Introduction to Applied Science

Rex W. Douglass

11/4/22

Table of contents

1 Preface

This is a Quarto book.

To learn more about Quarto books visit https://quarto.org/docs/books.

```
1 + 1
```

```
[1] 2
```

Part I

Introduction

2 Introduction

This is a book created from markdown and executable code.

See (**knuth84?**) for additional discussion of literate programming.

```
1 + 1
```

```
[1] 2
```

3 Modeling Literature

Bayesian Workflow Andrew Gelman, Aki Vehtari, Daniel Simpson, Charles C. Margossian, Bob Carpenter, Yuling Yao, Lauren Kennedy, Jonah Gabry, Paul-Christian Bürkner, Martin Modrák https://arxiv.org/abs/2011.01808

How to avoid machine learning pitfalls: a guide for academic researchers Michael A. Lones https://arxiv.org/abs/2108.02497

Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure David R. Roberts,Volker Bahn,Simone Ciuti,Mark S. Boyce,Jane Elith,Gurutzeta Guillera-Arroita,Severin Hauenstein,José J. Lahoz-Monfort,Boris Schröder,Wilfried Thuiller,David I. Warton,Brendan A. Wintle,Florian Hartig,Carsten F. Dormann https://onlinelibrary.wiley.com/doi/full/10.1111/ecog.02881

Information geometry and divergences https://franknielsen.github.io/IG/#bookIG

Statistical Rethinking: A Bayesian Course with Examples in R and Stan (& PyMC3 & brms) https://xcelab.net/rm/statistical-rethinking/ https://www.youtube.com/playlist?list=PLDcUM9US4XdMROZOIRtIK0aOynbgZN

ML Frameworks Interoperability Cheat Sheet http://bl.ocks.org/miguelusque/raw/f44a8e729896a96d0a3e4b07b5

Regression and Other Stories, Andrew Gelman, Jennifer Hill, Aki Vehtari

tidybayes: Bayesian analysis + tidy data + geoms

http://www.gradaanwr.net/

Data Visualization A practical introduction, Kieran Healy

The garden of forking paths: Why multiple comparisons can be a problem, even when there is no "fishing expedition" or "p-hacking" and the research hypothesis was posited ahead of time∗, Andrew Gelman† and Eric Loken

Part II

Computation

4 Computation

4.1 Python

https://docs.python.org/3/

4.1.1 Numpy

https://numpy.org/

4.1.2 Pandas

https://pandas.pydata.org/docs/

Effective Pandas https://store.metasnake.com/effective-pandas-book

4.2 jax

https://github.com/google/jax

4.3 Numpyro

https://github.com/pyro-ppl/numpyro

4.4 SQL

postgresql https://www.postgresql.org/docs/

4.5 git

https://git-scm.com/doc

4.6 Stan

https://mc-stan.org/users/documentation/

4.7 brms

https://github.com/paul-buerkner/brms

4.8 pyro

https://pyro.ai/

The StatQuest Introduction to PyTorch

4.9 tensorflow

https://www.tensorflow.org/

5 R

https://www.r-project.org/other-docs.html

Hands-On Programming with R, Garrett Grolemund

5.0.1 Tidyverse

https://www.tidyverse.org/

R for Data Science

The Tidyverse Cookbook

Part III

Research Design

6 Domain

CHANNELLING FISHER: RANDOMIZATION TESTS AND THE STATISTICAL INSIGNIFICANCE OF SEEMINGLY SIGNIFICANT EXPERIMENTAL RESULTS

An Automatic Finite-Sample Robustness Metric: When Can Dropping a Little Data Make a Big Difference?

7 Outliers

An Automatic Finite-Sample Robustness Metric: When Can Dropping a Little Data Make a Big Difference?

8 Estimand

What Is Your Estimand? Defining the Target Quantity Connects Statistical Evidence to Theory

9

Partial Identification in Econometrics

10 Random Control Trials

Evaluating the replicability of social science experiments in Nature and Science between 2010 and 2015

CHANNELLING FISHER: RANDOMIZATION TESTS AND THE STATISTICAL INSIGNIFICANCE OF SEEMINGLY SIGNIFICANT EXPERIMENTAL RESULTS

11 Instrumental Varibales

How Much Should We Trust Instrumental Variable Estimates in Political Science? Practical Advice Based on Over 60 Replicated Studies

12 Difference in Difference

How Much Should We Trust Differences-In-Differences Estimates?

