore, it can be used to check whether the function $f(x)$ at the point x is increasing or decreasing at x.

1. **Increasing:** $f'(x) > 0$
2. **Decreasing:** $f'(x) < 0$
3. **Neither increasing nor decreasing**: $f'(x) = 0$ i.e. a maximum, minimum, or saddle point

So for example, $f(x) = x^2 + 2$ and $f'(x) = 2x$

Exercise 5.1 (Plotting a mazimum and minimum)**.** Plot $f(x) = x^3 + x^2 + 2$, plot its derivative, and identifiy where the derivative is zero. Is there a maximum or minimum?

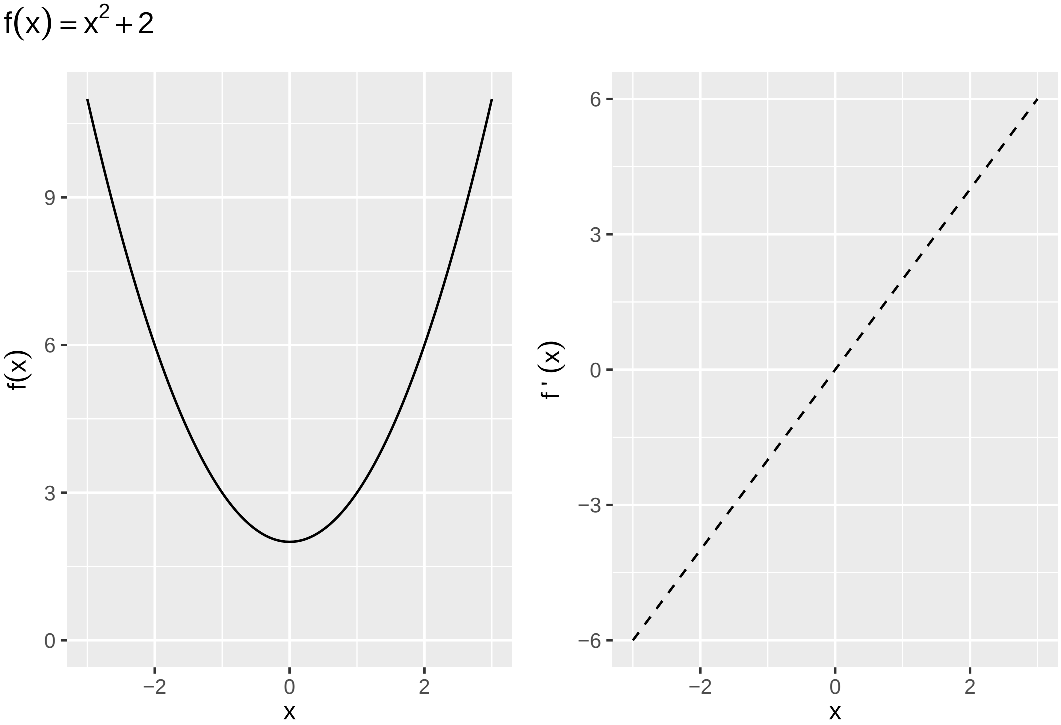


Figure 5.1: Maxima and Minima

The second derivative $f''(x)$ identifies whether the function $f(x)$ at the point x is

1. Concave down: $f''(x) < 0$
2. Concave up: $f''(x) > 0$

Maximum (Minimum): x_0 is a **local maximum (minimum)** if $f(x_0) > f(x)$ $(f(x_0) < f(x))$ for all x within some open interval containing x_0. x_0 is a **global maximum (minimum)** if $f(x_0) > f(x)$ $(f(x_0) < f(x))$ for all x in the domain of f.

Given the function f defined over domain D, all of the following are defined as **critical points**:

1. Any interior point of D where $f'(x) = 0$.
2. Any interior point of D where $f'(x)$ does not exist.
3. Any endpoint that is in D.

The maxima and minima will be a subset of the critical points.

Second Derivative Test of Maxima/Minima: We can use the second derivative to tell us whether a point is a maximum or minimum of $f(x)$.

1. Local Maximum: $f'(x) = 0$ and $f''(x) < 0$
2. Local Minimum: $f'(x) = 0$ and $f''(x) > 0$
3. Need more info: $f'(x) = 0$ and $f''(x) = 0$

Global Maxima and Minima Sometimes no global max or min exists — e.g., $f(x)$ not bounded above or below. However, there are three situations where we can fairly easily identify global max or min.

1. **Functions with only one critical point.** If x_0 is a local max or min of f and it is the only critical point, then it is the global max or min.
2. **Globally concave up or concave down functions.** If $f''(x)$ is never zero, then there is at most one critical point. That critical point is a global maximum if $f'' < 0$ and a global minimum if $f'' > 0$.
3. **Functions over closed and bounded intervals** must have both a global maximum and a global minimum.

Example 5.1 (Maxima and Minima by drawing)**.** Find any critical points and identify whether they are a max, min, or saddle point:

1. $f(x) = x^2 + 2$
2. $f(x) = x^3 + 2$
3. $f(x) = |x^2 - 1|$, $x \in [-2, 2]$

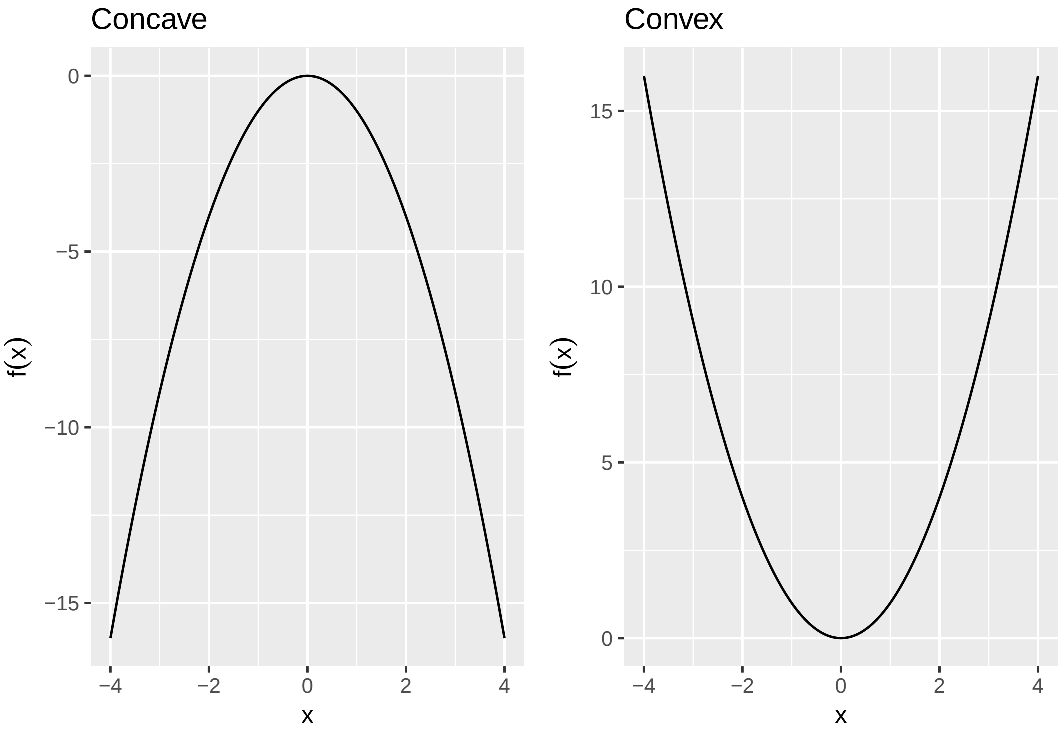
5.2 Concavity of a Function

Concavity helps identify the curvature of a function, $f(x)$, in 2 dimensional space.

Definition 5.1 (Concave Function)**.** A function f is strictly concave over the set S <u>if</u> $\forall x_1, x_2 \in S$ and $\forall a \in (0, 1)$,

$$f(ax_1 + (1 - a)x_2) > af(x_1) + (1 - a)f(x_2)$$

Any line connecting two points on a concave function will lie *below* the function.



Definition 5.2 (Convex Function)**.** Convex: A function f is strictly convex over the set S if $\forall x_1, x_2 \in S$ and $\forall a \in (0,1)$,

$$f(ax_1 + (1-a)x_2) < af(x_1) + (1-a)f(x_2)$$

Any line connecting two points on a convex function will lie above the function.

Sometimes, concavity and convexity are strict of a requirement. For most purposes of getting solutions, what we call quasi-concavity is enough.

Definition 5.3 (Quasiconcave Function)**.** A function f is quasiconcave over the set S if $\forall x_1, x_2 \in S$ and $\forall a \in (0,1)$,

$$f(ax_1 + (1-a)x_2) \geq \min(f(x_1), f(x_2))$$

No matter what two points you select, the *lowest* valued point will always be an end point.

Definition 5.4 (Quasiconvex)**.** A function f is quasiconvex over the set S if $\forall x_1, x_2 \in S$ and $\forall a \in (0,1)$,

$$f(ax_1 + (1-a)x_2) \leq \max(f(x_1), f(x_2))$$

No matter what two points you select, the *highest* valued point will always be an end point.

Second Derivative Test of Concavity: The second derivative can be used to understand concavity.

If

$$\begin{array}{lll} f''(x) < 0 & \Rightarrow & \text{Concave} \\ f''(x) > 0 & \Rightarrow & \text{Convex} \end{array}$$

Quadratic Forms

Quadratic forms is shorthand for a way to summarize a function. This is important for finding concavity because

1. Approximates local curvature around a point — e.g., used to identify max vs min vs saddle point.
2. They are simple to express even in n dimensions:
3. Have a matrix representation.

Quadratic Form: A polynomial where each term is a monomial of degree 2 in any number of variables:

$$\begin{aligned} \text{One variable: } & Q(x_1) = a_{11}x_1^2 \\ \text{Two variables: } & Q(x_1, x_2) = a_{11}x_1^2 + a_{12}x_1x_2 + a_{22}x_2^2 \\ \text{N variables: } & Q(x_1, \cdots, x_n) = \sum_{i \leq j} a_{ij}x_ix_j \end{aligned}$$

which can be written in matrix terms:

One variable

$$Q(\mathbf{x}) = x_1^\top a_{11} x_1$$

N variables:

$$\begin{aligned} Q(\mathbf{x}) &= \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix} \begin{pmatrix} a_{11} & \frac{1}{2}a_{12} & \cdots & \frac{1}{2}a_{1n} \\ \frac{1}{2}a_{12} & a_{22} & \cdots & \frac{1}{2}a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{2}a_{1n} & \frac{1}{2}a_{2n} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \\ &= \mathbf{x}^\top \mathbf{A} \mathbf{x} \end{aligned}$$

For example, the Quadratic on $\mathbf{R}^2$:

$$\begin{aligned} Q(x_1, x_2) &= \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} a_{11} & \frac{1}{2}a_{12} \\ \frac{1}{2}a_{12} & a_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ &= a_{11}x_1^2 + a_{12}x_1x_2 + a_{22}x_2^2 \end{aligned}$$

Definiteness of Quadratic Forms

When the function $f(\mathbf{x})$ has more than two inputs, determining whether it has a maxima and minima (remember, functions may have many inputs but they have only one output) is a bit more tedious. Definiteness helps identify the curvature of a function, $Q(\mathbf{x})$, in n dimensional space.

Definiteness: By definition, a quadratic form always takes on the value of zero when $x = 0$, $Q(\mathbf{x}) = 0$ at $\mathbf{x} = 0$. The definiteness of the matrix $\mathbf{A}$ is determined by whether the quadratic form $Q(\mathbf{x}) = \mathbf{x}^\top \mathbf{A}\mathbf{x}$ is greater than zero, less than zero, or sometimes both over all $\mathbf{x} \neq 0$.

5.3 FOC and SOC

We can see from a graphical representation that if a point is a local maxima or minima, it must meet certain conditions regarding its derivative. These are so commonly used that we refer these to “First Order Conditions” (FOCs) and “Second Order Conditions” (SOCs) in the economic tradition.

First Order Conditions

When we examined functions of one variable x, we found critical points by taking the first derivative, setting it to zero, and solving for x. For functions of n variables, the critical points are found in much the same way, except now we set the partial derivatives equal to zero. Note: We will only consider critical points on the interior of a function’s domain.

In a derivative, we only took the derivative with respect to one variable at a time. When we take the derivative separately with respect to all variables in the elements of $\mathbf{x}$ and then express the result as a vector, we use the term Gradient and Hessian.

Definition 5.5 (Gradient)**.** Given a function $f(\mathbf{x})$ in n variables, the gradient $\nabla f(\mathbf{x})$ (the greek letter nabla) is a column vector, where the ith element is the partial derivative of $f(\mathbf{x})$ with respect to x_i:

$$\nabla f(\mathbf{x}) = \begin{pmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_n} \end{pmatrix}$$

Before we know whether a point is a maxima or minima, if it meets the FOC it is a “Critical Point”.

Definition 5.6 (Critical Point)**.** $\mathbf{x}^*$ is a critical point if and only if $\nabla f(\mathbf{x}^*) = 0$. If the partial derivative of f(x) with respect to x^* is 0, then $\mathbf{x}^*$ is a critical point. To solve for $\mathbf{x}^*$, find the gradient, set each element equal to 0, and solve the system of equations.

$$\mathbf{x}^* = \begin{pmatrix} x_1^* \\ x_2^* \\ \vdots \\ x_n^* \end{pmatrix}$$

Example 5.2. Example: Given a function $f(\mathbf{x}) = (x_1 - 1)^2 + x_2^2 + 1$, find the (1) Gradient and (2) Critical point of $f(\mathbf{x})$.

Solution. Gradient

$$\begin{aligned} \nabla f(\mathbf{x}) &= \begin{pmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \end{pmatrix} \\ &= \begin{pmatrix} 2(x_1 - 1) \\ 2x_2 \end{pmatrix} \end{aligned}$$

Critical Point $\mathbf{x}^* =$

$$\begin{aligned} &\frac{\partial f(\mathbf{x})}{\partial x_1} = 2(x_1 - 1) = 0 \\ &\Rightarrow x_1^* = 1 \\ &\frac{\partial f(\mathbf{x})}{\partial x_2} = 2x_2 = 0 \\ &\Rightarrow x_2^* = 0 \end{aligned}$$

So

$$\mathbf{x}^* = (1, 0)$$

Second Order Conditions

When we found a critical point for a function of one variable, we used the second derivative as a indicator of the curvature at the point in order to determine whether the point was a min, max, or saddle (second derivative test of concavity). For functions of n variables, we use *second order partial derivatives* as an indicator of curvature.

Definition 5.7 (Hessian)**.** Given a function $f(\mathbf{x})$ in n variables, the hessian $\mathbf{H}(\mathbf{x})$ is an $n \times n$ matrix, where the (i, j)th element is the second order partial derivative of $f(\mathbf{x})$ with respect to x_i and x_j:

$$\mathbf{H}(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n^2} \end{pmatrix}$$

Note that the hessian will be a symmetric matrix because $\frac{\partial f(\mathbf{x})}{\partial x_1 \partial x_2} = \frac{\partial f(\mathbf{x})}{\partial x_2 \partial x_1}$.

Also note that given that $f(\mathbf{x})$ is of quadratic form, each element of the hessian will be a constant.

These definitions will be employed when we determine the **Second Order Conditions** of a function:

Given a function $f(\mathbf{x})$ and a point $\mathbf{x}^*$ such that $\nabla f(\mathbf{x}^*) = 0$,

1. Hessian is Positive Definite $\implies$ Strict Local Min
2. Hessian is Positive Semidefinite $\forall \mathbf{x} \in B(\mathbf{x}^*, \epsilon)\}$ $\implies$ Local Min
3. Hessian is Negative Definite $\implies$ Strict Local Max
4. Hessian is Negative Semidefinite $\forall \mathbf{x} \in B(\mathbf{x}^*, \epsilon)\}$ $\implies$ Local Max
5. Hessian is Indefinite $\implies$ Saddle Point

Example 5.3 (Max and min with two dimensions)**.** We found that the only critical point of $f(\mathbf{x}) = (x_1 - 1)^2 + x_2^2 + 1$ is at $\mathbf{x}^* = (1, 0)$. Is it a min, max, or saddle point?

Solution. The Hessian is

$$\mathbf{H}(\mathbf{x}) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

The Leading principal minors of the Hessian are $M_1 = 2; M_2 = 4$. Now we consider Definiteness. Since both leading principal minors are positive, the Hessian is positive definite.

Maxima, Minima, or Saddle Point? Since the Hessian is positive definite and the gradient equals 0, $x^\star = (1, 0)$ is a strict local minimum.

Note: Alternate check of definiteness. Is $\mathbf{H}(\mathbf{x}^*) \geq\leq 0 \quad \forall \quad \mathbf{x} \neq 0$

$$\begin{aligned} \mathbf{x}^\top H(\mathbf{x}^*)\mathbf{x} &= \begin{pmatrix} x_1 & x_2 \end{pmatrix} \\ &= \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \\ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= 2x_1^2 + 2x_2^2 \end{aligned}$$

For any $\mathbf{x} \neq 0$, $2(x_1^2 + x_2^2) > 0$, so the Hessian is positive definite and $\mathbf{x}^*$ is a strict local minimum.

Definiteness and Concavity

Although definiteness helps us to understand the curvature of an n-dimensional function, it does not necessarily tell us whether the function is globally concave or convex.

We need to know whether a function is globally concave or convex to determine whether a critical point is a global min or max. We can use the definiteness of the Hessian to determine whether a function is globally concave or convex:

1. Hessian is Positive Semidefinite $\forall \mathbf{x}\}$ $\implies$ Globally Convex
2. Hessian is Negative Semidefinite $\forall \mathbf{x}\}$ $\implies$ Globally Concave

Notice that the definiteness conditions must be satisfied over the entire domain.

5.4 Global Maxima and Minima

Global Max/Min Conditions: Given a function $f(\mathbf{x})$ and a point $\mathbf{x}^*$ such that $\nabla f(\mathbf{x}^*) = 0$,

1. $f(\mathbf{x})$ Globally Convex $\implies$ Global Min
2. $f(\mathbf{x})$ Globally Concave $\implies$ Global Max

Note that showing that $\mathbf{H}(\mathbf{x}^*)$ is negative semidefinite is not enough to guarantee $\mathbf{x}^*$ is a local max. However, showing that $\mathbf{H}(\mathbf{x})$ is negative semidefinite for all $\mathbf{x}$ guarantees that x^* is a global max. (The same goes for positive semidefinite and minima.)\

Example: Take $f_1(x) = x^4$ and $f_2(x) = -x^4$. Both have $x = 0$ as a critical point. Unfortunately, $f_1''(0) = 0$ and $f_2''(0) = 0$, so we can't tell whether $x = 0$ is a min or max for either. However, $f_1''(x) = 12x^2$ and $f_2''(x) = -12x^2$. For all x, $f_1''(x) \geq 0$ and $f_2''(x) \leq 0$ — i.e., $f_1(x)$ is globally convex and $f_2(x)$ is globally concave. So $x = 0$ is a global min of $f_1(x)$ and a global max of $f_2(x)$.

Exercise 5.2. Given $f(\mathbf{x}) = x_1^3 - x_2^3 + 9x_1x_2$, find any maxima or minima.

1. First order conditions.
   (a) Gradient $\nabla f(\mathbf{x}) =$

   (b) Critical Points $\mathbf{x}^* =$

2. Second order conditions.
   (a) Hessian $\mathbf{H}(\mathbf{x}) =$

   (b) Hessian $\mathbf{H}(\mathbf{x_1^*}) =$

   (c) Leading principal minors of $\mathbf{H}(\mathbf{x_1^*}) =$

   (d) Definiteness of $\mathbf{H}(\mathbf{x_1^*})$?

   (e) Maxima, Minima, or Saddle Point for $\mathbf{x_1^*}$?

   (f) Hessian $\mathbf{H}(\mathbf{x_2^*}) =$

   (g) Leading principal minors of $\mathbf{H}(\mathbf{x_2^*}) =$

   (h) Definiteness of $\mathbf{H}(\mathbf{x_2^*})$?

   (i) Maxima, Minima, or Saddle Point for $\mathbf{x}_2^*$?

3. Global concavity/convexity.
   (a) Is f(x) globally concave/convex?

   (b) Are any $\mathbf{x}^*$ global minima or maxima?

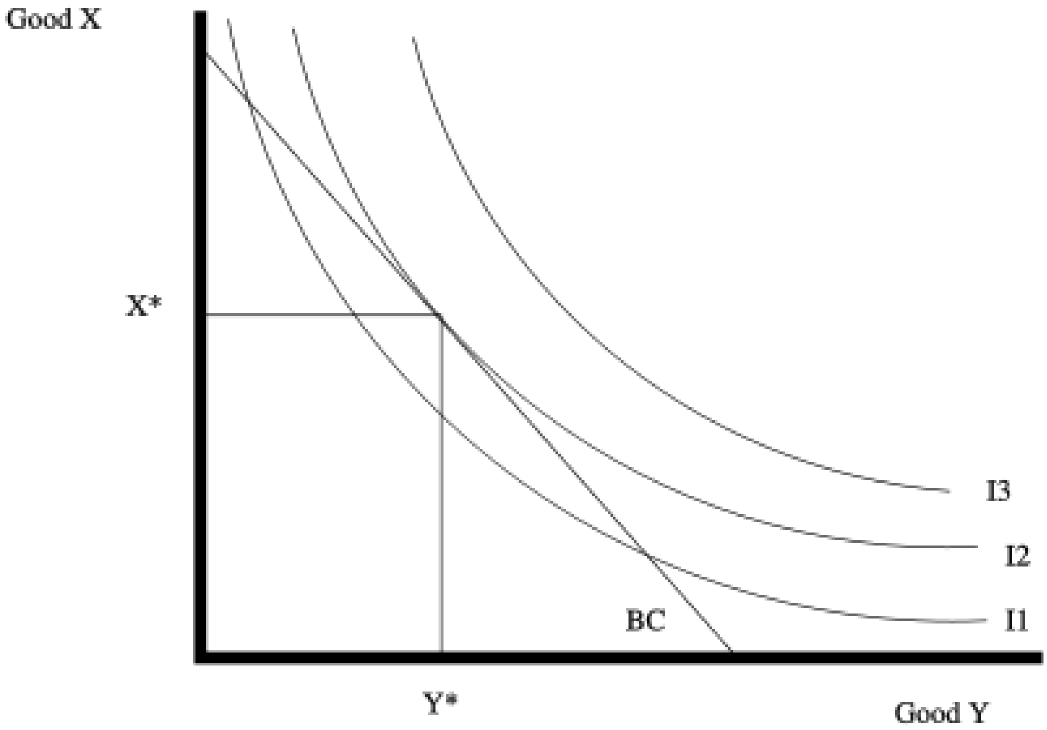


Figure 5.2: A typical Utility Function with a Budget Constraint

5.5 Constrained Optimization

We have already looked at optimizing a function in one or more dimensions over the whole domain of the function. Often, however, we want to find the maximum or minimum of a function over some restricted part of its domain.

ex: Maximizing utility subject to a budget constraint

Types of Constraints: For a function $f(x_1, \ldots, x_n)$, there are two types of constraints that can be imposed:

1. **Equality constraints:** constraints of the form $c(x_1, \ldots, x_n) = r$. Budget constraints are the classic example of equality constraints in social science.
2. **Inequality constraints:** constraints of the form $c(x_1, \ldots, x_n) \leq r$. These might arise from non-negativity constraints or other threshold effects.

In any constrained optimization problem, the constrained maximum will always be less than or equal to the unconstrained maximum. If the constrained maximum is less than the unconstrained maximum, then the constraint is binding. Essentially, this means that you can treat your constraint as an equality constraint rather than an inequality constraint.

For example, the budget constraint binds when you spend your entire budget. This generally happens because we believe that utility is strictly increasing in consumption, i.e. you always want more so you spend everything you have.

Any number of constraints can be placed on an optimization problem. When working with multiple constraints, always make sure that the set of constraints are not pathological; it must be possible for all of the constraints to be satisfied simultaneously.

Set-up for Constrained Optimization:

$$\max_{x_1,x_2} f(x_1,x_2) \text{ s.t. } c(x_1,x_2)$$

$$\min_{x_1,x_2} f(x_1,x_2) \text{ s.t. } c(x_1,x_2)$$

This tells us to maximize/minimize our function, $f(x_1, x_2)$, with respect to the choice variables, x_1, x_2, subject to the constraint.

Example:

$$\max_{x_1,x_2} f(x_1,x_2) = -(x_1^2 + 2x_2^2) \text{ s.t. } x_1 + x_2 = 4$$

It is easy to see that the *unconstrained* maximum occurs at $(x_1, x_2) = (0, 0)$, but that does not satisfy the constraint. How should we proceed?

Equality Constraints

Equality constraints are the easiest to deal with because we know that the maximum or minimum has to lie on the (intersection of the) constraint(s).

The trick is to change the problem from a constrained optimization problem in n variables to an unconstrained optimization problem in $n + k$ variables, adding *one* variable for *each* equality constraint. We do this using a lagrangian multiplier.

Lagrangian function: The Lagrangian function allows us to combine the function we want to optimize and the constraint function into a single function. Once we have this single function, we can proceed as if this were an *unconstrained* optimization problem.

For each constraint, we must include a **Lagrange multiplier** (λ_i) as an additional variable in the analysis. These terms are the link between the constraint and the Lagrangian function.

Given a *two dimensional* set-up:

$$\max_{x_1,x_2} / \min_{x_1,x_2} f(x_1,x_2) \text{ s.t. } c(x_1,x_2) = a$$

We define the Lagrangian function $L(x_1, x_2, \lambda_1)$ as follows:

$$L(x_1, x_2, \lambda_1) = f(x_1, x_2) - \lambda_1(c(x_1, x_2) - a)$$

More generally, in *n dimensions*:

$$L(x_1, \ldots, x_n, \lambda_1, \ldots, \lambda_k) = f(x_1, \ldots, x_n) - \sum_{i=1}^{k} \lambda_i(c_i(x_1, \ldots, x_n) - r_i)$$

Getting the sign right: Note that above we subtract the lagrangian term *and* we subtract the constraint constant from the constraint function. Occasionally, you may see the following alternative form of the Lagrangian, which is *equivalent*:

$$L(x_1, \ldots, x_n, \lambda_1, \ldots, \lambda_k) = f(x_1, \ldots, x_n) + \sum_{i=1}^{k} \lambda_i (r_i - c_i(x_1, \ldots, x_n))$$

Here we add the lagrangian term *and* we subtract the constraining function from the constraint constant.

Using the Lagrangian to Find the Critical Points: To find the critical points, we take the partial derivatives of lagrangian function, $L(x_1, \ldots, x_n, \lambda_1, \ldots, \lambda_k)$, with respect to each of its variables (all choice variables $\mathbf{x}$ *and* all lagrangian multipliers λ). At a critical point, each of these partial derivatives must be equal to zero, so we obtain a system of $n+k$ equations in $n+k$ unknowns:

$$\begin{aligned}
\frac{\partial L}{\partial x_1} &= \frac{\partial f}{\partial x_1} - \sum_{i=1}^{k} \lambda_i \frac{\partial c_i}{\partial x_1} = 0 \\
\vdots &= \vdots \\
\frac{\partial L}{\partial x_n} &= \frac{\partial f}{\partial x_n} - \sum_{i=1}^{k} \lambda_i \frac{\partial c_i}{\partial x_n} = 0 \\
\frac{\partial L}{\partial \lambda_1} &= c_1(x_i, \ldots, x_n) - r_1 = 0 \\
\vdots &= \vdots \\
\frac{\partial L}{\partial \lambda_k} &= c_k(x_i, \ldots, x_n) - r_k = 0
\end{aligned}$$

We can then solve this system of equations, because there are $n+k$ equations and $n+k$ unknowns, to calculate the critical point $(x_1^*, \ldots, x_n^*, \lambda_1^*, \ldots, \lambda_k^*)$.

Second-order Conditions and Unconstrained Optimization: There may be more than one critical point, i.e. we need to verify that the critical point we find is a maximum/minimum. Similar to unconstrained optimization, we can do this by checking the second-order conditions.

Example 5.4 (Constrained optimization with two goods and a budget constraint)**.** Find the constrained optimization of

$$\max_{x_1, x_2} f(x) = -(x_1^2 + 2x_2^2) \text{ s.t. } x_1 + x_2 = 4$$

Solution. 1. Begin by writing the Lagrangian:

$$L(x_1, x_2, \lambda) = -(x_1^2 + 2x_2^2) - \lambda(x_1 + x_2 - 4)$$

2. Take the partial derivatives and set equal to zero:

$$
\begin{aligned}
\frac{\partial L}{\partial x_1} &= -2x_1 - \lambda &&= 0 \\
\frac{\partial L}{\partial x_2} &= -4x_2 - \lambda &&= 0 \\
\frac{\partial L}{\partial \lambda} &= -(x_1 + x_2 - 4) &&= \qquad\qquad 0
\end{aligned}
$$

3. Solve the system of equations: Using the first two partials, we see that $\lambda = -2x_1$ and $\lambda = -4x_2$ Set these equal to see that $x_1 = 2x_2$. Using the third partial and the above equality, $4 = 2x_2 + x_2$ from which we get

$$x_2^* = 4/3, x_1^* = 8/3, \lambda = -16/3$$

4. Therefore, the only critical point is $x_1^* = \frac{8}{3}$ and $x_2^* = \frac{4}{3}$
5. This gives $f(\frac{8}{3}, \frac{4}{3}) = -\frac{96}{9}$, which is less than the unconstrained optimum $f(0, 0) = 0$

Notice that when we take the partial derivative of L with respect to the Lagrangian multiplier and set it equal to 0, we return exactly our constraint! This is why signs matter.

5.6 Inequality Constraints

Inequality constraints define the boundary of a region over which we seek to optimize the function. This makes inequality constraints more challenging because we do not know if the maximum/minimum lies along one of the constraints (the constraint binds) or in the interior of the region.

We must introduce more variables in order to turn the problem into an unconstrained optimization.

Slack: For each inequality constraint $c_i(x_1, \ldots, x_n) \leq a_i$, we define a slack variable s_i^2 for which the expression $c_i(x_1, \ldots, x_n) \leq a_i - s_i^2$ would hold with equality. These slack variables capture how close the constraint comes to binding. We use s^2 rather than s to ensure that the slack is positive.

Slack is just a way to transform our constraints.

Given a two-dimensional set-up and these edited constraints:

$$\max_{x_1,x_2} / \min_{x_1,x_2} f(x_1, x_2) \text{ s.t. } c(x_1, x_2) \leq a_1$$

Adding in Slack:

$$\max_{x_1,x_2} / \min_{x_1,x_2} f(x_1, x_2) \text{ s.t. } c(x_1, x_2) \leq a_1 - s_1^2$$

We define the Lagrangian function $L(x_1, x_2, \lambda_1, s_1)$ as follows:

$$L(x_1, x_2, \lambda_1, s_1) = f(x_1, x_2) - \lambda_1(c(x_1, x_2) + s_1^2 - a_1)$$

More generally, in n dimensions:

$$L(x_1, \ldots, x_n, \lambda_1, \ldots, \lambda_k, s_1, \ldots, s_k) = f(x_1, \ldots, x_n) - \sum_{i=1}^{k} \lambda_i(c_i(x_1, \ldots, x_n) + s_i^2 - a_i)$$

Finding the Critical Points: To find the critical points, we take the partial derivatives of the lagrangian function, $L(x_1, \ldots, x_n, \lambda_1, \ldots, \lambda_k, s_1, \ldots, s_k)$, with respect to each of its variables (all choice variables x, all lagrangian multipliers λ, and all slack variables s). At a critical point, *each* of these partial derivatives must be equal to zero, so we obtain a system of $n + 2k$ equations in $n + 2k$ unknowns:

$$\begin{aligned}
\frac{\partial L}{\partial x_1} &= \frac{\partial f}{\partial x_1} - \sum_{i=1}^{k} \lambda_i \frac{\partial c_i}{\partial x_1} = 0 \\
\vdots &= \vdots \\
\frac{\partial L}{\partial x_n} &= \frac{\partial f}{\partial x_n} - \sum_{i=1}^{k} \lambda_i \frac{\partial c_i}{\partial x_n} = 0 \\
\frac{\partial L}{\partial \lambda_1} &= c_1(x_i, \ldots, x_n) + s_1^2 - b_1 = 0 \\
\vdots &= \vdots \\
\frac{\partial L}{\partial \lambda_k} &= c_k(x_i, \ldots, x_n) + s_k^2 - b_k = 0 \\
\frac{\partial L}{\partial s_1} &= 2s_1\lambda_1 = 0 \\
\vdots &= \vdots \\
\frac{\partial L}{\partial s_k} &= 2s_k\lambda_k = 0
\end{aligned}$$

Complementary slackness conditions: The last set of first order conditions of the form $2s_i\lambda_i = 0$ (the partials taken with respect to the slack variables) are known as complementary slackness conditions. These conditions can be satisfied one of three ways:

1. $\lambda_i = 0$ and $s_i \neq 0$: This implies that the slack is positive and thus *the constraint does not bind.*
2. $\lambda_i \neq 0$ and $s_i = 0$: This implies that there is no slack in the constraint and *the constraint does bind.*
3. $\lambda_i = 0$ and $s_i = 0$: In this case, there is no slack but the *constraint binds trivially*, without changing the optimum.

Example: Find the critical points for the following constrained optimization:

$$\max_{x_1,x_2} f(x) = -(x_1^2 + 2x_2^2) \text{ s.t. } x_1 + x_2 \leq 4$$

1. Rewrite with the slack variables:

$$\max_{x_1,x_2} f(x) = -(x_1^2 + 2x_2^2) \text{ s.t. } x_1 + x_2 \leq 4 - s_1^2$$

2. Write the Lagrangian:

$$L(x_1, x_2, \lambda_1, s_1) = -(x_1^2 + 2x_2^2) - \lambda_1(x_1 + x_2 + s_1^2 - 4)$$

3. Take the partial derivatives and set equal to 0:

$$\begin{aligned}
\frac{\partial L}{\partial x_1} &= -2x_1 - \lambda_1 = 0 \\
\frac{\partial L}{\partial x_2} &= -4x_2 - \lambda_1 = 0 \\
\frac{\partial L}{\partial \lambda_1} &= -(x_1 + x_2 + s_1^2 - 4) = 0 \\
\frac{\partial L}{\partial s_1} &= -2s_1\lambda_1 = 0
\end{aligned}$$

4. Consider all ways that the complementary slackness conditions are solved:

Hypothesis	s_1	λ_1	x_1	x_2	$f(x_1, x_2)$
$s_1 = 0$ $\lambda_1 = 0$	No solution				
$s_1 \neq 0$ $\lambda_1 = 0$	2	0	0	0	0
$s_1 = 0$ $\lambda_1 \neq 0$	0	$\frac{-16}{3}$	$\frac{8}{3}$	$\frac{4}{3}$	$-\frac{32}{3}$
$s_1 \neq 0$ $\lambda_1 \neq 0$	No solution				

This shows that there are two critical points: $(0, 0)$ and $(\frac{8}{3}, \frac{4}{3})$.

5. Find maximum: Looking at the values of $f(x_1, x_2)$ at the critical points, we see that $f(x_1, x_2)$ is maximized at $x_1^* = 0$ and $x_2^* = 0$.

Exercise 5.3. Example: Find the critical points for the following constrained optimization:

$$\max_{x_1,x_2} f(x) = -(x_1^2 + 2x_2^2) \text{ s.t. } \begin{array}{l} x_1 + x_2 \leq 4 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{array}$$

1. Rewrite with the slack variables:

2. Write the Lagrangian:

3. Take the partial derivatives and set equal to zero:

4. Consider all ways that the complementary slackness conditions are solved:

Hypothesis	s_1	s_2	s_3	λ_1	λ_2	λ_3	x_1	x_2	$f(x_1, x_2)$
$s_1 = s_2 = s_3 = 0$									
$s_1 \neq 0, s_2 = s_3 = 0$									
$s_2 \neq 0, s_1 = s_3 = 0$									
$s_3 \neq 0, s_1 = s_2 = 0$									
$s_1 \neq 0, s_2 \neq 0, s_3 = 0$									
$s_1 \neq 0, s_3 \neq 0, s_2 = 0$									
$s_2 \neq 0, s_3 \neq 0, s_1 = 0$									
$s_1 \neq 0, s_2 \neq 0, s_3 \neq 0$									

5. Find maximum:

5.7 Kuhn-Tucker Conditions

As you can see, this can be a pain. When dealing explicitly with *non-negativity constraints*, this process is simplified by using the Kuhn-Tucker method.

Because the problem of maximizing a function subject to inequality and non-negativity constraints arises frequently in economics, the **Kuhn-Tucker conditions** provides a method that often makes it easier to both calculate the critical points and identify points that are (local) maxima.

Given a *two-dimensional set-up*:

$$\max_{x_1,x_2} / \min_{x_1,x_2} f(x_1,x_2) \text{ s.t. } \begin{array}{l} c(x_1,x_2) \leq a_1 \\ x_1 \geq 0 \\ gx_2 \geq 0 \end{array}$$

We define the Lagrangian function $L(x_1, x_2, \lambda_1)$ the same as if we did not have the non-negativity constraints:

$$L(x_1, x_2, \lambda_2) = f(x_1, x_2) - \lambda_1(c(x_1, x_2) - a_1)$$

More generally, in n dimensions:

$$L(x_1, \ldots, x_n, \lambda_1, \ldots, \lambda_k) = f(x_1, \ldots, x_n) - \sum_{i=1}^{k} \lambda_i(c_i(x_1, \ldots, x_n) - a_i)$$

Kuhn-Tucker and Complementary Slackness Conditions: To find the critical points, we first calculate the Kuhn-Tucker conditions by taking the partial derivatives of the lagrangian function, $L(x_1, \ldots, x_n, \lambda_1, \ldots, \lambda_k)$, with respect to each of its variables (all choice variable sx and all lagrangian multipliers λ) and we calculate the *complementary slackness conditions* by multiplying each partial derivative by its respective variable *and* include non-negativity conditions for all variables (choice variables x and lagrangian multipliers λ).

Kuhn-Tucker Conditions

$$\begin{aligned}\frac{\partial L}{\partial x_1} \leq 0, \ldots, \frac{\partial L}{\partial x_n} \leq 0\\ \frac{\partial L}{\partial \lambda_1} \geq 0, \ldots, \frac{\partial L}{\partial \lambda_m} \geq 0\end{aligned}$$

Complementary Slackness Conditions

$$\begin{aligned}x_1\frac{\partial L}{\partial x_1} = 0, \ldots, x_n\frac{\partial L}{\partial x_n} = 0\\ \lambda_1\frac{\partial L}{\partial \lambda_1} = 0, \ldots, \lambda_m\frac{\partial L}{\partial \lambda_m} = 0\end{aligned}$$

Non-negativity Conditions

$$\begin{array}{ccc} x_1 \geq 0 & \ldots & x_n \geq 0 \\ \lambda_1 \geq 0 & \ldots & \lambda_m \geq 0 \end{array}$$

Note that some of these conditions are set equal to 0, while others are set as inequalities!

Note also that to minimize the function $f(x_1, \ldots, x_n)$, the simplest thing to do is maximize the function $-f(x_1, \ldots, x_n)$; all of the conditions remain the same after reformulating as a maximization problem.

There are additional assumptions (notably, f(x) is quasi-concave and the constraints are convex) that are sufficient to ensure that a point satisfying the Kuhn-Tucker conditions is a global max; if these assumptions do not hold, you may have to check more than one point.

Finding the Critical Points with Kuhn-Tucker Conditions: Given the above conditions, to find the critical points we solve the above system of equations. To do so, we must check *all* border and interior solutions to see if they satisfy the above conditions.

