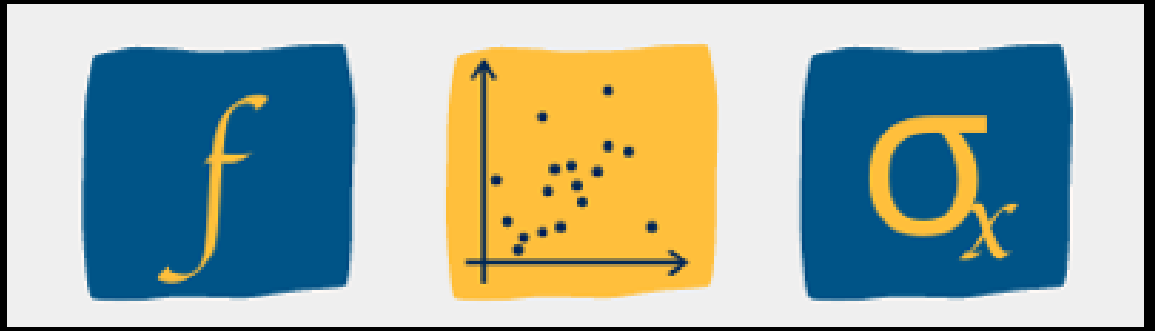
Centre—UB
Understanding Human Behaviour



Statistics with R: A practical and beginner friendly workshop for behavioural scientists

Dr. Lei Zhang

Adaptive Learning Psychology & Neuroscience (ALPN) Lab
Centre for Human Brain Health, School of Psychology
University of Birmingham

lei-zhang.net
@lei_zhang_lz

Edgbaston, 23.05.2025

Schedule

09:30 – 09:45	Arrival, check-in
09:45 – 10:45	Introduction
10:45 – 11:00	Coffee break 1
11:00 – 12:15	Basics in R
12:15 – 13:00	Lunch
13:00 – 13:30	R workflow showcase
13:30 – 14:00	Simple stats in R
14:00 – 14:15	Coffee break 2
14:15 – 15:00	Visualisation in R

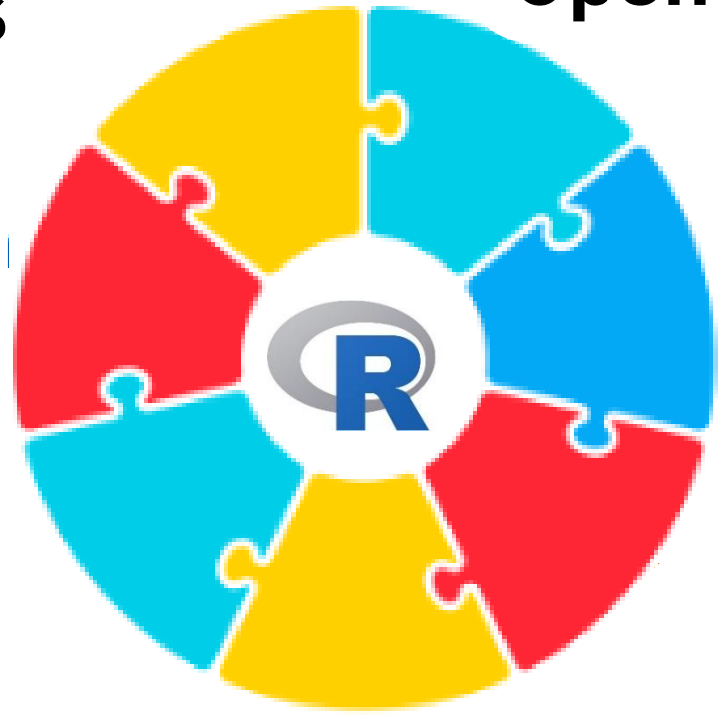
Goal of this course

- Practical R programming
- Practical scripting and stats in R
- (Enough) theory to ground you
- Be comfortable to use R for your own work: descriptive stats, inferential stats, visualisation

Why learn R?