How Much Should We Trust Staggered Difference-In-Differences Estimates?

Part IV

Estimation

Part V

Mathematical Objects

13 Set

Cites: Wikipedia; Wikidata; PlanetMath

14 List (Sequence)

AKA: Sequence, a_n where n is the nth element, (1,2,3, ….)

Distinct from: Set

Measure of:

Description: A list is a collection of objects with a specific ordering and where the same object can appear more than once. Call each object an element, and its location its index or rank. An index is a natural number counting upward from the first element in the list. Whether counting begins at 0 or 1 depends on local conventions.

Formalization:

Algorithm:

Cites: Wikipedia Wikidata Encyclopedia Of Math Wolfram PlanetMath

14.0.0.1 R

Documentation:

list: Lists – Generic and Dotted Pairs

Examples:

```
example_list = list(1,2,3)
example_list
```

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

[[2]]
[1] 2

[[3]]
[1] 3
```

14.0.0.2 Python

Documentation:

More on Lists

Examples:

```
example_list =  [1,2,3]
example_list
```

```
[1, 2, 3]
```

14.0.0.3 SQL

```
library(DBI)
# Create an ephemeral in-memory RSQLite database
con <- dbConnect(RSQLite::SQLite(), dbname = ":memory:")
dbListTables(con)
```

```
character(0)
```

```
dbWriteTable(con, "mtcars", mtcars)
dbListTables(con)
```

```
[1] "mtcars"
```

```
create table StatisticalNumbers(
 value int
)
```

```
SELECT * FROM mtcars LIMIT 5;
```

Table 14.1: 5 records

mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
21.0	6	160	110	3.90	2.875	17.02	0	1	4	4

mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
18.7	8	360	175	3.15	3.440	17.02	0	0	3	2

15 Vector/Matrix/Tensor

Instance of: algebraic object / data structure

AKA: array, matrices

Distinct from: list

English: Vectors, matrices, and tensors are like lists in that they are a collection of objects which are indexed. They differ in that the index can be multi-dimensional, where vectors are 1-d indexed, matrices are 2-d indexed, and tensors are m-d indexed. They also are typically constrained to have objects that share the same type, e.g. numbers or strings.

Formalization:

Cites:

Array:

Wikipedia

3Blue1Brown: Vectors | Chapter 1, Essence of linear algebra 3Blue1Brown: Linear combinations, span, and basis vectors | Chapter 2, Essence of linear algebra

Matrix:

Wikipedia

3Blue1Brown: Linear transformations and matrices | Chapter 3, Essence of linear algebra

Tensor:

Wikipedia

Code

Vector

Note unlike matrix and array, the basic vector function initializes an empty vector and you have to actually use as.vector to coerce something else to vector as the constructor.

vector: Vectors

```
example_vector <- as.vector(c(1,2,3,4))
class(example_vector)
```

```
[1] "numeric"
```

```
example_vector
```

```
[1] 1 2 3 4
```

Matrix

Note we can choose which direction to fill the matrix with, either by row1-col1, row1-col2, row1-col3, row1-col4

matrix: Matrices

```
example_matrix <- matrix(c(1,2,3,4,"A","B","C","D"), nrow = 2, ncol = 4, byrow = TRUE,
                dimnames = list(c("row1", "row2"),
                                c("C.1", "C.2", "C.3", "C.4")))
class(example_matrix)
```

```
[1] "matrix" "array"
```

```
example_matrix
```

```
     C.1 C.2 C.3 C.4
row1 "1" "2" "3" "4"
row2 "A" "B" "C" "D"
```

Arrays

Note array dimensions are ordered, row, column, depth, ..., M , and elements are filled row1-col1-depth1, row2-col1-depth1, row1-col2-depth1,... and so on. Note this was coerced to a string because any of the elements were a string.

array: Multi-way Arrays