In a two-dimensional set-up, this means we must check the following cases:

1. $x_1 = 0, x_2 = 0$ Border Solution
2. $x_1 = 0, x_2 \neq 0$ Border Solution
3. $x_1 \neq 0, x_2 = 0$ Border Solution
4. $x_1 \neq 0, x_2 \neq 0$ Interior Solution

Example 5.5 (Kuhn-Tucker with two variables)**.** Solve the following optimization problem with inequality constraints

$$\max_{x_1,x_2} f(x) = -(x_1^2 + 2x_2^2)$$

$$\text{s.t.} \begin{cases} x_1 + x_2 * \leq 4 \\ x_1 * \geq 0 \\ x_2 * \geq 0 \end{cases}$$

1. Write the Lagrangian:

$$L(x_1, x_2, \lambda) = -(x_1^2 + 2x_2^2) - \lambda(x_1 + x_2 - 4)$$

2. Find the First Order Conditions:

Kuhn-Tucker Conditions

$$\begin{aligned} \frac{\partial L}{\partial x_1} &= -2x_1 - \lambda \leq 0 \\ \frac{\partial L}{\partial x_2} &= -4x_2 - \lambda \leq 0 \\ \frac{\partial L}{\partial \lambda} &= -(x_1 + x_2 - 4) \geq 0 \end{aligned}$$

Complementary Slackness Conditions

$$\begin{aligned} x_1 \frac{\partial L}{\partial x_2} &= x_1(-2x_1 - \lambda) = 0 \\ x_2 \frac{\partial L}{\partial x_2} &= x_2(-4x_2 - \lambda) = 0 \\ \lambda \frac{\partial L}{\partial \lambda} &= -\lambda(x_1 + x_2 - 4) = 0 \end{aligned}$$

Non-negativity Conditions

$$\begin{aligned} x_1 &\geq 0 \\ x_2 &\geq 0 \\ \lambda &\geq 0 \end{aligned}$$

3. Consider all border and interior cases:

Hypothesis	λ	x_1	x_2	$f(x_1, x_2)$
$x_1 = 0, x_2 = 0$	0	0	0	0
$x_1 = 0, x_2 \neq 0$	-16	0	4	-32
$x_1 \neq 0, x_2 = 0$	-8	4	0	-16
$x_1 \neq 0, x_2 \neq 0$	$-\frac{16}{3}$	$\frac{8}{3}$	$\frac{4}{3}$	$-\frac{32}{3}$

4. Find Maximum: Three of the critical points violate the requirement that $\lambda \geq 0$, so the point $(0,0,0)$ is the maximum.

Exercise 5.4 (Kuhn-Tucker with logs)**.** Solve the constrained optimization problem,

$$\max_{x_1,x_2} f(x) = \frac{1}{3}\log(x_1+1) + \frac{2}{3}\log(x_2+1) \text{ s.t. } \begin{array}{l} x_1 + 2x_2 \leq 4 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{array}$$

1. Write the Lagrangian:

2. Find the First Order Conditions:
   Kuhn-Tucker Conditions

   Complementary Slackness Conditions

   Non-negativity Conditions

3. Consider all border and interior cases:

Hypothesis	λ	x_1	x_2	$f(x_1, x_2)$
$x_1 = 0, x_2 = 0$				
$x_1 = 0, x_2 \neq 0$				
$x_1 \neq 0, x_2 = 0$				
$x_1 \neq 0, x_2 \neq 0$				

4. Find Maximum:

5.8 Applications of Quadratic Forms

Curvature and The Taylor Polynomial as a Quadratic Form: The Hessian is used in a Taylor polynomial approximation to $f(\mathbf{x})$ and provides information about the curvature of $f(\mathbf{x})$ at $\mathbf{x}$ — e.g., which tells us whether a critical point $\mathbf{x}^*$ is a min, max, or saddle point.

1. The second order Taylor polynomial about the critical point $\mathbf{x}^*$ is

$$f(\mathbf{x}^* + \mathbf{h}) = \mathbf{f}(\mathbf{x}^*) + \nabla \mathbf{f}(\mathbf{x}^*)\mathbf{h} + \frac{\mathbf{1}}{\mathbf{2}}\mathbf{h}^\top \mathbf{H}(\mathbf{x}^*)\mathbf{h} + \mathbf{R}(\mathbf{h})$$

2. Since we're looking at a critical point, $\nabla f(\mathbf{x}^*) = 0$; and for small $\mathbf{h}$, $R(\mathbf{h})$ is negligible. Rearranging, we get

$$f(\mathbf{x}^* + \mathbf{h}) - \mathbf{f}(\mathbf{x}^*) \approx \frac{\mathbf{1}}{\mathbf{2}}\mathbf{h}^\top \mathbf{H}(\mathbf{x}^*)\mathbf{h}$$

3. The Righthand side here is a quadratic form and we can determine the definiteness of $\mathbf{H}(\mathbf{x}^*)$.

Chapter 6

Probability Theory

Probability and Inferences are mirror images of each other, and both are integral to social science. Probability quantifies uncertainty, which is important because many things in the social world are at first uncertain. Inference is then the study of how to learn about facts you don't observe from facts you do observe.

6.1 Counting rules

Fundamental Theorem of Counting: If an object has j different characteristics that are independent of each other, and each characteristic i has n_i ways of being expressed, then there are $\prod_{i=1}^{j} n_i$ possible unique objects.

Example: Cards can be either red or black and can take on any of 13 values.

$j =$

$n_{\text{color}} =$

$n_{\text{number}} =$

Number of Outcomes =

We often need to count the number of ways to choose a subset from some set of possibilities. The number of outcomes depends on two characteristics of the process: does the order matter and is replacement allowed?

Sampling Table: If there are n objects which are numbered 1 to n and we select $k < n$ of them, how many different outcomes are possible?

If the order in which a given object is selected matters, selecting 4 numbered objects in the following order (1, 3, 7, 2) and selecting the same four objects but in a different order such as (7, 2, 1, 3) will be counted as different outcomes.

If replacement is allowed, there are always the same n objects to select from. However, if replacement is not allowed, there is always one less option than the previous round when making a selection. For example, if replacement is not allowed and I am selecting 3 elements from the following set $\{1, 2, 3, 4, 5, 6\}$, I will have 6 options at first, 5 options as I make my second selection, and 4 options as I make my third.

So in counting how many different outcomes are possible, if ***order matters*** AND we are sampling ***with replacement***, the number of different outcomes is n^k.

If ***order matters*** AND we are sampling ***without replacement***, the number of different outcomes is $n(n-1)(n-2)...(n-k+1) = \frac{n!}{(n-k)!}$.

If ***order doesn't matter*** AND we are sampling ***without replacement***, the number of different outcomes is $\binom{n}{k} = \frac{n!}{(n-k)!k!}$.

Expression $\binom{n}{k}$ is read as "n choose k" and denotes $\frac{n!}{(n-k)!k!}$. Also, note that $0! = 1$.

Example 6.1 (Counting)**.** There are five balls numbered from 1 through 5 in a jar. Three balls are chosen. How many possible choices are there?

1. Ordered, with replacement =
2. Ordered, without replacement =
3. Unordered, without replacement =

Exercise 6.1 (Counting)**.** Four cards are selected from a deck of 52 cards. Once a card has been drawn, it is not reshuffled back into the deck. Moreover, we care only about the complete hand that we get (i.e. we care about the set of selected cards, not the sequence in which it was drawn). How many possible outcomes are there?

6.2 Sets

Set : A set is any well defined collection of elements. If x is an element of S, $x \in S$.

Sample Space (S): A set or collection of all possible outcomes from some process. Outcomes in the set can be discrete elements (countable) or points along a continuous interval (uncountable).

Examples:

1. Discrete: the numbers on a die, whether a vote cast is republican or democrat.
2. Continuous: GNP, arms spending, age.

Event: Any collection of possible outcomes of an experiment. Any subset of the full set of possibilities, including the full set itself. Event A ⊂ S.

Empty Set: a set with no elements. $S = \{\}$. It is denoted by the symbol $\emptyset$.

Set operations:

1. **Union**: The union of two sets A and B, $A \cup B$, is the set containing all of the elements in A or B.

$$A_1 \cup A_2 \cup \cdots \cup A_n = \bigcup_{i=1}^{n} A_i$$

2. **Intersection**: The intersection of sets A and B, $A \cap B$, is the set containing all of the elements in both A and B.

$$A_1 \cap A_2 \cap \cdots \cap A_n = \bigcap_{i=1}^{n} A_i$$

3. **Complement**: If set A is a subset of S, then the complement of A, denoted A^C, is the set containing all of the elements in S that are not in A.

Properties of set operations:

- **Commutative**: $A \cup B = B \cup A$; $A \cap B = B \cap A$
- **Associative**: $A \cup (B \cup C) = (A \cup B) \cup C$; $A \cap (B \cap C) = (A \cap B) \cap C$
- **Distributive**: $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$; $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
- **de Morgan's laws**: $(A \cup B)^C = A^C \cap B^C$; $(A \cap B)^C = A^C \cup B^C$
- **Disjointness**: Sets are disjoint when they do not intersect, such that $A \cap B = \emptyset$. A collection of sets is pairwise disjoint (**mutually exclusive**) if, for all $i \neq j$, $A_i \cap A_j = \emptyset$. A collection of sets form a partition of set S if they are pairwise disjoint and they cover set S, such that $\bigcup_{i=1}^{k} A_i = S$.

Example 6.2 (Sets)**.** Let set A be {1, 2, 3, 4}, B be {3, 4, 5, 6}, and C be {5, 6, 7, 8}. Sets A, B, and C are all subsets of the sample space S which is {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

Write out the following sets:

1. $A \cup B$
2. $C \cap B$
3. B^c
4. $A \cap (B \cup C)$

Exercise 6.2 (Sets)**.** Suppose you had a pair of four-sided dice. You sum the results from a single toss.

What is the set of possible outcomes (i.e. the sample space)?

Consider subsets A {2, 8} and B {2,3,7} of the sample space you found. What is

1. A^c
2. $(A \cup B)^c$

6.3 Probability

Historians think this totally meaningless and nonsensical statistic is the product of an early-modern epistemological shift in which numbers and quantifiable data became revered above other kinds of knowledge as the most useful and credible form of truth https://t.co/wVFyAQGxEv

— Gina Anne Tam (**?**) May 29, 2018

Probability Definitions: Formal and Informal

Many events or outcomes are random. In everyday speech, we say that we are *uncertain* about the outcome of random events. Probability is a formal model of uncertainty which provides a measure of uncertainty governed by a particular set of rules. A different model of uncertainty would, of course, have a different set of rules and measures. Our focus on probability is justified because it has proven to be a particularly useful model of uncertainty.

Probability Distribution Function: a mapping of each event in the sample space S to the real numbers that satisfy the following three axioms (also called Kolmogorov's Axioms).

Formally,

Definition 6.1 (Probability)**.** Probability is a function that maps events to a real number, obeying the axioms of probability.

The axioms of probability make sure that the separate events add up in terms of probability, and – for standardization purposes – that they add up to 1.

Definition 6.2 (Axioms of Probability)**.** 1. For any event A, $P(A) \geq 0$.

2. $P(S) = 1$
3. The Countable Additivity Axiom: For any sequence of *disjoint* (mutually exclusive) events $A_1, A_2, \ldots$ (of which there may be infinitely many),

$$P\left(\bigcup_{i=1}^{k} A_i\right) = \sum_{i=1}^{k} P(A_i)$$

The last axiom is an extension of a union to infinite sets. When there are only two events in the space, it boils down to:

$$P(A_1 \cup A_2) = P(A_1) + P(A_2) \quad \text{for disjoint } A_1, A_2$$

Probability Operations

Using these three axioms, we can define all of the common rules of probability.

1. $P(\emptyset) = 0$
2. For any event A, $0 \leq P(A) \leq 1$.
3. $P(A^C) = 1 - P(A)$
4. If $A \subset B$ (A is a subset of B), then $P(A) \leq P(B)$.
5. For *any* two events A and B, $P(A \cup B) = P(A) + P(B) - P(A \cap B)$
6. Boole's Inequality: For any sequence of n events (which need not be disjoint) $A_1, A_2, \ldots, A_n$, then $P\left(\bigcup\limits_{i=1}^{n} A_i\right) \leq \sum\limits_{i=1}^{n} P(A_i)$.

Example 6.3 (Probability)**.** Let's assume we have an evenly-balanced, six-sided die.

Then,

1. Sample space S =
2. $P(1) = \cdots = P(6) =$
3. $P(\emptyset) = P(7) =$
4. $P(\{1, 3, 5\}) =$
5. $P\left(\{1, 2\}^C\right) = P(\{3, 4, 5, 6\}) =$
6. Let $A = \{1, 2, 3, 4, 5\} \subset S$. Then $P(A) = 5/6 < P(S) =$
7. Let $A = \{1, 2, 3\}$ and $B = \{2, 4, 6\}$. Then $A \cup B$? $A \cap B$? $P(A \cup B)$?

Exercise 6.3 (Probability)**.** Suppose you had a pair of four-sided dice. You sum the results from a single toss. Let us call this sum, or the outcome, X.

1. What is $P(X = 5)$, $P(X = 3)$, $P(X = 6)$?

2. What is $P(X = 5 \cup X = 3)^C$?

6.4 Conditional Probability and Bayes Law

Conditional Probability: The conditional probability $P(A|B)$ of an event A is the probability of A, given that another event B has occurred. Conditional probability allows for the inclusion of other information into the calculation of the probability of an event. It is calculated as

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Note that conditional probabilities are probabilities and must also follow the Kolmagorov axioms of probability.

Example 6.4 (Conditional Probability 1)**.** Assume A and B occur with the following frequencies:

	A	A^c
B	n_{ab}	$n_{a^c b}$
B^C	n_{ab^c}	$n_{(ab)^c}$

and let $n_{ab} + n_{a^C b} + n_{ab^C} + n_{(ab)^C} = N$. Then

1. $P(A) =$
2. $P(B) =$
3. $P(A \cap B) =$
4. $P(A|B) = \frac{P(A \cap B)}{P(B)} =$
5. $P(B|A) = \frac{P(A \cap B)}{P(A)} =$

Example 6.5 (Conditional Probability 2)**.** A six-sided die is rolled. What is the probability of a 1, given the outcome is an odd number?

You could rearrange the fraction to highlight how a joint probability could be expressed as the product of a conditional probability.

Definition 6.3 (Multiplicative Law of Probability)**.** The probability of the intersection of two events A and B is $P(A \cap B) = P(A)P(B|A) = P(B)P(A|B)$ which follows directly from the definition of conditional probability. More generally,

$$P(A_1 \cap \cdots \cap A_k) = P(A_k|A_{k-1} \cap \cdots \cap A_1) \times P(A_{k-1}|A_{k-2} \cap \cdots A_1) \times \ldots \times P(A_2|A_1) \times P(A_1)$$

Sometimes it is easier to calculate these conditional probabilities and sum them than it is to calculate $P(A)$ directly.

Definition 6.4 (Law of Total Probability)**.** Let S be the sample space of some experiment and let the disjoint k events $B_1, \ldots, B_k$ partition S, such that $P(B_1 \cup ... \cup B_k) = P(S) = 1$. If A is some other event in S, then the events $A \cap B_1, A \cap B_2, \ldots, A \cap B_k$ will form a partition of A and we can write A as

$$A = (A \cap B_1) \cup \cdots \cup (A \cap B_k)$$

.

Since the k events are disjoint,

$$\begin{aligned} P(A) &= \sum_{i=1}^{k} P(A \cap B_i) \\ &= \sum_{i=1}^{k} P(B_i)P(A|B_i) \end{aligned}$$

Bayes Rule: Assume that events $B_1, \ldots, B_k$ form a partition of the space S. Then by the Law of Total Probability

$$P(B_j|A) = \frac{P(A \cap B_j)}{P(A)} = \frac{P(B_j)P(A|B_j)}{\sum\limits_{i=1}^{k} P(B_i)P(A|B_i)}$$

If there are only two states of B, then this is just

$$P(B_1|A) = \frac{P(B_1)P(A|B_1)}{P(B_1)P(A|B_1) + P(B_2)P(A|B_2)}$$

Bayes' rule determines the posterior probability of a state $P(B_j|A)$ by calculating the probability $P(A \cap B_j)$ that both the event A and the state B_j will occur and dividing it by the probability that the event will occur regardless of the state (by summing across all B_i). The states could be something like Normal/Defective, Healthy/Diseased, Republican/Democrat/Independent, etc. The event on which one conditions could be something like a sampling from a batch of components, a test for a disease, or a question about a policy position.

Prior and Posterior Probabilities: Above, $P(B_1)$ is often called the prior probability, since it's the probability of B_1 before anything else is known. $P(B_1|A)$ is called the posterior probability, since it's the probability after other information is taken into account.

Example 6.6 (Bayes' Rule)**.** In a given town, 40% of the voters are Democrat and 60% are Republican. The president's budget is supported by 50% of the Democrats and 90% of the Republicans. If a randomly (equally likely) selected voter is found to support the president's budget, what is the probability that they are a Democrat?

Exercise 6.4 (Conditional Probability)**.** Assume that 2% of the population of the U.S. are members of some extremist militia group. We develop a survey that positively classifies someone as being a member of a militia group given that they are a member 95% of the time and negatively classifies someone as not being a member of a militia group given that they are not a member 97% of the time. What is the probability that someone positively classified as being a member of a militia group is actually a militia member?

6.5 Independence

Definition 6.5 (Independence)**.** If the occurrence or nonoccurrence of either events A and B have no effect on the occurrence or nonoccurrence of the other, then A and B are independent.

If A and B are independent, then

1. $P(A|B) = P(A)$
2. $P(B|A) = P(B)$
3. $P(A \cap B) = P(A)P(B)$
4. More generally than the above, $P(\bigcap_{i=1}^{k} A_i) = \prod_{i=1}^{K} P(A_i)$

Are mutually exclusive events independent of each other?

No. If A and B are mutually exclusive, then they cannot happen simultaneously. If we know that A occurred, then we know that B couldn't have occurred. Because of this, A and B aren't independent.

Pairwise Independence: A set of more than two events $A_1, A_2, \ldots, A_k$ is pairwise independent if $P(A_i \cap A_j) = P(A_i)P(A_j)$, $\forall i \neq j$. Note that this does **not** necessarily imply joint independence.

Conditional Independence: If A and B are independent once you know the occurrence of a third event C, then A and B are conditionally independent (conditional on C):

1. $P(A|B \cap C) = P(A|C)$
2. $P(B|A \cap C) = P(B|C)$
3. $P(A \cap B|C) = P(A|C)P(B|C)$

Just because two events are conditionally independent does not mean that they are independent. Actually it is hard to think of real-world things that are "unconditionally" independent. That's why it's always important to ask about a finding: What was it conditioned on? For example, suppose that a graduate school admission decisions are done by only one professor, who picks a group of 50 bright students and flips a coin for each student to generate a class of about 25 students. Then the the probability that two students get accepted are conditionally independent, because they are determined by two separate coin tosses. However, this does not mean that their admittance is not completely independent. Knowing that student A got in gives us information about whether student B got in, if we think that the professor originally picked her pool of 50 students by merit.

Perhaps more counterintuitively: If two events are already independent, then it might seem that no amount of "conditioning" will make them dependent. But this is not always so. For example[1], suppose I only get a call from two people, Alice and Bob. Let A be the event that Alice calls, and B be the event that Bob calls. Alice and Bob do not communicate, so $P(A \mid B) = P(A)$. But now let C be the event that your phone rings. For conditional independence to hold here, then $P(A \mid C)$ must be equal to $P(A \mid B \cap C)$. But this is not true – $A \mid C$ may or may not be true, but $P(A \mid B \cap C)$ certainly is true.

6.6 Random Variables

Most questions in the social sciences involve events, rather than numbers per se. To analyze and reason about events quantitatively, we need a way of mapping events to numbers. A random variable does exactly that.

Definition 6.6 (Random Variable)**.** A random variable is a measurable function X that maps from the sample space S to the set of real numbers R. It assigns a real number to every outcome $s \in S$.

It might seem strange to define a random variable as a function – which is neither random nor variable. The randomness comes from the realization of an event from the sample space s.

Randomness means that the outcome of some experiment is not deterministic, i.e. there is some probability $(0 < P(A) < 1)$ that the event will occur.

The support of a random variable is all values for which there is a positive probability of occurrence.

Example: Flip a fair coin two times. What is the sample space?

A random variable must map events to the real line. For example, let a random variable X be the number of heads. The event (H, H) gets mapped to 2 $X(s) = 2$, and the events $\{(H, T), (T, H)\}$ gets mapped to 1 $(X(s) = 1)$, the event (T, T) gets mapped to 0 $(X(s) = 0)$.

[1] Example taken from Blitzstein and Hwang, Example 2.5.10

What are other possible random variables?

6.7 Distributions

We now have two main concepts in this section – probability and random variables. Given a sample space S and the same experiment, both probability and random variables take events as their inputs. But they output different things (probabilities measure the "size" of events, random variables give a number in a way that the analyst chose to define the random variable). How do the two concepts relate?

The concept of distributions is the natural bridge between these two concepts.

Definition 6.7 (Distribution of a random variable)**.** A distribution of a random variable is a function that specifies the probabilities of all events associated with that random variable. There are several types of distributions: A probability mass function for a discrete random variable and probability density function for a continuous random variable.

Notice how the definition of distributions combines two ideas of random variables and probabilities of events. First, the distribution considers a random variable, call it X. X can take a number of possible numeric values.

Example 6.7 (Total Number of Occurrences)**.** Consider three binary outcomes, one for each patient recovering from a disease: R_i denotes the event in which patient i ($i = 1, 2, 3$) recovers from a disease. R_1, R_2, and R_3. How would we represent the total number of people who end up recovering from the disease?

Solution. Define the random variable X be the total number of people (out of three) who recover from the disease. Random variables are functions, that take as an input a set of events (in the sample space S) and deterministically assigns them to a number of the analyst's choice.

Recall that with each of these numerical values there is a class of *events*. In the previous example, for $X = 3$ there is one outcome (R_1, R_2, R_3) and for $X = 1$ there are multiple $(\{(R_1, R_2^c, R_3^c), (R_1^c, R_2, R_3^c), (R_1^c, R_2^c, R_3), \})$. Now, the thing to notice here is that each of these events naturally come with a probability associated with them. That is, $P(R_1, R_2, R_3)$ is a number from 0 to 1, as is $P(R_1, R_2^c, R_3^c)$. These all have probabilities because they are in the sample space S. The function that tells us these probabilities that are associated with a numerical value of a random variable is called a distribution.

In other words, a random variable X *induces a probability distribution* P (sometimes written P_X to emphasize that the probability density is about the r.v. X)

Discrete Random Variables

The formal definition of a random variable is easier to given by separating out two cases: discrete random variables when the numeric summaries of the events are discrete, and continuous random variables when they are continuous.

Definition 6.8 (Discrete Random Variable)**.** X is a discrete random variable if it can assume only a finite or countably infinite number of distinct values. Examples: number of wars per year, heads or tails.

The distribution of a discrete r.v. is a PMF:

Definition 6.9 (Probability Mass Function)**.** For a discrete random variable X, the probability mass function (Also referred to simply as the "probability distribution.") (PMF), $p(x) = P(X = x)$, assigns probabilities to a countable number of distinct x values such that

1. $0 \leq p(x) \leq 1$
2. $\sum\limits_{y} p(x) = 1$

Example: For a fair six-sided die, there is an equal probability of rolling any number. Since there are six sides, the probability mass function is then $p(y) = 1/6$ for $y = 1, \ldots, 6$, 0 otherwise.}

In a discrete random variable, **cumulative density function** (Also referred to simply as the "cumulative distribution" or previously as the "density function"), $F(x)$ or $P(X \leq x)$, is the probability that X is less than or equal to some value x, or

$$P(X \leq x) = \sum_{i \leq x} p(i)$$

Properties a CDF must satisfy:

1. $F(x)$ is non-decreasing in x.
2. $\lim\limits_{x \to -\infty} F(x) = 0$ and $\lim\limits_{x \to \infty} F(x) = 1$
3. $F(x)$ is right-continuous.

Note that $P(X > x) = 1 - P(X \leq x)$.

Example 6.8. For a fair die with its value as Y, What are the following?

1. $P(Y \leq 1)$
2. $P(Y \leq 3)$
3. $P(Y \leq 6)$

Continuous Random Variables

We also have a similar definition for *continuous* random variables.

Definition 6.10 (Continuous Random Variable)**.** X is a continuous random variable if there exists a nonnegative function $f(x)$ defined for all real $x \in (-\infty, \infty)$, such that for any interval A, $P(X \in A) = \int\limits_{A} f(x)dx$. Examples: age, income, GNP, temperature.

Definition 6.11 (Probability Density Function)**.** The function f above is called the probability density function (pdf) of X and must satisfy

$$f(x) \geq 0$$

$$\int\limits_{-\infty}^{\infty} f(x)dx = 1$$

Note also that $P(X = x) = 0$ — i.e., the probability of any point y is zero.

For both discrete and continuous random variables, we have a unifying concept of another measure: the cumulative distribution:

Definition 6.12 (Cumulative Density Function)**.** Because the probability that a continuous random variable will assume any particular value is zero, we can only make statements about the probability of a continuous random variable being within an interval. The cumulative distribution gives the probability that Y lies on the interval $(-\infty, y)$ and is defined as

$$F(x) = P(X \leq x) = \int\limits_{-\infty}^{x} f(s)ds$$

Note that $F(x)$ has similar properties with continuous distributions as it does with discrete - non-decreasing, continuous (not just right-continuous), and $\lim\limits_{x \to -\infty} F(x) = 0$ and $\lim\limits_{x \to \infty} F(x) = 1$.

We can also make statements about the probability of Y falling in an interval $a \leq y \leq b$.

$$P(a \leq x \leq b) = \int\limits_{a}^{b} f(x)dx$$

The PDF and CDF are linked by the integral: The CDF of the integral of the PDF:

$$f(x) = F'(x) = \frac{dF(x)}{dx}$$

Example 6.9. For $f(y) = 1, \quad 0 < y < 1$, find: (1) The CDF $F(y)$ and (2) The probability $P(0.5 < y < 0.75)$.

6.8 Joint Distributions

Often, we are interested in two or more random variables defined on the same sample space. The distribution of these variables is called a joint distribution. Joint distributions can be made up of any combination of discrete and continuous random variables.

Joint Probability Distribution: If both X and Y are random variable, their joint probability mass/density function assigns probabilities to each pair of outcomes

Discrete:

$$p(x,y) = P(X = x, Y = y)$$

such that $p(x,y) \in [0,1]$ and

$$\sum \sum p(x,y) = 1$$

Continuous:

$$f(x,y); P((X,Y) \in A) = \iint_A f(x,y)dxdy$$

s.t. $f(x,y) \geq 0$ and

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y)dxdy = 1$$

If X and Y are independent, then $P(X = x, Y = y) = P(X = x)P(Y = y)$ and $f(x,y) = f(x)f(y)$

Marginal Probability Distribution: probability distribution of only one of the two variables (ignoring information about the other variable), we can obtain the marginal distribution by summing/integrating across the variable that we don't care about:

- Discrete: $p_X(x) = \sum_i p(x, y_i)$
- Continuous: $f_X(x) = \int_{-\infty}^{\infty} f(x,y)dy$

Conditional Probability Distribution: probability distribution for one variable, holding the other variable fixed. Recalling from the previous lecture that $P(A|B) = \frac{P(A\cap B)}{P(B)}$, we can write the conditional distribution as

- Discrete: $p_{Y|X}(y|x) = \frac{p(x,y)}{p_X(x)}, \quad p_X(x) > 0$
- Continuous: $f_{Y|X}(y|x) = \frac{f(x,y)}{f_X(x)}, \quad f_X(x) > 0$

Exercise 6.5 (Discrete Outcomes)**.** Suppose we are interested in the outcomes of flipping a coin and rolling a 6-sided die at the same time. The sample space for this process contains 12 elements:

$$\{(H,1),(H,2),(H,3),(H,4),(H,5),(H,6),(T,1),(T,2),(T,3),(T,4),(T,5),(T,6)\}$$

We can define two random variables X and Y such that $X = 1$ if heads and $X = 0$ if tails, while Y equals the number on the die.

We can then make statements about the joint distribution of X and Y. What are the following?

1. $P(X = x)$

2. $P(Y = y)$
3. $P(X = x, Y = y)$
4. $P(X = x|Y = y)$
5. Are X and Y independent?

6.9 Expectation

We often want to summarize some characteristics of the distribution of a random variable. The most important summary is the expectation (or expected value, or mean), in which the possible values of a random variable are weighted by their probabilities.

Definition 6.13 (Expectation of a Discrete Random Variable)**.** The expected value of a discrete random variable Y is

$$E(Y) = \sum_y yP(Y = y) = \sum_y yp(y)$$

In words, it is the weighted average of all possible values of Y, weighted by the probability that y occurs. It is not necessarily the number we would expect Y to take on, but the average value of Y after a large number of repetitions of an experiment.

Example 6.10 (Expectation of a Discrete Random Variable)**.** What is the expectation of a fair, six-sided die?

Expectation of a Continuous Random Variable: The expected value of a continuous random variable is similar in concept to that of the discrete random variable, except that instead of summing using probabilities as weights, we integrate using the density to weight. Hence, the expected value of the continuous variable Y is defined by

$$E(Y) = \int_y yf(y)dy$$

Example 6.11 (Expectation of a Continuous Random Variable)**.** Find $E(Y)$ for $f(y) = \frac{1}{1.5}, \quad 0 < y < 1.5$.

Expected Value of a Function

Remember: An Expected Value is a type of weighted average. We can extend this to composite functions. For random variable Y,

If Y is Discrete wiht PMF $p(y)$,

$$E[g(Y)] = \sum_y g(y)p(y)$$

If Y is Continuous with PDF $f(y)$,

$$E[g(Y)] = \int_{-\infty}^{\infty} g(y)f(y)dy$$

Properties of Expected Values

Dealing with Expecations is easier when the thing inside is a sum. The intuition behind this that Expectation is an integral, which is a type of sum.

1. Expectation of a constant is a constant
$$E(c) = c$$
2. Constants come out
$$E(cg(Y)) = cE(g(Y))$$
3. Expectation is Linear
$$E(g(Y_1) + \cdots + g(Y_n)) = E(g(Y_1)) + \cdots + E(g(Y_n)),$$
regardless of independence
4. Expected Value of Expected Values:
$$E(E(Y)) = E(Y)$$
(because the expected value of a random variable is a constant)

Finally, if X and Y are independent, even products are easy:

$$E(XY) = E(X)E(Y)$$

Conditional Expectation: With joint distributions, we are often interested in the expected value of a variable Y if we could hold the other variable X fixed. This is the conditional expectation of Y given $X = x$:

1. Y discrete: $E(Y|X = x) = \sum_y yp_{Y|X}(y|x)$
2. Y continuous: $E(Y|X = x) = \int_y yf_{Y|X}(y|x)dy$

The conditional expectation is often used for prediction when one knows the value of X but not Y

6.10 Variance and Covariance

We can also look at other summaries of the distribution, which build on the idea of taking expectations. Variance tells us about the "spread" of the distribution; it is the expected value of the squared deviations from the mean of the distribution. The standard deviation is simply the square root of the variance.

Definition 6.14 (Variance)**.** The Variance of a Random Variable Y is

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

The Standard Deviation is the square root of the variance :

$$SD(Y) = \sigma_Y = \sqrt{\text{Var}(Y)}$$

Example 6.12 (Variance)**.** Given the following PMF:

$$f(x) = \begin{cases} \frac{3!}{x!(3-x)!}(\frac{1}{2})^3 & x = 0, 1, 2, 3 \\ 0 & otherwise \end{cases}$$

What is Var(x)?

Hint: First calculate $E(X)$ and $E(X^2)$

Definition 6.15 (Covariance and Correlation)**.** The covariance measures the degree to which two random variables vary together; if the covariance between X and Y is positive, X tends to be larger than its mean when Y is larger than its mean.

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

We can also write this as

$$\begin{aligned} \text{Cov}(X, Y) &= E\left(XY - XE(Y) - E(X)Y + E(X)E(Y)\right) \\ &= E(XY) - E(X)E(Y) - E(X)E(Y) + E(X)E(Y) \\ &= E(XY) - E(X)E(Y) \end{aligned}$$

The covariance of a variable with itself is the variance of that variable.

The Covariance is unfortunately hard to interpret in magnitude. The correlation is a standardized version of the covariance, and always ranges from -1 to 1.

Definition 6.16 (Correlation)**.** The correlation coefficient is the covariance divided by the standard deviations of X and Y. It is a unitless measure and always takes on values in the interval $[-1, 1]$.

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

Properties of Variance and Covariance:

1. $\text{Var}(c) = 0$
2. $\text{Var}(cY) = c^2\text{Var}(Y)$

3. $\text{Cov}(Y, Y) = \text{Var}(Y)$
4. $\text{Cov}(X, Y) = \text{Cov}(Y, X)$
5. $\text{Cov}(aX, bY) = ab\text{Cov}(X, Y)$
6. $\text{Cov}(X + a, Y) = \text{Cov}(X, Y)$
7. $\text{Cov}(X + Z, Y + W) = \text{Cov}(X, Y) + \text{Cov}(X, W) + \text{Cov}(Z, Y) + \text{Cov}(Z, W)$
8. $\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y) + 2\text{Cov}(X, Y)$

Exercise 6.6 (Expectation and Variance)**.** Suppose we have a PMF with the following characteristics:

$$\begin{aligned} P(X = -2) &= \frac{1}{5} \\ P(X = -1) &= \frac{1}{6} \\ P(X = 0) &= \frac{1}{5} \\ P(X = 1) &= \frac{1}{15} \\ P(X = 2) &= \frac{11}{30} \end{aligned}$$

1. Calculate the expected value of X

Define the random variable $Y = X^2$.

2. Calculate the expected value of Y. (Hint: It would help to derive the PMF of Y first in order to calculate the expected value of Y in a straightforward way)

3. Calculate the variance of X.

Exercise 6.7 (Expectation and Variance 2)**.** 1. Find the expectation and variance

Given the following PDF:

$$f(x) = \begin{cases} \frac{3}{10}(3x - x^2) & 0 \leq x \leq 2 \\ 0 & otherwise \end{cases}$$

Exercise 6.8 (Expectation and Variance 3)**.** 1. Find the mean and standard deviation of random variable X. The PDF of this X is as follows:

$$f(x) = \begin{cases} \frac{1}{4}x & 0 \leq x \leq 2 \\ \frac{1}{4}(4 - x) & 2 \leq x \leq 4 \\ 0 & otherwise \end{cases}$$

2. Next, calculate $P(X < \mu - \sigma)$ Remember, μ is the mean and σ is the standard deviation

6.11 Special Distributions

Two *discrete* distributions used often are:

Definition 6.17 (Binomial Distribution)**.** Y is distributed binomial if it represents the number of "successes" observed in n independent, identical "trials," where the probability of success in any trial is p and the probability of failure is $q = 1 - p$.

For any particular sequence of y successes and $n - y$ failures, the probability of obtaining that sequence is $p^y q^{n-y}$ (by the multiplicative law and independence). However, there are $\binom{n}{y} = \frac{n!}{(n-y)!y!}$ ways of obtaining a sequence with y successes and $n - y$ failures. So the binomial distribution is given by

$$p(y) = \binom{n}{y} p^y q^{n-y}, \quad y = 0, 1, 2, \ldots, n$$

with mean $\mu = E(Y) = np$ and variance $\sigma^2 = \text{Var}(Y) = npq$.

Example 6.13. Republicans vote for Democrat-sponsored bills 2% of the time. What is the probability that out of 10 Republicans questioned, half voted for a particular Democrat-sponsored bill? What is the mean number of Republicans voting for Democrat-sponsored bills? The variance? 1. $P(Y = 5) =$ 1. $E(Y) =$ 1. $\text{Var}(Y) = 6$

Definition 6.18 (Poisson Distribution)**.** A random variable Y has a Poisson distribution if

$$P(Y = y) = \frac{\lambda^y}{y!} e^{-\lambda}, \quad y = 0, 1, 2, \ldots, \quad \lambda > 0$$

The Poisson has the unusual feature that its expectation equals its variance: $E(Y) = \text{Var}(Y) = \lambda$. The Poisson distribution is often used to model rare event counts: counts of the number of events that occur during some unit of time. λ is often called the "arrival rate."

Example 6.14. Border disputes occur between two countries through a Poisson Distribution, at a rate of 2 per month. What is the probability of 0, 2, and less than 5 disputes occurring in a month?

Two *continuous* distributions used often are:

Definition 6.19 (Uniform Distribution)**.** A random variable Y has a continuous uniform distribution on the interval (α, β) if its density is given by

$$f(y) = \frac{1}{\beta - \alpha}, \quad \alpha \le y \le \beta$$

The mean and variance of Y are $E(Y) = \frac{\alpha+\beta}{2}$ and $\text{Var}(Y) = \frac{(\beta-\alpha)^2}{12}$.

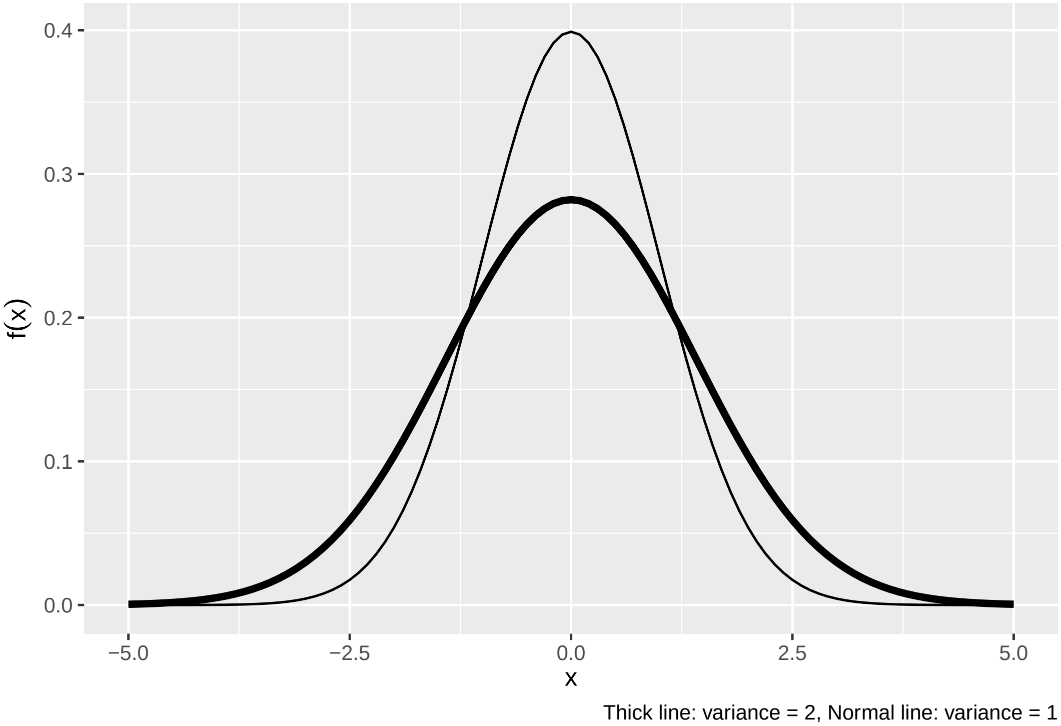


Figure 6.1: Normal Distribution Density

Example 6.15. For Y uniformly distributed over $(1, 3)$, what are the following probabilities?