Extensive Libraries

Open Source



Cross-platform Support

Data Visualisation

Integration with HPC

Statistical Computing

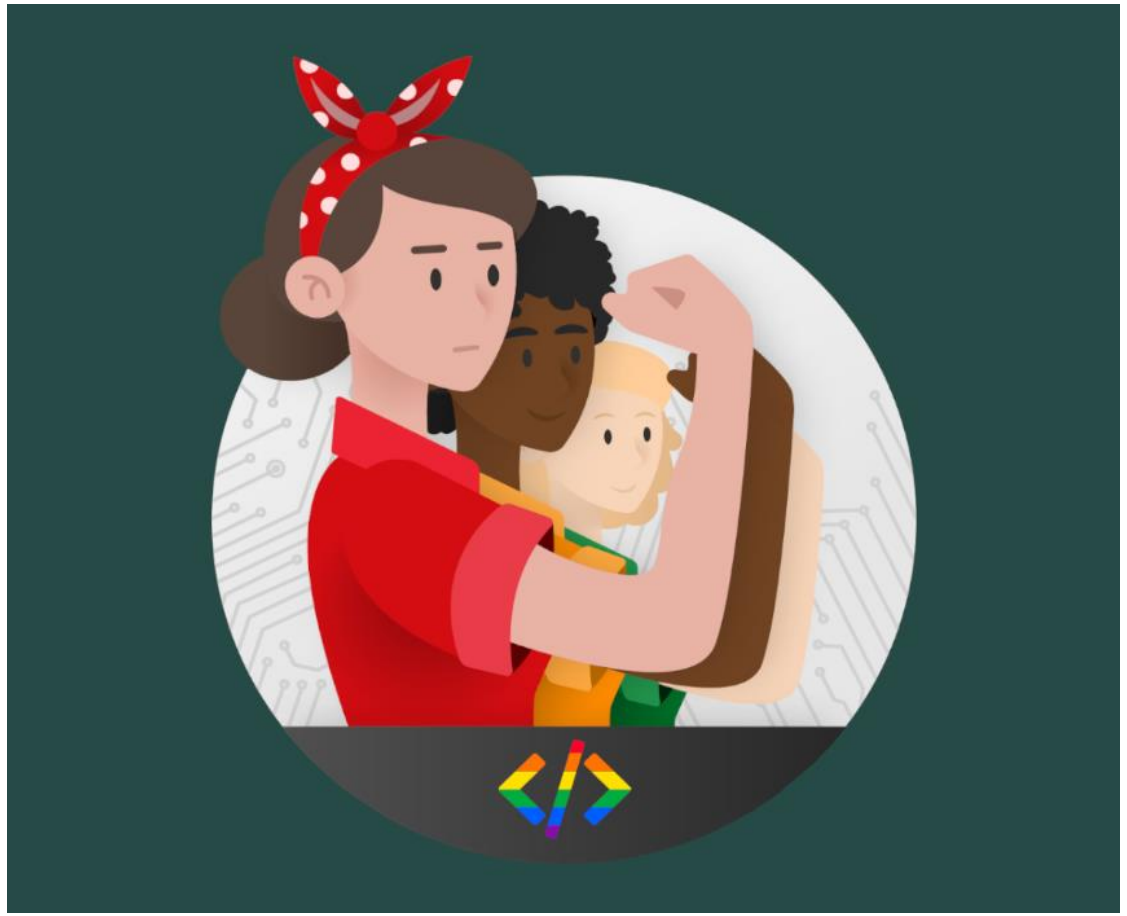
Data Handling

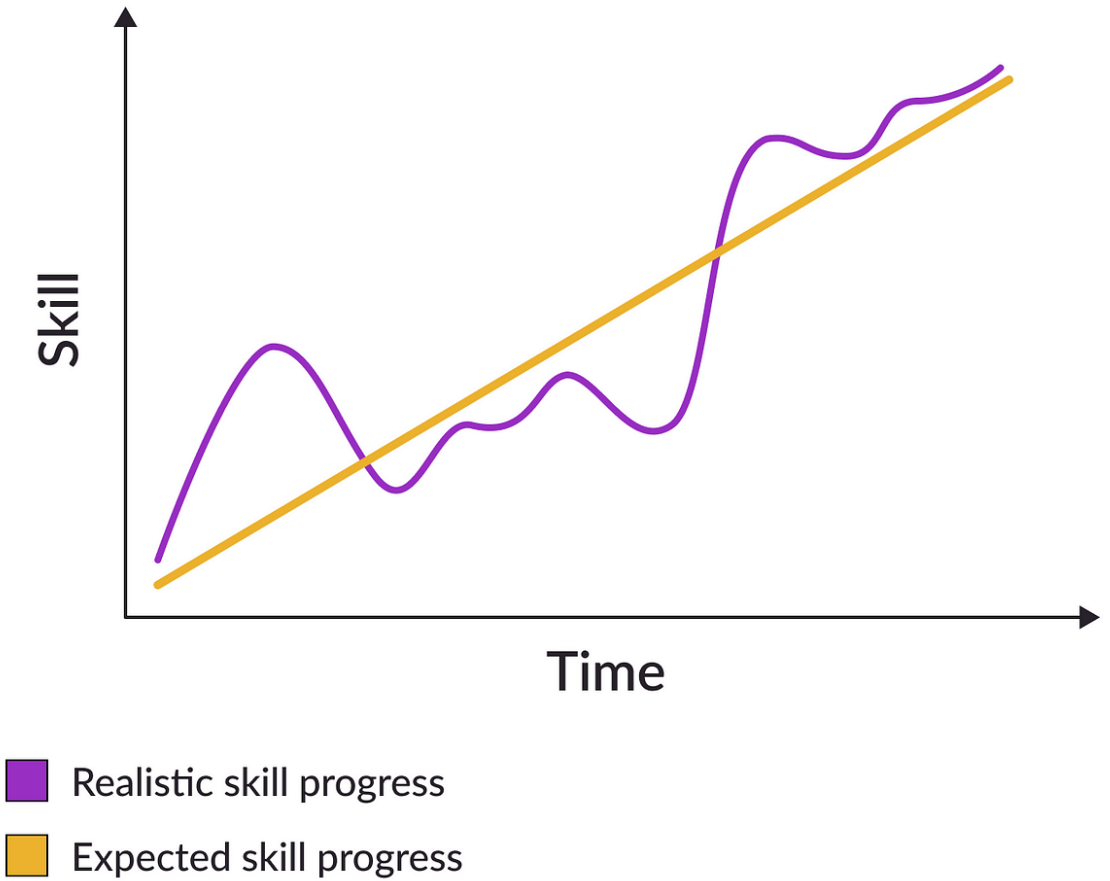
BEAR
BIRMINGHAM ENVIRONMENT FOR ACADEMIC RESEARCH

Applications of R

R is powerful for understanding behaviour

Everyone can code!





Why Does Statistics Seem Hard?

Math Anxiety

Stats involves math, but usually just basic algebra, and software helps with the rest.

Abstract Concepts

Ideas like probability can feel vague, but become clearer with examples and visuals

Uncertainty

Stats deals with likelihoods, not certainties, which takes getting used to

Symbols

Greek letters and formulas seem confusing but are just helpful shortand

Bad Teaching

Poor instruction can make stats harder than it is—better resources help

Course Pressure

It's often required, which adds stress, but it teaches useful real-world skills

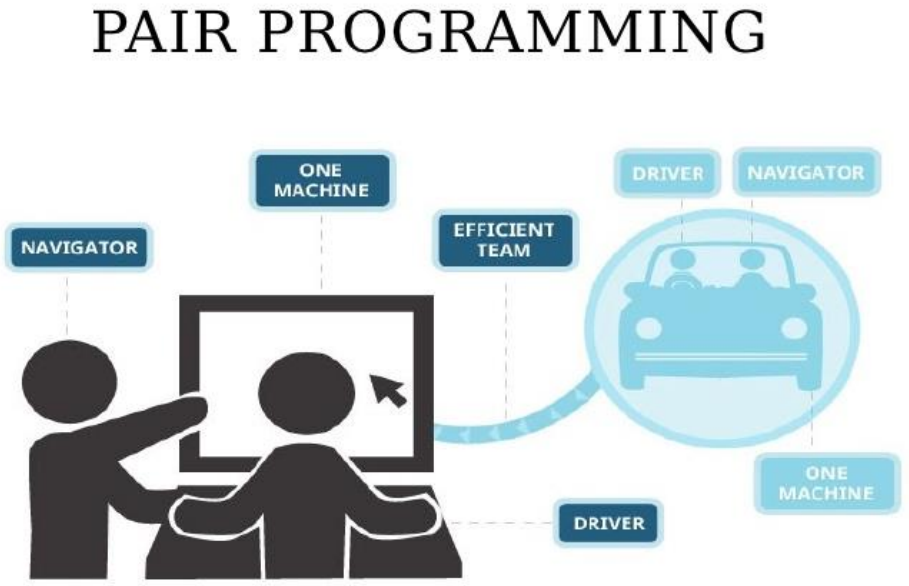
Richard McElreath
@rlmcelreath

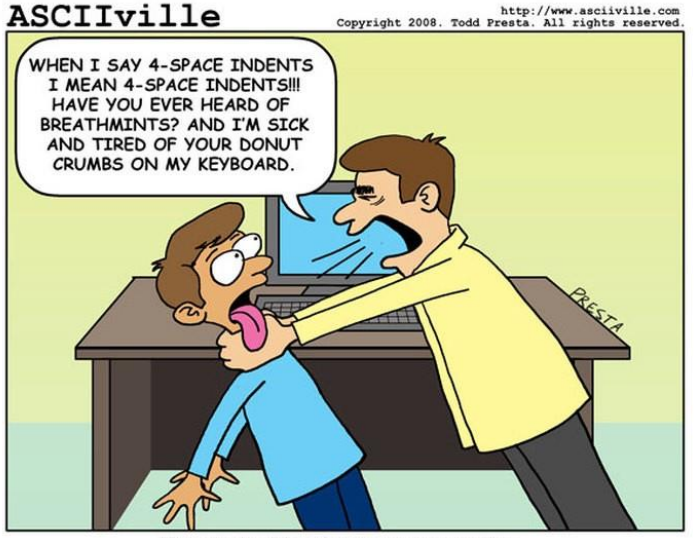
I say this a lot, bc I am also confused quite often.

> 
> **Anna Jacobson** @AnnaChingChing · Feb 21
>
> "If you are confused, it is only because you are trying to understand." - @rlmcelreath in Statistical Rethinking

How to Get the Most out of the course

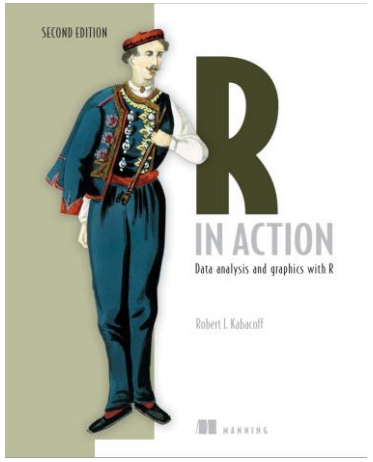
- Workshop structure: interleaved theory/demo + exercise
- Work in pairs: Talk to each other & help each other
- Ask questions
- Try the exercises

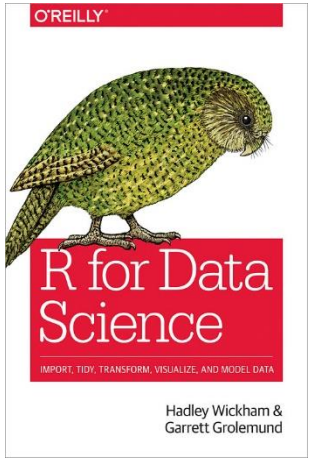




The dark side of pair programming.

Resources





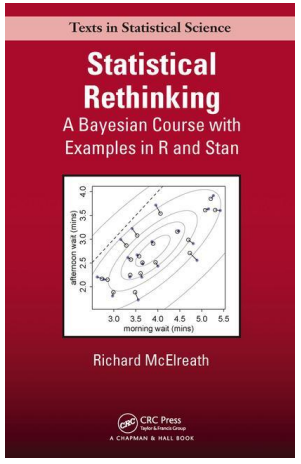
statsthinking21

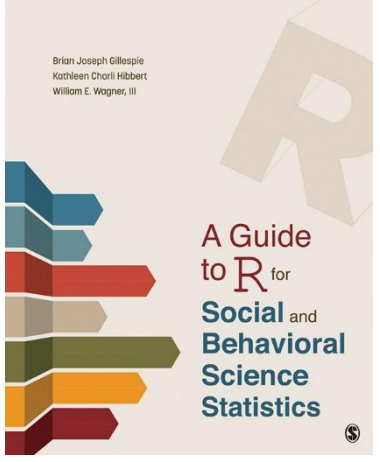
Main web site for Statistical Thinking for the 21st Century

http://statsthinking21.org/

datacamp

https://www.datacamp.com/





ChatGPT

https://chatgpt.com/

Now let’s begin!