```
example_tensor= array(c(1,2,3,4,"A","B","C","D","+","-","*","/"),dim=c(2,3,2,2))
class(example_tensor)
```

```
[1] "array"
```

```r
example_tensor
```

```
, , 1, 1

     [,1] [,2] [,3]
[1,] "1"  "3"  "A"
[2,] "2"  "4"  "B"

, , 2, 1

     [,1] [,2] [,3]
[1,] "C"  "+"  "*"
[2,] "D"  "-"  "/"

, , 1, 2

     [,1] [,2] [,3]
[1,] "1"  "3"  "A"
[2,] "2"  "4"  "B"

, , 2, 2

     [,1] [,2] [,3]
[1,] "C"  "+"  "*"
[2,] "D"  "-"  "/"
```

15.0.0.1 Python

Documentation:

Examples:

15.0.0.2 SQL

Documentation:

```r
library(DBI)
# Create an ephemeral in-memory RSQLite database
#con <- dbConnect(RSQLite::SQLite(), dbname = ":memory:")
#dbListTables(con)
```

```
#dbWriteTable(con, "mtcars", mtcars)
#dbListTables(con)

#Configuration failed because libpq was not found. Try installing:
#* deb: libpq-dev libssl-dev (Debian, Ubuntu, etc)
#install.packages('RPostgres')
#remotes::install_github("r-dbi/RPostgres")
#Took forever because my file permissions were broken
#pg_lsclusters
require(RPostgres)
```

```
Loading required package: RPostgres
```

```
# Connect to the default postgres database
#I had to follow these instructions and create both a username and database that matched m
#https://www.digitalocean.com/community/tutorials/how-to-install-postgresql-on-ubuntu-20-0
con <- dbConnect(RPostgres::Postgres())
```

```

```

15.0.0.3 Jax

15.0.0.4 Torch

```
import torch
```

16 Table

Instance of: arrangement of information or data

AKA: Dataframe

Distinct from:

English: A collection of rows and columns, where rows represent specific instances (AKA records, k-tuple, n-tuple, or a vector), and columns represent features (AKA variables, parameters, properties, attributes, or stanchions). The intersection of a row and column is called a sell.

Formalization:

Cites: Wikipedia Table (information) ; Wikipedia Table Table (database) ; Wikidata ; Wolfram

ML Frameworks Interoperability Cheat Sheet

Code

16.0.0.1 R

Documentation: data.frame: Data Frames

Examples:

```
df=data.frame(a=c(1,2,3,4), b=c('a','b','c','d'))
df
```

```
  a b
1 1 a
2 2 b
3 3 c
4 4 d
```

16.0.0.2 Python

Documentation: pandas.DataFrame

Examples:

```
import pandas as pd
df = pd.DataFrame({'a': [1, 2,3,4], 'b': ['a','b','c','d']})
df
```

```
   a  b
0  1  a
1  2  b
2  3  c
3  4  d
```

16.0.0.3 SQL

Documentation: PostgreSQL AVG Function

```
library(DBI)
# Create an ephemeral in-memory RSQLite database
#con <- dbConnect(RSQLite::SQLite(), dbname = ":memory:")
#dbListTables(con)
#dbWriteTable(con, "mtcars", mtcars)
#dbListTables(con)

#Configuration failed because libpq was not found. Try installing:
#* deb: libpq-dev libssl-dev (Debian, Ubuntu, etc)
#install.packages('RPostgres')
#remotes::install_github("r-dbi/RPostgres")
#Took forever because my file permissions were broken
#pg_lsclusters
require(RPostgres)
```

```
Loading required package: RPostgres
```

```
# Connect to the default postgres database
#I had to follow these instructions and create both a username and database that matched m
#https://www.digitalocean.com/community/tutorials/how-to-install-postgresql-on-ubuntu-20-0
```

```
con <- dbConnect(RPostgres::Postgres())
```

```
DROP TABLE IF EXISTS df;
```

```
CREATE TABLE IF NOT EXISTS df (
    a INTEGER,
    b CHAR
);
```

```
INSERT INTO df (a, b)
VALUES
    (1,'a'),
    (2,'b'),
    (3,'c'),
    (4,'d');
```