1. $P(Y = 2)$
2. Its density evaluated at 2, or $f(2)$
3. $P(Y \leq 2)$
4. $P(Y > 2)$

Definition 6.20 (Normal Distribution)**.** A random variable Y is normally distributed with mean $E(Y) = \mu$ and variance $\text{Var}(Y) = \sigma^2$ if its density is

$$f(y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-\mu)^2}{2\sigma^2}}$$

See Figure 6.1 are various Normal Distributions with the same $\mu = 1$ and two versions of the variance.

6.12 Summarizing Observed Events (Data)

So far, we've talked about distributions in a theoretical sense, looking at different properties of random variables. We don't observe random variables; we observe realizations of the random variable. These realizations of events are roughly equivalent to what we mean by "data".

Sample mean: This is the most common measure of central tendency, calculated by summing across the observations and dividing by the number of observations.

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

The sample mean is an *estimate* of the expected value of a distribution.

Example:

X	6	3	7	5	5	5	6	4	7	2
Y	1	2	1	2	2	1	2	0	2	0

1. $\bar{x} =$ $\bar{y} =$
2. median(x) = median(y) =
3. $m_x =$ $m_y =$

Dispersion: We also typically want to know how spread out the data are relative to the center of the observed distribution. There are several ways to measure dispersion.

Sample variance: The sample variance is the sum of the squared deviations from the sample mean, divided by the number of observations minus 1.

$$\hat{\text{Var}}(X) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

Again, this is an *estimate* of the variance of a random variable; we divide by $n-1$ instead of n in order to get an unbiased estimate.

Standard deviation: The sample standard deviation is the square root of the sample variance.

$$\hat{SD}(X) = \sqrt{\hat{\text{Var}}(X)} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Example: Using table above, calculate:

1. $\text{Var}(X) =$ $\text{Var}(Y) =$
2. $\text{SD}(X) =$ $\text{SD}(Y) =$

Covariance and Correlation: Both of these quantities measure the degree to which two variables vary together, and are estimates of the covariance and correlation of two random variables as defined above.

1. **Sample covariance**: $\hat{\text{Cov}}(X, Y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$
2. **Sample correlation**: $\hat{\text{Corr}} = \frac{\hat{\text{Cov}}(X,Y)}{\sqrt{\hat{\text{Var}}(X)\hat{\text{Var}}(Y)}}$

Example 6.16. Example: Using the above table, calculate the sample versions of:

1. $\text{Cov}(X, Y)$
2. $\text{Corr}(X, Y)$

6.13 Asymptotic Theory

In theoretical and applied research, asymptotic arguments are often made. In this section we briefly introduce some of this material.

What are asymptotics? In probability theory, asymptotic analysis is the study of limiting behavior. By limiting behavior, we mean the behavior of some random process as the number of observations gets larger and larger.

Why is this important? We rarely know the true process governing the events we see in the social world. It is helpful to understand how such unknown processes theoretically must behave and asymptotic theory helps us do this.

6.13.1 CLT and LLN

We are now finally ready to revisit, with a bit more precise terms, the two pillars of statistical theory we motivated Section 3 with.

Theorem 6.1 (Central Limit Theorem (i.i.d. case))**.** *Let $\{X_n\} = \{X_1, X_2, \ldots\}$ be a sequence of i.i.d. random variables with finite mean (μ) and variance (σ^2). Then, the sample mean $\bar{X}_n = \frac{X_1+X_2+\cdots+X_n}{n}$ increasingly converges into a Normal distribution.*

$$\frac{\bar{X}_n - \mu}{\sigma/\sqrt{n}} \xrightarrow{d} Normal(0, 1),$$

Another way to write this as a probability statement is that for all real numbers a,

$$P\left(\frac{\bar{X}_n - \mu}{\sigma/\sqrt{n}} \leq a\right) \to \Phi(a)$$

as $n \to \infty$, where

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}\, dx$$

is the CDF of a Normal distribution with mean 0 and variance 1.

This result means that, as n grows, the distribution of the sample mean $\bar{X}_n = \frac{1}{n}(X_1 + X_2 + \cdots + X_n)$ is approximately normal with mean μ and standard deviation $\frac{\sigma}{\sqrt{n}}$, i.e.,

$$\bar{X}_n \approx \mathcal{N}\left(\mu, \frac{\sigma^2}{n}\right).$$

The standard deviation of $\bar{X}_n$ (which is roughly a measure of the precision of $\bar{X}_n$ as an estimator of μ) decreases at the rate $1/\sqrt{n}$, so, for example, to increase its precision by 10 (i.e., to get one more digit right), one needs to collect $10^2 = 100$ times more units of data.

Intuitively, this result also justifies that whenever a lot of small, independent processes somehow combine together to form the realized observations, practitioners often feel comfortable assuming Normality.

Theorem 6.2 (Law of Large Numbers (LLN))**.** *For any draw of independent random variables with the same mean μ, the sample average after n draws, $\bar{X}_n = \frac{1}{n}(X_1 + X_2 + \ldots + X_n)$, converges in probability to the expected value of X, μ as $n \to \infty$:*

$$\lim_{n\to\infty} P(|\bar{X}_n - \mu| > \varepsilon) = 0$$

A shorthand of which is $\bar{X}_n \xrightarrow{p} \mu$, where the arrow is read as "converges in probability to".

as $n \to \infty$. In other words, $P(\lim_{n\to\infty} \bar{X}_n = \mu) = 1$. This is an important motivation for the widespread use of the sample mean, as well as the intuition link between averages and expected values.

More precisely this version of the LLN is called the *weak* law of large numbers because it leaves open the possibility that $|\bar{X}_n - \mu| > \varepsilon$ occurs many times. The *strong* law of large numbers states that, undero a few more conditions, the probability that the limit of the sample average is the true mean is 1 (and other possibilities occur with probability 0), but the difference is rarely consequential in practice.

The Strong Law of Large Numbers holds so long as the expected value exists; no other assumptions are needed. However, the rate of convergence will differ greatly depending on the distribution underlying the observed data. When extreme observations occur often (i.e. kurtosis is large), the rate of convergence is much slower. Cf. The distribution of financial returns.

6.13.2 Big $\mathcal{O}$ Notation

Some of you may encounter "big-OH"-notation. If f, g are two functions, we say that $f = \mathcal{O}(g)$ if there exists some constant, c, such that $f(n) \leq c \times g(n)$ for large enough n. This notation is useful for simplifying complex problems in game theory, computer science, and statistics.

Example.

What is $\mathcal{O}(5\exp(0.5n) + n^2 + n/2)$? Answer: $\exp(n)$. Why? Because, for large n,

$$\frac{5\exp(0.5n) + n^2 + n/2}{\exp(n)} \leq \frac{c\exp(n)}{\exp(n)} = c.$$

whenever $n > 4$ and where $c = 1$.

Answers to Examples and Exercises

Answer to Example 6.1:

1. $5 \times 5 \times 5 = 125$
2. $5 \times 4 \times 3 = 60$
3. $\binom{5}{3} = \frac{5!}{(5-3)!3!} = \frac{5\times4}{2\times1} = 10$

Answer to Exercise 6.1:

1. $\binom{52}{4} = \frac{52!}{(52-4)!4!} = 270725$

Answer to Example 6.2:

1. {1, 2, 3, 4, 5, 6}
2. {5, 6}
3. {1, 2, 7, 8, 9, 10}
4. {3, 4}

Answer to Exercise 6.2:

Sample Space: {2, 3, 4, 5, 6, 7, 8}

1. {3, 4, 5, 6, 7}
2. {4, 5, 6}

Answer to Example 6.3:

1. $1, 2, 3, 4, 5, 6$
2. $\frac{1}{6}$
3. 0
4. $\frac{1}{2}$
5. $\frac{4}{6} = \frac{2}{3}$
6. 1
7. $A \cup B = \{1, 2, 3, 4, 6\}$, $A \cap B = \{2\}$, $\frac{5}{6}$

Answer to Exercise 6.3:

1. $P(X = 5) = \frac{4}{16}$, $P(X = 3) = \frac{2}{16}$, $P(X = 6) = \frac{3}{16}$
2. What is $P(X = 5 \cup X = 3)^C = \frac{10}{16}$?

Answer to Example 6.4:

1. $\frac{n_{ab}+n_{ab^c}}{N}$
2. $\frac{n_{ab}+n_{a^cb}}{N}$
3. $\frac{n_{ab}}{N}$
4. $\frac{\frac{n_{ab}}{N}}{\frac{n_{ab}+n_{a^cb}}{N}} = \frac{n_{ab}}{n_{ab}+n_{a^cb}}$
5. $\frac{\frac{n_{ab}}{N}}{\frac{n_{ab}+n_{ab^c}}{N}} = \frac{n_{ab}}{n_{ab}+n_{ab^c}}$

Answer to Example 6.5:

$P(1|Odd) = \frac{P(1 \cap Odd)}{P(Odd)} = \frac{\frac{1}{6}}{\frac{1}{2}} = \frac{1}{3}$

Answer to Example 6.6:

We are given that

$$P(D) = .4, P(D^c) = .6, P(S|D) = .5, P(S|D^c) = .9$$

Using this, Bayes' Law and the Law of Total Probability, we know:

$$P(D|S) = \frac{P(D)P(S|D)}{P(D)P(S|D) + P(D^c)P(S|D^c)}$$

$$P(D|S) = \frac{.4 \times .5}{.4 \times .5 + .6 \times .9} = .27$$

Answer to Exercise 6.4:

We are given that

$$P(M) = .02, P(C|M) = .95, P(C^c|M^c) = .97$$

$$P(M|C) = \frac{P(C|M)P(M)}{P(C)}$$

$$= \frac{P(C|M)P(M)}{P(C|M)P(M) + P(C|M^c)P(M^c)}$$

$$= \frac{P(C|M)P(M)}{P(C|M)P(M) + [1 - P(C^c|M^c)]P(M^c)}$$

$$= \frac{.95 \times .02}{.95 \times .02 + .03 \times .98} = .38$$

Answer to Example 6.10:

$E(Y) = 7/2$

We would never expect the result of a rolled die to be 7/2, but that would be the average over a large number of rolls of the die.

Answer to Example 6.11

0.75

Answer to Example 6.12:

$E(X) = 0 \times \frac{1}{8} + 1 \times \frac{3}{8} + 2 \times \frac{3}{8} + 3 \times \frac{1}{8} = \frac{3}{2}$

Since there is a 1 to 1 mapping from X to $X^2 : E(X^2) = 0\times\frac{1}{8}+1\times\frac{3}{8}+4\times\frac{3}{8}+9\times\frac{1}{8} = \frac{24}{8} = 3$

$$\begin{aligned}
\mathrm{Var}(x) &= E(X^2) - E(x)^2 \\
&= 3 - (\frac{3}{2})^2 \\
&= \frac{3}{4}
\end{aligned}$$

Answer to Exercise 6.6:

1. $E(X) = -2(\frac{1}{5}) + -1(\frac{1}{6}) + 0(\frac{1}{5}) + 1(\frac{1}{15}) + 2(\frac{11}{30}) = \frac{7}{30}$
2. $E(Y) = 0(\frac{1}{5}) + 1(\frac{7}{30}) + 4(\frac{17}{30}) = \frac{5}{2}$
3.

$$\begin{aligned}
\mathrm{Var}(X) &= E[X^2] - E[X]^2 \\
&= E(Y) - E(X)^2 \\
&= \frac{5}{2} - \frac{7}{30}^2 \approx 2.45
\end{aligned}$$

Answer to Exercise 6.7:

1. expectation $= \frac{6}{5}$, variance $= \frac{6}{25}$

Answer to Exercise 6.8:

1. mean $= 2$, standard deviation $= \sqrt{(\frac{2}{3})}$
2. $\frac{1}{8}(2 - \sqrt{(\frac{2}{3})})^2$

Part II

Programming

Chapter 7

Orientation and Reading in Data[1]

Up till now, you should have

- Completed the R Visualization and Programming primers at `https://rstudio.cloud/primers/`
- Make an account at RStudio Cloud and join the Math Prefresher 2018 Space
- Set up wi-fi to work on your laptops: `https://getonline.harvard.edu/` (Access Harvard Secure with your HarvardKey. Try to get a HarvardKey as soon as possible.)

Where are we? Where are we headed?

Today we'll cover:

- What's what in RStudio
- Reading in Data
- Comment on coding style on the way

Check your understanding

- What is the difference between a file and a folder?
- In the RStudio windows, what is the difference between the "Source" Pane and the "Console"? What is a "code chunk"?
- How do you read a R help page? What is the `Usage` section, the `Values` section, and the `Examples` section?
- What use is the "Environment" Pane?
- How would you read in a spreadsheet in R?
- How would you figure out what variables are in the data? size of the data?
- How would you read in a `csv` file, a `dta` file, a `sav` file?

[1] Module originally written by Shiro Kuriwaki

7.1 Motivation: Data and You

The modal social science project starts by importing existing datasets. Datasets come in all shapes and sizes. As you search for new data you may encounter dozens of file extensions – csv, xlsx, dta, sav, por, Rdata, Rds, txt, xml, json, shp … the list continues. Although these files can often be cumbersome, its a good to be able to find a way to encounter any file that your research may call for.

Reviewing data import will allow us to get on the same page on how computer systems work.

7.2 Orienting

1. We will be using a cloud version of RStudio at `https://rstudio.cloud`. You should join the Math Prefresher Space 2018 from the link that was emailed to you. Each day, click on the project with the day's date on it.

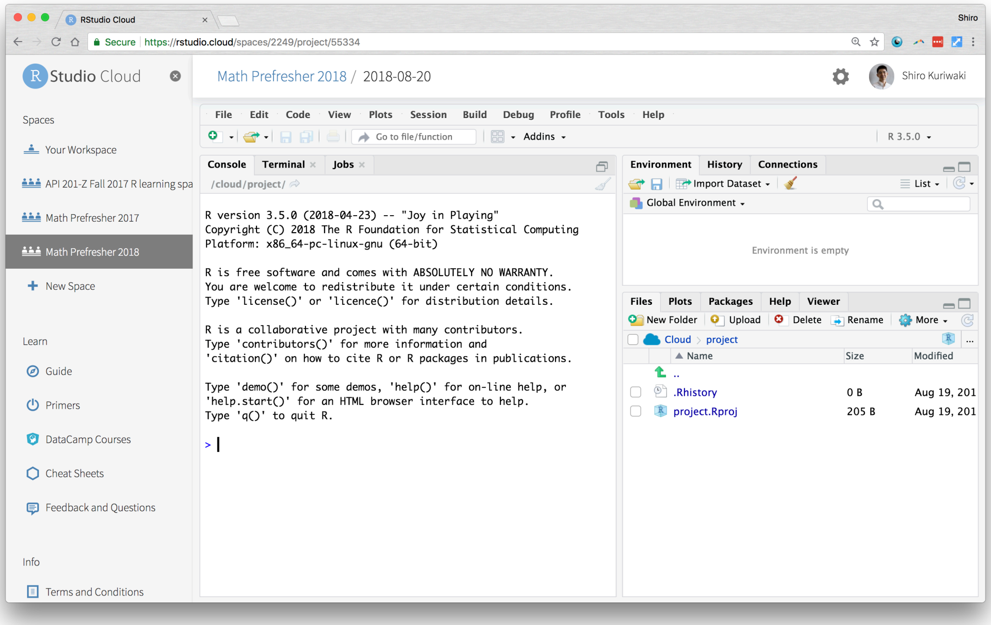
   Although most of you will probably doing your work on RStudio local rather than cloud, we are trying to use cloud because it makes it easier to standardize people's settings.

2. RStudio (either cloud or desktop) is a **GUI** and an IDE for the programming language R. A Graphical User Interface allows users to interface with the software (in this case R) using graphical aids like buttons and tabs. Often we don't think of GUIs because to most computer users, everything is a GUI (like Microsoft Word or your "Control Panel"), but it's always there! A Integrated Development Environment just says that the software to interface with R comes with useful useful bells and whistles to give you shortcuts.

   The **Console** is kind of a the core window through which you see your GUI actually operating through R. It's not graphical so might not be as intuitive. But all your results, commands, errors, warnings.. you see them in here. A console tells you what's going on now.

3. via the GUI, you the analyst needs to sends instructions, or **commands**, to the R application. The verb for this is "run" or "execute" the command. Computer programs ask users to provide instructions in very specific formats. While a English-speaking human can understand a sentence with a few typos in it by filling in the blanks, the same typo or misplaced character would halt a computer program. Each program has its own requirements for how commands should be typed; after all, each of these is its own language. We refer to the way a program needs its commands to be formatted as its **syntax**.

4. Theoretically, one could do all their work by typing in commands into the Console. But that would be a lot of work, because you'd have to give instructions each time you start your data analysis. Moreover, you'll have no record of what you did. That's why you need a **script**. This is a type of **code**. It can be referred to as a **source** because that is the source of your commands. Source is also used as a verb; "source



```
R version 3.5.0 (2018-04-23) -- "Joy in Playing"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 
```

Name	Size	Modified
..		
.Rhistory	0 B	Aug 19, 201
project.Rproj	205 B	Aug 19, 201

Figure 7.1: A Typical RStudio Window at Startup

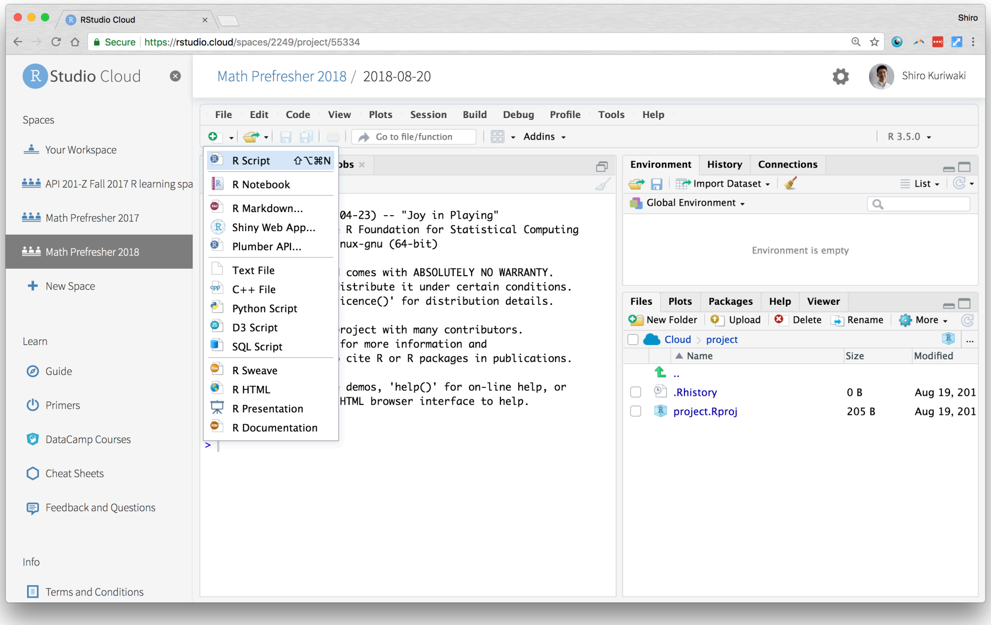
the script" just means execute it. RStudio doesn't start out with a script, so you can make one from "File > New" or the New file icon.

4. You can also open scripts that are in folders in your computer. A script is a type of File. Find your Files in the bottom-right "Files" pane.

   To load a dataset, you need to specify where that file is. Computer files (data, documents, programs) are organized hiearchically, like a branching tree. Folders can contain files, and also other folders. The GUI toolbar makes this lineaer and hiearchical relationship apparent. When we turn to locate the file in our commands, we need another set of syntax. Importantly, denote the hierarchy of a folder by the `/` (slash) symbol. `data/input/2018-08` indicates the `2018-08` folder, which is included in the `input` folder, which is in turn included in the `data` folder.

   Files (but not folders) have "file extensions" which you are probably familiar with already: `.docx`, `.pdf`, and `.pdf` already. The file extensions you will see in a stats or quantitative social science class are:

   - `.pdf`: PDF, a convenient format to view documents in, regardless of Mac/Windows.
   - `.csv`: A comma separated values file
   - `.xlsx`: Microsoft Excel file
   - `.dta`: Stata data



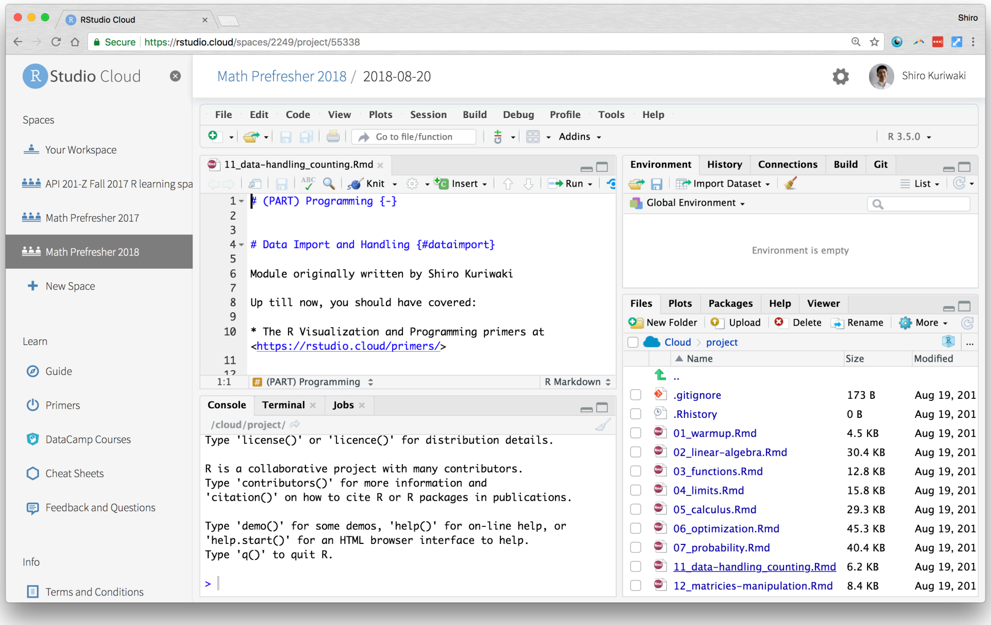
```
04-23) -- "Joy in Playing"
 R Foundation for Statistical Computing
nux-gnu (64-bit)

 comes with ABSOLUTELY NO WARRANTY.
istribute it under certain conditions.
icence()' for distribution details.

roject with many contributors.
for more information and
 cite R or R packages in publications.

 demos, 'help()' for on-line help, or
HTML browser interface to help.

>
```

Figure 7.2: Opening New Script (as opposed to the Console)

- `.sav`: SPSS data
- `.R`: R code (script)
- `.Rmd`: Rmarkdown code (text + code)
- `.do`: Stata code (script)

5. In R, there are two main types of scripts. A classic `.R` file and a `.Rmd` file (for Rmarkdown). A .R file is just lines and lines of R code that is meant to be inserted right into the Console. A .Rmd tries to weave code and English together, to make it easier for users to create reports that interact with data and intersperse R code with explanation. For example, we built this book in Rmds.

   The Rmarkdown facilitates is the use of **code chunks**, which are used here. These start and end with three back-ticks. In the beginning, we can add options in curly braces (`{}`). Specifying `r` in the beginning tells to render it as R code. Options like `echo = TRUE` switch between showing the code that was executed or not; `eval = TRUE` switch between evaluating the code. More about Rmarkdown in Section 13. For example, this code chunk would evaluate `1 + 1` and show its output when compiled, but not display the code that was executed.

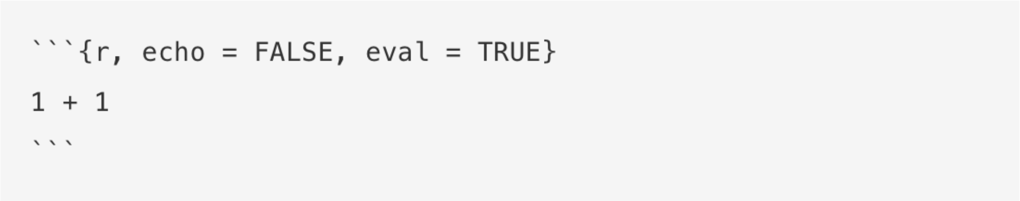


```
# (PART) Programming {-}


# Data Import and Handling {#dataimport}

Module originally written by Shiro Kuriwaki

Up till now, you should have covered:

* The R Visualization and Programming primers at
<https://rstudio.cloud/primers/>

```

```
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Name	Size	Modified
..		
.gitignore	173 B	Aug 19, 201
.Rhistory	0 B	Aug 19, 201
01_warmup.Rmd	4.5 KB	Aug 19, 201
02_linear-algebra.Rmd	30.4 KB	Aug 19, 201
03_functions.Rmd	12.8 KB	Aug 19, 201
04_limits.Rmd	15.8 KB	Aug 19, 201
05_calculus.Rmd	29.3 KB	Aug 19, 201
06_optimization.Rmd	45.3 KB	Aug 19, 201
07_probability.Rmd	40.4 KB	Aug 19, 201
11_data-handling_counting.Rmd	6.2 KB	Aug 19, 201
12_matricies-manipulation.Rmd	8.4 KB	Aug 19, 201

Figure 7.3: Opening an Existing Script from Files



~~~
```{r, echo = FALSE, eval = TRUE}
1 + 1
```
~~~

Figure 7.4: A code chunk in Rmarkdown (before rendering)

7.3 The Computer and You: Giving Instructions

We'll do the Peanut Butter and Jelly Exercise in class as an introduction to programming for those who are new.[2]

Assignment: Take 5 minutes to write down on a piece of paper, how to make a peanut butter and jelly sandwich. Be as concise and unambiguous as possible so that a robot (who doesn't know what a PBJ is) would understand. You can assume that there will be loaf of sliced bread, a jar of jelly, a jar of peanut butter, and a knife.

7.4 Base-R vs. tidyverse

One last thing before we jump into data. Many things in R and other open source packages have competing standards. A lecture on a technique inevitably biases one standard over another. Right now among R users in this area, there are two families of functions: base-R and tidyverse. R instructors thus face a dilemma about which to teach primarily.[3]

In this prefresher, we try our best to choose the one that is most useful to the modal task of social science researchers, and make use of the tidyverse functions in most applications. but feel free to suggest changes to us or to the booklet.

Although you do not need to choose one over the other, for beginners it is confusing what is a tidyverse function and what is not. Many of the tidyverse *packages* are covered in this 2017 graphic below, and the cheat-sheets that other programmers have written: `https://www.rstudio.com/resources/cheatsheets/`

The following side-by-side comparison of commands for a particular function compares some tidyverse and non-tidyverse functions (which we refer to loosely as base-R). This list is not meant to be comprehensive and more to give you a quick rule of thumb.

Dataframe subsetting

In order to …	in tidyverse:	in base-R:
Count each category	`count(df, var)`	`table(df$var)`
Filter rows by condition	`filter(df, var == "Female")`	`df[df$var == "Female", ]` or `subset(df, var == "Female")`
Extract columns	`select(df, var1, var2)`	`df[, c("var1", "var2")]`
Extract a single column as a vector	`pull(df, var)`	`df[["var"]]` or `df[, "var"]`
Combine rows	`bind_rows()`	`rbind()`

[2] This Exercise is taken from Harvard's Introductory Undergraduate Class, CS50 (`https://www.youtube.com/watch?v=kcbT3hrEi9s`), and many other writeups.

[3] See for example this community discussion: `https://community.rstudio.com/t/base-r-and-the-tidyverse/2965/17`

In order to …	in tidyverse:	in base-R:
Combine columns	`bind_cols()`	`cbind()`
Create a dataframe	`tibble(x = vec1, y = vec2)`	`data.frame(x = vec1, y = vec2)`
Turn a dataframe into a tidyverse dataframe	`tbl_df(df)`	

Remember that tidyverse applies to *dataframes* only, not vectors. For subsetting vectors, use the base-R functions with the square brackets.

Read data

Some non-tidyverse functions are not quite "base-R" but have similar relationships to tidyverse. For these, we recommend using the *tidyverse* functions as a general rule due to their common format, simplicity, and scalability.

In order to …	in tidyverse:	in base-R:
Read a Excel file	`read_excel()`	`read.xlsx()`
Read a csv	`read_csv()`	`read.csv()`
Read a Stata file	`read_dta()`	`read.dta()`
Substitute strings	`str_replace()`	`gsub()`
Return matching strings	`str_subset()`	`grep(., value = TRUE)`
Merge `data1` and `data2` on variables `x1` and `x2`	`left_join(data1, data2, by = c("x1", "x2"))`	`merge(data1, data2, by.x = "x1", by.y = "x2", all.x = TRUE)`

Visualization

Plotting by ggplot2 (from your tutorials) is also a tidyverse family.

In order to …	in tidyverse:	in base-R:
Make a scatter plot	`ggplot(data, aes(x, y)) + geom_point()`	`plot(data$x, data$y)`
Make a line plot	`ggplot(data, aes(x, y)) + geom_line()`	`plot(data$x, data$y, type = "l")`
Make a histogram	`ggplot(data, aes(x, y)) + geom_histogram()`	`hist(data$x, data$y)`
Make a barplot	See Section 9	See Section 9

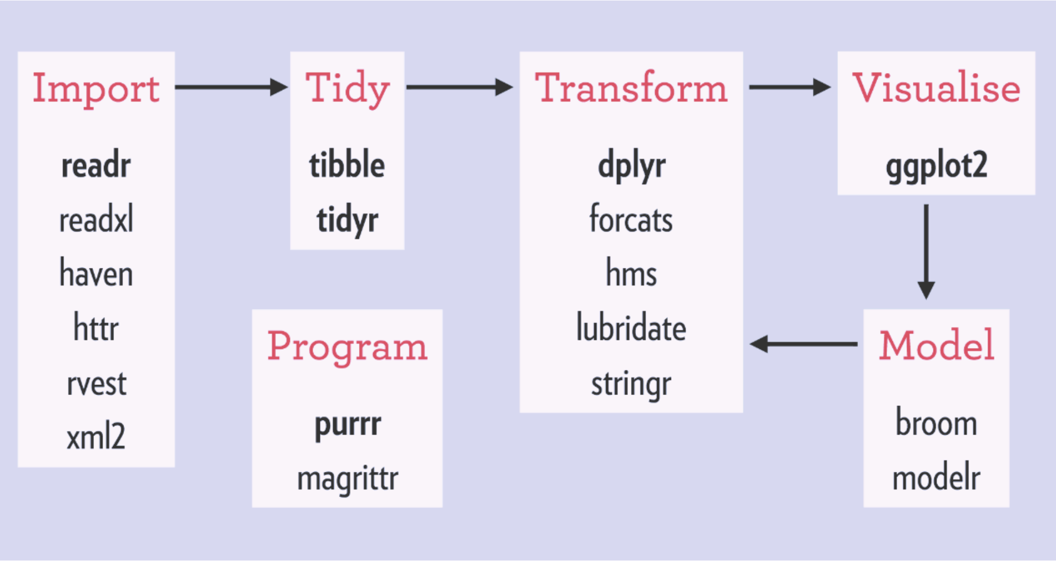


Figure 7.5: Names of Packages in the tidyverse Family

7.5 A is for Athens

For our first dataset, let's try reading in a dataset on the Ancient Greek world. Political Theorists and Political Historians study the domestic systems, international wars, cultures and writing of this era to understand the first instance of democracy, the rise and overturning of tyranny, and the legacies of political institutions.

This POLIS dataset was generously provided by Professor Josiah Ober of Stanford University. This dataset includes information on city states in the Ancient Greek world, parts of it collected by careful work by historians and archaeologists. It is part of his recent books on Greece (Ober 2015), "The Rise and Fall of Classical Greece"[4] and Institutions in Ancient Athens (Ober 2010) , "Democracy and Knowledge: Innovation and Learning in Classical Athens."[5]

7.5.1 Locating the Data

What files do we have in the `input` folder?

```
## input/Nunn_Wantchekon_AER_2011.dta
## input/Nunn_Wantchekon_sample.dta
## input/acs2015_1percent.csv
## input/complete_greek_data 5 DISTRIBUTION 2018.xlsx
## input/gapminder_wide.Rds
## input/gapminder_wide.tab
```

[4] Ober, Josiah (2015). *The Rise and Fall of Classical Greece.* Princeton University Press.

[5] Ober, Josiah (2010). *Democracy and Knowledge: Innovation and Learning in Classical Athens.* Princeton University Press.

```
## input/german_credit.sav
## input/justices_court-median.csv
## input/sample_mid.csv
## input/sample_polity.csv
## input/upshot-siena-polls.csv
## input/usc2010_001percent.csv
## input/usc2010_1percent.Rds
```

A typical file format is Microsoft Excel. Although this is not usually the best format for R because of its highly formatted structure as opposed to plain text (more on this in Section **??**(sec:wysiwyg)), recent packages have made this fairly easy.

For the first time using an outside package, you first need to install it.

```
install.packages("readxl")
```

After that, you don't need to install it again. But you **do** need to load it each time.

```
library(readxl)
```

The package `readxl` has a website: `https://readxl.tidyverse.org/`. Other packages are not as user-friendly, but they have a help page with a table of contents of all their functions.

```
help(package = readxl)
```

7.5.2 Reading in Data

From the help page, we see that `read_excel()` is the function that we want to use. Look at the help page. How do you read a help page?

Let's try it.

```
ober <- read_excel("input/complete_greek_data 5 DISTRIBUTION 2018.xlsx")
```

Review: what does the `/` mean? Why do we need the `input` term first? Does the argument need to be in quotes?

7.5.3 Inspecting

For almost any dataset, you usually want to do a couple of standard checks first to understand what you loaded.

```
ober
```

```
## # A tibble: 1,035 x 35
##    polis_number Name  Polisity Hellenicity  Fame `In/out`  Size
##           <dbl> <chr>    <dbl>       <dbl> <dbl>    <dbl> <dbl>
##  1            1 Alal~        1           1  1.12        1     3
##  2            2 Empo~        1           2  2.12        2     2
##  3            3 Mass~        1           1  4           2     2
##  4            4 Rhode        2           1  0.87        3     0
##  5            5 Abak~        2           2  1           2     0
##  6            6 Adra~        2           1  1           3     0
##  7            7 Agyr~        2           1  1.25        2     0
##  8            8 Aitna        1           1  3.25        1     4
##  9            9 Akra~        1           1  6.37        4     5
## 10           10 Akrai        3           1  1.25        2     0
## # ... with 1,025 more rows, and 28 more variables: `Silver-1st` <dbl>,
## #   `Bronze 1st` <dbl>, Grid <dbl>, Colonies <dbl>, Victors <dbl>,
## #   Proxenoi <dbl>, Walls <dbl>, `Delian L` <dbl>, Koinon <dbl>,
## #   Regime <dbl>, `Region #` <dbl>, `Region name` <chr>, Latitude <dbl>,
## #   Longitude <dbl>, Source <chr>, `Elevation m` <dbl>, `Pleiades
## #   Control` <chr>, `Hanson Appearance` <dbl>, `Hanson Name` <chr>,
## #   `Hanson #` <dbl>, `Occupied From (Hanson)` <dbl>, `Size (ha.)` <chr>,
## #   `Hanson Longitude` <dbl>, `Hanson Latitude` <dbl>, `Lat O-H` <dbl>,
## #   `Long O-H` <dbl>, `Absolute Lat+Long O-H` <dbl>, X__1 <chr>
```

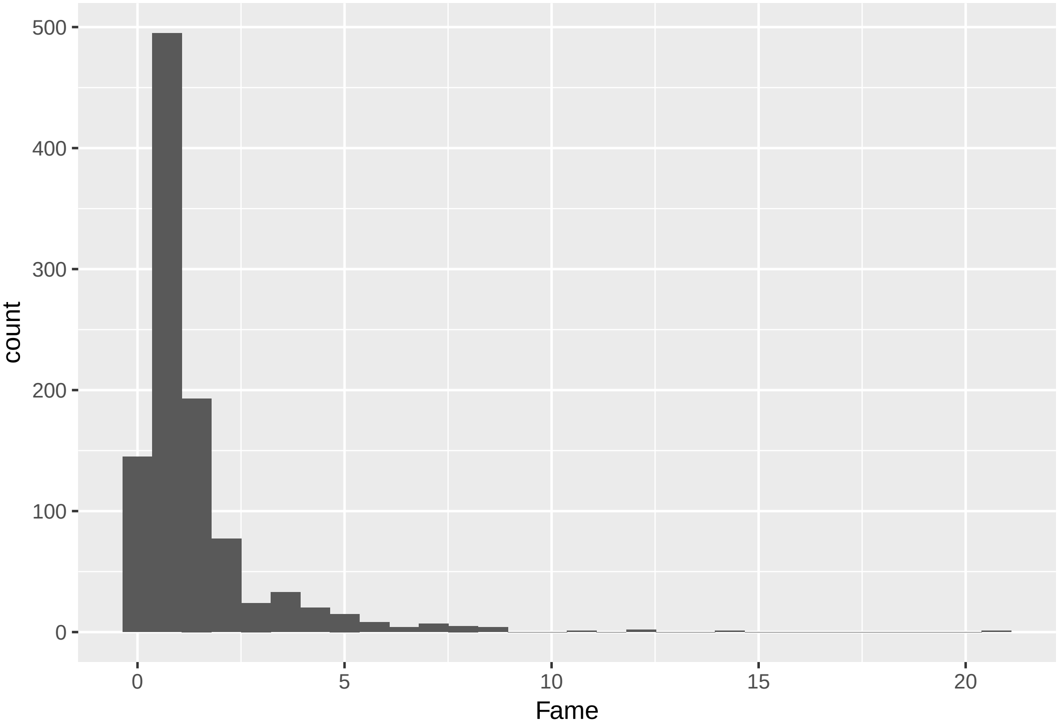
```r
dim(ober)
```

```
## [1] 1035   35
```

From your tutorials, you also know how to do graphics! Graphics are useful for grasping your data, but we will cover them more deeply in Chapter 9.

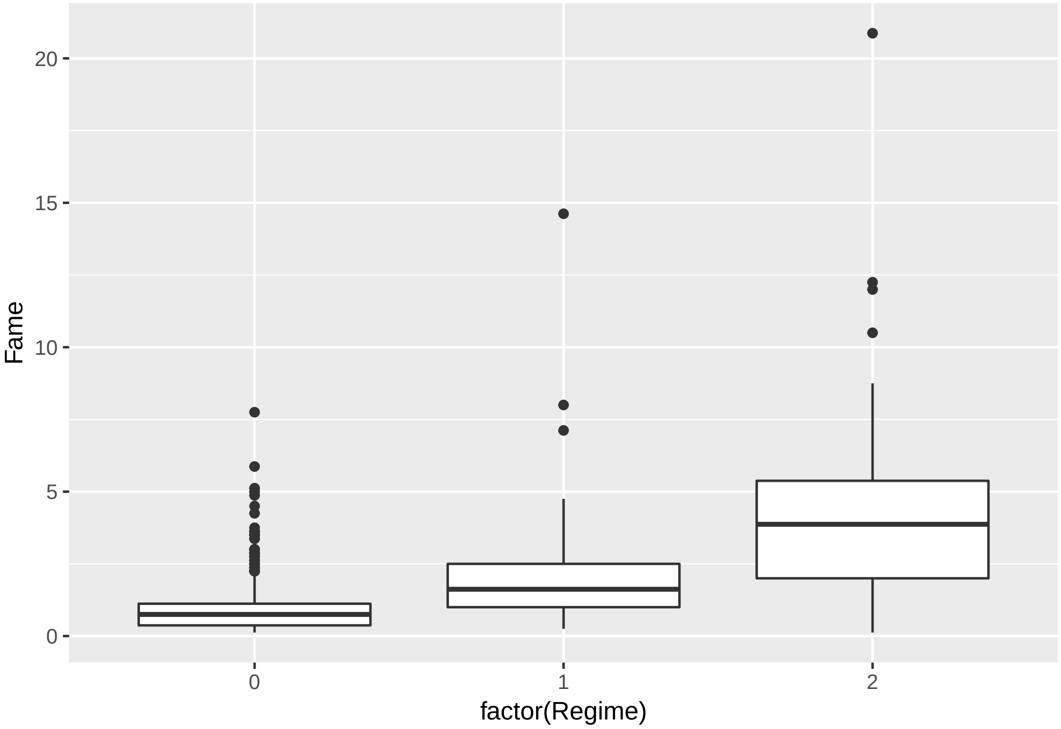
```r
ggplot(ober, aes(x = Fame)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



What about the distribution of fame by regime?

```
ggplot(ober, aes(y = Fame, x = factor(Regime), group = factor(Regime))) +
  geom_boxplot()
```



What do the 1's, 2's, and 3's stand for?