BASICS OF R PROGRAMMING

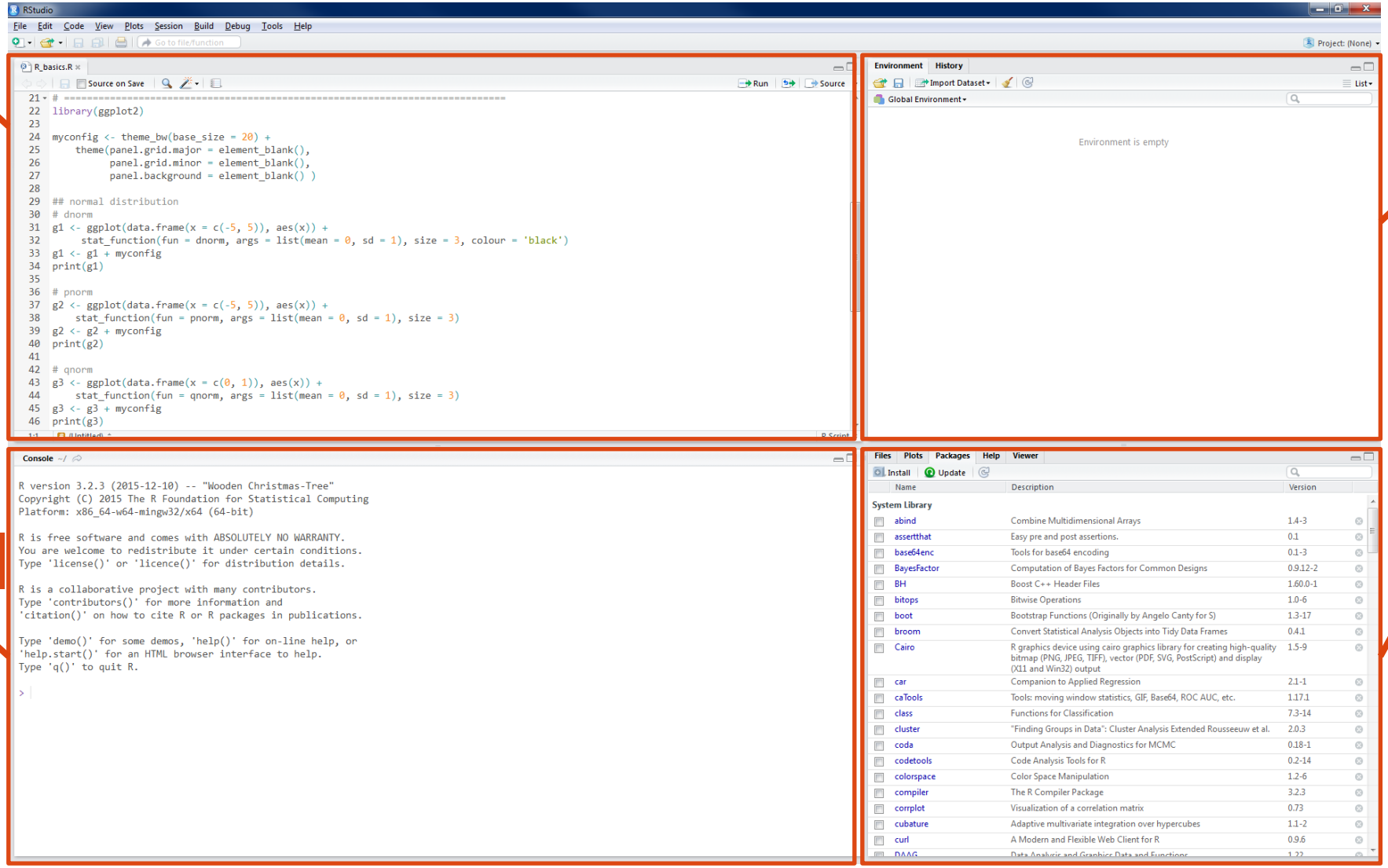
RStudio®

R Basics

- R
  - a programming language for statistical computing
  - R has its own user interface
  - freely available on Windows, Mac, and Linux

- R Studio
  - integrated development environment (IDE) for R
  - a more sophisticated R-friendly editor, with helpful syntax highlight

script editor



console

environment/
command history

file/pkg/img/
etc.

Know your R

```
>R.version

               _
platform       x86_64-w64-mingw32
arch           x86_64
os             mingw32
system         x86_64, mingw32
status
major          3
minor          5.1
year           2018
month          07
day            02
svn rev        74947
language       R
version.string R version 3.5.1 (2018-07-02)
nickname       Feather Spray
```

R Console as a Calculator

Addition and Subtraction

```
> 3+2
[1] 5

> 3-2
[1] 1
```

Multiplication and Division

```
> 3*2
[1] 6

> 3/2
[1] 1.5
```

Exponents in R

```
> 3^2
[1] 9

> 2^3
[1] 8
```

Constants in R

```
> pi
[1] 3.141593

> exp(1)
[1] 2.718282
```

base of the natural logarithm

Special values

Infinite Values

```
> Inf
[1] Inf

> 1+Inf
[1] Inf
```

Empty Values

```
> NULL
NULL

> 1+NULL
numeric(0)
```

Machine Epsilon

```
> .Machine$double.eps
[1] 2.220446e-16

> 0>.Machine$double.eps
[1] FALSE
```

Missing Values

```
> NA
[1] NA

> 1+NA
[1] NA
```

Storing and manipulating variables

Define objects `x` and `y` with values of 3 and 2, respectively:

```
> x=3
> y=2
```

Some calculations with the defined objects `x` and `y`:

```
> x+y
[1] 5

> x*y
[1] 6
```

Warning: R is case sensitve, so `x` and `X` are not the same object.

Basic R functions

Combine

```
> c(1,3,-2)
[1]  1  3 -2
```

```
> c("a","a","b","b","a")
[1] "a" "a" "b" "b" "a"
```

Sum and Mean

```
> sum(c(1,3,-2))
[1] 2
```

```
> mean(c(1,3,-2))
[1] 0.6666667
```

Variance and Std. Dev.

```
> var(c(1,3,-2))
[1] 6.333333
```

```
> sd(c(1,3,-2))
[1] 2.516611
```

Minimum and Maximum

```
> min(c(1,3,-2))
[1] -2
```

```
> max(c(1,3,-2))
[1] 3
```

Basic R functions (cont.)

Define objects `x` and `y`:

```
> x=c(1,3,4,6,8)
> y=c(2,3,5,7,9)
```

Calculate the correlation:

```
> cor(x,y)
[1] 0.988765
```

Calculate the covariance:

```
> cov(x,y)
[1] 7.65
```

Combine as columns

```
> cbind(x,y)
     x y
[1,] 1 2
[2,] 3 3
[3,] 4 5
[4,] 6 7
[5,] 8 9
```

Combine as rows

```
> rbind(x,y)
  [,1] [,2] [,3] [,4] [,5]
x    1    3    4    6    8
y    2    3    5    7    9
```

Basic Commands

```
getwd()
setwd('E:/teaching/2025_Centre-UB_R_Workshop/')
dir() # folders/files in the wd
ls()  # anything in the environment/workspace
print('Hello World!')
cat('Hello', 'World!')
paste0('C:/', 'Group1')
help(func)
? func # and Google!
a <- 5
a = 5
head(d) # first 6 entries
tail(d) # last 6 entries
save(varname, file = "pathname/varname.RData")
load("pathname/varname.RData")
rm(list = ls())
q()
```

RStudio - Shortcuts

Ctrl + L: clean console

Ctrl + Shift + N: create a new script

↑: command history

Ctrl(hold) + ↑: command history with certain starts

Ctrl + Enter: execute selected codes (in a script)

Editor (WIN general) - Shortcuts

Ctrl + home/Pos: go to the very top of a script

Ctrl + end/Ende: go to the very end of a script

Shift(hold) + ↑/↓: select line(s)

Ctrl(hold) + ←/→: select word(s)

Data Classes

numeric: 1.1 2.0

integer: 1 2 3

character / string: "hello world!"

logical: TRUE FALSE

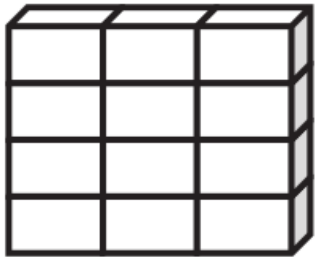
factors: "male" / "female"

(complex: 1+2i)