```
SELECT  * FROM  df;
```

Table 16.1: 4 records

a	b
1	a
2	b
3	c
4	d

16.0.0.4 Torch

```
import torch
```

Part VI

Operations of Arithmetic

17 Addition

Instance of: operation of arithmetic

17.1 Frequentist

AKA: + ; add

Distinct from:

English:

Formalization:

Cites: Wikipedia ; Wikidata ; Wolfram

Code

17.1.0.1 R

Documentation: mean: Arithmetic Mean

Examples:

17.1.0.2 Python

Documentation: numpy.mean

Examples:

17.1.0.3 SQL

Documentation: PostgreSQL AVG Function

```r
library(DBI)
# Create an ephemeral in-memory RSQLite database
#con <- dbConnect(RSQLite::SQLite(), dbname = ":memory:")
#dbListTables(con)
#dbWriteTable(con, "mtcars", mtcars)
#dbListTables(con)

#Configuration failed because libpq was not found. Try installing:
#* deb: libpq-dev libssl-dev (Debian, Ubuntu, etc)
#install.packages('RPostgres')
#remotes::install_github("r-dbi/RPostgres")
#Took forever because my file permissions were broken
#pg_lsclusters
require(RPostgres)
```

```
Loading required package: RPostgres
```

```r
# Connect to the default postgres database
#I had to follow these instructions and create both a username and database that matched m
#https://www.digitalocean.com/community/tutorials/how-to-install-postgresql-on-ubuntu-20-0
con <- dbConnect(RPostgres::Postgres())
```

17.1.0.4 Torch

```r
import torch
```

17.2 Bayesian

English: **Formalization**:

Cites:

Code

18 Introduction

Instance of: operation of arithmetic

18.1 Frequentist

AKA: - ; minus

Distinct from:

English:

Formalization:

Cites: Wikipedia ; Wikidata ; Wolfram

Code

18.1.0.1 R

Documentation: mean: Arithmetic Mean

Examples:

18.1.0.2 Python

Documentation: numpy.mean

Examples:

18.1.0.3 SQL

Documentation: PostgreSQL AVG Function

```r
library(DBI)
# Create an ephemeral in-memory RSQLite database
#con <- dbConnect(RSQLite::SQLite(), dbname = ":memory:")
#dbListTables(con)
#dbWriteTable(con, "mtcars", mtcars)
#dbListTables(con)

#Configuration failed because libpq was not found. Try installing:
#* deb: libpq-dev libssl-dev (Debian, Ubuntu, etc)
#install.packages('RPostgres')
#remotes::install_github("r-dbi/RPostgres")
#Took forever because my file permissions were broken
#pg_lsclusters
require(RPostgres)
```

```
Loading required package: RPostgres
```

```r
# Connect to the default postgres database
#I had to follow these instructions and create both a username and database that matched m
#https://www.digitalocean.com/community/tutorials/how-to-install-postgresql-on-ubuntu-20-0
con <- dbConnect(RPostgres::Postgres())
```

```r

```

18.1.0.4 Torch

```r
import torch
```

18.2 Bayesian

English: **Formalization**:

Cites:

Code

19 Multiplication

Instance of: operation of arithmetic

19.1 Frequentist

AKA: * ; × ; ; multiply

Distinct from:

English:

Formalization:

Cites: Wikipedia ; Wikidata ; Wolfram

3Blue1Brown: Matrix multiplication as composition | Chapter 4, Essence of linear algebra
3Blue1Brown: Cross products in the light of linear transformations | Chapter 11, Essence of linear algebra

Code

19.1.0.1 R

Documentation: mean: Arithmetic Mean

Examples:

19.1.0.2 Python

Documentation: numpy.mean

Examples:

19.1.0.3 SQL

Documentation: PostgreSQL AVG Function

```
library(DBI)
# Create an ephemeral in-memory RSQLite database
#con <- dbConnect(RSQLite::SQLite(), dbname = ":memory:")
#dbListTables(con)
#dbWriteTable(con, "mtcars", mtcars)
#dbListTables(con)