7.5.4 Finding observations

These `tidyverse` commands from the `dplyr` package are newer and not built-in, but they are one of the increasingly more popular ways to wrangle data.

- 80 percent of your data wrangling needs might be doable with these basic `dplyr` functions: `select`, `mutate`, `group_by`, `summarize`, and `arrange`.
- These verbs roughly correspond to the same commands in SQL, another important language in data science.
- The `%>%` symbol is a pipe. It takes the thing on the left side and pipes it down to the function on the right side. We could have done `count(cen10, race)` as `cen10 %>% count(race)`. That means take `cen10` and pass it on to the function `count`, which will count observations by race and return a collapsed dataset with the categories in its own variable and their respective counts in `n`.

7.5.5 Extra: A sneak peak at Ober's data

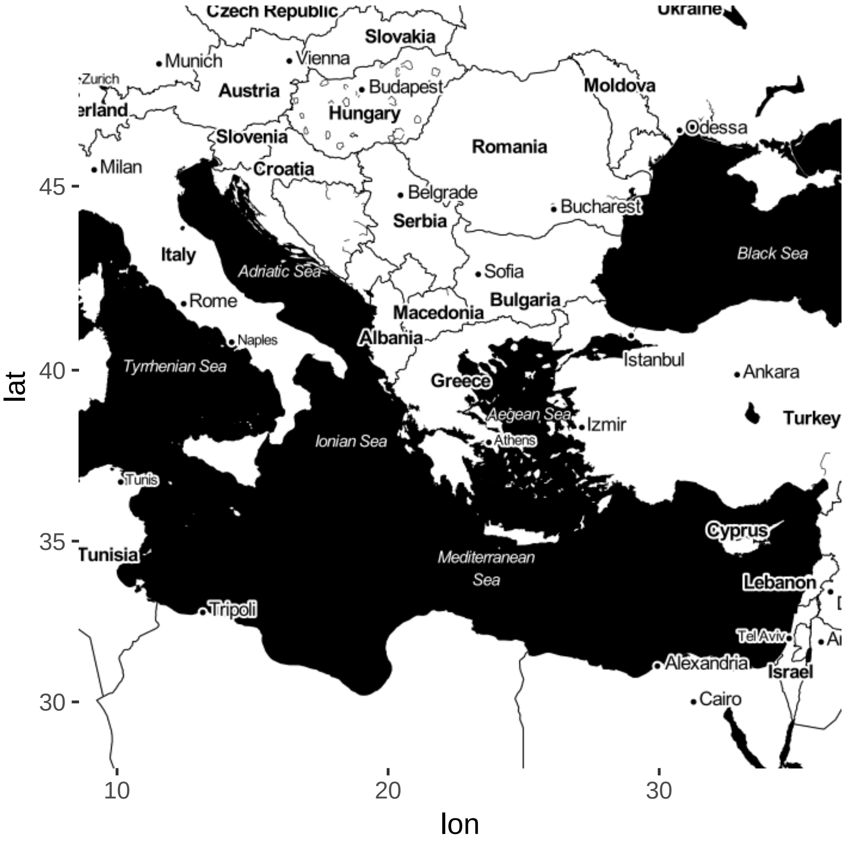
Although this is a bit beyond our current stage, it's hard to resist the temptation to see what you can do with data like this. For example, you can map it.

Using the `ggmap` package

```r
library(ggmap)
```

First get a map of the Greek world.

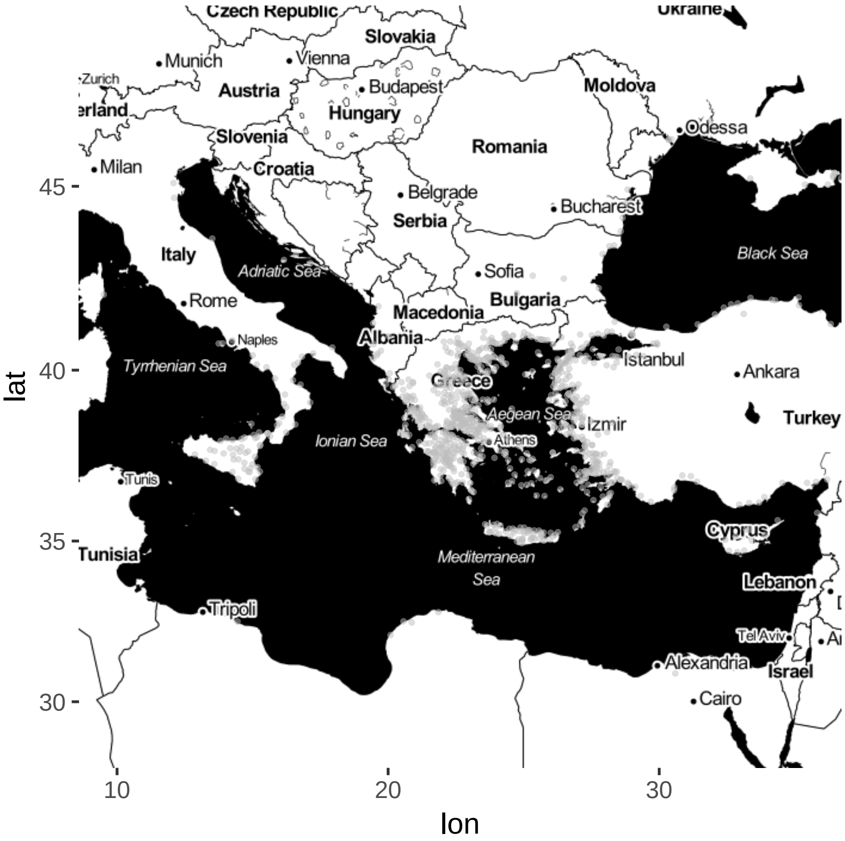
```r
greece <- get_map(location = c(lat = 39.543287, lon = 22.6382849),
                  zoom = 5,
                  source = "stamen",
                  maptype = "toner")
ggmap(greece)
```



I chose the specifications for arguments `zoom` and `maptype` by looking at the webpage and Googling some examples/

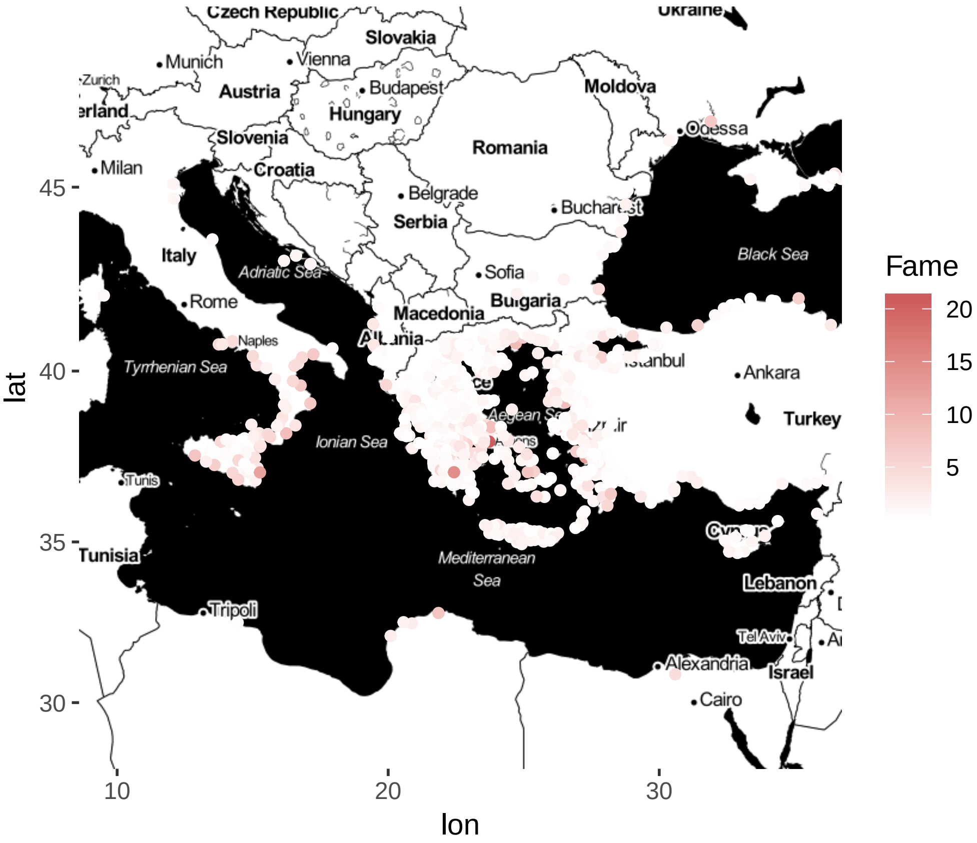
Ober's data has the latitude and longitude of each polis. Because the map of Greece has the same coordinates, we can add the polei on the same map.

```
gg_ober <- ggmap(greece) +
  geom_point(data = ober,
             aes(y = Latitude, x = Longitude),
             size = 0.5,
             alpha = 0.5,
             color = "gray")
gg_ober
```



We can also color the points by another variable, like Fame.

```
gg_ober_fame <- ggmap(greece) +
  geom_point(data = ober,
             aes(y = Latitude, x = Longitude, color = Fame)) +
  scale_color_gradient(low = "white", high = "indianred")
gg_ober_fame
```



Exercises

1

What is the Fame value of Delphoi?

```
# Enter here
```

2

Find the polis with the top 10 Fame values.

```
# Enter here
```

3

Make a scatterplot with the number of colonies on the x-axis and Fame on the y-axis.

```
# Enter here
```

4

Find the right function to read the following datasets into your R window.

- `input/acs2015_1percent.csv`: A one percent sample of the American Community Survey
- `input/gapminder_wide.tab`: Country-level wealth and health from Gapminder[6]
- `input/gapminder_wide.Rds`: A Rds version of the Gapminder (What is a Rds file? What's the difference?)
- `input/Nunn_Wantchekon_sample.dta`: A sample from the Afrobarometer survey (which we'll explore tomorrow). `.dta` is a Stata format.
- `input/german_credit.sav`: A hypothetical dataset on consumer credit. `.sav` is a SPSS format.

Our Recommendations: Look at the packages `haven` and `readr`

```
# Enter here, perhaps making a chunk for each file.
```

5

Read Ober's codebook and find a variable that you think is interesting. Check the distribution of that variable in your data, get a couple of statistics, and summarize it in English.

```
# Enter here
```

6

This is day 1 and we covered a lot of material. Some of you might have found this completely new; others not so. Please click through this survey before you leave so we can adjust accordingly on the next few days.

`https://harvard.az1.qualtrics.com/jfe/form/SV_8As7Y7C83iBiQzH`

[6] Formatted and taken from `https://doi.org/10.7910/DVN/GJQNEQ`

Chapter 8

Manipulating Vectors and Matrices[1]

Where are we? Where are we headed?

Up till now, you should have covered:

- R basic programming
- Data Import
- Statistical Summaries.

Today we'll cover

- Matrices & Dataframes in R
- Manipulating variables
- And other `R` tips

8.1 Basics - Matrices

Let's take a look at Matrices in the context of R

```
cen10 <- read_csv("input/usc2010_001percent.csv")
head(cen10)
```

```
## # A tibble: 6 x 13
##    year serial pernum region state countyfips city  cpuma0010 sex     age
##   <int>  <int>  <int> <chr>  <chr>      <int> <chr>     <int> <chr> <int>
## 1  2010 8.80e6      4 Middl~ New ~          0 Not ~       636 Fema~     8
## 2  2010 9.80e6      1 East ~ Ohio         103 Not ~       802 Male     24
## 3  2010 8.69e6      1 Mount~ Neva~          3 Not ~       582 Male     37
## 4  2010 6.35e6      3 East ~ Mich~          0 Not ~       476 Fema~    12
```

[1] Module originally written by Shiro Kuriwaki and Yon Soo Park

```
## 5  2010 6.15e6      2 South~ Mary~         33 Not ~       449 Fema~    18
## 6  2010 8.10e6      1 New E~ New ~          0 Not ~       586 Male     50
## # ... with 3 more variables: race <chr>, hhtype <chr>, relate <chr>
```

What is the dimension of this dataframe? What does the number of rows represent? What does the number of columns represent?

```
dim(cen10)
```

```
## [1] 30871    13
```

```
nrow(cen10)
```

```
## [1] 30871
```

```
ncol(cen10)
```

```
## [1] 13
```

What variables does this dataset hold? What kind of information does it have?

```
colnames(cen10)
```

```
##  [1] "year"       "serial"     "pernum"     "region"     "state"
##  [6] "countyfips" "city"       "cpuma0010"  "sex"        "age"
## [11] "race"       "hhtype"     "relate"
```

We can access column vectors, or vectors that contain values of variables by using the $ sign

```
head(cen10$state)
```

```
## [1] "New York"      "Ohio"          "Nevada"        "Michigan"
## [5] "Maryland"      "New Hampshire"
```

```
head(cen10$city)
```

```
## [1] "Not in identifiable city (or size group)"
## [2] "Not in identifiable city (or size group)"
## [3] "Not in identifiable city (or size group)"
## [4] "Not in identifiable city (or size group)"
## [5] "Not in identifiable city (or size group)"
## [6] "Not in identifiable city (or size group)"
```

We can look at a unique set of variable values by calling the unique function

```
unique(cen10$state)
```

```
##  [1] "New York"              "Ohio"                  "Nevada"
##  [4] "Michigan"              "Maryland"              "New Hampshire"
##  [7] "Iowa"                  "Missouri"              "New Jersey"
## [10] "California"            "Texas"                 "Pennsylvania"
## [13] "Washington"            "West Virginia"         "Idaho"
## [16] "North Carolina"        "Massachusetts"         "Connecticut"
## [19] "Arkansas"              "Indiana"               "Wisconsin"
## [22] "Maine"                 "Tennessee"             "Minnesota"
## [25] "Florida"               "Oklahoma"              "Montana"
## [28] "Georgia"              "Arizona"              "Colorado"
## [31] "Virginia"             "Illinois"             "Oregon"
## [34] "Kentucky"             "South Carolina"       "Kansas"
## [37] "Louisiana"            "Alabama"              "District of Columbia"
## [40] "Mississippi"          "Utah"                 "Delaware"
## [43] "Nebraska"             "Alaska"               "New Mexico"
## [46] "South Dakota"         "Hawaii"               "Vermont"
## [49] "Rhode Island"         "Wyoming"              "North Dakota"
```

How many different states are represented (this dataset includes DC as a state)?

```
length(unique(cen10$state))
```

```
## [1] 51
```

Matrices are rectangular structures of numbers (they have to be numbers, and they can't be characters).

A cross-tab can be considered a matrix:

```
table(cen10$race, cen10$sex)
```

```
## 
##                                    Female  Male
##   American Indian or Alaska Native    142   153
##   Black/Negro                        2070  1943
##   Chinese                             192   162
##   Japanese                             51    26
##   Other Asian or Pacific Islander     587   542
##   Other race, nec                     877   962
##   Three or more major races            37    51
##   Two major races                     443   426
##   White                             11252 10955
```

```
cross_tab <- table(cen10$race, cen10$sex)
dim(cross_tab)
```

```
## [1] 9 2
```

```
cross_tab[6, 2]
```

```
## [1] 962
```

But a subset of your data – individual values– can be considered a matrix too.

```
# First 20 rows of the entire data
# Below two lines of code do the same thing
cen10[1:20, ]
```

```
## # A tibble: 20 x 13
##     year serial pernum region state countyfips city  cpuma0010 sex     age
##    <int>  <int>  <int> <chr>  <chr>      <int> <chr>     <int> <chr> <int>
##  1  2010 8.80e6      4 Middl~ New ~          0 Not ~       636 Fema~     8
##  2  2010 9.80e6      1 East ~ Ohio         103 Not ~       802 Male     24
##  3  2010 8.69e6      1 Mount~ Neva~          3 Not ~       582 Male     37
##  4  2010 6.35e6      3 East ~ Mich~          0 Not ~       476 Fema~    12
##  5  2010 6.15e6      2 South~ Mary~         33 Not ~       449 Fema~    18
##  6  2010 8.10e6      1 New E~ New ~          0 Not ~       586 Male     50
##  7  2010 4.06e6      1 West ~ Iowa           0 Not ~       362 Fema~    51
##  8  2010 7.03e6      2 West ~ Miss~          0 Not ~       550 Fema~    41
##  9  2010 8.16e6      2 Middl~ New ~          3 Not ~       592 Male     62
## 10  2010 1.12e6      3 Pacif~ Cali~         37 Los ~        81 Male     25
## 11  2010 1.25e7      1 West ~ Texas        453 Not ~       995 Fema~    23
## 12  2010 1.07e7      2 Middl~ Penn~          3 Not ~       871 Fema~    66
## 13  2010 9.31e5      2 Pacif~ Cali~         31 Not ~        69 Fema~    57
## 14  2010 1.18e7      6 West ~ Texas          0 Not ~       968 Fema~    73
## 15  2010 1.41e6      1 Pacif~ Cali~         59 Not ~       106 Male     43
## 16  2010 1.34e7      1 Pacif~ Wash~         33 Seat~      1040 Male     29
## 17  2010 1.17e7      2 West ~ Texas        381 Not ~       956 Male      8
## 18  2010 7.21e6      1 West ~ Miss~        189 Not ~       558 Male     78
## 19  2010 1.39e7      4 South~ West~          0 Not ~      1052 Male     10
## 20  2010 4.21e6      3 Mount~ Idaho          0 Not ~       290 Fema~     9
## # ... with 3 more variables: race <chr>, hhtype <chr>, relate <chr>
```

```r
cen10 %>% slice(1:20)
```

```
## # A tibble: 20 x 13
##     year serial pernum region state countyfips city  cpuma0010 sex     age
##    <int>  <int>  <int> <chr>  <chr>       <int> <chr>      <int> <chr> <int>
##  1  2010 8.80e6      4 Middl~ New ~          0 Not ~       636 Fema~     8
##  2  2010 9.80e6      1 East ~ Ohio         103 Not ~       802 Male     24
##  3  2010 8.69e6      1 Mount~ Neva~          3 Not ~       582 Male     37
##  4  2010 6.35e6      3 East ~ Mich~          0 Not ~       476 Fema~    12
##  5  2010 6.15e6      2 South~ Mary~         33 Not ~       449 Fema~    18
##  6  2010 8.10e6      1 New E~ New ~          0 Not ~       586 Male     50
##  7  2010 4.06e6      1 West ~ Iowa           0 Not ~       362 Fema~    51
##  8  2010 7.03e6      2 West ~ Miss~          0 Not ~       550 Fema~    41
##  9  2010 8.16e6      2 Middl~ New ~          3 Not ~       592 Male     62
## 10  2010 1.12e6      3 Pacif~ Cali~         37 Los ~        81 Male     25
## 11  2010 1.25e7      1 West ~ Texas        453 Not ~       995 Fema~    23
## 12  2010 1.07e7      2 Middl~ Penn~          3 Not ~       871 Fema~    66
## 13  2010 9.31e5      2 Pacif~ Cali~         31 Not ~        69 Fema~    57
## 14  2010 1.18e7      6 West ~ Texas          0 Not ~       968 Fema~    73
## 15  2010 1.41e6      1 Pacif~ Cali~         59 Not ~       106 Male     43
## 16  2010 1.34e7      1 Pacif~ Wash~         33 Seat~      1040 Male     29
## 17  2010 1.17e7      2 West ~ Texas        381 Not ~       956 Male      8
## 18  2010 7.21e6      1 West ~ Miss~        189 Not ~       558 Male     78
## 19  2010 1.39e7      4 South~ West~          0 Not ~      1052 Male     10
## 20  2010 4.21e6      3 Mount~ Idaho          0 Not ~       290 Fema~     9
## # ... with 3 more variables: race <chr>, hhtype <chr>, relate <chr>
```

```r
# Of the first 20 rows of the entire data, look at values of just year and age
# Below two lines of code do the same thing
```

```
cen10[1:20, c("year", "age")]
```

```
## # A tibble: 20 x 2
##     year   age
##    <int> <int>
##  1  2010     8
##  2  2010    24
##  3  2010    37
##  4  2010    12
##  5  2010    18
##  6  2010    50
##  7  2010    51
##  8  2010    41
##  9  2010    62
## 10  2010    25
## 11  2010    23
## 12  2010    66
## 13  2010    57
## 14  2010    73
## 15  2010    43
## 16  2010    29
## 17  2010     8
## 18  2010    78
## 19  2010    10
## 20  2010     9
```

```
cen10 %>% slice(1:20) %>% select(c("year", "age"))
```

```
## # A tibble: 20 x 2
##     year   age
##    <int> <int>
##  1  2010     8
##  2  2010    24
##  3  2010    37
##  4  2010    12
##  5  2010    18
##  6  2010    50
##  7  2010    51
##  8  2010    41
##  9  2010    62
## 10  2010    25
## 11  2010    23
## 12  2010    66
## 13  2010    57
## 14  2010    73
## 15  2010    43
## 16  2010    29
## 17  2010     8
## 18  2010    78
## 19  2010    10
## 20  2010     9
```

A vector is a special type of matrix with only one column or only one row

```
# One column
cen10[1:10, c("age")]
```

```
## # A tibble: 10 x 1
##      age
##    <int>
##  1     8
##  2    24
##  3    37
##  4    12
##  5    18
##  6    50
##  7    51
##  8    41
##  9    62
## 10    25
```

```
cen10 %>% slice(1:10) %>% select(c("age"))
```

```
## # A tibble: 10 x 1
##      age
##    <int>
##  1     8
##  2    24
##  3    37
##  4    12
##  5    18
##  6    50
##  7    51
##  8    41
##  9    62
## 10    25
```

```
# One row
cen10[2, ]
```

```
## # A tibble: 1 x 13
##    year serial pernum region state countyfips city  cpuma0010 sex     age
##   <int>  <int>  <int> <chr>  <chr>       <int> <chr>     <int> <chr> <int>
## 1  2010 9.80e6      1 East ~ Ohio          103 Not ~       802 Male     24
## # ... with 3 more variables: race <chr>, hhtype <chr>, relate <chr>
```

```
cen10 %>% slice(2)
```

```
## # A tibble: 1 x 13
##    year serial pernum region state countyfips city  cpuma0010 sex     age
##   <int>  <int>  <int> <chr>  <chr>      <int> <chr>     <int> <chr> <int>
## 1  2010 9.80e6      1 East ~ Ohio         103 Not ~       802 Male     24
## # ... with 3 more variables: race <chr>, hhtype <chr>, relate <chr>
```

What if we want a special subset of the data? For example, what if I only want the records of individuals in California? What if I just want the age and race of individuals in California?

```
# subset for CA rows
ca_subset <- cen10[cen10$state == "California", ]

ca_subset_tidy <- cen10 %>% filter(state == "California")

all_equal(ca_subset, ca_subset_tidy)
```

```
## [1] TRUE
```

```
# subset for CA rows and select age and race
ca_subset_age_race <- cen10[cen10$state == "California", c("age", "race")]

ca_subset_age_race_tidy <- cen10 %>% filter(state == "California") %>% select(age, race)

all_equal(ca_subset_age_race, ca_subset_age_race_tidy)
```

```
## [1] TRUE
```

Some common operators that can be used to filter or to use as a condition. Remember, you can use the unique function to look at the set of all values a variable holds in the dataset.

```
# all individuals older than 30 and younger than 70
s1 <- cen10[cen10$age > 30 & cen10$age < 70, ]
s2 <- cen10 %>% filter(age > 30 & age < 70)
all_equal(s1, s2)
```

```
## [1] TRUE
```

```
# all individuals in either New York or California
s3 <- cen10[cen10$state == "New York" | cen10$state == "California", ]
s4 <- cen10 %>% filter(state == "New York" | state == "California")
all_equal(s3, s4)
```

```
## [1] TRUE
```

```
# all individuals NOT in the Pacific Division region
s5 <- cen10[cen10$region != "Pacific Division", ]
s6 <- cen10 %>% filter(region != "Pacific Division")
all_equal(s5, s6)
```

```
## [1] TRUE
```

```
# all individuals in any of the following states: California, Ohio, Nevada, Michigan
s7 <- cen10[cen10$state %in% c("California", "Ohio", "Nevada", "Michigan"), ]
s8 <- cen10 %>% filter(state %in% c("California", "Ohio", "Nevada", "Michigan"))
all_equal(s7, s8)
```

```
## [1] TRUE
```

```
# all individuals NOT in any of the following states: California, Ohio, Nevada, Michigan
s9 <- cen10[!(cen10$state %in% c("California", "Ohio", "Nevada", "Michigan")), ]
s10 <- cen10 %>% filter(!state %in% c("California", "Ohio", "Nevada", "Michigan"))
all_equal(s9, s10)
```

```
## [1] TRUE
```

Checkpoint

1

Get the subset of cen10 for non-white individuals (Hint: look at the set of values for the race variable by using the unique function)

```
# Enter here
```

2

Get the subset of cen10 for females over the age of 40

```
# Enter here
```

3

Get all the serial numbers for black, male individuals who don't live in Ohio or Nevada.

```
# Enter here
```

8.1.1 data frames

You can think of data frames maybe as matrices-plus, because a column can take on characters as well as numbers. As we just saw, this is often useful for real data analyses.

```
cen10
```

```
## # A tibble: 30,871 x 13
##     year serial pernum region state countyfips city  cpuma0010 sex     age
##    <int>  <int>  <int> <chr>  <chr>      <int> <chr>     <int> <chr> <int>
##  1  2010 8.80e6      4 Middl~ New ~          0 Not ~       636 Fema~     8
##  2  2010 9.80e6      1 East ~ Ohio         103 Not ~       802 Male     24
##  3  2010 8.69e6      1 Mount~ Neva~          3 Not ~       582 Male     37
##  4  2010 6.35e6      3 East ~ Mich~          0 Not ~       476 Fema~    12
##  5  2010 6.15e6      2 South~ Mary~         33 Not ~       449 Fema~    18
##  6  2010 8.10e6      1 New E~ New ~          0 Not ~       586 Male     50
##  7  2010 4.06e6      1 West ~ Iowa           0 Not ~       362 Fema~    51
##  8  2010 7.03e6      2 West ~ Miss~          0 Not ~       550 Fema~    41
##  9  2010 8.16e6      2 Middl~ New ~          3 Not ~       592 Male     62
## 10  2010 1.12e6      3 Pacif~ Cali~         37 Los ~        81 Male     25
## # ... with 30,861 more rows, and 3 more variables: race <chr>,
## #   hhtype <chr>, relate <chr>
```

Another way to think about data frames is that it is a type of list. Try the `str()` code below and notice how it is organized in slots. Each slot is a vector. They can be vectors of numbers or characters.

```
# enter this on your console
str(cen10)
```

8.2 Motivation

Nunn and Wantchekon (2011) – "The Slave Trade and the Origins of Mistrust in Africa"[2] – argues that across African countries, the distrust of co-ethnics fueled by the slave trade has had long-lasting effects on modern day trust in these territories. They argued that the slave trade created distrust in these societies in part because as some African groups were employed by European traders to capture their neighbors and bring them to the slave ships.

Nunn and Wantchekon use a variety of statistical tools to make their case (adding controls, ordered logit, instrumental variables, falsification tests, causal mechanisms), many of which will be covered in future courses. In this module we will only touch on their first set of analysis that use Ordinary Least Squares (OLS). OLS is likely the most common application of linear algebra in the social sciences. We will cover some linear algebra, matrix manipulation, and vector manipulation from this data.

8.3 Read Data

```
library(haven)
nunn_full <- read_dta("input/Nunn_Wantchekon_AER_2011.dta")
```

Nunn and Wantchekon's main dataset has more than 20,000 observations. Each observation is a respondent from the Afrobarometer survey.

```
head(nunn_full)
```

```
## # A tibble: 6 x 59
##   respno ethnicity murdock_name isocode region district townvill
##   <chr>  <chr>     <chr>        <chr>   <chr>  <chr>    <chr>
## 1 BEN00~ fon       FON          BEN     atlna~ KPOMASSE TOKPA-D~
## 2 BEN00~ fon       FON          BEN     atlna~ KPOMASSE TOKPA-D~
## 3 BEN00~ fon       FON          BEN     atlna~ OUIDAH   3ARROND
```

[2] Nunn, Nathan, and Leonard Wantchekon. 2011. "The Slave Trade and the Origins of Mistrust in Africa." American Economic Review 101(7): 3221–52.

```
## 4 BEN00~ fon       FON          BEN     atlna~ OUIDAH   3ARROND
## 5 BEN00~ fon       FON          BEN     atlna~ OUIDAH   PAHOU
## 6 BEN00~ fon       FON          BEN     atlna~ OUIDAH   PAHOU
## # ... with 52 more variables: location_id <dbl>, trust_relatives <dbl>,
## #   trust_neighbors <dbl>, intra_group_trust <dbl>,
## #   inter_group_trust <dbl>, trust_local_council <dbl>,
## #   ln_export_area <dbl>, export_area <dbl>, export_pop <dbl>,
## #   ln_export_pop <dbl>, age <dbl>, age2 <dbl>, male <dbl>,
## #   urban_dum <dbl>, occupation <dbl>, religion <dbl>,
## #   living_conditions <dbl>, education <dbl>, near_dist <dbl>,
## #   distsea <dbl>, loc_murdock_name <chr>, loc_ln_export_area <dbl>,
## #   local_council_performance <dbl>, council_listen <dbl>,
## #   corrupt_local_council <dbl>, school_present <dbl>,
## #   electricity_present <dbl>, piped_water_present <dbl>,
## #   sewage_present <dbl>, health_clinic_present <dbl>,
## #   district_ethnic_frac <dbl>, frac_ethnicity_in_district <dbl>,
## #   townvill_nonethnic_mean_exports <dbl>,
## #   district_nonethnic_mean_exports <dbl>,
## #   region_nonethnic_mean_exports <dbl>,
## #   country_nonethnic_mean_exports <dbl>, murdock_centr_dist_coast <dbl>,
## #   centroid_lat <dbl>, centroid_long <dbl>, explorer_contact <dbl>,
## #   railway_contact <dbl>, dist_Saharan_node <dbl>,
## #   dist_Saharan_line <dbl>, malaria_ecology <dbl>, v30 <dbl+lbl>,
## #   v33 <dbl+lbl>, fishing <dbl>, exports <dbl>, ln_exports <dbl>,
## #   total_missions_area <dbl>, ln_init_pop_density <dbl>,
## #   cities_1400_dum <dbl>
```

```r
colnames(nunn_full)
```

```
##  [1] "respno"                          "ethnicity"
##  [3] "murdock_name"                    "isocode"
##  [5] "region"                          "district"
##  [7] "townvill"                        "location_id"
##  [9] "trust_relatives"                 "trust_neighbors"
## [11] "intra_group_trust"               "inter_group_trust"
## [13] "trust_local_council"             "ln_export_area"
## [15] "export_area"                     "export_pop"
## [17] "ln_export_pop"                   "age"
## [19] "age2"                            "male"
## [21] "urban_dum"                       "occupation"
## [23] "religion"                        "living_conditions"
## [25] "education"                       "near_dist"
## [27] "distsea"                         "loc_murdock_name"
## [29] "loc_ln_export_area"              "local_council_performance"
## [31] "council_listen"                  "corrupt_local_council"
## [33] "school_present"                  "electricity_present"
## [35] "piped_water_present"             "sewage_present"
## [37] "health_clinic_present"           "district_ethnic_frac"
## [39] "frac_ethnicity_in_district"      "townvill_nonethnic_mean_exports"
## [41] "district_nonethnic_mean_exports" "region_nonethnic_mean_exports"
## [43] "country_nonethnic_mean_exports"  "murdock_centr_dist_coast"
## [45] "centroid_lat"                    "centroid_long"
## [47] "explorer_contact"                "railway_contact"
## [49] "dist_Saharan_node"               "dist_Saharan_line"
## [51] "malaria_ecology"                 "v30"
## [53] "v33"                             "fishing"
## [55] "exports"                         "ln_exports"
## [57] "total_missions_area"             "ln_init_pop_density"
## [59] "cities_1400_dum"
```

First, let's consider a small subset of this dataset.

```
nunn <- read_dta("./input/Nunn_Wantchekon_sample.dta")
```

```
nunn
```

```
## # A tibble: 10 x 5
##    trust_neighbors exports ln_exports export_area ln_export_area
##              <dbl>   <dbl>      <dbl>       <dbl>          <dbl>
##  1               3   0.388      0.328     0.00407        0.00406
##  2               3   0.631      0.489     0.0971         0.0926
##  3               3   0.994      0.690     0.0125         0.0124
##  4               0 183.         5.21      1.82           1.04
##  5               3   0          0         0              0
##  6               2   0          0         0              0
##  7               2 666.         6.50     14.0            2.71
##  8               0   0.348      0.298     0.00608        0.00606
##  9               3   0.435      0.361     0.0383         0.0376
## 10               3   0          0         0              0
```

8.4 data.frame vs. matricies

This is a `data.frame` object.

```
class(nunn)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

But it can be also consider a matrix in the linear algebra sense. What are the dimensions of this matrix?

```
nrow(nunn)
```

```
## [1] 10
```

`data.frames` and matrices have much overlap in `R`, but to explicitly treat an object as a matrix, you'd need to coerce its class. Let's call this matrix `X`.

```
X <- as.matrix(nunn)
```

What is the difference between a `data.frame` and a matrix? A `data.frame` can have columns that are of different types, whereas — in a matrix — all columns must be of the same type (usually either “numeric” or “character”).

8.5 Speed considerations

```
Nrow <- 100
Ncol <- 5
Xmat <- matrix(rnorm(Nrow * Ncol), nrow = Nrow, ncol = Ncol)
Xdf <- as.data.frame(Xmat)

system.time(replicate(50000, colMeans(Xmat)))
```

```
##    user  system elapsed
##   0.551   0.011   0.640
```

```
system.time(replicate(50000, colMeans(Xdf)))
```

```
##    user  system elapsed
##   7.283   0.128   8.751
```

8.6 Handling matricies in `R`

You can easily transpose a matrix

```
X
```

```
##       trust_neighbors     exports ln_exports  export_area ln_export_area
##  [1,]               3   0.3883497  0.3281158  0.004067405    0.004059155
##  [2,]               3   0.6311236  0.4892691  0.097059444    0.092633367
##  [3,]               3   0.9941893  0.6902376  0.012524694    0.012446908
##  [4,]               0 182.5891266  5.2127004  1.824284434    1.038255095
##  [5,]               3   0.0000000  0.0000000  0.000000000    0.000000000
##  [6,]               2   0.0000000  0.0000000  0.000000000    0.000000000
##  [7,]               2 665.9652100  6.5027380 13.975566864    2.706419945
##  [8,]               0   0.3476418  0.2983562  0.006082553    0.006064130
##  [9,]               3   0.4349871  0.3611559  0.038332380    0.037615947
## [10,]               3   0.0000000  0.0000000  0.000000000    0.000000000
```

```
t(X)
```

```
##                        [,1]       [,2]       [,3]       [,4] [,5] [,6]
## trust_neighbors 3.000000000 3.00000000 3.00000000   0.000000    3    2
## exports         0.388349682 0.63112360 0.99418926 182.589127    0    0
## ln_exports      0.328115761 0.48926911 0.69023758   5.212700    0    0
## export_area     0.004067405 0.09705944 0.01252469   1.824284    0    0
## ln_export_area  0.004059155 0.09263337 0.01244691   1.038255    0    0
##                       [,7]        [,8]       [,9] [,10]
## trust_neighbors   2.000000 0.000000000 3.00000000     3
## exports         665.965210 0.347641766 0.43498713     0
## ln_exports        6.502738 0.298356235 0.36115587     0
## export_area      13.975567 0.006082553 0.03833238     0
## ln_export_area    2.706420 0.006064130 0.03761595     0
```

What are the values of all rows in the first column?

```
X[, 1]
```

```
##  [1] 3 3 3 0 3 2 2 0 3 3
```

What are all the values of “exports”? (i.e. return the whole “exports” column)

```
X[, "exports"]
```

```
##  [1]   0.3883497   0.6311236   0.9941893 182.5891266   0.0000000
##  [6]   0.0000000 665.9652100   0.3476418   0.4349871   0.0000000
```

What is the first observation (i.e. first row)?

```
X[1, ]
```

```
## trust_neighbors         exports      ln_exports     export_area
##     3.000000000     0.388349682     0.328115761     0.004067405
##  ln_export_area
##     0.004059155
```

What is the value of the first variable of the first observation?

```
X[1, 1]
```

```
## trust_neighbors
##               3
```

Pause and consider the following problem on your own. What is the following code doing?

```
X[X[, "trust_neighbors"] == 0, "export_area"]
```

```
## [1] 1.824284434 0.006082553
```

Why does it give the same output as the following?