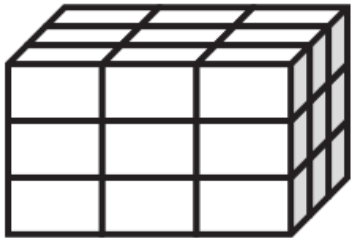
Data Types

(a) Vector

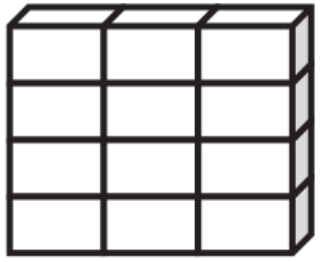
(b) Matrix



(c) Array



(d) Data frame



Columns can be different modes

(e) List

- Vectors
- Arrays
- Data frames
- Lists

Kabacoff (2015)

Exercise I

```
…/01.R_basics/_scripts/R_basics.R

up to “Control Flow”
```

TASK: practise basic R commands and data type

TIP: `class()`, `str()`

Side note: folder structure

click this to start each exercise,
then no need to set directory

Logical Operators

Operator	Summary
$<$	Less than
$>$	Greater than
$<=$	Less than or equal to
$>=$	Greater than or equal to
$==$	Equal to
$!=$	Not equal to
$!x$	NOT x
$x \mid y$	x OR y
$x \& y$	x AND y

Control Flow

- if-else

```
if (cond) {
    ..statement..
}
```

```
if (cond) {
    ..statement..
} else {
    ..statement..
}
```

```
if (cond) {
    ..statement..
} else if (cond) {
    ..statement..
} else {
    ..statement..
}
```

- for-loop

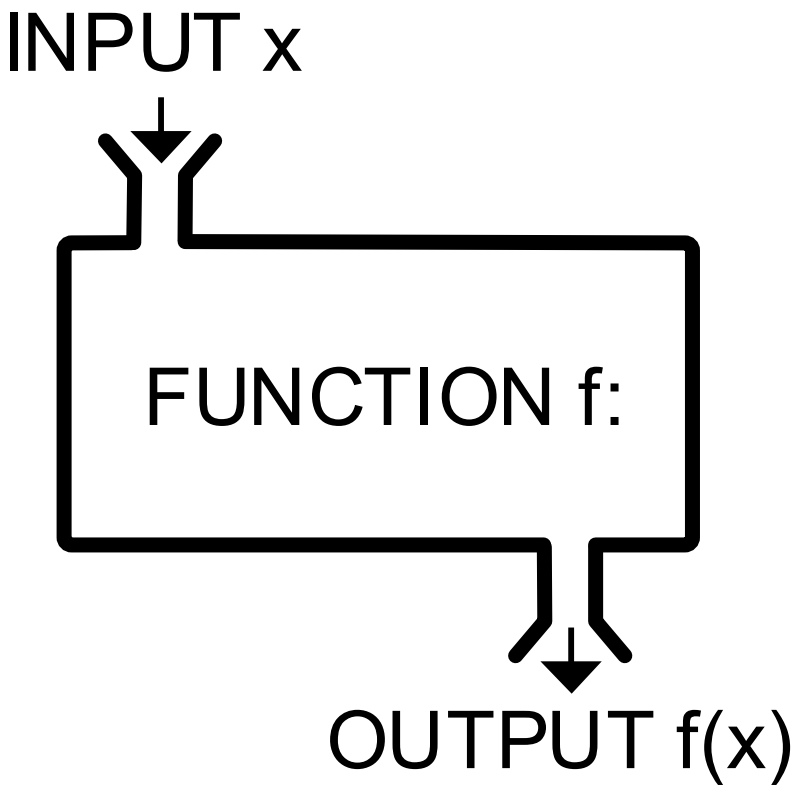
```
for ( j in 1:J) {
    ..statement..
}
```

```
for ( j in 1:J ) {
    for ( k in 1:K ) {
        ..statement..
    }
}
```

Functions

The operation(s) to obtain some quantity, based on another quantity.

- built-in functions
- external functions (packages)
- user-defined functions



User-defined Function

```
funname <- function (input_arges) {
    .. function body ..
    .. function body ..
    return(output_arges)
}
```

$$sem = \sqrt{\frac{s^2}{n-1}}$$

```
sem <- function(x) {
    sqrt( var(x,na.rm=TRUE) / (length(na.omit(x))-1) )
}
```

Exercise II

```
…/01.R_basics/_scripts/R_basics.R
```

TASK: practise control flow and user-defined function

Exercise II

- Generate a random number between 0 and 1
- Compare it against 1/3 and 2/3
- Print the random number and its position relative to 1/3 and 2/3.

```r
# if-else
t <- runif(1) # random number between 0 and 1
if (t <= 1/3) {
    cat("t =", , ", t <= 1/3. \n")
} else if () {
    cat("t =", t, ", t > 2/3. \n")
} else {
    cat("t =", t, ", 1/3 < t <= 2/3. \n")
}
```

Example outcome:

```
t = 0.895 , t > 2/3.
```

- Get the name of each month
- Print it one by one

```r
# for-loop
month_name <- format(ISOdate(2018,1:12,1),"%B")
for (j in 1:length(month_name) ) {
    cat()
}
```

```
The month is January
The month is February
The month is March
The month is April
The month is May
The month is June
The month is July
The month is August
The month is September
The month is October
The month is November
The month is December
```

Packages in R

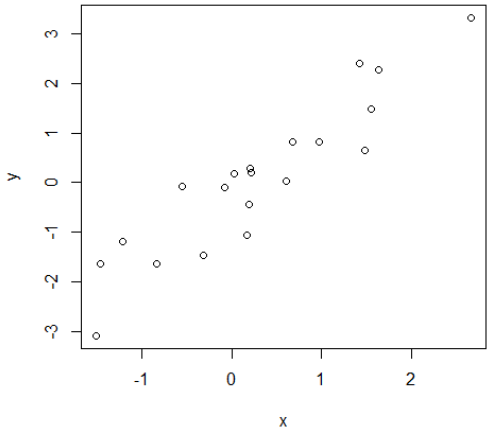
R packages are collections of functions and data sets developed by the community, to make your life a lot easier!

```
install.packages('ggplot2')
library(ggplot2)
detach('package:ggplot2')
```

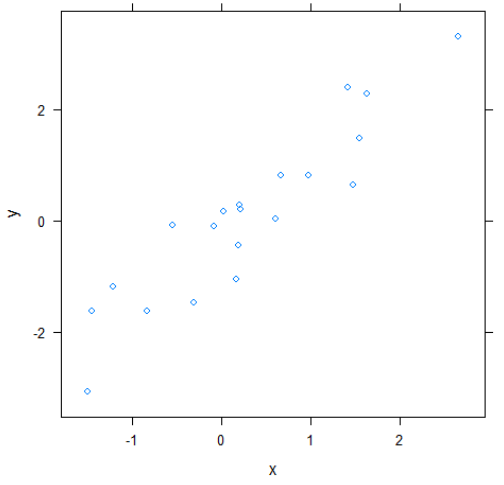
Visualization

- built-in plotting functions – first attempt / quick look / exploratory
- {lattice} – making nicer, similar to basic plotting functions (takes lm formulae)
- {ggplot2} – making nicer, a layering philosophy

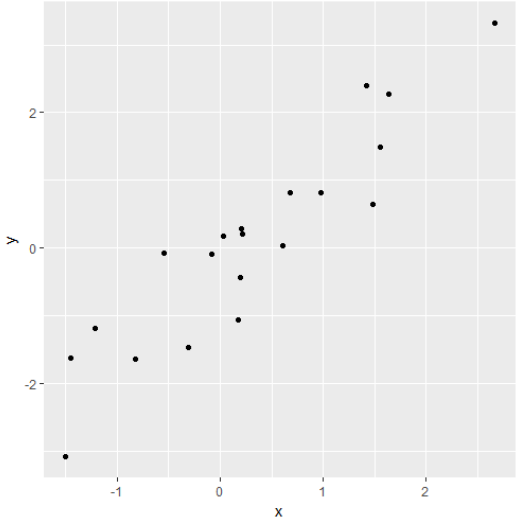
`plot(x,y)`



`lattice::xyplot(y~x)`



`ggplot2::qplot(x,y)`

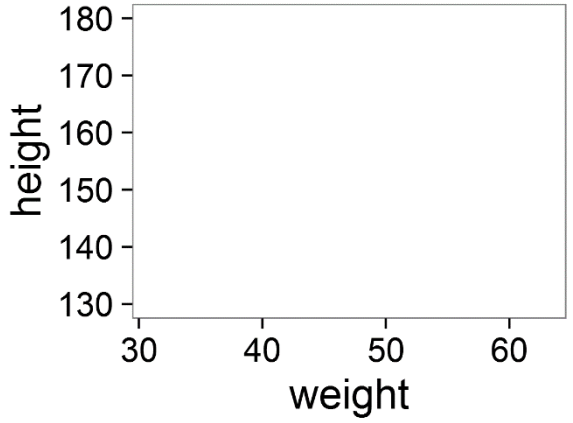


Brief Intro to ggplot2

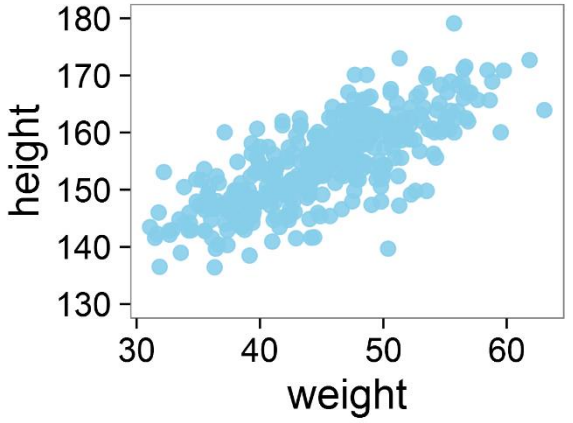
```
plot = geometric (points, lines, bars) + aesthetic (color, shape, size)
```

game of adding layers!