#Configuration failed because libpq was not found. Try installing:
#* deb: libpq-dev libssl-dev (Debian, Ubuntu, etc)
#install.packages('RPostgres')
#remotes::install_github("r-dbi/RPostgres")
#Took forever because my file permissions were broken
#pg_lsclusters
require(RPostgres)
```

```
Loading required package: RPostgres
```

```
# Connect to the default postgres database
#I had to follow these instructions and create both a username and database that matched m
#https://www.digitalocean.com/community/tutorials/how-to-install-postgresql-on-ubuntu-20-0
con <- dbConnect(RPostgres::Postgres())
```

19.1.0.4 Torch

```
import torch
```

19.2 Bayesian

English: **Formalization**:

Cites:

Code

20 Division

Instance of: operation of arithmetic

20.1 Frequentist

AKA: / ; $\frac{numerator}{denominator}$; $\div$

Distinct from:

English:

Formalization:

Cites: Wikipedia ; Wikidata ; Wolfram

Code

20.1.0.1 R

Documentation: mean: Arithmetic Mean

Examples:

20.1.0.2 Python

Documentation: numpy.mean

Examples:

20.1.0.3 SQL

Documentation: PostgreSQL AVG Function

```r
library(DBI)
# Create an ephemeral in-memory RSQLite database
#con <- dbConnect(RSQLite::SQLite(), dbname = ":memory:")
#dbListTables(con)
#dbWriteTable(con, "mtcars", mtcars)
#dbListTables(con)

#Configuration failed because libpq was not found. Try installing:
#* deb: libpq-dev libssl-dev (Debian, Ubuntu, etc)
#install.packages('RPostgres')
#remotes::install_github("r-dbi/RPostgres")
#Took forever because my file permissions were broken
#pg_lsclusters
require(RPostgres)
```

```
Loading required package: RPostgres
```

```r
# Connect to the default postgres database
#I had to follow these instructions and create both a username and database that matched m
#https://www.digitalocean.com/community/tutorials/how-to-install-postgresql-on-ubuntu-20-0
con <- dbConnect(RPostgres::Postgres())
```

20.1.0.4 Torch

```r
import torch
```

20.2 Bayesian

English: **Formalization**:

Cites:

Code

Part VII

Operations of Algebra

21 Dot product

Instance of: algebraic operation

AKA: scalar product; inner product ; projection product ; $ · $

Distinct from:

English:

Formalization:

$$a \cdot b$$

Cites: Wikipedia ; Wikidata ; Wolfram

3Blue1Brown: Dot products and duality | Chapter 9, Essence of linear algebra

Code

21.0.0.1 R

Documentation:

Examples:

21.0.0.2 Python

Documentation: numpy.mean

Examples:

21.0.0.3 SQL

Documentation: PostgreSQL AVG Function

```
library(DBI)
# Create an ephemeral in-memory RSQLite database
```

```r
#con <- dbConnect(RSQLite::SQLite(), dbname = ":memory:")
#dbListTables(con)
#dbWriteTable(con, "mtcars", mtcars)
#dbListTables(con)

#Configuration failed because libpq was not found. Try installing:
#* deb: libpq-dev libssl-dev (Debian, Ubuntu, etc)
#install.packages('RPostgres')
#remotes::install_github("r-dbi/RPostgres")
#Took forever because my file permissions were broken
#pg_lsclusters
require(RPostgres)
```

```
Loading required package: RPostgres
```

```r
# Connect to the default postgres database
#I had to follow these instructions and create both a username and database that matched m
#https://www.digitalocean.com/community/tutorials/how-to-install-postgresql-on-ubuntu-20-0
con <- dbConnect(RPostgres::Postgres())
```

```r

```

21.0.0.4 Torch

```r
import torch
```

21.1 Bayesian

English: **Formalization**:

Cites:

Code

Part VIII

Moments of a Distribution

22 Mean

Measure of: Central tendency

22.1 Frequentist

AKA: Arithmetic mean; average; $\bar{x}$ (sample mean); μ (population mean); μ_x (population mean)

Distinct from: Geometric mean (GM); Harmonic mean (HM); generalized mean/ Power mean; weighted arithmetic mean

English: Take a list of numbers, sum those numbers, and then divide by the number of numbers.

Formalization:

$$\bar{x} = \frac{1}{n}(\sum_{i=1}^{n} x_i) = \frac{x_1 + x_2 + ... + x_n}{n}$$

Cites: Wikipedia ; Wikidata ; Wolfram

Code

22.1.0.1 R

Documentation: mean: Arithmetic Mean

Examples:

```
x = c(1,2,3,4)
x
```

```
[1] 1 2 3 4
```

```r
#Algorithm
x_bar = sum(x, na.rm=T)/length(x)
x_bar
```

```
[1] 2.5
```

```r
#Base Function
x_bar = mean(x, na.rm=T)
x_bar
```

```
[1] 2.5
```

22.1.0.2 Python

Documentation: numpy.mean

Examples:

```python
x = [1,2,3,4]
print(x)
```

```
[1, 2, 3, 4]
```

```python
#Algorithm
x_bar= sum(x)/len(x)
x_bar
```

```
2.5
```

```python
#statistics Function
import statistics
x_bar = statistics.mean(x)
x_bar
```

```
2.5
```

```
#scipy Function
#<string>:1: DeprecationWarning: scipy.mean is deprecated and will be removed in SciPy 2.0
import scipy
x_bar = scipy.mean(x)
```

```
<string>:1: DeprecationWarning: scipy.mean is deprecated and will be removed in SciPy 2.0.0,
```

```
x_bar
```

```
2.5
```

```
#numpy Function
import numpy as np
x = np.array(x)
x_bar = x.mean()
x_bar
```

```
2.5
```

22.1.0.3 SQL

Documentation: PostgreSQL AVG Function

```
library(DBI)
# Create an ephemeral in-memory RSQLite database
#con <- dbConnect(RSQLite::SQLite(), dbname = ":memory:")
#dbListTables(con)
#dbWriteTable(con, "mtcars", mtcars)
#dbListTables(con)

#Configuration failed because libpq was not found. Try installing:
#* deb: libpq-dev libssl-dev (Debian, Ubuntu, etc)
#install.packages('RPostgres')
#remotes::install_github("r-dbi/RPostgres")
#Took forever because my file permissions were broken
#pg_lsclusters
require(RPostgres)
```

```
Loading required package: RPostgres
```

```
# Connect to the default postgres database
#I had to follow these instructions and create both a username and database that matched m
#https://www.digitalocean.com/community/tutorials/how-to-install-postgresql-on-ubuntu-20-0
con <- dbConnect(RPostgres::Postgres())
```

```
DROP TABLE IF EXISTS t1;
```

```
CREATE TABLE IF NOT EXISTS t1 (
    id serial PRIMARY KEY,
    amount INTEGER
);
```

```
INSERT INTO t1 (amount)
VALUES
    (10),
    (NULL),
    (30);
```

```
SELECT  * FROM  t1;
```

Table 22.1: 3 records

id	amount
1	10
2	NA
3	30

```
SELECT AVG(amount)::numeric(10,2)
FROM t1;
```

Table 22.2: 1 records

avg
20

22.1.0.4 Torch

```
import torch
```

22.2 Bayesian

Bayesian average; Solving an age-old problem using Bayesian Average; Of bayesian average and star ratings; Bayesian Average Ratings ;

English: The Bayesian average is the weighted average of a prior and the observed sample average. When would you want this? When you have strong beliefs about the true mean, or when sample size is too small to reliable calculate a mean. For example a movie rating website where a movie may have only a single 5 star rating and so would rank higher than the Godfather with over a 100 almost all 5 star ratings.

Formalization:

$$\bar{x} = \frac{C * m + (\sum_{i=1}^{n} x_i)}{c + n}$$

Where m is a prior for true mean, and C is a constant representing how many elements would be necessary to reliably estimate a sample mean.

Code

Part IX

Supervised Learning

23

24

25

26

Videos

StatQuest with Josh Starmer Gradient Boost Part 1 (of 4): Regression Main Ideas XGBoost Part 1 (of 4): Regression

XGBoost LightGBM

Part X

Unsupervised Learning

27

References