```
X[which(X[, "trust_neighbors"] == 0), "export_area"]
```

```
## [1] 1.824284434 0.006082553
```

Some more manipulation

```
X + X
```

```
##       trust_neighbors     exports ln_exports export_area ln_export_area
##  [1,]               6   0.7766994  0.6562315 0.008134809     0.00811831
##  [2,]               6   1.2622472  0.9785382 0.194118887     0.18526673
##  [3,]               6   1.9883785  1.3804752 0.025049388     0.02489382
##  [4,]               0 365.1782532 10.4254007 3.648568869     2.07651019
##  [5,]               6    0.0000000  0.0000000  0.000000000     0.00000000
##  [6,]               4    0.0000000  0.0000000  0.000000000     0.00000000
##  [7,]               4 1331.9304199 13.0054760 27.951133728     5.41283989
##  [8,]               0    0.6952835  0.5967125  0.012165107     0.01212826
##  [9,]               6    0.8699743  0.7223117  0.076664761     0.07523189
## [10,]               6    0.0000000  0.0000000  0.000000000     0.00000000
```

```
X - X
```

```
##       trust_neighbors exports ln_exports export_area ln_export_area
##  [1,]               0       0          0           0              0
##  [2,]               0       0          0           0              0
##  [3,]               0       0          0           0              0
##  [4,]               0       0          0           0              0
##  [5,]               0       0          0           0              0
##  [6,]               0       0          0           0              0
##  [7,]               0       0          0           0              0
##  [8,]               0       0          0           0              0
##  [9,]               0       0          0           0              0
## [10,]               0       0          0           0              0
```

```
t(X) %*% X
```

```
##                 trust_neighbors    exports ln_exports export_area
## trust_neighbors       62.000000   1339.276   18.61181    28.40709
## exports             1339.276369 476850.298 5283.76294  9640.42990
## ln_exports            18.611811   5283.763   70.50077   100.46202
## export_area           28.407085   9640.430  100.46202   198.65558
## ln_export_area         5.853106   1992.047   23.08189    39.72847
##                 ln_export_area
## trust_neighbors       5.853106
## exports            1992.046502
## ln_exports           23.081893
## export_area          39.728468
## ln_export_area        8.412887
```

```
cbind(X, 1:10)
```

```
##       trust_neighbors     exports ln_exports  export_area ln_export_area
##  [1,]               3   0.3883497  0.3281158  0.004067405    0.004059155
##  [2,]               3   0.6311236  0.4892691  0.097059444    0.092633367
##  [3,]               3   0.9941893  0.6902376  0.012524694    0.012446908
##  [4,]               0 182.5891266  5.2127004  1.824284434    1.038255095
##  [5,]               3   0.0000000  0.0000000  0.000000000    0.000000000
##  [6,]               2   0.0000000  0.0000000  0.000000000    0.000000000
##  [7,]               2 665.9652100  6.5027380 13.975566864    2.706419945
##  [8,]               0   0.3476418  0.2983562  0.006082553    0.006064130
##  [9,]               3   0.4349871  0.3611559  0.038332380    0.037615947
## [10,]               3   0.0000000  0.0000000  0.000000000    0.000000000
##
```

```
##  [1,]  1
##  [2,]  2
##  [3,]  3
##  [4,]  4
##  [5,]  5
##  [6,]  6
##  [7,]  7
##  [8,]  8
##  [9,]  9
## [10,] 10
```

```
cbind(X, 1)
```

```
##       trust_neighbors     exports ln_exports  export_area ln_export_area  
##  [1,]               3   0.3883497  0.3281158  0.004067405    0.004059155 1
##  [2,]               3   0.6311236  0.4892691  0.097059444    0.092633367 1
##  [3,]               3   0.9941893  0.6902376  0.012524694    0.012446908 1
##  [4,]               0 182.5891266  5.2127004  1.824284434    1.038255095 1
##  [5,]               3   0.0000000  0.0000000  0.000000000    0.000000000 1
##  [6,]               2   0.0000000  0.0000000  0.000000000    0.000000000 1
##  [7,]               2 665.9652100  6.5027380 13.975566864    2.706419945 1
##  [8,]               0   0.3476418  0.2983562  0.006082553    0.006064130 1
##  [9,]               3   0.4349871  0.3611559  0.038332380    0.037615947 1
## [10,]               3   0.0000000  0.0000000  0.000000000    0.000000000 1
```

```
colnames(X)
```

```
## [1] "trust_neighbors" "exports"         "ln_exports"      "export_area"    
## [5] "ln_export_area"
```

8.7 Variable Transformations

`exports` is the total number of slaves that were taken from the individual's ethnic group between Africa's four slave trades between 1400-1900.

What is `ln_exports`? The article describes this as the natural log of one plus the `exports`. This is a transformation of one column by a particular function

```
log(1 + X[, "exports"])
```

```
##  [1] 0.3281158 0.4892691 0.6902376 5.2127003 0.0000000 0.0000000 6.5027379
##  [8] 0.2983562 0.3611559 0.0000000
```

Question for you: why add the 1?

Verify that this is the same as `X[, "ln_exports"]`

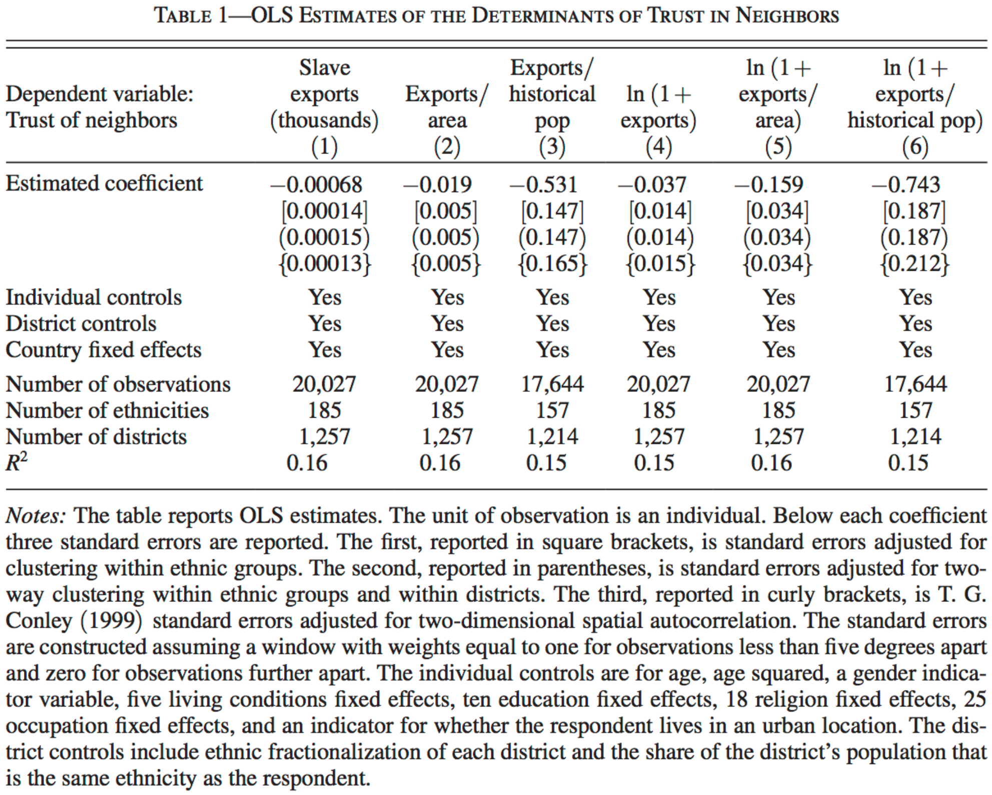


TABLE 1—OLS ESTIMATES OF THE DETERMINANTS OF TRUST IN NEIGHBORS

Dependent variable: Trust of neighbors	Slave exports (thousands) (1)	Exports/ area (2)	Exports/ historical pop (3)	ln (1+ exports) (4)	ln (1+ exports/ area) (5)	ln (1+ exports/ historical pop) (6)
Estimated coefficient	−0.00068 [0.00014] (0.00015) {0.00013}	−0.019 [0.005] (0.005) {0.005}	−0.531 [0.147] (0.147) {0.165}	−0.037 [0.014] (0.014) {0.015}	−0.159 [0.034] (0.034) {0.034}	−0.743 [0.187] (0.187) {0.212}
Individual controls	Yes	Yes	Yes	Yes	Yes	Yes
District controls	Yes	Yes	Yes	Yes	Yes	Yes
Country fixed effects	Yes	Yes	Yes	Yes	Yes	Yes
Number of observations	20,027	20,027	17,644	20,027	20,027	17,644
Number of ethnicities	185	185	157	185	185	157
Number of districts	1,257	1,257	1,214	1,257	1,257	1,214
R^2	0.16	0.16	0.15	0.15	0.16	0.15

Notes: The table reports OLS estimates. The unit of observation is an individual. Below each coefficient three standard errors are reported. The first, reported in square brackets, is standard errors adjusted for clustering within ethnic groups. The second, reported in parentheses, is standard errors adjusted for two-way clustering within ethnic groups and within districts. The third, reported in curly brackets, is T. G. Conley (1999) standard errors adjusted for two-dimensional spatial autocorrelation. The standard errors are constructed assuming a window with weights equal to one for observations less than five degrees apart and zero for observations further apart. The individual controls are for age, age squared, a gender indicator variable, five living conditions fixed effects, ten education fixed effects, 18 religion fixed effects, 25 occupation fixed effects, and an indicator for whether the respondent lives in an urban location. The district controls include ethnic fractionalization of each district and the share of the district's population that is the same ethnicity as the respondent.

Figure 8.1

8.8 Linear Combinations

In Table 1 we see "OLS Estimates". These are estimates of OLS coefficients and standard errors. You do not need to know what these are for now, but it doesn't hurt to getting used to seeing them.

A very crude way to describe regression is through linear combinations. The simplest linear combination is a one-to-one transformation.

Take the first number in Table 1, which is -0.00068. Now, multiply this by `exports`

```
-0.00068 * X[, "exports"]
```

```
##  [1] -0.0002640778 -0.0004291640 -0.0006760487 -0.1241606061  0.0000000000
##  [6]  0.0000000000 -0.4528563428 -0.0002363964 -0.0002957912  0.0000000000
```

Now, just one more step. Make a new matrix with just exports and the value 1

```
X2 <- cbind(1, X[, "exports"])
```

name this new column "intercept"

```
colnames(X2)
```

```
## NULL
```

```
colnames(X2) <- c("intercept", "exports")
```

What are the dimensions of the matrix `X2`?

```
dim(X2)
```

```
## [1] 10  2
```

Now consider a new matrix, called `B`.

```
B <- matrix(c(1.62, -0.00068))
```

What are the dimensions of `B`?

```
dim(B)
```

```
## [1] 2 1
```

What is the product of `X2` and `B`? From the dimensions, can you tell if it will be conformable?

```
X2 %*% B
```

```
##           [,1]
##  [1,] 1.619736
##  [2,] 1.619571
##  [3,] 1.619324
##  [4,] 1.495839
##  [5,] 1.620000
##  [6,] 1.620000
##  [7,] 1.167144
##  [8,] 1.619764
##  [9,] 1.619704
## [10,] 1.620000
```

What is this multiplication doing in terms of equations?

Exercises

1

Let

$$\mathbf{A} = \begin{bmatrix} 0.6 & 0.2 \\ 0.4 & 0.8 \end{bmatrix}$$

Use R to write code that will create the matrix A, and then consecutively multiply A to itself 4 times. What is the value of A^4?

```
## Enter yourself
```

Note that R notation of matrices is different from the math notation. Simply trying `X^n` where `X` is a matrix will only take the power of each element to `n`. Instead, this problem asks you to perform matrix multiplication.

2

Let's apply what we learned about subsetting or filtering/selecting. Use the `nunn_full` dataset you have already loaded

a) First, show all observations (rows) that have a `"male"` variable higher than 0.5

```
## Enter yourself
```

b) Next, create a matrix / dataframe with only two columns: `"trust_neighbors"` and `"age"`

```
## Enter yourself
```

c) Lastly, show all values of `"trust_neighbors"` and `"age"` for observations (rows) that have the "male" variable value that is higher than 0.5

```
## Enter yourself
```

3

Find a way to generate a vector of "column averages" of the matrix `X` from the Nunn and Wantchekon data in one line of code. Each entry in the vector should contain the sample average of the values in the column. So a 100 by 4 matrix should generate a length-4 matrix.

4

Similarly, generate a vector of "column medians".

5

Consider the regression that was run to generate Table 1:

```
form <- "trust_neighbors ~ exports + age + age2 +  male + urban_dum + factor(education) + factor
lm_1_1 <- lm(as.formula(form), nunn_full)

# The below coef function returns a vector of OLS coefficiants
coef(lm_1_1)
```

```
##                (Intercept)                      exports
##               1.619913e+00                -6.791360e-04
##                        age                       age2
##               8.395936e-03              -5.473436e-05
##                       male                  urban_dum
##               4.550246e-02              -1.404551e-01
##         factor(education)1         factor(education)2
##               1.709816e-02              -5.224591e-02
##         factor(education)3         factor(education)4
##              -1.373770e-01              -1.889619e-01
##         factor(education)5         factor(education)6
##              -1.893494e-01              -2.400767e-01
##         factor(education)7         factor(education)8
##              -2.850748e-01              -1.232085e-01
##         factor(education)9        factor(occupation)1
##              -2.406437e-01               6.185655e-02
##        factor(occupation)2        factor(occupation)3
##               7.392168e-02               3.356158e-02
##        factor(occupation)4        factor(occupation)5
##               7.942048e-03               6.661126e-02
##        factor(occupation)6        factor(occupation)7
##              -7.563297e-02               1.699699e-02
##        factor(occupation)8        factor(occupation)9
##              -9.428177e-02              -9.981440e-02
##       factor(occupation)10       factor(occupation)11
##              -3.307068e-02              -2.300045e-02
##       factor(occupation)12       factor(occupation)13
##              -1.564540e-01              -1.441370e-02
##       factor(occupation)14       factor(occupation)15
##              -5.566414e-02              -2.343762e-01
##       factor(occupation)16       factor(occupation)18
##              -1.306947e-02              -1.729589e-01
##       factor(occupation)19       factor(occupation)20
##              -1.770261e-01              -2.457800e-02
##       factor(occupation)21       factor(occupation)22
##              -4.936813e-02              -1.068511e-01
##       factor(occupation)23       factor(occupation)24
##              -9.712205e-02               1.292371e-02
##       factor(occupation)25      factor(occupation)995
##               2.623186e-02              -1.195063e-03
##          factor(religion)2          factor(religion)3
##               5.395953e-02               7.887878e-02
##          factor(religion)4          factor(religion)5
##               4.749150e-02               4.318455e-02
##          factor(religion)6          factor(religion)7
##              -1.787694e-02              -3.616542e-02
##         factor(religion)10         factor(religion)11
##               6.015041e-02               2.237845e-01
##         factor(religion)12         factor(religion)13
##               2.627086e-01              -6.812813e-02
##         factor(religion)14         factor(religion)15
##               4.673681e-02               3.844555e-01
##        factor(religion)360        factor(religion)361
##               3.656843e-01               3.416413e-01
##        factor(religion)362        factor(religion)363
##               8.230393e-01               3.856565e-01
##        factor(religion)995 factor(living_conditions)2
##               4.161301e-02               4.395862e-02
## factor(living_conditions)3 factor(living_conditions)4
##               8.627372e-02               1.197428e-01
## factor(living_conditions)5       district_ethnic_frac
##               1.203606e-01              -1.553648e-02
## frac_ethnicity_in_district                 isocodeBWA
##               1.011222e-01              -4.258953e-01
##                 isocodeGHA                 isocodeKEN
##               1.135307e-02              -1.819556e-01
##                 isocodeLSO                 isocodeMDG
##              -5.511200e-01              -3.315727e-01
##                 isocodeMLI                 isocodeMOZ
##               7.528101e-02               8.223730e-02
##                 isocodeMWI                 isocodeNAM
##               3.062497e-01              -1.397541e-01
##                 isocodeNGA                 isocodeSEN
##              -2.381525e-01               3.867371e-01
##                 isocodeTZA                 isocodeUGA
##               2.079366e-01              -6.443732e-02
##                 isocodeZAF                 isocodeZMB
##              -2.179153e-01              -2.172868e-01
```

First, get a small subset of the nunn_full dataset. This time, sample 20 rows and select for variables `exports`, `age`, `age2`, `male`, and `urban_dum`. To this small subset, add (`bind_cols()` in tidyverse or `cbind()` in base R) a column of 1's; this represents the intercept. If you need some guidance, look at how we sampled 10 rows selected for a different set of variables above in the lecture portion.

```
# Enter here
```

Next let's try calculating predicted values of levels of trust in neighbors by multiplying coefficients for the intercept, `exports`, `age`, `age2`, `male`, and `urban_dum` to the actual observed values for those variables in the small subset you've just created.

```
# Hint: You can get just selected elements from the vector returned by coef(lm_1_1)

# For example, the below code gives you the first 3 elements of the original vector
coef(lm_1_1)[1:3]
```

```
##  (Intercept)      exports         age
##  1.619913146 -0.000679136  0.008395936
```

```
# Also, the below code gives you the coefficient elements for intercept and male
coef(lm_1_1)[c("(Intercept)", "male")]
```

```
## (Intercept)        male
##  1.61991315  0.04550246
```

Chapter 9

Visualization[1]

Where are we? Where are we headed?

Up till now, you should have covered:

- The R Visualization and Programming primers at `https://rstudio.cloud/primers/`
- Reading and handling data
- Matrices and Vectors
- What does `:` mean in R? What about `==`? `,`?, `!=` , `&`, `|`, `%in%`
- What does `%>%` do?

Today we'll cover:

- Visualization
- A bit of data wrangling

Check your understanding

- How do you make a barplot, in base-R and in ggplot?
- How do you add layers to a ggplot?
- How do you change the axes of a ggplot?
- How do you make a histogram?
- How do you make a graph that looks like this?

Other review

9.1 Motivation: The Law of the Census

In this module, let's visualize some cross-sectional stats with an actual Census. Then, we'll do an example on time trends with Supreme Court ideal points.

[1] Module originally written by Shiro Kuriwaki

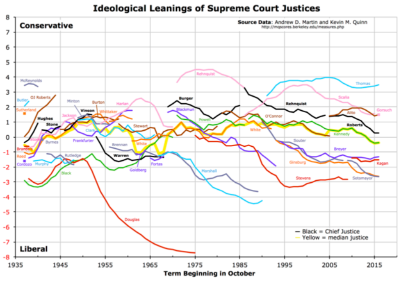


Figure 9.1: By Randy Schutt - Own work, CC BY-SA 3.0, Wikimedia.

Why care about the Census? The Census is one of the fundamental acts of a government. See the Law Review article by Persily (2011), "The Law of the Census."[2] The Census is government's primary tool for apportionment (allocating seats to districts), appropriations (allocating federal funding), and tracking demographic change. See[3] for example Hochschild and Powell (2008) on how the categorizations of race in the Census during 1850-1930. Notice also that both of these pieces are not inherently "quantitative" — the Persily article is a Law Review and the Hochschild and Powell article is on American Historical Development — but data analysis would be certainly relevant.

Time series data is a common form of data in social science data, and there is growing methodological work on making causal inferences with time series.[4] We will use the the ideological estimates of the Supreme court, which has been in the news with Brett Kavanaugh's nomination.

9.2 Read data

First, the census. Read in a subset of the 2010 Census. This is a 1 percent sample of the entire U.S. Census (100 times larger than the previous version).

```r
cen10 <- readRDS("input/usc2010_1percent.Rds")
```

The data comes from IPUMS[5], a great source to extract and analyze Census and Census-conducted survey (ACS, CPS) data.

[2] Persily, Nathaniel. 2011. "The Law of the Census: How to Count, What to Count, Whom to Count, and Where to Count Them.". *Cardozo Law Review* 32(3): 755–91.

[3] Hochschild, Jennifer L., and Brenna Marea Powell. 2008. "Racial Reorganization and the United States Census 1850–1930: Mulattoes, Half-Breeds, Mixed Parentage, Hindoos, and the Mexican Race.". *Studies in American Political Development* 22(1): 59–96.

[4] Blackwell, Matthew, and Adam Glynn. 2018. "How to Make Causal Inferences with Time-Series Cross-Sectional Data under Selection on Observables." *American Political Science Review*

[5] Ruggles, Steven, Katie Genadek, Ronald Goeken, Josiah Grover, and Matthew Sobek. 2015. Integrated Public Use Microdata Series: Version 6.0 dataset

9.3 Counting

How many people are in your sample?

```
nrow(cen10)
```

```
## [1] 3087108
```

This and all subsequent tasks involve manipulating and summarizing data, sometimes called "wrangling". As per last time, there are both "base-R" and "tidyverse" approaches.

Yesterday we saw several functions from the tidyverse:

- `select` selects columns
- `filter` selects rows based on a logical (boolean) statement
- `slice` selects rows based on the row number
- `arrange` reordered the rows in descending order.

In this visualization section, we'll make use of the pair of functions `group_by()` and `summarize()`.

9.4 Tabulating

Summarizing data is the key part of communication; good data viz gets the point across.[6] Summaries of data come in two forms: tables and figures.

Here are two ways to count by group, or to tabulate.

In base-R Use the `table` function, that provides how many rows exist for an unique value of the vector (remember `unique` from yesterday?)

```
table(cen10$race)
```

```
## 
## American Indian or Alaska Native    Black/African American/Negro 
##                            29584                          390640 
##                          Chinese                        Japanese 
##                            33537                            7629 
##  Other Asian or Pacific Islander                 Other race, nec 
##                           110325                          190319 
##         Three or more major races                 Two major races 
##                             8421                           82724 
##                            White 
##                          2233929
```

With tidyverse, a quick convenience function is `count`, with the variable to count on included.

```
count(cen10, race)
```

[6] Kastellec, Jonathan P., and Eduardo L. Leoni. 2007. "Using Graphs Instead of Tables in Political Science.". *Perspectives on Politics* 5 (4): 755–71.

```
## # A tibble: 9 x 2
##   race                                   n
##   <chr>                              <int>
## 1 American Indian or Alaska Native   29584
## 2 Black/African American/Negro      390640
## 3 Chinese                            33537
## 4 Japanese                            7629
## 5 Other Asian or Pacific Islander   110325
## 6 Other race, nec                   190319
## 7 Three or more major races           8421
## 8 Two major races                    82724
## 9 White                            2233929
```

We can check out the arguments of `count` and see that there is a `sort` option. What does this do?

```
count(cen10, race, sort = TRUE)
```

```
## # A tibble: 9 x 2
##   race                                   n
##   <chr>                              <int>
## 1 White                            2233929
## 2 Black/African American/Negro      390640
## 3 Other race, nec                   190319
## 4 Other Asian or Pacific Islander   110325
## 5 Two major races                    82724
## 6 Chinese                            33537
## 7 American Indian or Alaska Native   29584
## 8 Three or more major races           8421
## 9 Japanese                            7629
```

`count` is a kind of shorthand for `group_by()` and `summarize`. This code would have done the same.

```
cen10 %>%
  group_by(race) %>%
  summarize(n = n())
```

```
## # A tibble: 9 x 2
##   race                                   n
##   <chr>                              <int>
## 1 American Indian or Alaska Native   29584
## 2 Black/African American/Negro      390640
## 3 Chinese                            33537
## 4 Japanese                            7629
## 5 Other Asian or Pacific Islander   110325
## 6 Other race, nec                   190319
## 7 Three or more major races           8421
## 8 Two major races                    82724
## 9 White                            2233929
```

If you are new to tidyverse, what would you *think* each row did? Reading the function help page, verify if your intuition was correct.

where `n()` is a function that counts rows.

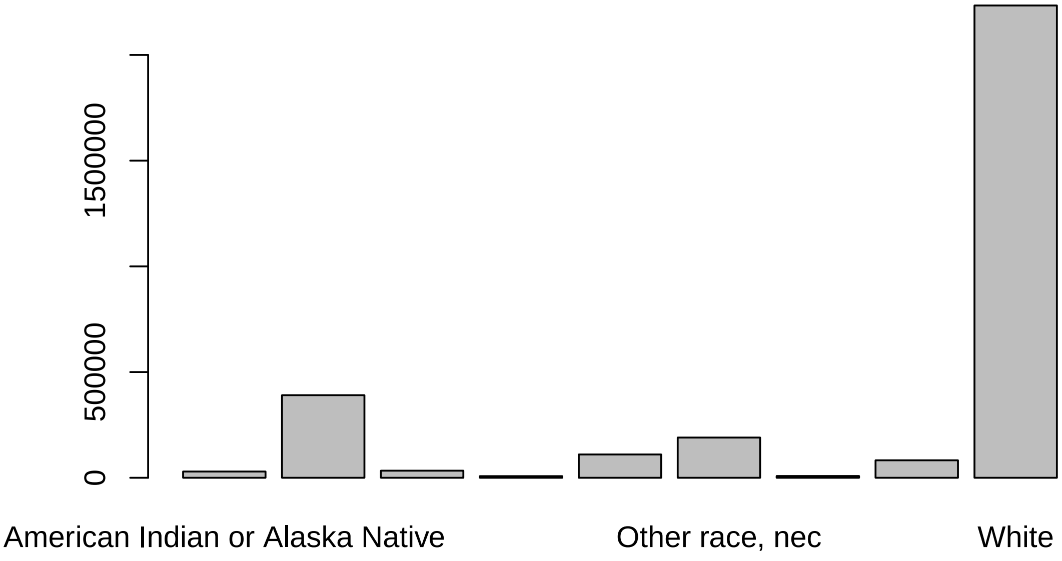
9.5 base R graphics and ggplot

Two prevalent ways of making graphing are referred to as "base-R" and "ggplot".

9.5.1 base R

"Base-R" graphics are graphics that are made with R's default graphics commands. First, let's assign our tabulation to an object, then put it in the `barplot()` function.

```
barplot(table(cen10$race))
```



9.5.2 ggplot

A popular alternative a `ggplot` graphics, that you were introduced to in the tutorial. `gg` stands for grammar of graphics by Hadley Wickham, and it has a new semantics of explaining graphics in R. Again, first let's set up the data.

Although the tutorial covered making scatter plots as the first cut, often data requires summaries before they made into graphs.

For this example, let's group and count first like we just did. But assign it to a new object.

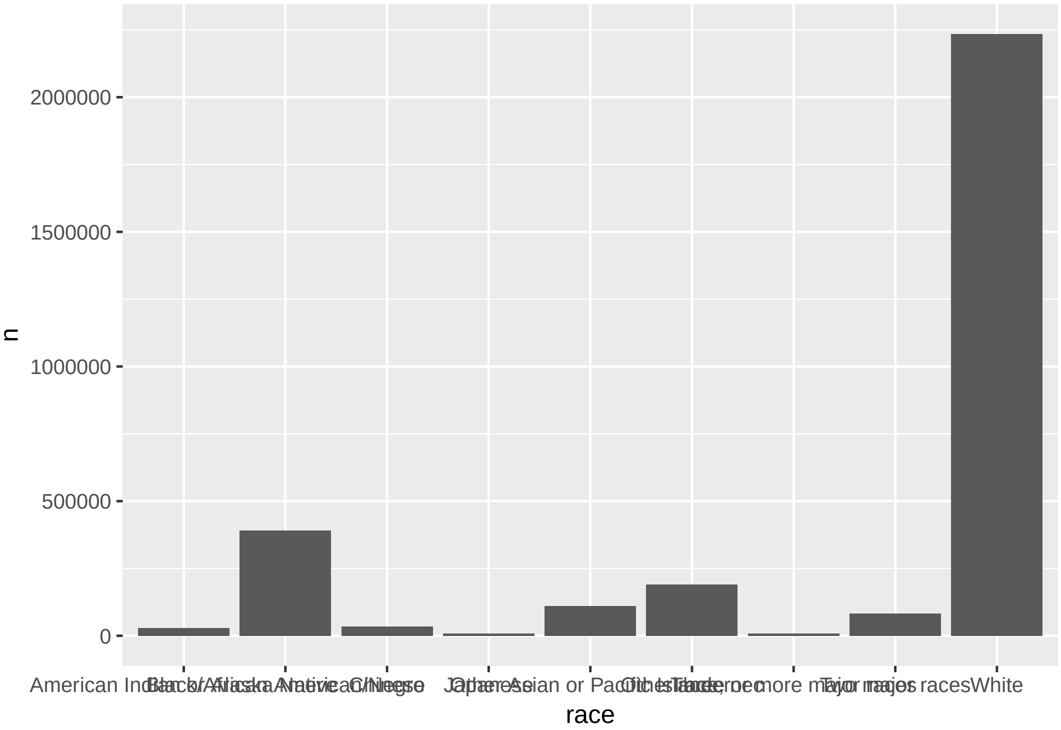
```
grp_race <- count(cen10, race)
```

We will now plot this grouped set of numbers. Recall that the `ggplot()` function takes two main arguments, `data` and `aes`.

1. First enter a single dataframe from which you will draw a plot.
2. Then enter the `aes`, or aesthetics. This defines which variable in the data the plotting functions should take for pre-set dimensions in graphics. The dimensions `x` and `y` are the most important. We will assign `race` and `count` to them, respectively,
3. After you close `ggplot()` .. add **layers** by the plus sign. A `geom` is a layer of graphical representation, for example `geom_histogram` renders a histogram, `geom_point` renders a scatter plot. For a barplot, we can use `geom_col()`

What is the right geometry layer to make a barplot? Turns out:

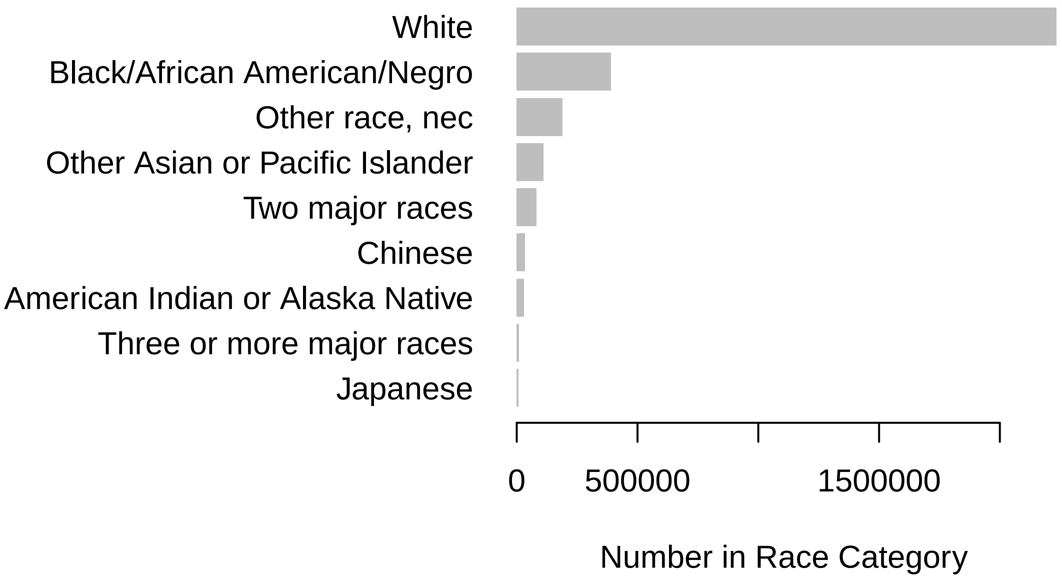
```
ggplot(data = grp_race, aes(x = race, y = n)) + geom_col()
```



9.6 Improving your graphics

Adjusting your graphics to make the point clear is an important skill. Here is a base-R example of showing the same numbers but with a different design, in a way that aims to maximize the "data-to-ink ratio".

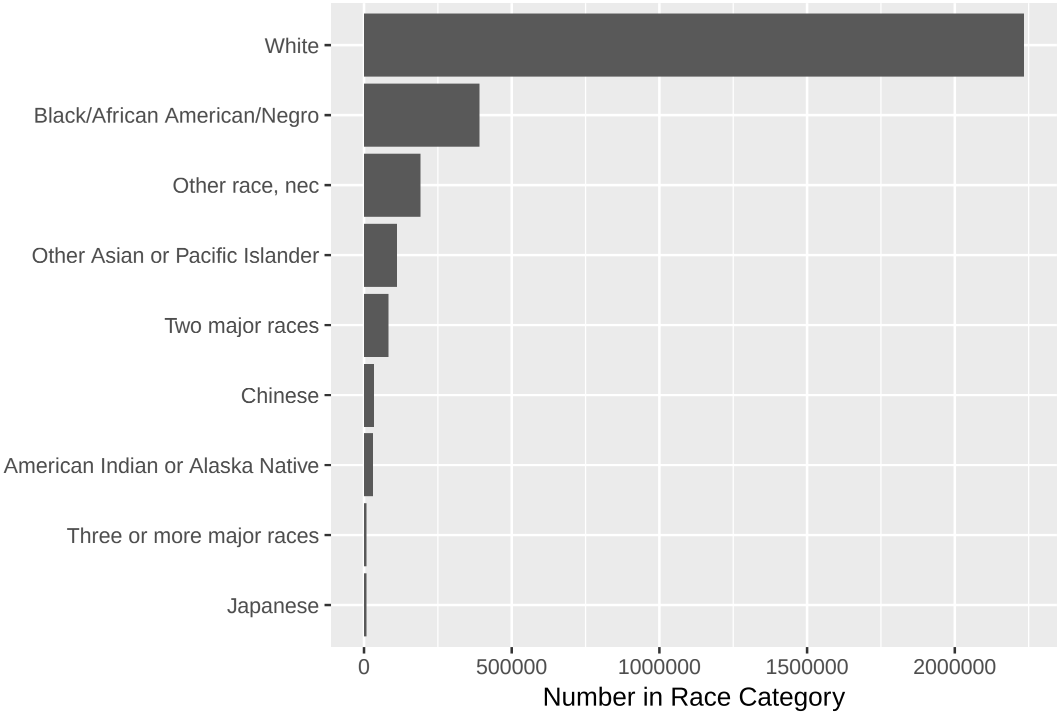
```
par(oma = c(1, 11, 1, 1))
barplot(sort(table(cen10$race)), # sort numbers
        horiz = TRUE, # flip
        border = NA, # border is extraneous
        xlab = "Number in Race Category",
        bty = "n", # no box
        las = 1) # alignment of axis labels is horizontal
```



Notice that we applied the `sort()` function to order the bars in terms of their counts. The default ordering of a categorical variable / factor is alphabetical. Alphabetical ordering is uninformative and almost never the way you should order variables.

In ggplot you might do this by:

```
library(forcats)

grp_race_ordered <- arrange(grp_race, n) %>%
  mutate(race = as_factor(race))

ggplot(data = grp_race_ordered, aes(x = race, y = n)) +
  geom_col() +
  coord_flip() +
  labs(y = "Number in Race Category",
       x = "",
       caption = "Source: 2010 U.S. Census sample")
```



Source: 2010 U.S. Census sample

The data ink ratio was popularized by Ed Tufte (originally a political economy scholar who has recently become well known for his data visualization work). See Tufte (2001), *The Visual Display of Quantitative Information* and his website `https://www.edwardtufte.com/tufte/`. For a R and ggplot focused example using social science examples, check out Healy (2018), *Data Visualization: A Practical Introduction* with a draft at `https://socviz.co/`[7]. There are a growing number of excellent books on data visualization.

9.7 Cross-tabs

Visualizations and Tables each have their strengths. A rule of thumb is that more than a dozen numbers on a table is too much to digest, but less than a dozen is too few for a figure to be worth it. Let's look at a table first.

A cross-tab is counting with two types of variables, and is a simple and powerful tool to show the relationship between multiple variables.

```
xtab_race_state <- table(cen10$state, cen10$race)
```

What would the dimensions of this table be?

What if we care about proportions within states, rather than counts. We want to compare the racial composition of a small state (like Delaware) and a large state (like California).

[7] Healy, Kieran. forthcoming. *Data Visualization: A Practical Introduction*. Princeton University Press

One way to transform a table of counts to a table of proportions is the function `prop.table`. Be careful what you want to take proportions of – this is set by the `margin` argument. In R, the first margin (`margin = 1`) is *rows* and the second (`margin = 2`) is *columns*.

```
ptab_race_state <- prop.table(xtab_race_state, margin = 2)
```

9.8 Composition Plots

How would you make the same figure with `ggplot()`? First, we want a count for each state × race combination. So group by those two factors and count how many observations are in each two-way categorization. `group_by()` can take any number of variables, separated by commas.

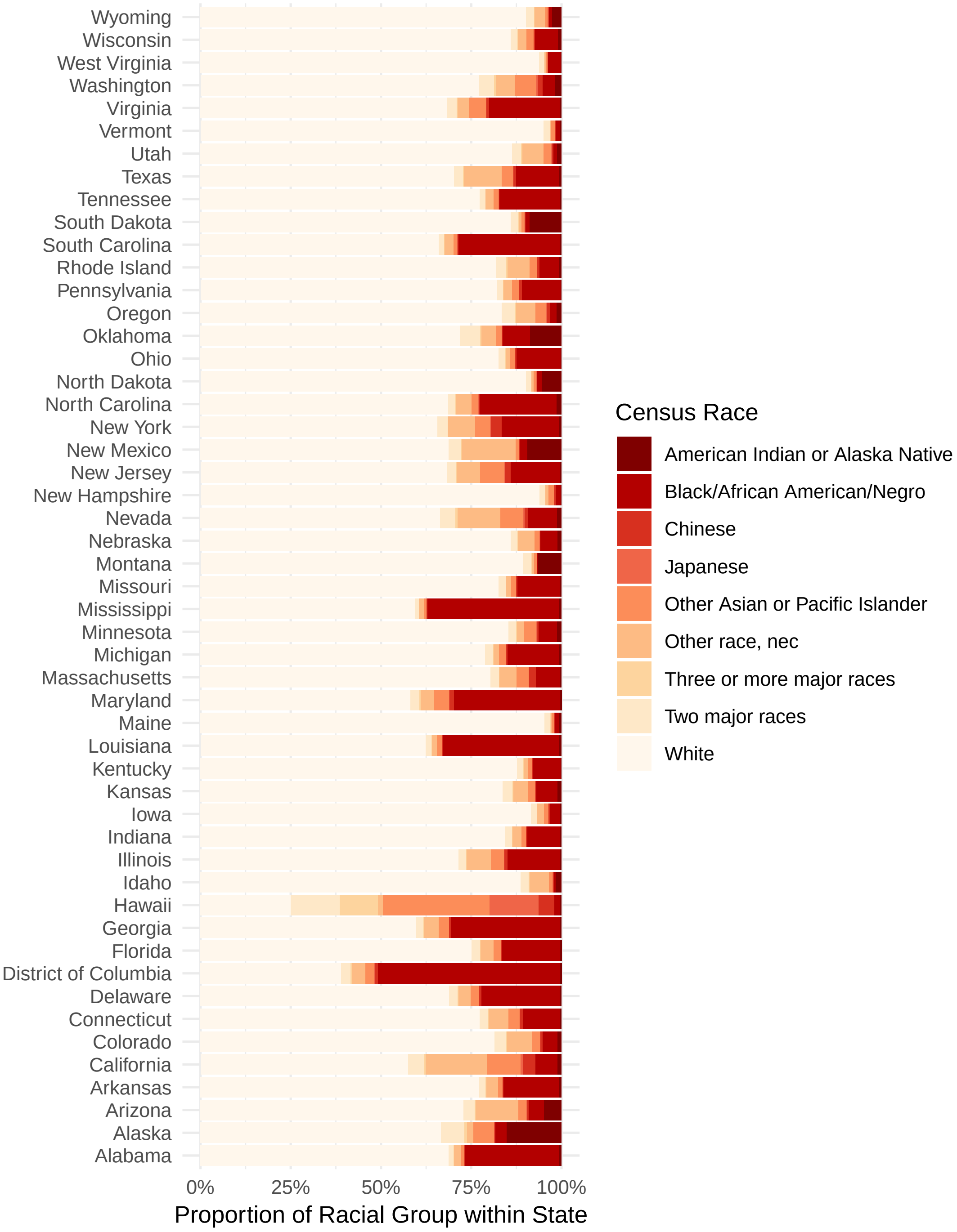
```
grp_race_state <- cen10 %>%
  count(race, state)
```

Can you tell from the code what `grp_race_state` will look like?