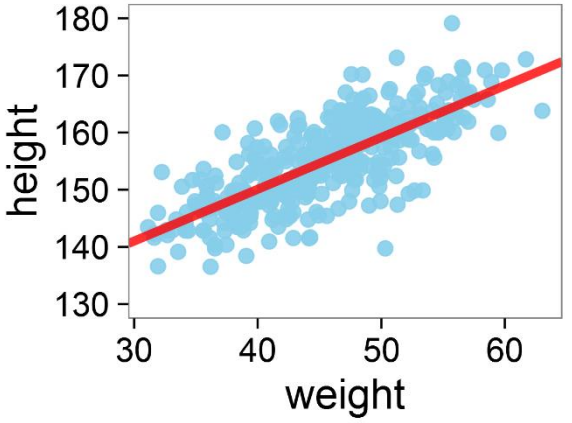
background



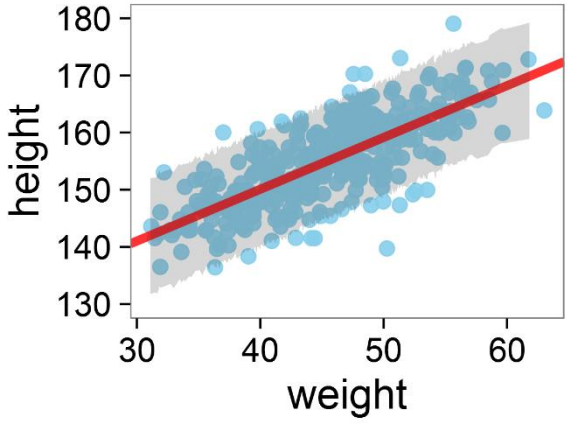
add scatters



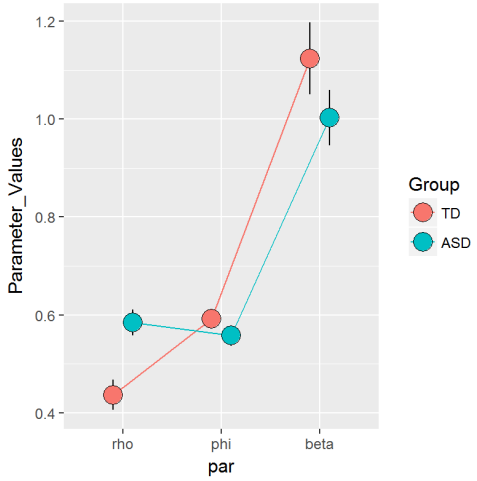
add regression line

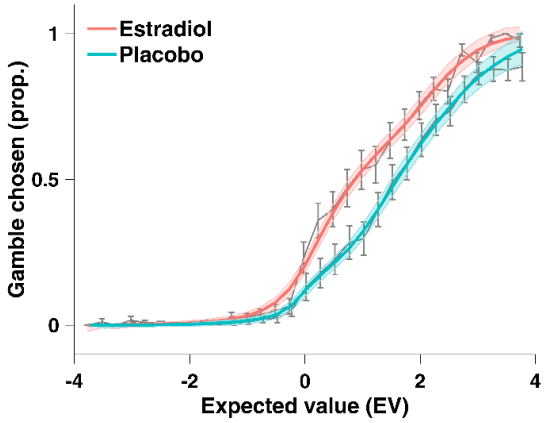


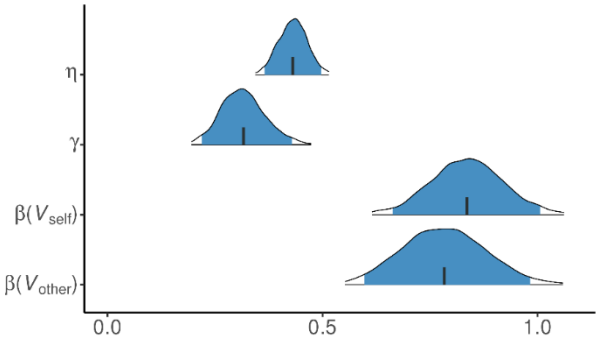
add uncertainty

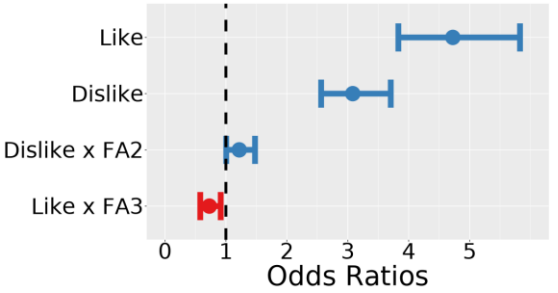


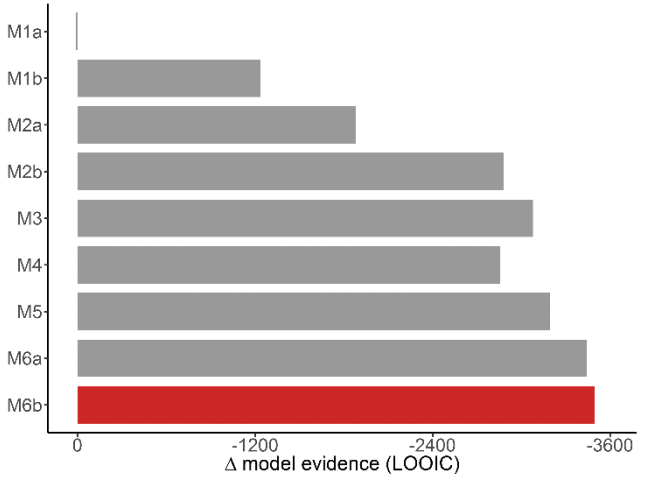
A taste of ggplot2

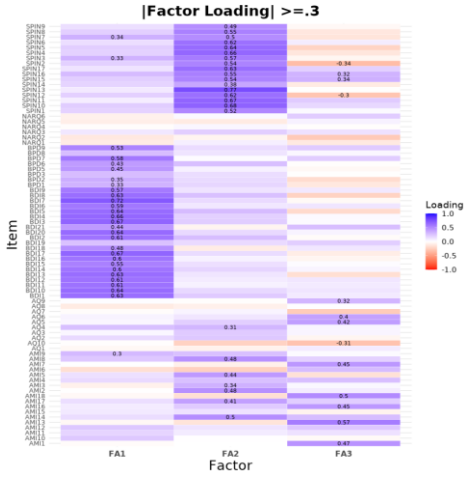


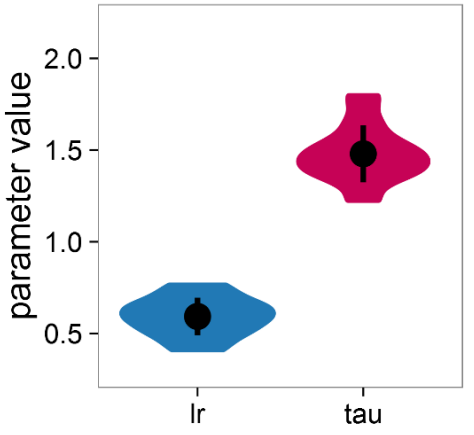


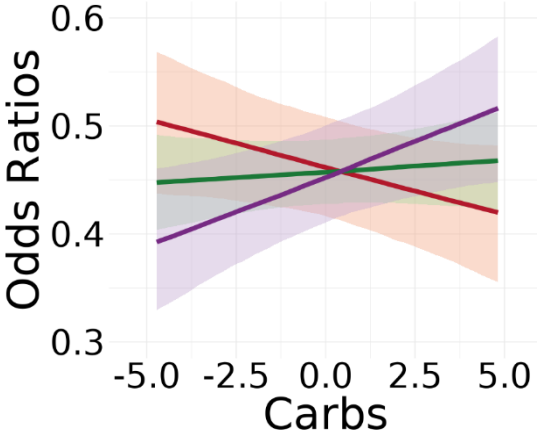


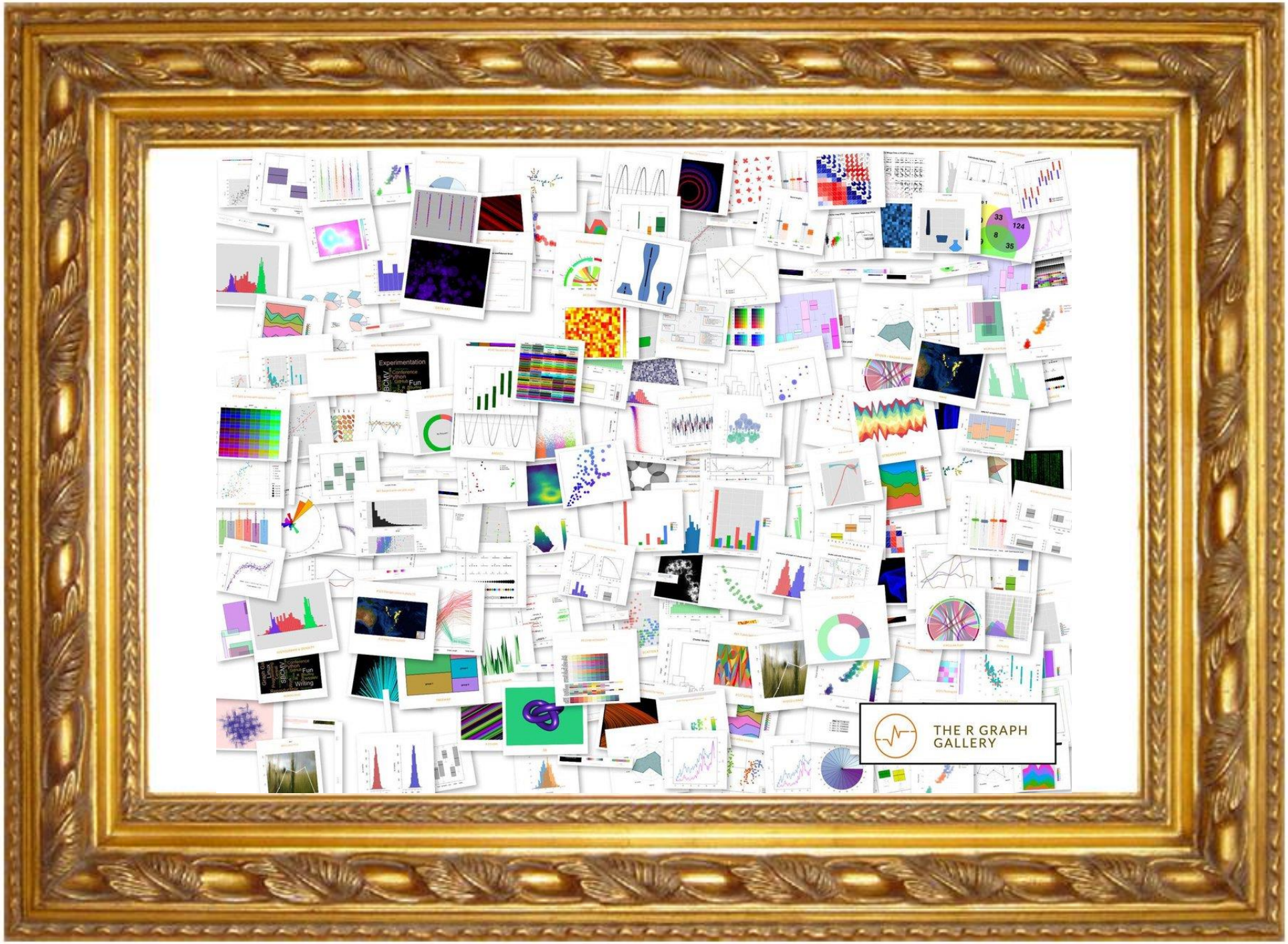






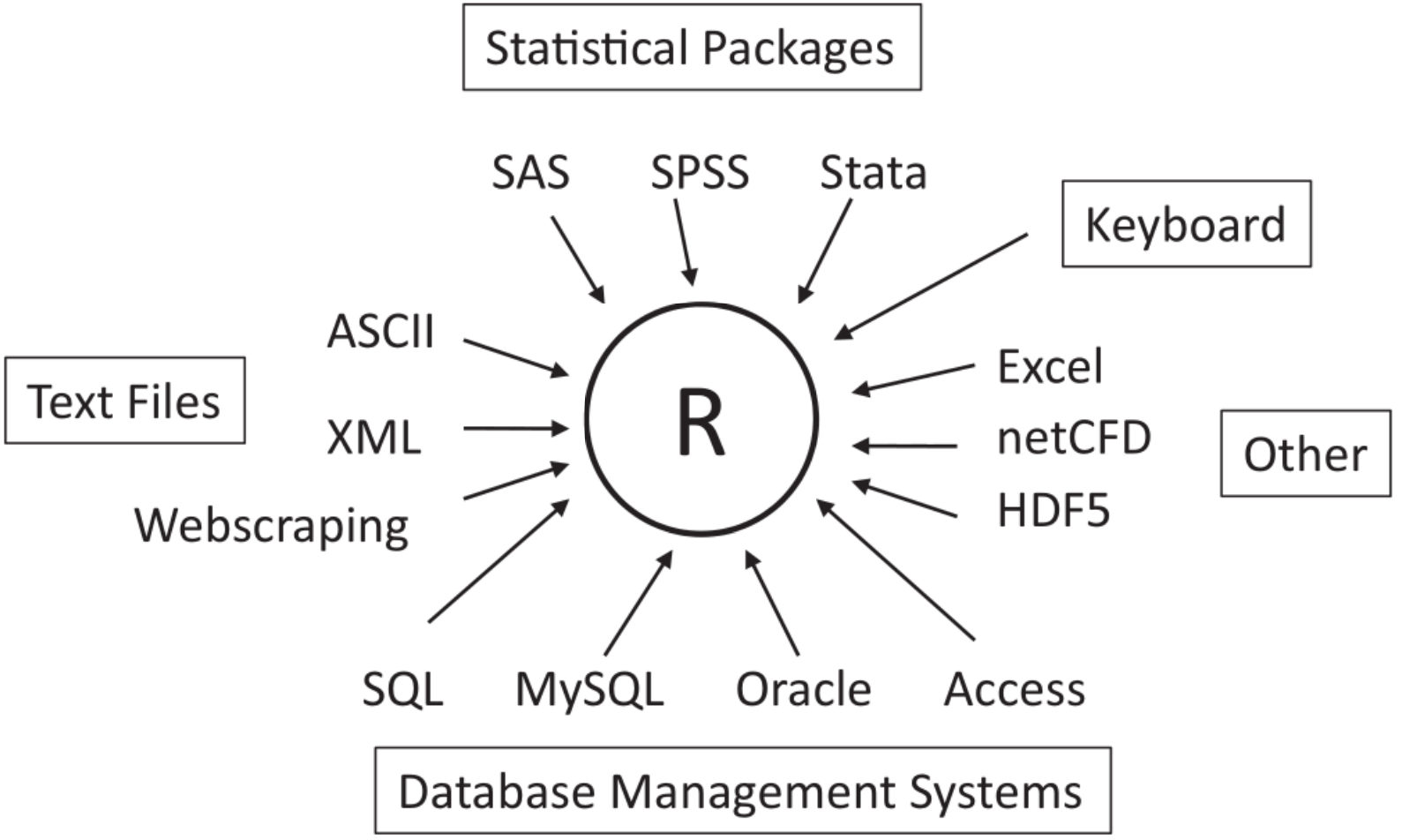






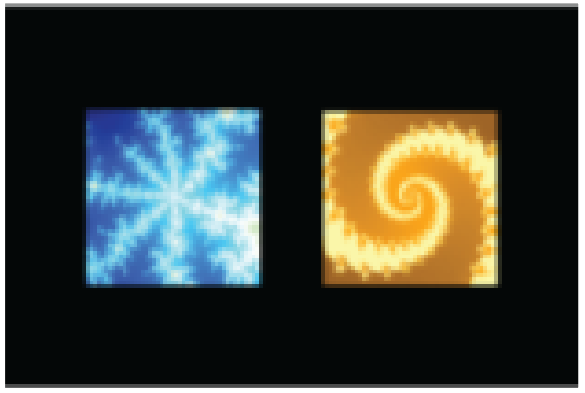
https://www.r-graph-gallery.com/

Data management

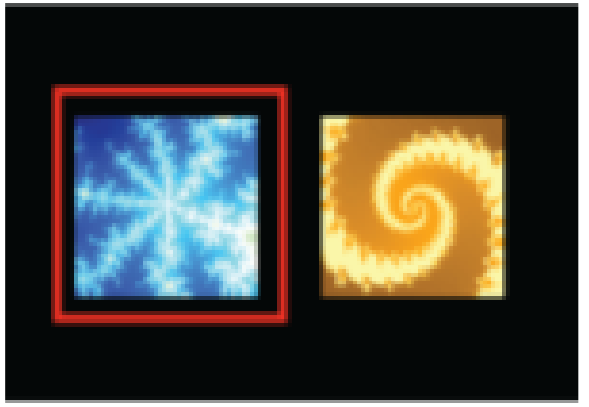


Kabacoff (2015)

One simple experiment



choice
presentation



action
selection

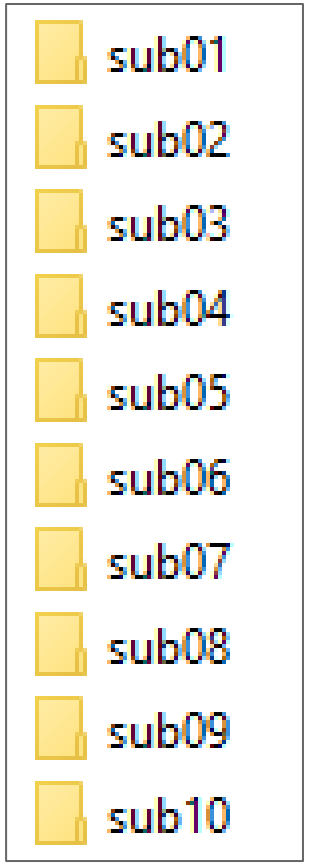
outcome

reward contingency – 80:20

The data

- nSub = 10
- nTrial = 80

./_data/_raw_data/sub01/raw_data_sub01.txt




```
subjID, trialID, choice, outcome, correct
1,1,2,-1,1
1,2,1,1,1
1,3,1,1,1
1,4,1,1,1
1,5,2,-1,1
1,6,1,1,1
1,7,1,1,1
1,8,1,1,1
1,9,1,-1,1
1,10,2,-1,1
1,11,1,1,1
1,12,1,1,1
1,13,1,-1,2
```

Import some data!

```
data_dir = ('_data/RL_raw_data/sub01/raw_data_sub01.txt')
data = read.table(data_dir, header = T, sep = ",")
head(data)
```

```
  subjID trialID choice outcome correct
1      1       1      1       1       1
2      1       2      1       1       1
3      1       3      1       1       1
4      1       4     NA       1       1
5      1       5      1      -1       1
6      1       6      2      -1       1
```

Indexing

```
data[1,1]
data[1,]
data[,1]
data[1:10,]
data[,1:2]
data[1:10, 1:2]
data[c(1,3,5,6), c(2,4)]

data$choice
```

```
> data
   subjID trialID choice outcome correct
1       1       1      1       1       1
2       1       2      1       1       1
3       1       3      1       1       1
5       1       5      1      -1       1
6       1       6      2      -1       1
7       1       7      1       1       1
8       1       8      1       1       1
9       1       9      1       1       1
10      1      10      1       1       1
11      1      11      1       1       1
```

Import some data!

```
data_dir = ('_data/RL_raw_data/sub01/raw_data_sub01.txt')
data = read.table(data_dir, header = T, sep = ",")
head(data)
```

```
  subjID trialID choice outcome correct
1      1       1      1       1       1
2      1       2      1       1       1
3      1       3      1       1       1
4      1       4     NA       1       1
5      1       5      1      -1       1
6      1       6      2      -1       1
```

```
sum(complete.cases(data))   # number of valid trials
data = data[complete.cases(data),]
dim(data[complete.cases(data),])
```

Exercise III

`…/01.R_basics/_scripts/R_basics.R`

TASK:

write a for loop

… which reads in each participant's raw data

… and reshape it in the "long format" by subj

TIP: complete line 173; consider `sprintf()`

```
for ( j in 1:n ) {
   read.table(file, header = T, sep = ",")
}
```

subID	Choice
sub01	1
sub01	2
…	
sub02	2
sub02	2
…	
sub10	2
sub10	1

Read all the data!

```
ns = 10
data_dir = '_data/RL_raw_data'