```
# run on your own
grp_race_state
```

Now, we want to tell `ggplot2` something like the following: I want bars by state, where heights indicate racial groups. Each bar should be colored by the race. With some googling, you will get something like this:

```
ggplot(data = grp_race_state, aes(x =  state, y = n,  fill = race)) +
  geom_col(position = "fill") + # the position is dertemined by the fill ae
  scale_fill_brewer(name = "Census Race", palette = "OrRd", direction = -1) + # choose palette
  coord_flip() + # flip axes
  scale_y_continuous(labels = percent) + # label numbers as percentage
  labs(y = "Proportion of Racial Group within State",
       x = "",
       source = "Source: 2010 Census  sample") +
  theme_minimal()
```


Wyoming
Wisconsin
West Virginia
Washington
Virginia
Vermont
Utah
Texas
Tennessee
South Dakota
South Carolina
Rhode Island
Pennsylvania
Oregon
Oklahoma
Ohio
North Dakota
North Carolina
New York
New Mexico
New Jersey
New Hampshire
Nevada
Nebraska
Montana
Missouri
Mississippi
Minnesota
Michigan
Massachusetts
Maryland
Maine
Louisiana
Kentucky
Kansas
Iowa
Indiana
Illinois
Idaho
Hawaii
Georgia
Florida
District of Columbia
Delaware
Connecticut
Colorado
California
Arkansas
Arizona
Alaska
Alabama
0%
25%
50%
75%
100%
Proportion of Racial Group within State
Census Race
American Indian or Alaska Native
Black/African American/Negro
Chinese
Japanese
Other Asian or Pacific Islander
Other race, nec
Three or more major races
Two major races
White

9.9 Line graphs

Line graphs are useful for plotting time trends.

The Census does not track individuals over time. So let's take up another example: The U.S. Supreme Court. Take the dataset justices_median.csv.

This data is adapted from the estimates of Martin and Quinn on their website http://mqscores.lsa.umich.edu/.[8]

```
justice <- read_csv("input/justices_court-median.csv")
```

What does the data look like? How do you think it is organized? What does each row represent?

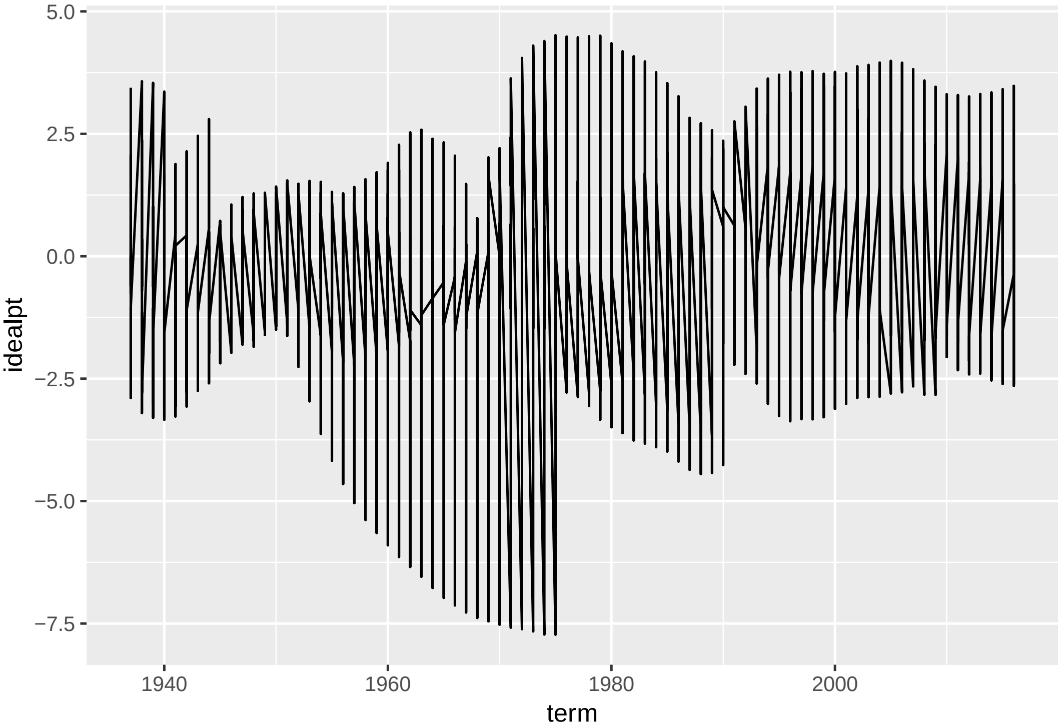
```
justice
```

```
## # A tibble: 728 x 7
##     term justice_id justice idealpt idealpt_sd median_idealpt
##    <int>      <int> <chr>     <dbl>      <dbl>          <dbl>
##  1  1937         67 McReyn~   3.44       0.546         NA
##  2  1938         67 McReyn~   3.57       0.561         NA
##  3  1939         67 McReyn~   3.54       0.616         -1.07
##  4  1940         67 McReyn~   3.36       0.714         -0.734
##  5  1937         68 Brande~  -0.611      0.273         NA
##  6  1938         68 Brande~  -0.616      0.313         NA
##  7  1937         71 Suther~   1.58       0.551         NA
##  8  1937         72 Butler    2.07       0.425         NA
##  9  1938         72 Butler    2.37       0.429         NA
## 10  1937         74 Stone    -0.769      0.259         NA
## # ... with 718 more rows, and 1 more variable: median_justice <chr>
```

As you might have guessed, these data can be shown in a time trend from the range of the term variable. As there are only nine justices at any given time and justices have life tenure, there times on the court are staggered. With a common measure of "preference", we can plot time trends of these justices ideal points on the same y-axis scale.

```
ggplot(justice, aes(x = term, y = idealpt)) +
  geom_line()
```

[8] This exercise inspired from Princeton's R Camp Assignment.



Why does the above graph not look like the the put in the beginning? Fix it by adding just one aesthetic to the graph.

```
# enter a correction that draws separate lines by group.
```

If you got the right aesthetic, this seems to "work" off the shelf. But take a moment to see why the code was written as it is and how that maps on to the graphics. What is the `group` aesthetic doing for you?

Now, this graphic already indicates a lot, but let's improve the graphics so people can actually read it. This is left for a Exercise.

As social scientists, we should also not forget to ask ourselves whether these numerical measures are fit for what we care about, or actually succeeds in measuring what we'd like to measure. The estimation of these "ideal points" is a subfield of political methodology beyond this prefresher. For more reading, skim through the original paper by Martin and Quinn (2002).[9] Also for a methodological discussion on the difficulty of measuring time series of preferences, check out Bailey (2013).[10]

Exercises

In the time remaining, try the following exercises. Order doesn't matter.

[9] Martin, Andrew D. and Kevin M. Quinn. 2002. "Dynamic Ideal Point Estimation via Markov Chain Monte Carlo for the U.S. Supreme Court, 1953-1999". *Political Analysis.* 10(2): 134-153.

[10] Bailey, Michael A. 2013. "Is Today's Court the Most Conservative in Sixty Years? Challenges and Opportunities in Measuring Judicial Preferences.". *Journal of Politics* 75(3): 821-834

1: Rural states

Make a well-labelled figure that plots the proportion of the state's population (as per the census) that reside in their file as a renter. Each state should be visualized as a point, rather than a bar, and there should be 51 points, ordered by their value. All labels should be readable.

```
# Enter yourself
```

- Alternatively, you can for instead plot the proportion of residents who do not reisde in a specified city.

2: The swing justice

Using the `justice` dataset and building off of the plot that was given, make an improved plot by implementing as many of the following changes (which hopefully improves the graph):

- Label axes
- Use a black-white background.
- Change the breaks of the x-axis to print numbers for every decade, not just every two decades.
- Plots each line in translucent gray, so the overlapping lines can be visualized clearly. (Hint: in ggplot the `alpha` argument controls the degree of transparency)
- Limit the scale of the y-axis to [-5, 5] so that the outlier justice in the 60s is trimmed and the rest of the data can be seen more easily (also, who is that justice?)
- Plot the ideal point of the justice who holds the "median" ideal point in a given term. To distinguish this with the others, plot this line separately in a very light red *below* the individual justice's lines.
- Highlight the trend-line of only the nine justices who are *currently* sitting on SCOTUS. Make sure this is clearer than the other past justices.
- Add the current nine justice's names to the right of the endpoint of the 2016 figure, alongside their ideal point.
- Make sure the text labels do not overlap with each other for readability using the `ggrepel` package.
- Extend the x-axis label to about 2020 so the text labels of justices are to the right of the trend-lines.
- Add a caption to your text describing the data briefly, as well as any features relevant for the reader (such as the median line and the trimming of the y-axis)

```
# Enter yourself
```

3: Don't sort by the alphabet

The Figure we made that shows racial composition by state has one notable shortcoming: it orders the states alphabetically, which is not particularly useful if you want see an overall pattern, without having particular states in mind.

Find a way to modify the figures so that the states are ordered by the *proportion* of White residents in the sample.

```
# Enter yourself
```

4 What to show and how to show it

As a student of politics our goal is not necessarily to make pretty pictures, but rather make pictures that tell us something about politics, government, or society. If you could augment either the census dataset or the justices dataset in some way, what would be an substantively significant thing to show as a graphic?

Chapter 10

Objects, Functions, Loops

Where are we? Where are we headed?

Up till now, you should have covered:

- R basic programming
- Data Import
- Statistical Summaries
- Visualization

Today we'll cover

- Objects
- Functions
- Loops

10.1 What is an object?

Now that we have covered some hands-on ways to use graphics, let's go into some fundamentals of the R language.

Let's first set up

```
library(dplyr)
library(readr)
library(haven)
library(ggplot2)
```

```
cen10 <- read_csv("input/usc2010_001percent.csv", col_types = cols())
```

Objects are abstract symbols in which you store data. Here we will create an object from copy, and assign cen10 to it.

```
copy <- cen10
```

This looks the same as the original dataset:

```
copy
```

```
## # A tibble: 30,871 x 13
##     year serial pernum region state countyfips city  cpuma0010 sex     age
##    <int>  <int>  <int> <chr>  <chr>      <int> <chr>     <int> <chr> <int>
##  1  2010 8.80e6      4 Middl~ New ~          0 Not ~       636 Fema~     8
##  2  2010 9.80e6      1 East ~ Ohio         103 Not ~       802 Male     24
##  3  2010 8.69e6      1 Mount~ Neva~          3 Not ~       582 Male     37
##  4  2010 6.35e6      3 East ~ Mich~          0 Not ~       476 Fema~    12
##  5  2010 6.15e6      2 South~ Mary~         33 Not ~       449 Fema~    18
##  6  2010 8.10e6      1 New E~ New ~          0 Not ~       586 Male     50
##  7  2010 4.06e6      1 West ~ Iowa           0 Not ~       362 Fema~    51
##  8  2010 7.03e6      2 West ~ Miss~          0 Not ~       550 Fema~    41
##  9  2010 8.16e6      2 Middl~ New ~          3 Not ~       592 Male     62
## 10  2010 1.12e6      3 Pacif~ Cali~         37 Los ~        81 Male     25
## # ... with 30,861 more rows, and 3 more variables: race <chr>,
## #   hhtype <chr>, relate <chr>
```

What happens if you do this next?

```
copy <- ""
```

It got reassigned:

```
copy
```

```
## [1] ""
```

10.1.1 lists

Lists are one of the most generic and flexible type of object. You can make an empty list by the function `list()`

```
my_list <- list()
my_list
```

```
## list()
```

And start filling it in. Slots on the list are invoked by double square brackets `[[]]`

```
my_list[[1]] <- "contents of the first slot -- this is a string"
my_list[["slot 2"]] <- "contents of slot named slot 2"
my_list
```

```
## [[1]]
## [1] "contents of the first slot -- this is a string"
##
## $`slot 2`
## [1] "contents of slot named slot 2"
```

each slot can be anything. What are we doing here? We are defining the 1st slot of the list `my_list` to be a vector `c(1, 2, 3, 4, 5)`

```
my_list[[1]] <- c(1, 2, 3, 4, 5)
my_list
```

```
## [[1]]
## [1] 1 2 3 4 5
##
## $`slot 2`
## [1] "contents of slot named slot 2"
```

You can even make nested lists. Let's say we want the 1st slot of the list to be another list of three elements.

```
my_list[[1]][[1]] <- "subitem 1 in slot 1 of my_list"
my_list[[1]][[2]] <- "subitem 1 in slot 2 of my_list"
my_list[[1]][[3]] <- "subitem 1 in slot 3 of my_list"

my_list
```

```
## [[1]]
## [1] "subitem 1 in slot 1 of my_list" "subitem 1 in slot 2 of my_list"
## [3] "subitem 1 in slot 3 of my_list" "4"
## [5] "5"
##
## $`slot 2`
## [1] "contents of slot named slot 2"
```

10.2 Making your own objects

We've covered one type of object, which is a list. You saw it was quite flexible. How many types of objects are there?

There are an infinite number of objects, because people make their own class of object. You can detect the type of the object (the class) by the function `class`

Object can be said to be an instance of a class.

Analogies:

class - Pokemon, **object** - Pikachu

class - Book, **object** - To Kill a Mockingbird

class - DataFrame, **object** - 2010 census data

class - Character, **object** - "Programming is Fun"

What is type (class) of object is `cen10`?

```
class(cen10)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

What about this text?

```
class("some random text")
```

```
## [1] "character"
```

To change or create the class of any object, you can *assign* it. To do this, assign the name of your class to character to an object's `class()`.

We can start from a simple list. For example, say we wanted to store data about pokemon. Because there is no pre-made package for this, we decide to make our own class.

```
pikachu <- list(name = "Pikachu",
                number = 25,
                type = "Electric",
                color = "Yellow")
```

and we can give it any class name we want.

```
class(pikachu) <- "Pokemon"
str(pikachu)
```

```
## List of 4
##  $ name  : chr "Pikachu"
##  $ number: num 25
##  $ type  : chr "Electric"
##  $ color : chr "Yellow"
##  - attr(*, "class")= chr "Pokemon"
```

```
pikachu$type
```

```
## [1] "Electric"
```

10.2.1 Seeing R through objects

Most of the R objects that you will see as you advance are their own objects. For example, here's a linear regression object (which you will learn more about in Gov 2000):

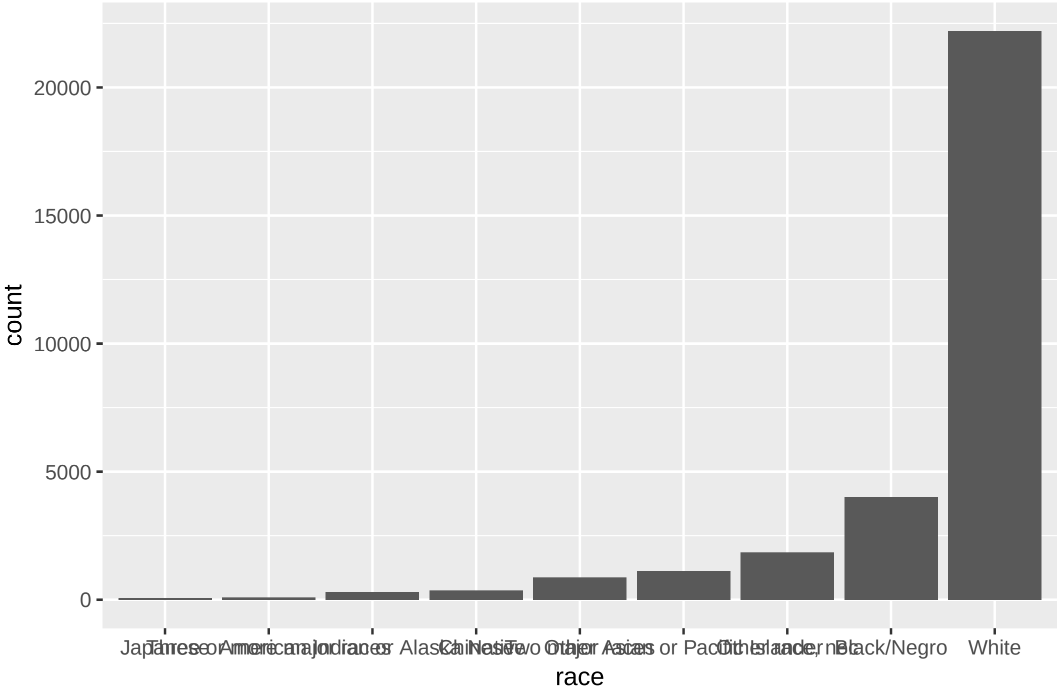
```
ols <- lm(mpg ~ wt + vs + gear + carb, mtcars)
class(ols)
```

```
## [1] "lm"
```

Anything can be an object! Even graphs (in `ggplot`) can be assigned, re-assigned, and edited.

```
grp_race <- group_by(cen10, race)%>%
  summarize(count = n())
```

```r
grp_race_ordered <- arrange(grp_race, count) %>%
  mutate(race = forcats::as_factor(race))

gg_tab <- ggplot(data = grp_race_ordered) +
  aes(x = race, y = count) +
  geom_col() +
  labs(caption = "Source: U.S. Census 2010")

gg_tab
```



Source: U.S. Census 2010

You can change the orientation

```r
gg_tab<- gg_tab + coord_flip()
```

10.2.2 Parsing an object by `str()`s

It can be hard to understand an R object because it's contents are unknown. The function `str`, short for structure, is a quick way to look into the innards of an object

```r
str(my_list)
```

```
## List of 2
##  $       : chr [1:5] "subitem 1 in slot 1 of my_list" "subitem 1 in slot 2 of my_list" "subite
##  $ slot 2: chr "contents of slot named slot 2"
```

```
class(my_list)
```

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

Same for the object we just made

```
str(pikachu)
```

```
## List of 4
##  $ name  : chr "Pikachu"
##  $ number: num 25
##  $ type  : chr "Electric"
##  $ color : chr "Yellow"
##  - attr(*, "class")= chr "Pokemon"
```

What does a `ggplot` object look like? Very complicated, but at least you can see it:

```
# enter this on your console
str(gg_tab)
```

10.3 Types of variables

In the social science we often analyze variables. As you saw in the tutorial, different types of variables require different care.

A key link with what we just learned is that variables are also types of R objects.

10.3.1 scalars

One number. How many people did we count in our Census sample?

```
nrow(cen10)
```

```
## [1] 30871
```

Question: What proportion of our census sample is Native American? This number is also a scalar

```
# Enter yourself
unique(cen10$race)
```

```
## [1] "White"                          "Black/Negro"
## [3] "Other race, nec"                "American Indian or Alaska Native"
## [5] "Chinese"                        "Other Asian or Pacific Islander"
## [7] "Two major races"                "Three or more major races"
## [9] "Japanese"
```

```
mean(cen10$race == "American Indian or Alaska Native")
```

```
## [1] 0.009555894
```

Hint: you can use the function `mean()` to calcualte the sample mean. The sample proportion is the mean of a sequence of number, where your event of interest is a 1 (or `TRUE`) and others are 0 (or `FALSE`).

10.3.2 numeric vectors

A sequence of numbers.

```
grp_race_ordered$count
```

```
## [1]    77    88   295   354   869  1129  1839  4013 22207
```

```
class(grp_race_ordered$count)
```

```
## [1] "integer"
```

Or even, all the ages of the millions of people in our Census. Here are just the first few numbers of the list.

```
head(cen10$age)
```

```
## [1]  8 24 37 12 18 50
```

10.3.3 characters (aka strings)

This can be just one stretch of characters

```
my_name <- "Yon Soo"
my_name
```

```
## [1] "Yon Soo"
```

```
class(my_name)
```

```
## [1] "character"
```

or more characters. Notice here that there's a difference between a vector of individual characters and a length-one object of characters.

```
my_name_letters <- c("S", "h", "i", "r", "o")
my_name_letters
```

```
## [1] "S" "h" "i" "r" "o"
```

```
class(my_name_letters)
```

```
## [1] "character"
```

Finally, remember that lower vs. upper case matters in R!

```
my_name2 <- "shiro"
my_name == my_name2
```

```
## [1] FALSE
```

10.4 What is a function?

Most of what we do in R is executing a function. `read_csv()`, `nrow()`, `ggplot()` .. pretty much anything with a parentheses is a function. And even things like `<-` and `[` are functions as well.

A function is a set of instructions with specified ingredients. It takes an **input**, then **manipulates** it – changes it in some way – and then returns the manipulated product.

One way to see what a function actually does is to enter it without parentheses.

```
# enter this on your console
table
```

You'll see below that the most basic functions are quite complicated internally.

You'll notice that functions contain other functions. *wrapper* functions are functions that "wrap around" existing functions. This sounds redundant, but it's an important feature of programming. If you find yourself repeating a command more than two times, you should make your own function, rather than writing the same type of code.

10.4.1 Write your own function

It's worth remembering the basic structure of a function. You create a new function, call it `my_fun` by this:

```
my_fun <- function() {

}
```

If we wanted to generate a function that computed the number of men in your data, what would that look like?

```
count_men <- function(data) {

  nmen <- sum(data$sex == "Male")

  return(nmen)
}
```

Then all we need to do is feed this function a dataset

```
count_men(cen10)
```

```
## [1] 15220
```

The point of a function is that you can use it again and again without typing up the set of constituent manipulations. So, what if we wanted to figure out the number of men in California?

```
count_men(cen10[cen10$state == "California",])
```

```
## [1] 1876
```

Let's go one step further. What if we want to know the proportion of non-whites in a state, just by entering the name of the state? There's multiple ways to do it, but it could look something like this

```
nw_in_state <- function(data, state) {

  s.subset <- data[data$state == state,]
  total.s <- nrow(s.subset)
  nw.s <- sum(s.subset$race != "White")

  nw.s / total.s
}
```

The last line is what gets generated from the function. To be more explicit you can wrap the last line around `return()`. (as in `return(nw.s/total.s)`. `return()` is used when you want to break out of a function in the middle of it and not wait till the last line.

Try it on your favorite state!

```
nw_in_state(cen10, "Massachusetts")
```

```
## [1] 0.2040185
```

Checkpoint

1

Try making your own function, `average_age_in_state`, that will give you the average age of people in a given state.

```
# Enter on your own
```

2

Try making your own function, `asians_in_state`, that will give you the number of `Chinese`, `Japanese`, and `Other Asian or Pacific Islander` people in a given state.

```
# Enter on your own
```

3

Try making your own function, 'top_10_oldest_cities', that will give you the names of cities whose population's average age is top 10 oldest.

```
# Enter on your own
```

10.5 What is a package?

You can think of a package as a suite of functions that other people have already built for you to make your life easier.

```r
help(package = "ggplot2")
```

To use a package, you need to do two things: (1) install it, and then (2) load it.

Installing is a one-time thing

```r
install.packages("ggplot2")
```

But you need to load each time you start a R instance. So always keep these commands on a script.

```r
library(ggplot2)
```

In `rstudio.cloud`, we already installed a set of packages for you. But when you start your own R instance, you need to have installed the package at some point.

10.6 Conditionals

Sometimes, you want to execute a command only under certain conditions. This is done through the almost universal function, `if()`. Inside the `if` function we enter a logical statement. The line that is adjacent to, or follows, the `if()` statement only gets executed if the statement returns `TRUE`.

For example,

For example,

```r
x <- 5
if (x >0) {
  print("positive number")
} else if (x == 0)  {
  print ("zero")
} else {
  print("negative number")
}
```

```
## [1] "positive number"
```

You can wrap that whole things in a function

```r
is_positive <- function(number) {
  if (number >0) {
    print("positive number")
  } else if (number == 0)  {
    print ("zero")
  } else {
    print("negative number")
  }
}

is_positive(5)
```

```
## [1] "positive number"
```

```
is_positive(-3)
```

```
## [1] "negative number"
```

10.7 For-loops

Loops repeat the same statement, although the statement can be "the same" only in an abstract sense. Use the `for(x in X)` syntax to repeat the subsequent command as many times as there are elements in the right-hand object `X`. Each of these elements will be referred to the left-hand index `x`

First, come up with a vector.

```
fruits <- c("apples", "oranges", "grapes")
```

Now we use the `fruits` vector in a `for` loop.

```
for (fruit in fruits) {
  print(paste("I love", fruit))
}
```

```
## [1] "I love apples"
## [1] "I love oranges"
## [1] "I love grapes"
```

Here `for()` and `in` must be part of any for loop. The right hand side `fruits` must be a thing that exists. Finally the `left-hand` side object is "Pick your favor name." It is analogous to how we can index a sum with any letter. $\sum_{i=1}^{10} i$ and `sum_{j = 1}^{10}j` are in fact the same thing.

```
for (i in 1:length(fruits)) {
  print(paste("I love", fruits[i]))
}
```

```
## [1] "I love apples"
## [1] "I love oranges"
## [1] "I love grapes"
```

```
states_of_interest <- c("California", "Massachusetts", "New Hampshire", "Washington")

for( state in states_of_interest){
  state_data <- cen10[cen10$state == state,]
  nmen <- sum(state_data$sex == "Male")
  n <- nrow(state_data)
  men_perc <- round(100*(nmen/n), digits=2)
  print(paste("Percentage of men in",state, "is", men_perc))

}
```

```
## [1] "Percentage of men in California is 49.85"
## [1] "Percentage of men in Massachusetts is 47.6"
## [1] "Percentage of men in New Hampshire is 48.55"
## [1] "Percentage of men in Washington is 48.19"
```

Instead of printing, you can store the information in a vector

```
states_of_interest <- c("California", "Massachusetts", "New Hampshire", "Washington")
male_percentages <- c()
iter <-1

for( state in states_of_interest){
  state_data <- cen10[cen10$state == state,]
  nmen <- sum(state_data$sex == "Male")
  n <- nrow(state_data)
  men_perc <- round(100*(nmen/n), digits=2)

  male_percentages <- c(male_percentages, men_perc)
  names(male_percentages)[iter] <- state
  iter <- iter + 1
}

male_percentages
```

```
##    California Massachusetts New Hampshire    Washington
##         49.85         47.60         48.55         48.19
```

10.8 Nested Loops

What if I want to calculate the population percentage of a race group for all race groups in states of interest? You could probably use tidyverse functions to do this, but let's try using loops!

```
states_of_interest <- c("California", "Massachusetts", "New Hampshire", "Washington")
for (state in states_of_interest) {
  for (race in unique(cen10$race)) {
    race_state_num <- nrow(cen10[cen10$race == race & cen10$state == state, ])
    state_pop <- nrow(cen10[cen10$state == state, ])
    race_perc <- round(100*(race_state_num/(state_pop)), digits=2)
    print(paste("Percentage of ", race , "in", state, "is", race_perc))
  }
}
```

```
## [1] "Percentage of  White in California is 57.61"
## [1] "Percentage of  Black/Negro in California is 6.72"
## [1] "Percentage of  Other race, nec in California is 15.55"
## [1] "Percentage of  American Indian or Alaska Native in California is 1.12"
## [1] "Percentage of  Chinese in California is 3.75"
## [1] "Percentage of  Other Asian or Pacific Islander in California is 9.54"
## [1] "Percentage of  Two major races in California is 4.62"
## [1] "Percentage of  Three or more major races in California is 0.37"
## [1] "Percentage of  Japanese in California is 0.72"
## [1] "Percentage of  White in Massachusetts is 79.6"
## [1] "Percentage of  Black/Negro in Massachusetts is 5.87"
## [1] "Percentage of  Other race, nec in Massachusetts is 4.02"
## [1] "Percentage of  American Indian or Alaska Native in Massachusetts is 0.77"
## [1] "Percentage of  Chinese in Massachusetts is 2.32"
## [1] "Percentage of  Other Asian or Pacific Islander in Massachusetts is 4.33"
## [1] "Percentage of  Two major races in Massachusetts is 2.78"
## [1] "Percentage of  Three or more major races in Massachusetts is 0"
## [1] "Percentage of  Japanese in Massachusetts is 0.31"
## [1] "Percentage of  White in New Hampshire is 93.48"
## [1] "Percentage of  Black/Negro in New Hampshire is 0.72"
## [1] "Percentage of  Other race, nec in New Hampshire is 0.72"
## [1] "Percentage of  American Indian or Alaska Native in New Hampshire is 0.72"
## [1] "Percentage of  Chinese in New Hampshire is 0.72"
## [1] "Percentage of  Other Asian or Pacific Islander in New Hampshire is 2.17"
## [1] "Percentage of  Two major races in New Hampshire is 0.72"
## [1] "Percentage of  Three or more major races in New Hampshire is 0"
## [1] "Percentage of  Japanese in New Hampshire is 0.72"
## [1] "Percentage of  White in Washington is 76.05"
## [1] "Percentage of  Black/Negro in Washington is 2.9"
## [1] "Percentage of  Other race, nec in Washington is 5.37"
## [1] "Percentage of  American Indian or Alaska Native in Washington is 2.03"
## [1] "Percentage of  Chinese in Washington is 1.31"
## [1] "Percentage of  Other Asian or Pacific Islander in Washington is 6.68"
## [1] "Percentage of  Two major races in Washington is 4.79"
## [1] "Percentage of  Three or more major races in Washington is 0.29"
## [1] "Percentage of  Japanese in Washington is 0.58"
```

Exercises

Exercise 1: Counting CVAP

A issue raised in Persily's article is that the full-count U.S. Census does not record whether the residents are citizens of the United States[1]. Instead, this question is asked in a survey, the American Community Survey. The two are fundamentally different exercises: the Census counts everyone by definition, a survey samples its data. Load the 1 percent sample of the 2015 ACS (`acs2015_1percent.csv`, in the `input` folder) and give an estimate of the proportion of a state's ACS respondents that are reportedly U.S. citizens.

```
acs<- read_csv("input/acs2015_1percent.csv", col_types = cols())
set.seed(02138)
sample_acs <- sample_frac(acs, 0.01)

# Enter yourself
```

Exercise 2: Write your own function

Write your own function that makes some task of data analysis simpler. Ideally, it would be a function that helps you do either of the previous tasks in fewer lines of code. You can use the three lines of code that was provided in exercise 1 to wrap that into another function too!

```
# Enter yourself
```

Exercise 3: Using Loops

Using a loop, create a crosstab of sex and race for each state in the set "states_of_interest"

```
states_of_interest <- c("California", "Massachusetts", "New Hampshire", "Washington")
# Enter yourself
```

Exercise 4: Storing information derived within loops in a global dataframe

Recall the following nested loop

```
states_of_interest <- c("California", "Massachusetts", "New Hampshire", "Washington")
for (state in states_of_interest) {
  for (race in unique(cen10$race)) {
    race_state_num <- nrow(cen10[cen10$race == race & cen10$state == state, ])
    state_pop <- nrow(cen10[cen10$state == state, ])
```

[1] Here is that argument of his again, more recently in the popular press. "The Mysterious Number of American Citizens". June 2, 2015. *POLITICO*

```
    race_perc <- round(100*(race_state_num/(state_pop)), digits=2)
    print(paste("Percentage of ", race , "in", state, "is", race_perc))
  }
}
```

```
## [1] "Percentage of  White in California is 57.61"
## [1] "Percentage of  Black/Negro in California is 6.72"
## [1] "Percentage of  Other race, nec in California is 15.55"
## [1] "Percentage of  American Indian or Alaska Native in California is 1.12"
## [1] "Percentage of  Chinese in California is 3.75"
## [1] "Percentage of  Other Asian or Pacific Islander in California is 9.54"
## [1] "Percentage of  Two major races in California is 4.62"
## [1] "Percentage of  Three or more major races in California is 0.37"
## [1] "Percentage of  Japanese in California is 0.72"
## [1] "Percentage of  White in Massachusetts is 79.6"
## [1] "Percentage of  Black/Negro in Massachusetts is 5.87"
## [1] "Percentage of  Other race, nec in Massachusetts is 4.02"
## [1] "Percentage of  American Indian or Alaska Native in Massachusetts is 0.77"
## [1] "Percentage of  Chinese in Massachusetts is 2.32"
## [1] "Percentage of  Other Asian or Pacific Islander in Massachusetts is 4.33"
## [1] "Percentage of  Two major races in Massachusetts is 2.78"
## [1] "Percentage of  Three or more major races in Massachusetts is 0"
## [1] "Percentage of  Japanese in Massachusetts is 0.31"
## [1] "Percentage of  White in New Hampshire is 93.48"
## [1] "Percentage of  Black/Negro in New Hampshire is 0.72"
## [1] "Percentage of  Other race, nec in New Hampshire is 0.72"
## [1] "Percentage of  American Indian or Alaska Native in New Hampshire is 0.72"
## [1] "Percentage of  Chinese in New Hampshire is 0.72"
## [1] "Percentage of  Other Asian or Pacific Islander in New Hampshire is 2.17"
## [1] "Percentage of  Two major races in New Hampshire is 0.72"
## [1] "Percentage of  Three or more major races in New Hampshire is 0"
## [1] "Percentage of  Japanese in New Hampshire is 0.72"
## [1] "Percentage of  White in Washington is 76.05"
## [1] "Percentage of  Black/Negro in Washington is 2.9"
## [1] "Percentage of  Other race, nec in Washington is 5.37"
## [1] "Percentage of  American Indian or Alaska Native in Washington is 2.03"
## [1] "Percentage of  Chinese in Washington is 1.31"
## [1] "Percentage of  Other Asian or Pacific Islander in Washington is 6.68"
## [1] "Percentage of  Two major races in Washington is 4.79"
## [1] "Percentage of  Three or more major races in Washington is 0.29"
## [1] "Percentage of  Japanese in Washington is 0.58"
```

Instead of printing the percentage of each race in each state, create a dataframe, and store all that information in that dataframe. (Hint: look at how I stored information about male percentage in each state of interest in a vector.)

Chapter 11

Joins and Merges, Wide and Long[1]

Where are we? Where are we headed?

Up till now, you should have covered:

- R basic programming
- Counting.
- Visualization.
- Objects and Classes.
- Matrix algebra in R
- Functions.

Today you will work on your own, but feel free to ask a fellow classmate nearby or the instructor. The objective for this session is to get more experience using R, but in the process (a) test a prominent theory in the political science literature and (b) explore related ideas of interest to you.

11.1 Motivation

The "Democratic Peace" is one of the most widely discussed propositions in political science, covering the fields of International Relations and Comparative Politics, with insights to domestic politics of democracies (e.g. American Politics). The one-sentence idea is that democracies do not fight with each other. There have been much theoretical debate – for example in earlier work, Oneal and Russet (1999) argue that the democratic peace is not due to the hegemony of strong democracies like the U.S. and attempt to distinguish between

[1] Module originally written by Shiro Kuriwaki, Connor Jerzak, and Yon Soo Park

realist and what they call Kantian propositions (e.g. democratic governance, international organizations)[2].

An empirical demonstration of the democratic peace is also a good example of a **Time Series Cross Sectional** (or panel) dataset, where the same units (in this case countries) are observed repeatedly for multiple time periods. Experience in assembling and analyzing a TSCS dataset will prepare you for any future research in this area.

11.2 Setting up

```r
library(dplyr)
library(tidyr)
library(readr)
library(data.table)
library(foreach)
library(readxl)
library(ggplot2)
```

11.3 Create a project directory

First start a directory for this project. This can be done manually or through RStudio's Project feature(`File > New Project...`)

Directories is the computer science / programming name for folders. While advice about how to structure your working directories might strike you as petty, we believe that starting from some well-tested guides will go a long way in improving the quality and efficiency of your work.

Chapter 4 of Gentzkow and Shapiro's memo, Code and Data for the Social Scientist] provides a good template.

11.4 Data Sources

Most projects you do will start with downloading data from elsewhere. For this task, you'll probably want to track down and download the following:

- **Correlates of war dataset (COW):** Find and download the Militarized Interstate Disputes (MIDs) data from the Correlates of War website: `http://www.correlatesofwar.org/data-sets`. Or a dyad-version on dataverse: `https://dataverse.harvard.edu/dataset.xhtml?persistentId=hdl:1902.1/11489`

[2] The Kantian Peace: The Pacific Benefits of Democracy, Interdependence, and International Organizations, 1885-1992. *World Politics* 52(1):1-37

- **PRIO Data on Armed Conflict:** Find and download the Uppsala Conflict Data Program (UCDP) and PRIO dyad-year data on armed conflict(`https://www.prio.org`) or this link to to the flat csv file (`http://ucdp.uu.se/downloads/dyadic/ucdp-dyadic-171.csv`).
- **Polity:** The Polity data can be downloaded from their website (`http://www.systemicpeace.org/inscrdata.html`). Look for the newest version of the time series that has the widest coverage.

11.5 Example with 2 Datasets

Let's read in a sample dataset.

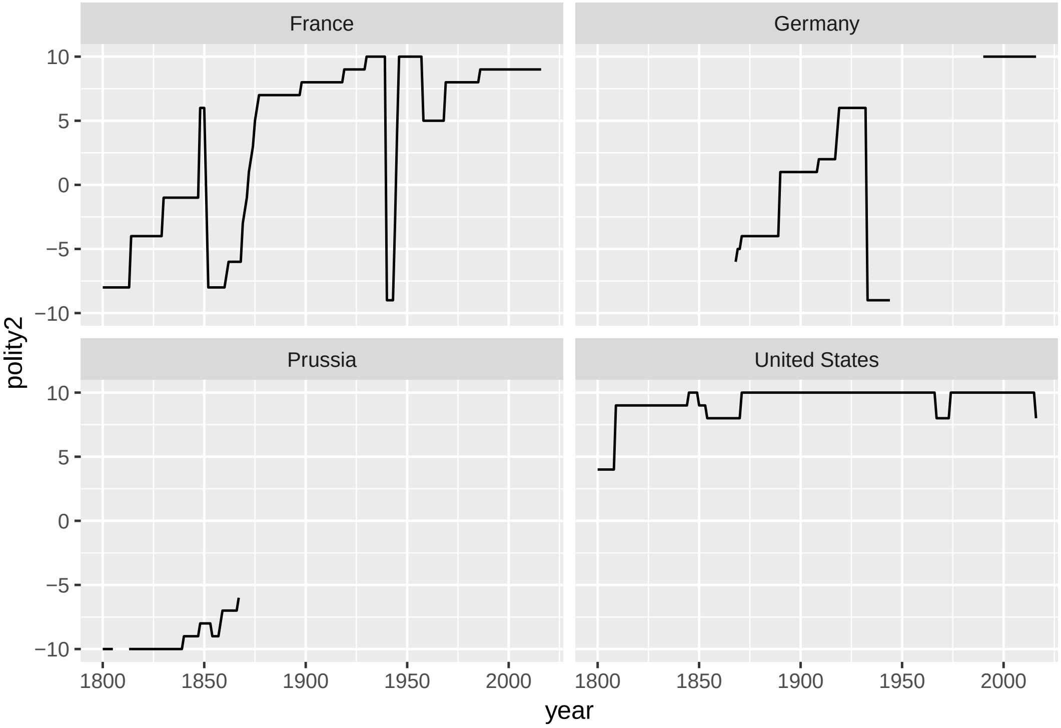
```
polity <- read_csv("input/sample_polity.csv")
mid <- read_csv("input/sample_mid.csv")
```

What does `polity` look like?

```
unique(polity$country)
```

```
## [1] "France"        "Prussia"       "Germany"       "United States"
```

```
ggplot(polity, aes(x = year, y = polity2)) +
  facet_wrap(~ country) +
  geom_line()
```



```
head(polity)
```

```
## # A tibble: 6 x 5
##   scode ccode country  year polity2
##   <chr> <int> <chr>   <dbl>   <int>
## 1 FRN     220 France   1800      -8
## 2 FRN     220 France   1801      -8
## 3 FRN     220 France   1802      -8
## 4 FRN     220 France   1803      -8
## 5 FRN     220 France   1804      -8
## 6 FRN     220 France   1805      -8
```

MID is a dataset that captures a `dispute` for a given country and year.

```
mid
```

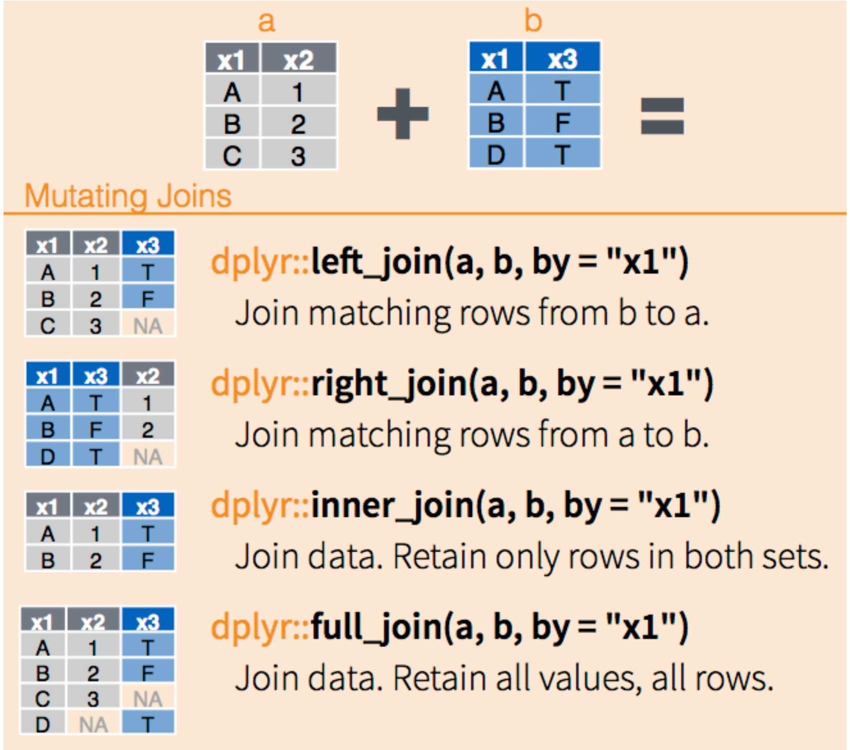
```
## # A tibble: 6,132 x 5
##    ccode polity_code dispute StYear EndYear
##    <int> <chr>         <int>  <int>   <int>
##  1   200 UKG               1   1902    1903
##  2     2 USA               1   1902    1903
##  3   345 YGS               1   1913    1913
##  4   300 <NA>              1   1913    1913
##  5   339 ALB               1   1946    1946
##  6   200 UKG               1   1946    1946
##  7   200 UKG               1   1951    1952
##  8   651 EGY               1   1951    1952
##  9   630 IRN               1   1856    1857
## 10   200 UKG               1   1856    1857
## # ... with 6,122 more rows
```

11.6 Loops

Notice that in the `mid` data, we have a start of a dispute vs. an end of a dispute.In order to combine this into the `polity` data, we want a way to give each of the interval years a row.

There are many ways to do this, but one is a loop. We go through one row at a time, and then for each we make a new dataset. that has `year` as a sequence of each year.

```
mid_year_by_year <- data_frame(ccode = numeric(),
                               year = numeric(),
                               dispute = numeric())

for(i in 1:nrow(mid)) {
  x <- data_frame(ccode = mid$ccode[i], ## row i's country
                  year = mid$StYear[i]:mid$EndYear[i],  ## sequence of years for dispute in row i
                  dispute = 1)
  mid_year_by_year <- rbind(mid_year_by_year, x)
```



a

x1	x2
A	1
B	2
C	3

b

x1	x3
A	T
B	F
D	T

Mutating Joins

x1	x2	x3
A	1	T
B	2	F
C	3	NA

dplyr::left_join(a, b, by = "x1")
Join matching rows from b to a.

x1	x3	x2
A	T	1
B	F	2
D	T	NA

dplyr::right_join(a, b, by = "x1")
Join matching rows from a to b.

x1	x2	x3
A	1	T
B	2	F

dplyr::inner_join(a, b, by = "x1")
Join data. Retain only rows in both sets.

x1	x2	x3
A	1	T
B	2	F
C	3	NA
D	NA	T

dplyr::full_join(a, b, by = "x1")
Join data. Retain all values, all rows.

Figure 11.1

```
}

head(mid_year_by_year)
```

```
## # A tibble: 6 x 3
##   ccode  year dispute
##   <int> <int>   <dbl>
## 1   200  1902       1
## 2   200  1903       1
## 3     2  1902       1
## 4     2  1903       1
## 5   345  1913       1
## 6   300  1913       1
```

11.7 Merging

We want to combine these two datasets by merging. Base-R has a function called `merge`. `dplyr` has several types of `joins` (the same thing). Those names are based on SQL syntax.

Here we can do a `left_join` matching rows from `mid` to `polity`. We want to keep the rows in polity that do not match in mid, and label them as non-disputes.

```
p_m <- left_join(polity,
                 distinct(mid_year_by_year),
                 by = c("ccode", "year"))

head(p_m)
```

```
## # A tibble: 6 x 6
##   scode ccode country  year polity2 dispute
##   <chr> <int> <chr>   <dbl>   <int>   <dbl>
## 1 FRN     220 France   1800      -8      NA
## 2 FRN     220 France   1801      -8      NA
## 3 FRN     220 France   1802      -8      NA
## 4 FRN     220 France   1803      -8      NA
## 5 FRN     220 France   1804      -8      NA
## 6 FRN     220 France   1805      -8      NA
```

Replace dispute = NA rows with a zero.

```
p_m$dispute[is.na(p_m$dispute)] <- 0
```

long to wide

```
p_m_wide <- dcast(data = p_m,
                  formula = ccode ~ year,
                  value.var = "polity2")
```

11.8 Main Project

Try building a panel that would be useful in answering the Democratic Peace Question, perhaps in these steps.

Task 1: Data Input and Standardization

Often, files we need are saved in the .xls or xlsx format. It is possible to read these files directly into R, but experience suggests that this process is slower than converting them first to .csv format and reading them in as .csv files.

readxl/readr/haven packages(https://github.com/tidyverse/tidyverse) is constantly expanding to capture more file types. In day 1, we used the package readxl, using the read_excel() function.

Task 2: Data Merging

We will use data to test a version of the Democratic Peace Thesis (DPS). Democracies are said to go to war less because the leaders who wage wars are accountable to voters who have to bear the costs of war. Are democracies less likely to engage in militarized interstate disputes?

To start, let's download and merge some data.

- Load in the Militarized Interstate Dispute (MID) files. Militarized interstate disputes are hostile action between two formally recognized states. Examples of this would be threats to use force, threats to declare war, beginning war, fortifying a border with troops, and so on.
- Find a way to **merge** the Polity IV dataset and the MID data. This process can be a bit tricky.
- An *advanced* version of this task would be to download the dyadic form of the data and try merging that with polity.

Task 3: Tabulations and Visualization

1. Calculate the mean Polity2 score by year. Plot the result. Use graphical indicators of your choosing to show where key events fall in this timeline (such as 1914, 1929, 1939, 1989, 2008). Speculate on why the behavior from 1800 to 1920 seems to be qualitatively different than behavior afterwards.
2. Do the same but only among state-years that were invovled in a MID. Plot this line together with your results from 1.
3. Do the same but only among state years that were *not* involved in a MID.
4. Arrive at a tentative conclusion for how well the Democratic Peace argument seems to hold up in this dataset. Visualize this conclusion.

Chapter 12

Simulation[1]

Where are we? Where are we headed?

Up till now, you should have covered:

- `R` basics
- Visualization
- Matrices and vectors
- Functions, objects, loops
- Joining real data

In this module, we will start to work with generating data within R, from thin air, as it were. Doing simulation also strengthens your understanding of Probability (Section @ref{probability}).

Check your Understanding

- What does the `sample()` function do?
- What does `runif()` stand for?
- What is a `seed`?
- What is a Monte Carlo?

Check if you have an idea of how you might code the following tasks:

- Simulate 100 rolls of a die
- Simulate one random ordering of 25 numbers
- Simulate 100 values of white noise (uniform random variables)
- Generate a "bootstrap" sample of an existing dataset

We're going to learn about this today!

[1] Module originally written by Connor Jerzak and Shiro Kuriwaki

12.1 Motivation: Simulation as an Analytical Tool

An increasing amount of political science contributions now include a simulation.

- Axelrod (1977) demonstrated via simulation how atomized individuals evolve to be grouped in similar clusters or countries, a model of culture.[2]
- Chen and Rodden (2013) argued in a 2013 article that the vote-seat inequality in U.S. elections that is often attributed to intentional partisan gerrymandering can actually attributed to simply the reality of "human geography" – Democratic voters tend to be concentrated in smaller area. Put another way, no feasible form of gerrymandering could spread out Democratic voters in such a way to equalize their vote-seat translation effectiveness. After demonstrating the empirical pattern of human geography, they advance their key claim by simulating thousands of redistricting plans and record the vote-seat ratio.[3]
- Gary King, James Honaker, and multiple other authors propose a way to analyze missing data with a method of multiple imputation, which uses a lot of simulation from a researcher's observed dataset.[4] (Software: Amelia[5])

Statistical methods also incorporate simulation:

- The bootstrap: a statistical method for estimating uncertainty around some parameter by re-sampling observations.
- Bagging: a method for improving machine learning predictions by re-sampling observations, storing the estimate across many re-samples, and averaging these estimates to form the final estimate. A variance reduction technique.
- Statistical reasoning: if you are trying to understand a quantitative problem, a wonderful first-step to understand the problem better is to simulate it! The analytical solution is often very hard (or impossible), but the simulation is often much easier :-)

12.2 Pick a sample, any sample

12.3 The `sample()` function

The core functions for coding up stochastic data revolves around several key functions, so we will simply review them here.

Suppose you have a vector of values `x` and from it you want to randomly sample a sample of length `size`. For this, use the `sample` function

```
sample(x = 1:10, size = 5)
```

```
## [1]  3  1  8 10  2
```

[2] Axelrod, Robert. 1997. "The Dissemination of Culture." *Journal of Conflict Resolution* 41(2): 203–26.

[3] Chen, Jowei, and Jonathan Rodden. "Unintentional Gerrymandering: Political Geography and Electoral Bias in Legislatures. *Quarterly Journal of Political Science*, 8:239-269"

[4] King, Gary, et al. "Analyzing Incomplete Political Science Data: An Alternative Algorithm for Multiple Imputation". *American Political Science Review*, 95: 49-69.

[5] James Honaker, Gary King, Matthew Blackwell (2011). Amelia II: A Program for Missing Data. Journal of Statistical Software, 45(7), 1-47.

There are two subtypes of sampling – with and without replacement.

1. Sampling without replacement (`replace = FALSE`) means once an element of `x` is chosen, it will not be considered again:

```
sample(x = 1:10, size = 10, replace = FALSE) ## no number appears more than once
```

```
##  [1]  4  5  9  2  3  7  1  6 10  8
```

2. Sampling with replacement (`replace = TRUE`) means that even if an element of `x` is chosen, it is put back in the pool and may be chosen again.

```
sample(x = 1:10, size = 10, replace = TRUE) ## any number can appear more than once
```

```
##  [1] 7 4 9 7 2 2 6 8 6 9
```

It follows then that you cannot sample without replacement a sample that is larger than the pool.

```
sample(x = 1:10, size = 100, replace = FALSE)
```

```
## Error in sample.int(length(x), size, replace, prob): cannot take a sample larger than the popu
```

So far, every element in `x` has had an equal probability of being chosen. In some application, we want a sampling scheme where some elements are more likely to be chosen than others. The argument `prob` handles this.

For example, this simulates 20 fair coin tosses (each outcome is equally likely to happen)

```
sample(c("Head", "Tail"), size = 20, prob = c(0.5, 0.5), replace = TRUE)
```

```
##  [1] "Tail" "Head" "Tail" "Tail" "Head" "Head" "Tail" "Head" "Head" "Head"
## [11] "Head" "Tail" "Tail" "Tail" "Head" "Head" "Tail" "Tail" "Head" "Tail"
```

But this simulates 20 biased coin tosses, where say the probability of Tails is 4 times more likely than the number of Heads

```
sample(c("Head", "Tail"), size = 20, prob = c(0.2, 0.8), replace = TRUE)
```

```
##  [1] "Tail" "Tail" "Tail" "Head" "Tail" "Tail" "Tail" "Head" "Tail" "Tail"
## [11] "Head" "Tail" "Tail" "Head" "Tail" "Head" "Head" "Head" "Tail" "Head"
```

12.3.1 Sampling rows from a dataframe

In tidyverse, there is a convenience function to sample rows randomly: `sample_n()` and `sample_frac()`.

For example, load the dataset on cars, `mtcars`, which has 32 observations.

```
mtcars
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

sample_n picks a user-specified number of rows from the dataset:

```
sample_n(mtcars, 3)
```

```
##                   mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Pontiac Firebird 19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Maserati Bora    15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Toyota Corolla   33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
```

Sometimes you want a X percent sample of your dataset. In this case use `sample_frac()`

```
sample_frac(mtcars, 0.10)
```

```
##                    mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## AMC Javelin       15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Merc 450SE        16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Chrysler Imperial 14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
```

As a side-note, these functions have very practical uses for any type of data analysis:

- Inspecting your dataset: using `head()` all the same time and looking over the first few rows might lead you to ignore any issues that end up in the bottom for whatever reason.

- Testing your analysis with a small sample: If running analyses on a dataset takes more than a handful of seconds, change your dataset upstream to a fraction of the size so the rest of the code runs in less than a second. Once verifying your analysis code runs, then re-do it with your full dataset (by simply removing the `sample_n` / `sample_frac` line of code in the beginning). While three seconds may not sound like much, they accumulate and eat up time.

12.4 Random numbers from specific distributions

`rbinom()`

`rbinom` builds upon `sample` as a tool to help you answer the question – what is the *total number of successes* I would get if I sampled a binary (Bernoulli) result from a test with `size` number of trials each, with a event-wise probability of `prob`. The first argument `n` asks me how many such numbers I want.

For example, I want to know how many Heads I would get if I flipped a fair coin 100 times.

```
rbinom(n = 1, size = 100, prob = 0.5)
```

```
## [1] 46
```

Now imagine this I wanted to do this experiment 10 times, which would require I flip the coin 10 x 100 = 1000 times! Helpfully, we can do this in one line

```
rbinom(n = 10, size = 100, prob = 0.5)
```

```
##  [1] 42 47 41 51 52 57 44 49 53 45
```

`runif()`

`runif` also simulates a stochastic scheme where each event has equal probability of getting chosen like `sample`, but is a continuous rather than discrete system. We will cover this more in the next math module.

The intuition to emphasize here is that one can generate potentially infinite amounts (size `n`) of noise that is a essentially random

```
runif(n = 5)
```

```
## [1] 0.04820592 0.18489989 0.54155194 0.83747694 0.97833988
```

`rnorm()`

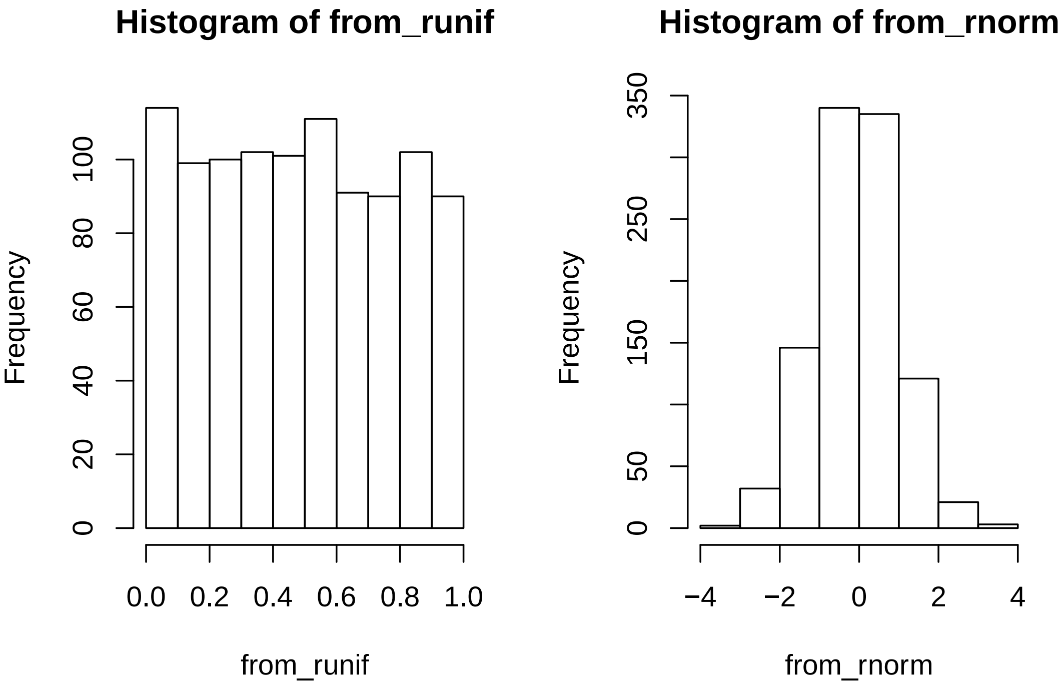
`rnorm` is also a continuous distribution, but draws from a Normal distribution – perhaps the most important distribution in statistics. It runs the same way as `runif`

```
rnorm(n = 5)
```

```
## [1]  0.94704868  0.05728521  0.32914573 -2.90922109  0.08422519
```

To better visualize the difference between the output of `runif` and `rnorm`, let's generate lots of each and plot a histogram.

```
from_runif <- runif(n = 1000)
from_rnorm <- rnorm(n = 1000)

par(mfrow = c(1, 2)) ## base-R parameter for two plots at once
hist(from_runif)
hist(from_rnorm)
```



12.5 r, p, and d

Each distribution can do more than generate random numbers (the prefix `r`). We can compute the cumulative probability by the function `pbinom()`, `punif()`, and `pnorm()`. Also the density – the value of the PDF – by `dbinom()`, `dunif()` and `dnorm()`.

12.6 `set.seed()`

`R` doesn't have the ability to generate truly random numbers! Random numbers are actually very hard to generate. (Think: flipping a coin –> can be perfectly predicted if I know wind speed, the angle the coin is flipped, etc.). Some people use random noise in the atmosphere or random behavior in quantum systems to generate "truly" (?) random numbers. Conversely, R uses deterministic algorithms which take as an input a "seed" and which then perform a series of operations to generate a sequence of random-seeming numbers (that is, numbers whose sequence is sufficiently hard to predict).

Let's think about this another way. Sampling is a stochastic process, so every time you run `sample()` or `runif()` you are bound to get a different output (because different random seeds are used). This is intentional in some cases but you might want to avoid it in others. For example, you might want to diagnose a coding discrepancy by setting the random number generator to give the same number each time. To do this, use the function `set.seed()`.

In the function goes any number. When you run a sample function in the same command as a preceding `set.seed()`, the sampling function will always give you the same sequence of numbers. In a sense, the sampler is no longer random (in the sense of unpredictable to use; remember: it never was "truly" random in the first place)

```r
set.seed(02138)
runif(n = 10)
```

```
##  [1] 0.51236144 0.61530551 0.37451441 0.43541258 0.21166530 0.17812129
##  [7] 0.04420775 0.45567854 0.88718264 0.06970056
```

The random number generator should give you the exact same sequence of numbers if you precede the function by the same seed,

```r
set.seed(02138)
runif(n = 10)
```

```
##  [1] 0.51236144 0.61530551 0.37451441 0.43541258 0.21166530 0.17812129
##  [7] 0.04420775 0.45567854 0.88718264 0.06970056
```

whereas a true random number generator would give you the exact same sequence of output with probability 0!

Exercises

Census Sampling

What can we learn from surveys of populations, and how wrong do we get if our sampling is biased?[6] Suppose we want to estimate the proportion of U.S. residents who are non-white (`race != "White"`). In reality, we do not have any population dataset to utilize and so we *only see the sample survey*. Here, however, to understand how sampling works, let's conveniently use the Census extract in some cases and pretend we didn't in others.

(a) First, load `usc2010_001percent.csv` into your R session. After loading the `library(tidyverse)`, browse it. Although this is only a 0.01 percent extract, treat

[6] This example is inspired from Meng, Xiao-Li (2018). Statistical paradises and paradoxes in big data (I): Law of large populations, big data paradox, and the 2016 US presidential election. *Annals of Applied Statistics* 12:2, 685–726. doi:10.1214/18-AOAS1161SF.

this as your population for pedagogical purposes. What is the population proportion of non-White residents?

(b) Setting a seed to `1669482`, sample 100 respondents from this sample. What is the proportion of non-White residents in this *particular* sample? By how many percentage points are you off from (what we labelled as) the true proportion?

(c) Now imagine what you did above was one survey. What would we get if we did 20 surveys?

To simulate this, write a loop that does the same exercise 20 times, each time computing a sample proportion. Use the same seed at the top, but be careful to position the `set.seed` function such that it generates the same sequence of 20 samples, rather than 20 of the same sample.

Try doing this with a `for` loop and storing your sample proportions in a new length-20 vector. (Suggestion: make an empty vector first as a container). After running the loop, show a histogram of the 20 values. Also what is the average of the 20 sample estimates?

(d) Now, to make things more real, let's introduce some response bias. The goal here is not to correct response bias but to induce it and see how it affects our estimates. Suppose that non-White residents are 10 percent less likely to respond to enter your survey than White respondents. This is plausible if you think that the Census is from 2010 but you are polling in 2018, and racial minorities are more geographically mobile than Whites. Repeat the same exercise in (c) by modeling this behavior.

You can do this by creating a variable, e.g. `propensity`, that is 0.9 for non-Whites and 1 otherwise. Then, you can refer to it in the propensity argument.

(e) Finally, we want to see if more data ("Big Data") will improve our estimates. Using the same unequal response rates framework as (d), repeat the same exercise but instead of each poll collecting 100 responses, we collect 10,000.

(f) Optional - visualize your 2 pairs of 20 estimates, with a bar showing the "correct" population average.

Conditional Proportions

This example is not on simulation, but is meant to reinforce some of the probability discussion from math lecture.

Read in the Upshot Siena poll from Fall 2016, `input/upshot-siena-polls.csv`.

In addition to some standard demographic questions, we will focus on one called `vt_pres_2` in the csv. This is a two-way presidential vote question, asking respondents who they plan to vote for President if the election were held today – Donald Trump, the Republican, or Hilary Clinton, the Democrat, with options for Other candidates as well. For this problem, use the two-way vote question rather than the 4-way vote question.

(a) Drop the the respondents who answered the November poll (i.e. those for which `poll == "November"`). We do this in order to ignore this November population in all subsequent parts of this question because they were not asked the Presidential vote question.

(b) Using the dataset after the procedure in (a), find the proportion of *poll respondents* (those who are in the sample) who support Donald Trump.

(c) Among those who supported Donald Trump, what proportion of them has a Bachelor's degree or higher (i.e. have a Bachelor's, Graduate, or other Professional Degree)?

(d) Among those who did not support Donald Trump (i.e. including supporters of Hilary Clinton, another candidate, or those who refused to answer the question), what proportion of them has a Bachelor's degree or higher?

(e) Express the numbers in the previous parts as probabilities of specified events. Define your own symbols: For example, we can let T be the event that a randomly selected respondent in the poll supports Donald Trump, then the proportion in part (b) is the probability $P(T)$.

(f) Suppose we randomly sampled a person who participated in the survey and found that he/she had a Bachelor's degree or higher. Given this evidence, what is the probability that the same person supports Donald Trump? Use Bayes Rule and show your work – that is, do not use data or R to compute the quantity directly. Then, verify this is the case via R.

The Birthday problem

Write code that will answer the well-known birthday problem via simulation.[7]

The problem is fairly simple: Suppose k people gather together in a room. What is the probability at least two people share the same birthday?

To simplify reality a bit, assume that (1) there are no leap years, and so there are always 365 days in a year, and (2) a given individual's birthday is randomly assigned and independent from each other.

Step 1: Set `k` to a concrete number. Pick a number from 1 to 365 randomly, `k` times to simulate birthdays (would this be with replacement or without?).

```
# Your code
```

Step 2: Write a line (or two) of code that gives a `TRUE` or `FALSE` statement of whether or not at least two people share the same birth date.

```
# Your code
```

Step 3: The above steps will generate a `TRUE` or `FALSE` answer for your event of interest, but only for one realization of an event in the sample space. In order to estimate the *probability* of your event happening, we need a "stochastic", as opposed to "deterministic", method. To do this, write a loop that does Steps 1 and 2 repeatedly for many times, call that number of times `sims`. For each of `sims` iteration, your code should give you a `TRUE` or `FALSE` answer. Code up a way to store these estimates.

```
# Your code
```

[7] This exercise draws from Imai (2017)

Step 4: Finally, generalize the function further by letting `k` be a user-defined number. You have now created a *Monte Carlo simulation*!

```
# Your code
```

Step 5: Generate a table or plot that shows how the probability of sharing a birthday changes by `k` (fixing `sims` at a large number like `1000`). Also generate a similar plot that shows how the probability of sharing a birthday changes by `sims` (fixing `k` at some arbitrary number like `10`).

```
# Your code
```

Extra credit: Give an "analytical" answer to this problem, that is an answer through deriving the mathematical expressions of the probability.

```
# Your equations
```

Chapter 13

LaTeX and markdown[1]

Where are we? Where are we headed?

Up till now, you should have covered:

- Statistical Programming in `R`

This is only the beginning of `R` – programming is like learning a language, so learn more as we use it. And yet `R` is of likely not the only programming language you will want to use. While we cannot introduce everything, we'll pick out a few that we think are particularly helpful.

Here will cover

- Markdown
- LaTeX (and BibTeX)

as examples of a non-WYSIWYG editor

and the next chapter (you can read it without reading this LaTeX chapter) covers

- command-line
- git

command-line are a basic set of tools that you may have to use from time to time. It also clarifies what more complicated programs are doing. Markdown is an example of compiling a plain text file. LaTeX is a typesetting program and git is a version control program – both are useful for non-quantitative work as well.

Check your understanding

Check if you have an idea of how you might code the following tasks:

- What does "WYSIWYG" stand for? How would a non-WYSIWYG format text?
- How do you start a header in markdown?

[1] Module originally written by Shiro Kuriwaki

- What are some “plain text” editors?
- How do you start a document in `.tex`?
- How do you start a environment in `.tex`?
- How do you insert a figure in `.tex`?
- How do you reference a figure in `.tex`?
- What is a `.bib` file?
- Say you came across a interesting journal article. How would you want to maintain this reference so that you can refer to its citation in all your subsequent papers?

13.1 Motivation

Statistical programming is a fast-moving field. The beta version of `R` was released in 2000, `ggplot2` was released on 2005, and `RStudio` started around 2010. Of course, some programming technologies are quite “old”: (`C` in 1969, `C++` around 1989, `TeX` in 1978, `Linux` in 1991, Mac OS in 1984). But it is easy to feel you are falling behind in the recent developments of programming. Today we will do a **brief** and rough overview of some fundamental and new tools other than `R`, with the general aim of having you break out of your comfort zone so you won’t be shut out from learning these tools in the future.

13.2 Markdown

Markdown is the text we have been using throughout this course! At its core markdown is just plain text. Plain text does not have any formatting embedded in it. Instead, the formatting is coded up as text. Markdown is *not* a WYSIWYG (What you see is what you get) text editor like Microsoft Word or Google Docs. This will mean that you need to explicitly code for `bold{text}` rather than hitting Command+B and making your text look **bold** on your own computer.

Markdown is known as a “light-weight” editor, which means that it is relatively easy to write code that will compile. It is quick and easy and satisfies most presentation purposes; you might want to try `LaTeX` for more involved papers.

13.2.1 markdown commands

For italic and bold, use either the asterisks or the underlines,

```
*italic*   **bold**
_italic_   __bold__
```

And for headers use the hash symbols,

```
# Main Header
## Sub-headers
```

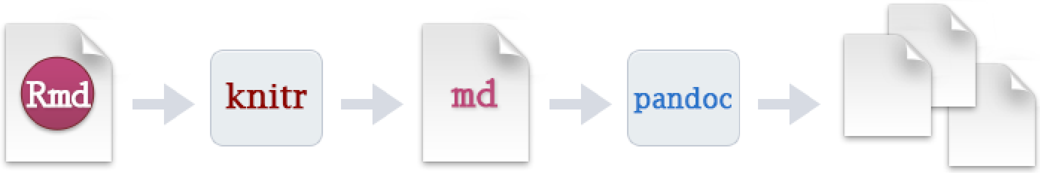


Figure 13.1: How Rmds become PDFs or HTMLs

13.2.2 your own markdown

RStudio makes it easy to compile your very first markdown file by giving you templates. Got to `New > R Markdown`, pick a document and click Ok. This will give you a skeleton of a document you can compile – or "knit".

Rmd is actually a slight modification of real markdown. It is a type of file that R reads and turns into a proper `md` file. Then, it uses a document-conversion called pandoc to compile your `md` into documents like PDF or HTML.

13.2.3 A note on plain-text editors

Multiple software exist where you can edit plain-text (roughly speaking, text that is not WYSIWYG).

- RStudio (especially for R-related links)
- TeXMaker, TeXShop (especially for TeX)
- emacs, aquamacs (general)
- vim (general)
- Sublime Text (general)

Each has their own keyboard shortcuts and special features. You can browse a couple and see which one(s) you like.

13.3 LaTeX

LaTeX is a typesetting program. You'd engage with LaTeX much like you engage with your `R` code. You will interact with LaTeX in a text editor, and will writing code which will be interpreted by the LaTeX compiler and which will finally be parsed to form your final PDF.

13.3.1 compile online

1. Go to `https://www.overleaf.com`
2. Scroll down and go to "CREATE A NEW PAPER" if you don't have an account.
3. Let's discuss the default template.
4. Make a new document, and set it as your main document. Then type in the Minimal Working Example (MWE):

```
\documentclass{article}
\begin{document}
Hello World
\end{document}
```

13.3.2 compile your first LaTeX document locally

LaTeX is a very stable system, and few changes to it have been made since the 1990s. The main benefit: better control over how your papers will look; better methods for writing equations or making tables; overall pleasing aesthetic.

1. Open a plain text editor. Then type in the MWE

```
\documentclass{article}
\begin{document}
Hello World
\end{document}
```

2. Save this as `hello_world.tex`. Make sure you get the file extension right.
3. Open this in your "LaTeX" editor. This can be `TeXMaker`, `Aqumacs`, etc..
4. Go through the click/dropdown interface and click compile.

13.3.3 main LaTeX commands

LaTeX can cover most of your typesetting needs, to clean equations and intricate diagrams.

Some main commands you'll be using are below, and a very concise cheat sheet here: `https://wch.github.io/latexsheet/latexsheet.pdf`

Most involved features require that you begin a specific "environment" for that feature, clearly demarcating them by the notation `\begin{figure}` and then `\end{figure}`, e.g. in the case of figures.

```
\begin{figure}
\includegraphics{histogram.pdf}
\end{figure}
```

where `histogram.pdf` is a path to one of your files.

Notice that each line starts with a backslash `\` – in LaTeX this is the symbol to run a command.

The following syntax at the endpoints are shorthand for math equations.

```
\[\int x^2 dx\]
```

these compile math symbols: $\int x^2 dx$.[2]

The `align` environment is useful to align your multi-line math, for example.

[2] Enclosing with `$$` instead of `\[` also has the same effect, so you may see it too. But this is now discouraged due to its inflexibility.

```
\begin{align}
P(A \mid B) &= \frac{P(A \cap B)}{P(B)}\\
&= \frac{P(B \mid A)P(A)}{P(B)}
\end{align}
```

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} \tag{13.1}$$

$$= \frac{P(B \mid A)P(A)}{P(B)} \tag{13.2}$$

Regression tables should be outputted as `.tex` files with packages like `xtable` and `stargazer`, and then called into LaTeX by `\input{regression_table.tex}` where `regression_table.tex` is the path to your regression output.

Figures and equations should be labelled with the tag (e.g. `label{tab:regression}` so that you can refer to them later with their tag `Table \ref{tab:regression}`, instead of hard-coding `Table 2`).

For some LaTeX commands you might need to load a separate package that someone else has written. Do this in your preamble (i.e. before `\begin{document}`):

```
\usepackage[options]{package}
```

where `package` is the name of the package and `options` are options specific to the package.

Further Guides

For a more comprehensive listing of LaTeX commands, Mayya Komisarchik has a great tutorial set of folders: `https://scholar.harvard.edu/mkomisarchik/tutorials-0`

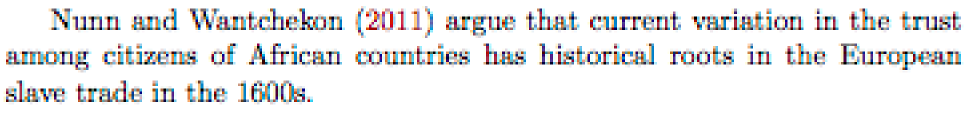
There is a version of LaTeX called Beamer, which is a popular way of making a slideshow. Slides in markdown is also a competitor. The language of Beamer is the same as LaTeX but has some special functions for slides.

13.4 BibTeX

BibTeX is a reference system for bibliographical tests. We have a `.bib` file separately on our computer. This is also a plain text file, but it encodes bibliographical resources with special syntax so that a program can rearrange parts accordingly for different citation systems.

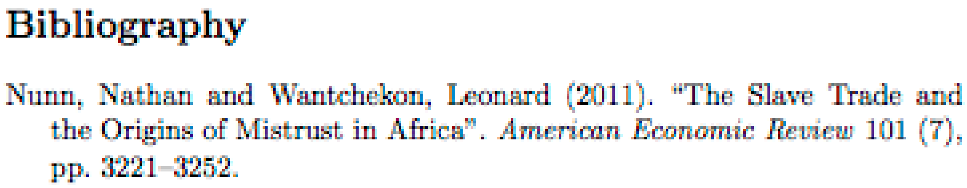
13.4.1 what is a `.bib` file?

For example, here is the Nunn and Wantchekon article entry in `.bib` form.



Nunn and Wantchekon (2011) argue that current variation in the trust among citizens of African countries has historical roots in the European slave trade in the 1600s.

Figure 13.2



Bibliography

Nunn, Nathan and Wantchekon, Leonard (2011). "The Slave Trade and the Origins of Mistrust in Africa". *American Economic Review* 101 (7), pp. 3221–3252.

Figure 13.3

```
@article{nunn2011slave,
  title={The Slave Trade and the Origins of Mistrust in Africa},
  author={Nunn, Nathan and Wantchekon, Leonard},
  journal={American Economic Review},
  volume={101},
  number={7},
  pages={3221--3252},
  year={2011}
}
```

The first entry, `nunn2011slave`, is "pick your favorite" – pick your own name for your reference system. The other slots in this `@article` entry are entries that refer to specific bibliographical text.

13.4.2 what does LaTeX do with .bib files?

Now, in LaTeX, if you type

```
  \textcite{nunn2011slave} argue that current variation in the trust among citizens of African co
```

as part of your text, then when the `.tex` file is compiled the PDF shows something like

in whatever citation style (APSA, APA, Chicago) you pre-specified!

Also at the end of your paper you will have a bibliography with entries ordered and formatted in the appropriate citation.

This is a much less frustrating way of keeping track of your references – no need to hand-edit formatting the bibliography to conform to citation rules (which biblatex already knows) and no need to update your bibliography as you add and drop references (biblatex will only show entries that are used in the main text).

13.4.3 stocking up on your .bib files

You should keep your own `.bib` file that has all your bibliographical resources. Storing entries is cheap (does not take much memory), so it is fine to keep all your references in one place (but you'll want to make a new one for collaborative projects where multiple people will compile a `.tex` file).

For example, Gary's BibTeX file is here: `https://github.com/iqss-research/gkbibtex/blob/master/gk.bib`

Citation management software (Mendeley or Zotero) automatically generates .bib entries from your library of PDFs for you, provided you have the bibliography attributes right.

Exercise

Create a LaTeX document for a hypothetical research paper on your laptop and, once you've verified it compiles into a PDF, come show it to either one of the instructors.

You can also use overleaf if you have preference for a cloud-based system. But don't swallow the built-in templates without understanding or testing them.

Each student will have slightly different substantive interests, so we won't impose much of a standard. But at a minimum, the LaTeX document should have:

- A title, author, date, and abstract
- Sections
- Italics and boldface
- A figure with a caption and in-text reference to it.

Depending on your subfield or interests, try to implement some of the following:

- A bibliographical reference drawing from a separate `.bib` file
- A table
- A math expression
- A different font
- Different page margins
- Different line spacing

Concluding the Prefresher

Math may not be the perfect tool for every aspiring political scientist, but hopefully it was useful background to have at the least:

> Historians think this totally meaningless and nonsensical statistic is the product of an early-modern epistemological shift in which numbers and quantifiable data became revered above other kinds of knowledge as the most useful and credible form of truth https://t.co/wVFyAQGxEv

— Gina Anne Tam (?) May 29, 2018

But we should be aware that too much slant towards math and programming can miss the point:

To be clear, PhD training in Econ (first year) is often a disaster– like how to prove the Central Limit Theorem (the LeBron James of Statistics) with polar-cooardinates. This is mostly a way to demoralize actual economists and select a bunch of unimaginative math jocks.

— Amitabh Chandra (**?**) August 14, 2018

Keep on learning, trying new techniques to improve your work, and learn from others!

What #rstats tricks did it take you way too long to learn? One of mine is using readRDS and saveRDS instead of repeatedly loading from CSV

— Emily Riederer (**?**) August 19, 2017

Your Feedback Matters

Please tell us how we can improve the Prefresher: The Prefresher is a work in progress, with material mainly driven by graduate students. Please tell us how we should change (or not change) each of its elements:

`https://harvard.az1.qualtrics.com/jfe/form/SV_esbzN8ZFAOPTqiV`

Chapter 14

Text[1]

Where are we? Where are we headed?

Up till now, you should have covered:

- Loading in data;
- `R` notation;
- Matrix algebra.

14.1 Review

- `"` and `'` are usually equivalent.
- `<-` and `=` are usually interchangeable[2]. (`x <- 3` is equivalent to `x = 3`, although the former is more preferred because it explicitly states the assignment).
- Use `( )` when you are giving input to a function:

```
# my_results <- FunctionName(FunctionInputs)
```

```
note  `c(1,2,3)` is inputting three numbers in the function `c`
```

- Use `{ }` when you are defining a function or writing a `for` loop:

```
#function
MyFunction <- function(InputMatrix){
  TempMat <- InputMatrix
  for(i in 1:5){
    TempMat <- t(TempMat)  %*% TempMat / 10
  }
  return( TempMat )
}
```

[1] Module originally written by Connor Jerzak

[2] Only equal signs are allowed to define the values of a functions' argument

```
myMat <- matrix(rnorm(100*5), nrow = 100, ncol = 5)
print( MyFunction(myMat) )
```

```
##            [,1]      [,2]       [,3]       [,4]      [,5]
## [1,]  342.3602  196.1668   856.7638  -732.7517  173.1954
## [2,]  196.1668  515.3176   762.8554  -277.1625  299.6710
## [3,]  856.7638  762.8554  2697.1230 -1868.8323  461.6741
## [4,] -732.7517 -277.1625 -1868.8323  1678.3580 -264.6936
## [5,]  173.1954  299.6710   461.6741  -264.6936  219.0823
```

```
# loop
x <- c()
for(i in 1:20){
  x[i] <- i
}
print(x)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

14.2 Goals for today

Today, we will learn more about using text data. Our objectives are:

- Reading and writing in text in R.
- To learn how to use paste and sprintf;
- To learn how to use regular expressions;
- To learn about other tools for representing + analyzing text in R.

14.3 Reading and writing text in R

- To read in a text file, use readLines

```
readLines("~/Downloads/Carboxylic acid - Wikipedia.html")
```

- To write a text file, use:

```
write.table(my_string_vector, "~/mydata.txt", sep="\t")
```

14.4 paste() and sprintf()

paste and sprintf are useful commands in text processing, such as for automatically naming files or automatically performing a series of command over a subset of your data. Table making also will often need these commands.

Paste concatenates vectors together.

```
#use collapse for inputs of length > 1
my_string <- c("Not", "one", "could", "equal")
paste(my_string, collapse = " ")
```

```
## [1] "Not one could equal"
```

```
#use sep for inputs of length == 1
paste("Not", "one", "could", "equal", sep = " ")
```

```
## [1] "Not one could equal"
```

For more sophisticated concatenation, use sprintf. This is very useful for automatically making tables.

```
sprintf("Coefficient for %s: %.3f (%.2f)", "Gender", 1.52324, 0.03143)
```

```
## [1] "Coefficient for Gender: 1.523 (0.03)"
```

```
#%s is replaced by a character string
#%.3f is replaced by a floating point digit with 3 decimal places
#%.2f is replaced by a floating point digit with 2 decimal places
```

14.5 Regular expressions

A regular expression is a special text string for describing a search pattern. They are most often used in functions for detecting, locating, and replacing desired text in a corpus.