rawdata = c();
for (s in 1:ns) {
    sub_file = file.path(data_dir, sprintf('sub%02i/raw_data_sub%02i.txt',s,s))
    sub_data = read.table(sub_file, header = T, sep = ",")
    rawdata = rbind(rawdata, sub_data)
}
rawdata = rawdata[complete.cases(rawdata),]
rawdata$accuracy = (rawdata$choice == rawdata$correct) * 1.0

acc_mean = aggregate(rawdata$accuracy, by = list(rawdata$subjID), mean)[,2]
```

mean choice accuracy across trials, per participant.

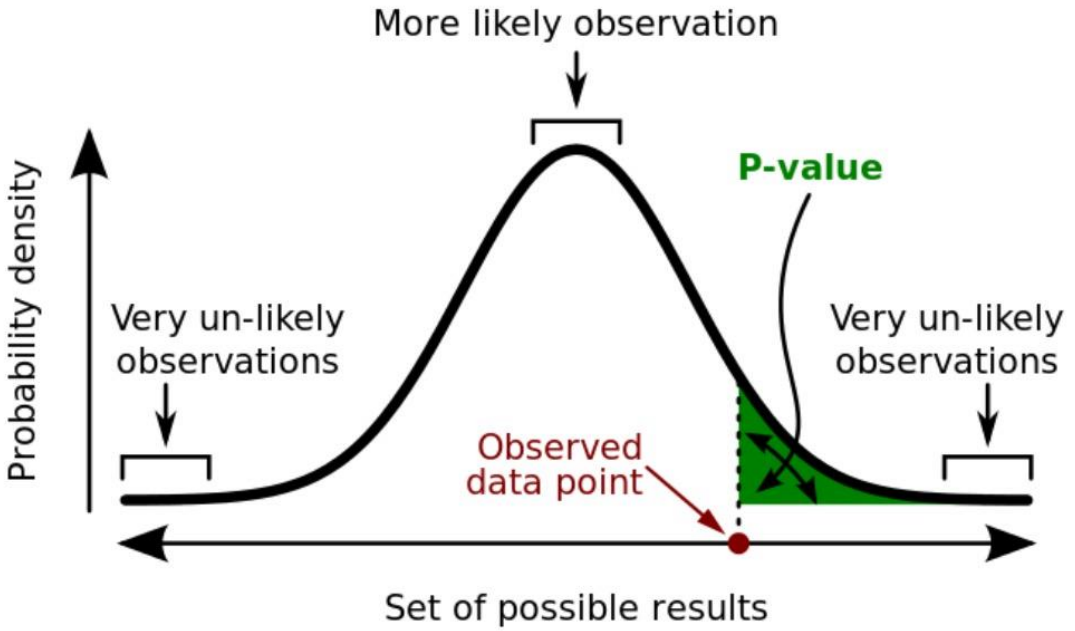
STATS REFRESHER

Null Hypothesis Significance Testing (NHTS)

- NHST is a method of statistical inference by which an experimental factor is tested against a hypothesis of no effect or no relationship based on a given observation.
- the *p*-value is used to reject the null hypothesis
- what is the *p*-value?

(A) The probability of failing to reject the null hypothesis, given the observed results.
(B) The probability that the null hypothesis is true, given the observed results.
(C) The probability of observing results as extreme or more extreme than currently observed, given that the null hypothesis is true.
(D) The probability that the observed results are statistically significant, given that the null hypothesis is true.

p-value



A **p-value** (shaded green area) is the probability of an observed (or more extreme) result assuming that the null hypothesis is true.

Example of a *p*-value computation. The vertical coordinate is the probability density of each outcome, computed under the null hypothesis. The *p*-value is the area under the curve past the observed data point.