Use cases:

1. TEXT PARSING. E.g. I have 10000 congressional speaches. Find all those which mention Iran.
2. WEB SCRAPING. E.g. Parse html code in order to extract research information from an online table.
3. CLEANING DATA. E.g. After loading in a dataset, we might need to remove mistakes from the dataset, orsubset the data using regular expression tools.

Example in `R`. Extract the tweet mentioning Indonesia.

```
s1 <- "If only Bradley's arm was longer. RT"
s2 <- "Share our love in Indonesia and in the World. RT if you agree."
my_string <- c(s1, s2)
grepl(my_string, pattern = "Indonesia")
```

```
## [1] FALSE  TRUE
```

```
my_string[ grepl(my_string, pattern = "Indonesia")]
```

```
## [1] "Share our love in Indonesia and in the World. RT if you agree."
```

Key point: Many R commands use regular expressions. See `?grepl`. Assume that `x` is a character vector and that `pattern` is the target pattern. In the earlier example, `x` could have been something like `my_string` and `pattern` would have been "`Indonesia`". Here are other key uses:

1. DETECT PATTERNS. `grepl(pattern, x)` goes through all the entries of `x` and returns a string of TRUE and FALSE values of the same size as `x`. It will return a `TRUE` whenever that string entry has the target pattern, and `FALSE` whenever it doesn't.

2. REPLACE PATTERNS. `gsub(pattern, x, replacement)` goes through all the entries of `x` replaces the `pattern` with `replacement`.

```
gsub(x = my_string,
     pattern = "o",
     replacement = "AAAA")
```

```
## [1] "If AAAAnly Bradley's arm was lAAAAnger. RT"
## [2] "Share AAAAur lAAAAve in IndAAAAnesia and in the WAAAArld. RT if yAAAAu agree."
```

3. LOCATE PATTERNS. `regexpr(pattern, text)` goes through each element of the character string. It returns a vector of the same length, with the entries of the vector corresponding to the location of the first pattern match, or a -1 if no match was obtained.

```
regex_object <- regexpr(pattern = "was",  text = my_string)
attr(regex_object, "match.length")
```

```
## [1]  3 -1
```

```
attr(regex_object, "useBytes")
```

```
## [1] TRUE
```

```
regexpr(pattern = "was", text = my_string)[1]
```

```
## [1] 23
```

```
regexpr(pattern = "was", text = my_string)[2]
```

```
## [1] -1
```

Seems simple? The problem: the patterns can get pretty complex!

14.5.1 Character classes

Some types of symbols are stand in for some more complex thing, rather than taken literally.

`[[:digit:]]` Matches with all digits.

`[[:lower:]]` Matches with lower case letters.

`[[:alpha:]]` Matches with all alphabetic characters.

`[[:punct:]]` Matches with all punctuation characters.

`[[:cntrl:]]` Matches with "control" characters such as `\n`, `\r`, etc.

Example in `R`:

```
my_string <- "Do you think that 34% of apples are red?"
gsub(my_string, pattern = "[[:digit:]]", replace ="DIGIT")
```

```
## [1] "Do you think that DIGITDIGIT% of apples are red?"
```

```
gsub(my_string, pattern = "[[:alpha:]]", replace ="")
```

```
## [1] "    34%    ?"
```

14.5.2 Special Characters.

Certain characters (such as `.`, `*`, `\`) have special meaning in the regular expressions framework (they are used to form conditional patterns as discussed below). Thus, when we want our pattern to explicitly include those characters as characters, we must "escape" them by using \ or encoding them in \Q…\E.

Example in `R`:

```
my_string <- "Do *really* think he will win?"
gsub(my_string, pattern = "\\*", replace ="")
```

```
## [1] "Do really think he will win?"
```

```
my_string <- "Now be brave! \n Dread what comrades say of you here in combat! "
gsub(my_string, pattern = "\\\n", replace ="")
```

```
## [1] "Now be brave!  Dread what comrades say of you here in combat! "
```

14.5.3 Conditional patterns

`[]` The target characters to match are located between the brackets. For example, `[aAbB]` will match with the characters `a, A, b, B`.

`[^...]` Matches with everything except the material between the brackets. For example, `[^aAbB]` will match with everything but the characters `a, A, b, B`.

`(?=)` Lookahead – match something that IS followed by the pattern.

`(?!)` Negative lookahead — match something that is NOT followed by the pattern.

`(?<=)` Lookbehind – match with something that follows the pattern.

```
my_string <- "Do you think that 34%of the 23%of apples are red?"
gsub(my_string, pattern = "(?<=%)", replace = " ", perl = TRUE)
```

```
## [1] "Do you think that 34% of the 23% of apples are red?"
```

```
my_string <- c("legislative1_term1.png",
               "legislative1_term1.pdf",
               "legislative1_term2.png",
               "legislative1_term2.pdf",
               "term2_presidential1.png",
               "presidential1.png",
               "presidential1_term2.png",
               "presidential1_term1.pdf",
               "presidential1_term2.pdf")

grepl(my_string, pattern = "^(?!presidential1).*\\.png", perl = TRUE)
```

```
## [1]  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
```

- Indicates which file names don't start with `presidential1` but do end in `.png`
- `^` indicates that the pattern should start at the beginning of the string.
- `?!` indicates negative lookahead – we're looking for any pattern NOT following presidential1 which meets the subsequent conditions. (see below)
- The first `.` indicates that, following the negative lookahead, there can be any characters and the * says that it doesn't matter how many. Note that we have to escape the . in `.png`. (by writing `\\.` instead of just `.`)

You will have the chance to try out some regular expressions for yourself at the end!

14.6 Representing Text

In courses and research, we often want to analyze text, to extract meaning out of it. One of the key decisions we need to make is how to represent the text as numbers. Once the text is represented numerically, we can then apply a host of statistical and machine learning methods to it. Those methods are discussed more in the Gov methods sequence (Gov 2000-2003). Here's a summary of the decisions you must make:

1. WHICH TEXT TO USE? Which text do I want to analyze? What is my universe of documents?
2. HOW TO REPRESENT THE TEXT NUMERICALLY? How do I use numbers to represent different things about the text?
3. HOW TO ANALYZE THE NUMERICAL REPRESENTATION? How do I extract meaning out of the numerical representation?

Representing text numerically.

1. Document term matrix. The document term matrix (DTM) is a common method for representing text. The DTM is a matrix. Each row of this matrix corresponds to a document; each column corresponds to a word. It is often useful to look at summary statistics such as the percentage of speaches in which a Democratic lawmaker used the word "inequality" compared to a Republican; the DTM would be very helpful for this and other tasks.

```
doc1 <- "Rage---Goddess, sing the rage of Peleus' son Achilles,
         murderous, doomed, that cost the Achaeans countless losses,
         hurling down to the House of Death so many sturdy souls,
         great fighters' souls."
doc2 <- "And fate? No one alive has ever escaped it,
         neither brave man nor coward, I tell you,
         it's born with us the day that we are born."
doc3 <- "Many cities of men he saw and learned their minds,
         many pains he suffered, heartsick on the open sea,
         fighting to save his life and bring his comrades home."
```

```
DocVec <- c(doc1, doc2, doc3)
```

Now we can use utility functions in the `tm` package:

```
library(tm)
DocCorpus <- Corpus(VectorSource(DocVec) )
DTM1 <-  inspect( DocumentTermMatrix(DocCorpus) )
```

Consider the effect of different “pre-processing” choices on the resulting DTM!

```
DocVec <- tolower(DocVec)
DocVec <- gsub(DocVec, pattern ="[[:punct:]]", replace = " ")
DocVec <- gsub(DocVec, pattern ="[[:cntrl:]]", replace = " ")
DocCorpus <- Corpus(VectorSource(DocVec) )
DTM2 <-  inspect(DocumentTermMatrix(DocCorpus,
                                    control = list(stopwords = TRUE,  stemming = TRUE)))
```

Stemming is the process of reducing inflected/derived words to their word stem or base (e.g. stemming, stemmed, stemmer –> stem*)

14.7 Important packages for parsing text

1. rvest – Useful for downloading and manipulating HTML and XM.
2. tm – Useful for converting text into a numerical representation (forming DTMs).
3. stringr – Useful for string parsing.

Exercises

1

Figure out why this command does what it does:

```
sprintf("%s of spontaneous events are %s in the mind.
        Really, %.2f?",
        "15.03322123", "puzzles", 15.03322123)
```

```
## [1] "15.03322123 of spontaneous events are puzzles in the mind. \n        Really, 15.03?"
```

2

Why does this command not work?

```r
try(sprintf("%s of spontaneous events are %s in the mind. Really, %.2f?",
            "15.03322123", "puzzles", "15.03322123" ), TRUE)
```

3

Using grepl, these materials, Google, and your friends, describe what the following command does. What changes when `value = FALSE`?

```r
grep('\'',
     c("To dare is to lose one's footing momentarily.",  "To not dare is to lose oneself."), valu
```

```
## [1] "To dare is to lose one's footing momentarily."
```

4

Write code to automatically extract the file names that DO end start with presidential and DO end in .pdf

```r
my_string <- c("legislative1_term1.png",
               "legislative1_term1.pdf",
               "legislative1_term2.png",
               "legislative1_term2.pdf",
               "term2_presidential1.png",
               "presidential1.png",
               "presidential1_term2.png",
               "presidential1_term1.pdf",
               "presidential1_term2.pdf")
```

5

Using the same string as in the above, write code to automatically extract the file names that end in .pdf and that contain the text `term2`.

```r
# Your code here
```

6

Combine these two strings into a single string separated by a "-". Desired output: "The carbonyl group in aldehydes and ketones is an oxygen analog of the carbon–carbon double bond."

```r
string1 <- "The carbonyl group in aldehydes and ketones
            is an oxygen analog of the carbon"
string2 <-  "-carbon double bond."
```

7

Challenge problem! Download this webpage `https://en.wikipedia.org/wiki/Odyssey`

- Read the html file into your R workspace.
- Remove all of the htlm tags (you may need Google to help with this one).
- Remove all punctuation.
- Make all the characters lower case.
- Do this same process with this webpage (`https://en.wikipedia.org/wiki/Iliad`).
- Form a document term matrix from the two resulting text strings.

```r
# Your code here
```

Chapter 15

Command-line, git[1]

15.1 Where are we? Where are we headed?

Up till now, you should have covered:

- Statistical Programming in `R`

In conjunction with the markdown/LaTeX chapter, which is mostly used for typesetting and presentation, here we'll introduce the command-line and git, more used for software extensions and version control

15.2 Check your understanding

Check if you have an idea of how you might code the following tasks:

- What is a GUI?
- What do the following commands stand for in shell: `ls` (or `dir` in Windows), `cd`, `rm`, `mv` (or `move` in windows), `cp` (or `copy` in Windows).
- What is the difference between a relative path and an absolute path?
- What paths do these refer to in shell/terminal: `~/`, `.`, `..`
- What is a *repository* in github?
- What does it mean to "clone" a repository?

15.3 command-line

Elementary programming operations are done on the command-line, or by entering commands into your computer. This is different from a UI or GUI – graphical user-interface – which are interfaces that allow you to click buttons and enter commands in more readable

[1] Module originally written by Shiro Kuriwaki

form. Although there are good enough GUIs for most of your needs, you still might need to go under the hood sometimes and run a command.

15.3.1 command-line commands

Open up `Terminal` in a Mac. (`Command Prompt` in Windows)

Running this command in a Mac (`dir` in Windows) should show you a list of all files in the directory that you are currently in.

```
ls
```

```
## 01_warmup.Rmd
## 02_linear-algebra.Rmd
## 03_functions.Rmd
## 04_limits.Rmd
## 05_calculus.Rmd
## 06_optimization.Rmd
## 07_probability.Rmd
## 11_data-handling_counting.Rmd
## 12_matricies-manipulation.Rmd
## 13_visualization.Rmd
## 14_functions_obj_loops.Rmd
## 15_project-dempeace.Rmd
## 16_simulation.Rmd
## 17_non-wysiwyg.Rmd
## 18_text.Rmd
## 19_command-line_git.Rmd
## 21_solutions-warmup.Rmd
## 23_solution_programming.Rmd
## CODE_OF_CONDUCT.md
## CONTRIBUTING.md
## Is America Headed for a New Kind of Civil War? | The New Yorker.pdf
## LICENSE
## README.md
## _book
## _bookdown.yml
## _bookdown_files
## _output.yml
## book.bib
## images
## index.Rmd
## input
## preamble.tex
## prefresher.Rmd
## prefresher.Rproj
## prefresher.bbl
## prefresher.blg
## prefresher_files
## rsconnect
## sample_library.bib
## style.css
## survey.R
```

`pwd` stands for present working directory (`cd` in Windows)

```
pwd
```

```
## /Users/shirokuriwaki/Dropbox/prefresher
```

`cd` means change directory. You need to give it what to change your current directory *to*. You can specify a name of another directory in your directory.

Or you can go up to your parent directory. The syntax for that are two periods, `..` . One period `.` refers to the current directory.

```
cd ..
pwd
```

```
## /Users/shirokuriwaki/Dropbox
```

`~/` stands for your home directory defined by your computer.

```
cd ~/
ls
```

```
## Applications
## Candidate Data File for 2018 Statewide General Election.xlsx
## Candidate Data File for 20180611_20578_11_6_2018 Statewide General Election.csv
## Desktop
## Documents
## Downloads
## Dropbox
## Google Drive File Stream
## Library
## Movies
## Music
## PaladinTemp
## Pictures
## Public
## my_directory_structure.txt
```

Using `..` and `.` are "relative" to where you are currently at. So are things like `figures/figure1.pdf`, which is implicitly writing `./figures/figure1.pdf`. These are called relative paths. In contrast, `/Users/shirokuriwaki/project1/figures/figure1.pdf` is an "absolute" path because it does not start from your current directory.

Relative paths are nice if you have a shared Dropbox, for example, and I had `/Users/shirokuriwaki/mathcamp` but Connor's path to the same folder is `/Users/connorjerzak/mathcamp`. To run the same code in `mathcamp`, we should be using relative paths that start from "`mathcamp`". Relative paths are also shorter, and they are invariant to higher-level changes in your computer.

15.3.2 running things via command-line

Suppose you have a simple Rscript, call it `hello_world.R`. This is simply a plain text file that contains

```
cat("Hello World")
```

Then in command-line, go to the directory that contains `hello_world.R` and enter

```
Rscript hello_world.R
```

This should give you the output `Hello World`, which verifies that you "executed" the file with R via the command-line.

15.3.3 why do command-line?

If you know exactly what you want to do your files and the changes are local, then command-line might be faster and be more sensible than navigating yourself through a GUI. For example, what if you wanted a single command that will run 10 R scripts successively at once (as Gentzkow and Shapiro suggest you should do in your research)? It is tedious to run each of your scripts on Rstudio, especially if running some take more than a few minutes. Instead you could write a "batch" script that you can run on the terminal,

```
Rscript 01_read_data.R
Rscript 02_merge_data.R
Rscript 03_run_regressions.R
Rscript 04_make_graphs.R
Rscript 05_maketable.R
```

Then run this single file, call it `run_all_Rscripts.sh`, on your terminal as

```
sh run_all_Rscripts.sh
```

On the other hand, command-line prompts may require more keystrokes, and is also less intuitive than a good GUI. It can also be dangerous for beginners, because it can allow you to make large irreversible changes inadvertently. For example, removing a file (`rm`) has no "Undo" feature.

15.4 git

Git is a tool for version control. It comes pre-installed on Macs, you will probably need to install it yourself on Windows.

15.4.1 why version control?

All version control software should be built to

- preserve all snapshots of your work

- and catalog them in such a way that you can refer back or even revert back your files to the past snapshot.
- makes it easy to see exactly which parts of your files you changed between directories.

Further, git is most commonly used for collaborative work.

- maintains "branches", or parallel universes of your files that people can switch back and forth on, doing version control on each one
- makes it easy to "merge" a sub-branch to a master branch when it is ready.

Note that Dropbox is useful for collaborative work too. But the added value of git's branches is that people can make different changes simultaneously on their computers and merge them to the master branch later. In Dropbox, there is only one copy of each thing so simultaneous editing is not possible.

15.4.2 open-source code at your fingertips

Some links to check out:

- `https://github.com/tidyverse/dplyr`
- `https://github.com/apple/swift`
- `https://github.com/kosukeimai/qss`

GitHub `https://github.com` is the GUI to git. Making an account there is free. Making an account will allow you to be a part of the collaborative programming community. It will also allow you to "fork" other people's "repositories". "Forking" is making your own copy of the project that forks off from the master project at a point in time. A "repository" is simply the name of your main project directory.

"cloning" someone else's repository is similar to forking – it gives you your own copy.

15.4.3 commands in git

As you might have noticed from all the quoted terms, git uses a lot of its own terms that are not intuitive and hard to remember at first. The nuts and bolts of maintaining your version control further requires "adding", "committing", and "push"ing, sometimes "pull"ing.

The tutorial `https://try.github.io/` is quite good. You'd want to have familiarity with command-line to fully understand this and use it in your work.

RStudio Projects has a great git GUI as well.

15.4.4 is git worth it?

While git is a powerful tool, you may choose to not use it for everything because

- git is mainly for code, not data. It has a fairly strict limit on the size of your dataset that you cover.
- your collaborators might want to work with Dropbox
- unless you get a paid account, all your repositories will be public.

Part III

Solutions

Solutions to Warmup Questions

Linear Algebra

Vectors

Define the vectors $u = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$, $v = \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix}$, and the scalar $c = 2$.

1. $u + v = \begin{pmatrix} 5 \\ 7 \\ 9 \end{pmatrix}$
2. $cv = \begin{pmatrix} 8 \\ 10 \\ 12 \end{pmatrix}$
3. $u \cdot v = 1(4) + 2(5) + 3(6) = 32$

If you are having trouble with these problems, please review Section 1.1 “Working with Vectors” in Chapter 1.

Are the following sets of vectors linearly independent?

1. $u = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$, $v = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$

$\rightsquigarrow$ No:

$$2u = \begin{pmatrix} 2 \\ 4 \end{pmatrix}, v = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$$

so infinitely many linear combinations of u and v that amount to 0 exist.

2. $u = \begin{pmatrix} 1 \\ 2 \\ 5 \end{pmatrix}$, $v = \begin{pmatrix} 3 \\ 7 \\ 9 \end{pmatrix}$

$\rightsquigarrow$ Yes: we cannot find linear combination of these two vectors that would amount to zero.

3. $a = \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix}$, $b = \begin{pmatrix} 3 \\ -4 \\ -2 \end{pmatrix}$, $c = \begin{pmatrix} 5 \\ -10 \\ -8 \end{pmatrix}$

$\rightsquigarrow$ No: After playing around with some numbers, we can find that

$$-2a = \begin{pmatrix} -4 \\ 2 \\ -2 \end{pmatrix}, 3b = \begin{pmatrix} 9 \\ -12 \\ -6 \end{pmatrix}, -1c = \begin{pmatrix} -5 \\ 10 \\ 8 \end{pmatrix}$$

So

$$-2a + 3b - c = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

i.e., a linear combination of these three vectors that would amount to zero exists.

If you are having trouble with these problems, please review Section 1.2.

Matrices

$$\mathbf{A} = \begin{pmatrix} 7 & 5 & 1 \\ 11 & 9 & 3 \\ 2 & 14 & 21 \\ 4 & 1 & 5 \end{pmatrix}$$

What is the dimensionality of matrix **A**? 4 × 3

What is the element a_{23} of **A**? 3

Given that

$$\mathbf{B} = \begin{pmatrix} 1 & 2 & 8 \\ 3 & 9 & 11 \\ 4 & 7 & 5 \\ 5 & 1 & 9 \end{pmatrix}$$

$$\mathbf{A} + \mathbf{B} = \begin{pmatrix} 8 & 7 & 9 \\ 14 & 18 & 14 \\ 6 & 21 & 26 \\ 9 & 2 & 14 \end{pmatrix}$$

Given that

$$\mathbf{C} = \begin{pmatrix} 1 & 2 & 8 \\ 3 & 9 & 11 \\ 4 & 7 & 5 \end{pmatrix}$$

$$\mathbf{A} + \mathbf{C} = \text{No solution, matrices non-conformable}$$

Given that

$$c = 2$$

$$c\mathbf{A} = \begin{pmatrix} 14 & 10 & 2 \\ 22 & 18 & 6 \\ 4 & 28 & 42 \\ 8 & 2 & 10 \end{pmatrix}$$

If you are having trouble with these problems, please review Section 1.3.

Operations

Summation

Simplify the following

1. $\sum_{i=1}^{3} i = 1 + 2 + 3 = 6$
2. $\sum_{k=1}^{3} (3k + 2) = 3 \sum_{k=1}^{3} k + \sum_{k=1}^{3} 2 = 3 \times 6 + 3 \times 2 = 24$
3. $\sum_{i=1}^{4} (3k + i + 2) = 3 \sum_{i=1}^{4} k + \sum_{i=1}^{4} i + \sum_{i=1}^{4} 2 = 12k + 10 + 8 = 12k + 18$

Products

1. $\prod_{i=1}^{3} i = 1 \cdot 2 \cdot 3 = 6$
2. $\prod_{k=1}^{3} (3k + 2) = (3 + 2) \cdot (6 + 2) \cdot (9 + 2) = 440$

To review this material, please see Section 2.1.

Logs and exponents

Simplify the following

1. $4^2 = 16$
2. $4^2 2^3 = 2^{2 \cdot 2} 2^3 = 2^{4+3} = 128$
3. $\log_{10} 100 = \log_{10} 10^2 = 2$
4. $\log_2 4 = \log_2 2^2 = 2$
5. when log is the natural log, $\log e = \log_e e^1 = 1$
6. when a, b, c are each constants, $e^a e^b e^c = e^{a+b+c}$,
7. $\log 0 =$ undefined – no exponentiation of anything will result in a 0.

8. $e^0 = 1$ – any number raised to the 0 is always 1.
9. $e^1 = e$ – any number raised to the 1 is always itself
10. $\log e^2 = \log_e e^2 = 2$

To review this material, please see Section 2.3

Limits

Find the limit of the following.

1. $\lim_{x\to 2}(x-1) = 1$
2. $\lim_{x\to 2} \frac{(x-2)(x-1)}{(x-2)} = 1$, though note that the original function $\frac{(x-2)(x-1)}{(x-2)}$ would have been undefined at $x = 2$ because of a divide by zero problem; otherwise it would have been equal to $x - 1$.
3. $\lim_{x\to 2} \frac{x^2-3x+2}{x-2} = 1$, same as above.

To review this material please see Section 3.3

Calculus

For each of the following functions $f(x)$, find the derivative $f'(x)$ or $\frac{d}{dx}f(x)$

1. $f(x) = c$, $f'(x) = 0$
2. $f(x) = x$, $f'(x) = 1$
3. $f(x) = x^2$, $f'(x) = 2x$
4. $f(x) = x^3$, $f'(x) = 3x^2$
5. $f(x) = 3x^2 + 2x^{1/3}$, $f'(x) = 6x + \frac{2}{3}x^{-2/3}$
6. $f(x) = (x^3)(2x^4)$, $f'(x) = \frac{d}{dx}2x^7 = 14x^6$

For a review, please see Section 4.1 - 4.2

Optimization

For each of the followng functions $f(x)$, does a maximum and minimum exist in the domain $x \in \mathbf{R}$? If so, for what are those values and for which values of x?

1. $f(x) = x \rightsquigarrow$ neither exists.
2. $f(x) = x^2 \rightsquigarrow$ a minimum $f(x) = 0$ exists at $x = 0$, but not a maximum.
3. $f(x) = -(x-2)^2 \rightsquigarrow$ a maximum $f(x) = 0$ exists at $x = 2$, but not a minimum.

If you are stuck, please try sketching out a picture of each of the functions.

Probability

1. If there are 12 cards, numbered 1 to 12, and 4 cards are chosen, $\binom{12}{4} = \frac{12\cdot 11\cdot 10\cdot 9}{4!} = 495$ possible hands exist (unordered, without replacement) .

2. Let $A = \{1, 3, 5, 7, 8\}$ and $B = \{2, 4, 7, 8, 12, 13\}$. Then $A \cup B = \{1, 2, 3, 4, 5, 7, 8, 12, 13\}$, $A \cap B = \{7, 8\}$? If A is a subset of the Sample Space $S = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, then the complement $A^C = \{2, 4, 6, 9, 10\}$

3. If we roll two fair dice, what is the probability that their sum would be 11? $\rightsquigarrow \frac{1}{18}$

4. If we roll two fair dice, what is the probability that their sum would be 12? $\rightsquigarrow \frac{1}{36}$. There are two independent dice, so $6^2 = 36$ options in total. While the previous question had two possibilities for a sum of 11 (5,6 and 6,5), there is only one possibility out of 36 for a sum of 12 (6,6).

For a review, please see Sections 6.2 - 6.3

Suggested Programming Solutions

```
library(tidyverse)
library(ggrepel)
library(forcats)
library(scales)
```

15.5 Chapter 9: Visualization

1 State Proportions

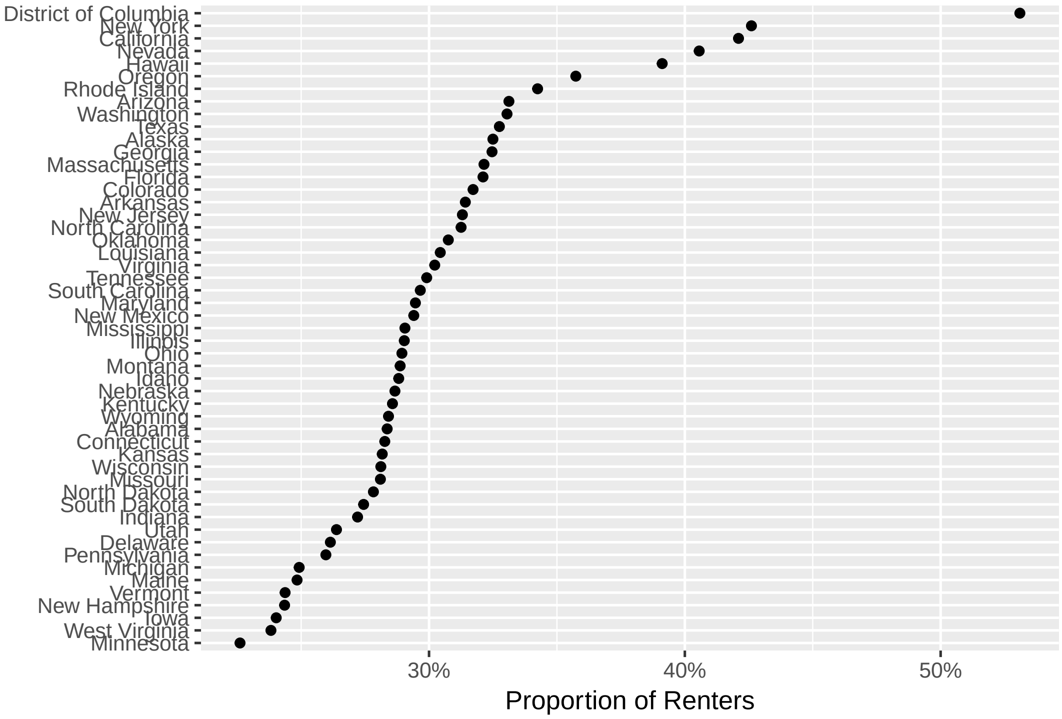
```
cen10 <- readRDS("input/usc2010_1percent.Rds")
```

Group by state, noting that the mean of a set of logicals is a mean of 1s (`TRUE`) and 0s (`FALSE`).

```
grp_st <- cen10 %>%
  group_by(state) %>%
  summarize(prop = mean(ownership == "Rented")) %>%
  arrange(prop) %>%
  mutate(state = as_factor(state))
```

Plot points

```
ggplot(grp_st, aes(x = state, y = prop)) +
  geom_point() +
  coord_flip() +
  scale_y_continuous(labels = percent) + # use the scales package to format percentages
  labs(
    y = "Proportion of Renters",
    x = "",
    caption = "Source: 2010 Census (1 percent)"
  )
```



Source: 2010 Census (1 percent)

2 Swing Justice

```
justices <- read_csv("input/justices_court-median.csv")
```

Keep justices who are in the dataset in 2016,

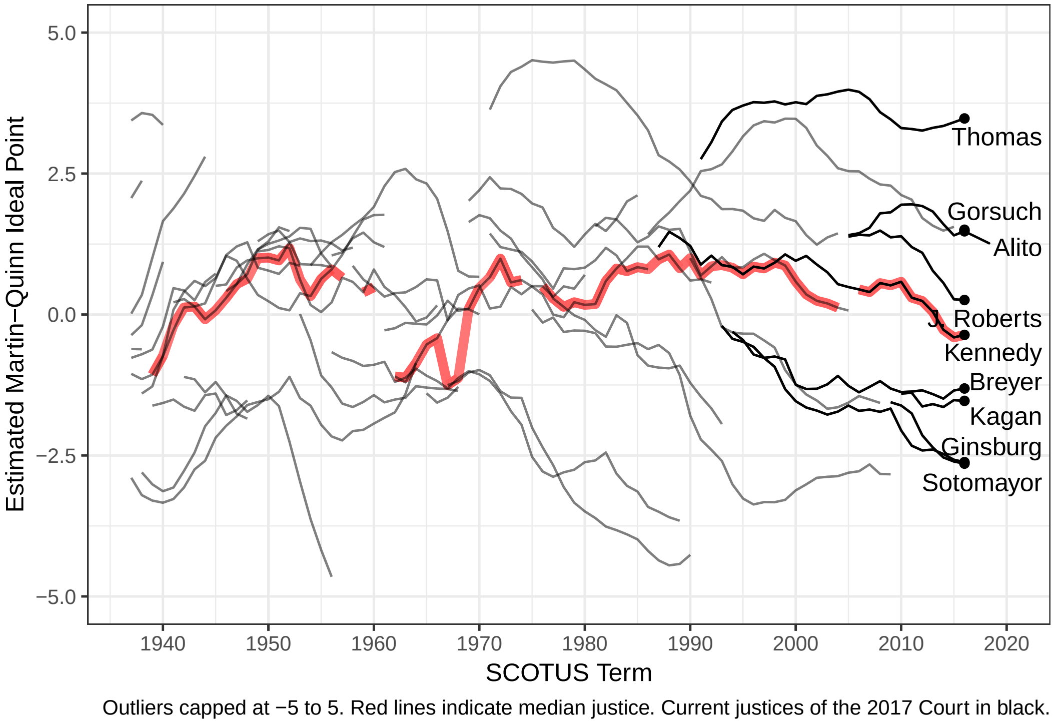
```
in_2017 <- justices %>%
  filter(term == 2016) %>%
  distinct(justice) %>% # unique values
  mutate(present_2016 = 1) # keep an indicator to distinguish from rest after merge

df_indicator <- justices %>%
  left_join(in_2017)
```

```
## Joining, by = "justice"
```

All together

```
ggplot(df_indicator, aes(x = term, y = idealpt, group = justice_id)) +
  geom_line(aes(y = median_idealpt), color = "red", size = 2, alpha = 0.1) +
  geom_line(alpha = 0.5) +
  geom_line(data = filter(df_indicator, !is.na(present_2016))) +
  geom_point(data = filter(df_indicator, !is.na(present_2016), term == 2016)) +
  geom_text_repel(
    data = filter(df_indicator, term == 2016), aes(label = justice),
    nudge_x = 10,
    direction = "y"
  ) + # labels nudged and vertical
  scale_x_continuous(breaks = seq(1940, 2020, 10), limits = c(1937, 2020)) + # axis breaks
  scale_y_continuous(limits = c(-5, 5)) + # axis limits
  labs(
    x = "SCOTUS Term",
    y = "Estimated Martin-Quinn Ideal Point",
    caption = "Outliers capped at -5 to 5. Red lines indicate median justice. Current justices o
  ) +
  theme_bw()
```

```
## Warning: Removed 30 rows containing missing values (geom_path).

## Warning: Removed 19 rows containing missing values (geom_path).
```



15.6 Chapter 10: Objects and Loops

```
cen10 <- read_csv("input/usc2010_001percent.csv")
sample_acs <- read_csv("input/acs2015_1percent.csv")
```

Checkpoint #3

```
cen10 %>%
  group_by(city) %>%
  summarise(avg_age = mean(age)) %>%
  arrange(desc(avg_age)) %>%
  slice(1:10)
```

```
## # A tibble: 10 x 2
##    city                 avg_age
##    <chr>                  <dbl>
##  1 Sterling Heights, MI    53.2
##  2 Waterbury, CT           49.3
##  3 Antioch, CA             49
##  4 Little Rock, AR         46.9
##  5 Roseville, CA           43.3
##  6 Port St. Lucie, FL      42.8
##  7 Huntington Beach, CA    42.3
##  8 Pittsburgh, PA          42.2
##  9 Cambridge, MA           41.8
## 10 Alexandria, VA          41.8
```

Exercise 1

```
colnames(sample_acs)
```

```
##  [1] "serial"            "pernum"            "hhwt"
##  [4] "perwt"             "state"             "county_identified"
##  [7] "puma"              "city"              "sex"
## [10] "age"               "birthyr"           "race"
## [13] "hispan"            "educ"              "citizen"
## [16] "yrnatur"
```

```
unique(sample_acs$citizen)
```

```
## [1] "Born in the US"
## [2] "US citizen by naturalization"
## [3] "Not a citizen of the US"
## [4] "Born abroad of American parent(s)"
## [5] "Born in Puerto Rico, Guam, the US Virgin Islands,or the Northern Marianas"
```

```
mean(sample_acs$citizen != "Not a citizen of the US")
```

```
## [1] 0.9419765
```

Exercise 3

```
states_of_interest <- c("California", "Massachusetts", "New Hampshire", "Washington")

for (state_i in states_of_interest) {
  state_subset <- cen10 %>% filter(state == state_i)

  print(state_i)

  print(table(state_subset$race, state_subset$sex))
}
```

```
## [1] "California"
##
##                                     Female Male
##   American Indian or Alaska Native      21   21
##   Black/Negro                          127  126
##   Chinese                               76   65
##   Japanese                              15   12
##   Other Asian or Pacific Islander      182  177
##   Other race, nec                      283  302
##   Three or more major races              7    7
##   Two major races                       91   83
##   White                               1085 1083
## [1] "Massachusetts"
##
##                                     Female Male
##   American Indian or Alaska Native       4    1
##   Black/Negro                           21   17
##   Chinese                                8    7
##   Japanese                               1    1
##   Other Asian or Pacific Islander       14   14
##   Other race, nec                        9   17
##   Two major races                       10    8
##   White                                272  243
## [1] "New Hampshire"
##
##                                     Female Male
##   American Indian or Alaska Native       1    0
##   Black/Negro                            0    1
##   Chinese                                0    1
##   Japanese                               1    0
##   Other Asian or Pacific Islander        2    1
##   Other race, nec                        1    0
##   Two major races                        0    1
##   White                                 66   63
## [1] "Washington"
##
```

```
##                                    Female Male
##   American Indian or Alaska Native      9    5
##   Black/Negro                          11    9
##   Chinese                               2    7
##   Japanese                              4    0
##   Other Asian or Pacific Islander      28   18
##   Other race, nec                      19   18
##   Three or more major races             0    2
##   Two major races                      17   16
##   White                               267  257
```

Exercise 4

```
race_d <- c()
state_d <- c()
proportion_d <- c()
answer <- data.frame(race_d, state_d, proportion_d)
```

Then

```
for (state in states_of_interest) {
  for (race in unique(cen10$race)) {
    race_state_num <- nrow(cen10[cen10$race == race & cen10$state == state, ])
    state_pop <- nrow(cen10[cen10$state == state, ])
    race_perc <- round(100 * (race_state_num / (state_pop)), digits = 2)
    line <- data.frame(race_d = race, state_d = state, proportion_d = race_perc)
    answer <- rbind(answer, line)
  }
}
```

15.7 Chapter 11: Demoratic Peace Project

Task 1: Data Input and Standardization

```
mid_b <- read_csv("input/MIDB_4.2.csv")
polity <- read_excel("input/p4v2017.xls")
```

Task 2: Data Merging

```
mid_y_by_y <- data_frame(ccode = numeric(),
                         year = numeric(),
                         dispute = numeric())
colnames(mid_b)
```

```
for(i in 1:nrow(mid_b)) {
    x <- data_frame(ccode = mid_b$ccode[i], ## row i's country
    year = mid_b$styear[i]:mid_b$endyear[i],  ## sequence of years for dispute in row i
    dispute = 1)## there was a dispute
    mid_y_by_y <- rbind(mid_y_by_y, x)
}

merged_mid_polity <- left_join(polity,
                 distinct(mid_y_by_y),
                 by = c("ccode", "year"))
```

Task 3: Tabulations and Visualization

```
#don't include the -88, -77, -66 values in calculating the mean of polity
mean_polity_by_year <- merged_mid_polity %>% group_by(year) %>% summarise(mean_polity = mean(poli

mean_polity_by_year_ordered <- arrange(mean_polity_by_year, year)

mean_polity_by_year_mid <- merged_mid_polity %>% group_by(year, dispute) %>% summarise(mean_polit

mean_polity_by_year_mid_ordered <- arrange(mean_polity_by_year_mid, year)

mean_polity_no_mid <- mean_polity_by_year_mid_ordered %>% filter(dispute == 0)
mean_polity_yes_mid <- mean_polity_by_year_mid_ordered %>% filter(dispute == 1)


answer <- ggplot(data = mean_polity_by_year_ordered, aes(x = year, y = mean_polity)) +
  geom_line() +
  labs(y = "Mean Polity Score",
       x = "") +
  geom_vline(xintercept = c(1914, 1929, 1939, 1989, 2008), linetype = "dashed")

answer + geom_line(data =mean_polity_no_mid, aes(x = year, y = mean_polity_mid), col = "indianred
```

15.8 Chapter 12: Simulation

15.8.1 Census Sampling

```
pop <- read_csv("input/usc2010_001percent.csv")
```

```
## Parsed with column specification:
## cols(
##   year = col_integer(),
##   serial = col_integer(),
##   pernum = col_integer(),
##   region = col_character(),
##   state = col_character(),
##   countyfips = col_integer(),
##   city = col_character(),
##   cpuma0010 = col_integer(),
##   sex = col_character(),
##   age = col_integer(),
##   race = col_character(),
##   hhtype = col_character(),
##   relate = col_character()
## )
```

```
mean(pop$race != "White")
```

```
## [1] 0.2806517
```

```
set.seed(1669482)
samp <- sample_n(pop, 100)
mean(samp$race != "White")
```

```
## [1] 0.31
```

```
ests <- c()
set.seed(1669482)

for (i in 1:20) {
  samp <- sample_n(pop, 100)
  ests[i] <- mean(samp$race != "White")
}


mean(ests)
```

```
pop_with_prop <- mutate(pop, propensity = ifelse(race != "White", 0.9, 1))
```

```
ests <- c()
set.seed(1669482)

for (i in 1:20) {
  samp <- sample_n(pop_with_prop, 100, weight = propensity)
  ests[i] <- mean(samp$race != "White")
}

mean(ests)
```

```
ests <- c()
set.seed(1669482)

for (i in 1:20) {
  samp <- sample_n(pop_with_prop, 10000, weight = propensity)
  ests[i] <- mean(samp$race != "White")
}

mean(ests)
```