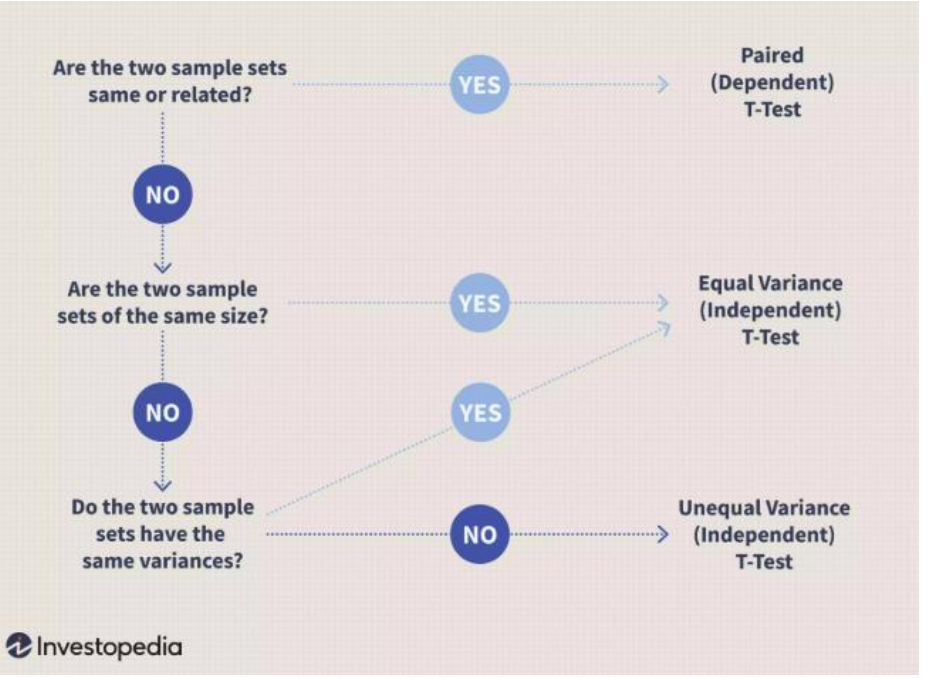
Type I, Type II errors

Decision made using inferential statistic	The reality (H_0 is either true or false)	
	H_0 is true	**H_0 is false**
Reject H_0	Type I error probability = α	Correct decision probability = $1-\beta$ = Power
Do not reject H_0	Correct decision probability = $1-\alpha$	Type II error probability = β

	Null hypothesis is TRUE	Null hypothesis is FALSE
Reject null hypothesis	Type I Error (False positive)	Correct outcome! (True positive)
Fail to reject null hypothesis	Correct outcome! (True negative)	Type II Error (False negative)

t-test

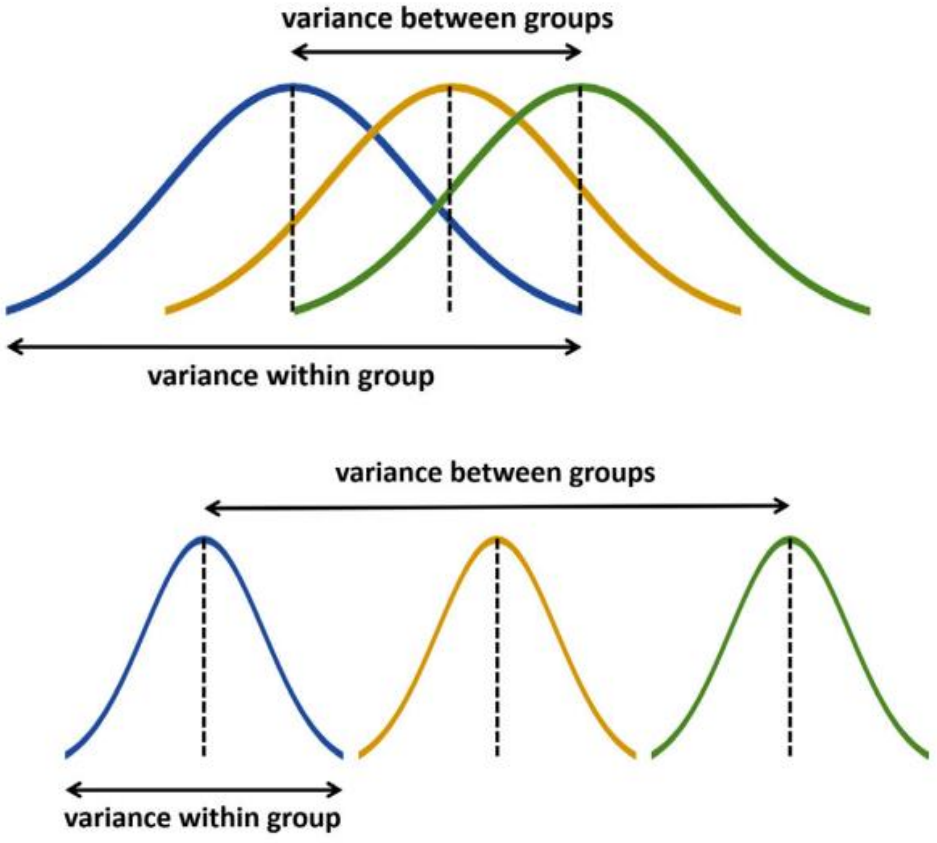
- one sample *t*-test
  - e.g., grades of a class is higher than the average of the entire university
- independent sample *t*-test
  - e,g., IQ performance between the "funny" condition and the "ordinary" condition; appropriate for between-subject designs, when there are two groups
- paired *t*-test
  - e.g., learning performance at the beginning vs. at the end of the semester; appropriate for within-subject designs, when there are two conditions

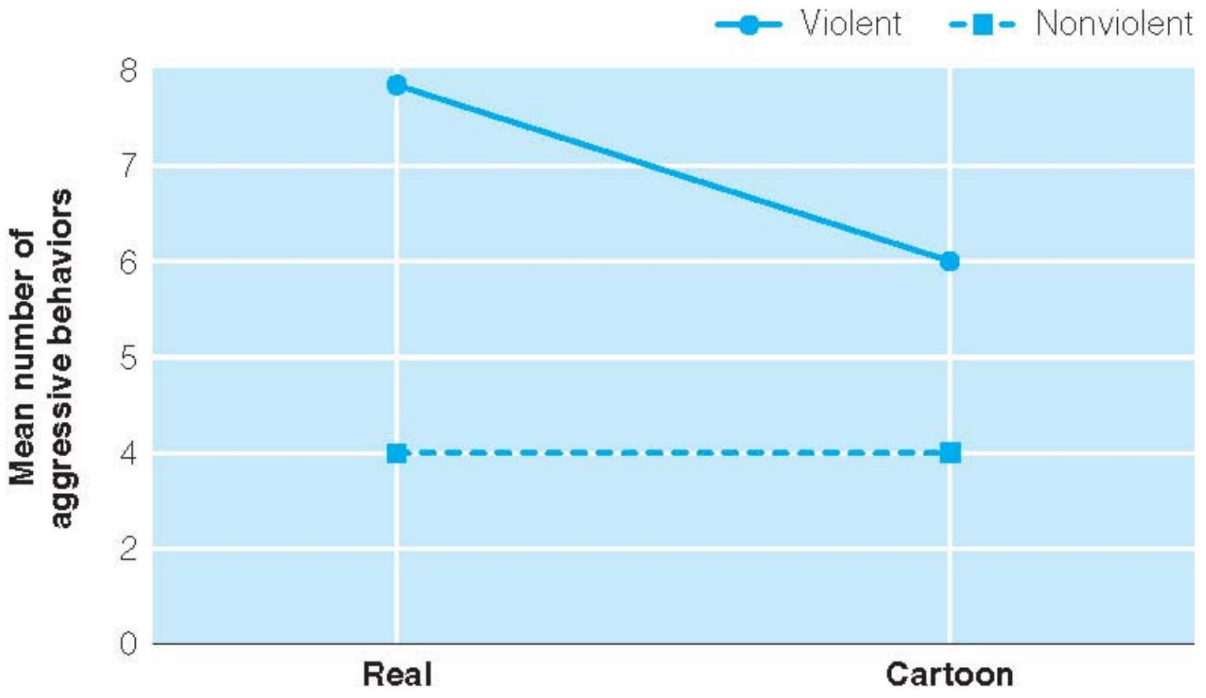


More reading: https://opentext.wsu.edu/carriecuttler/chapter/13-2-some-basic-null-hypothesis-tests/

Analysis of Variance (ANOVA)

- When there are more than two groups or condition means to be compared, the most common null hypothesis test is the analysis of variance (ANOVA).

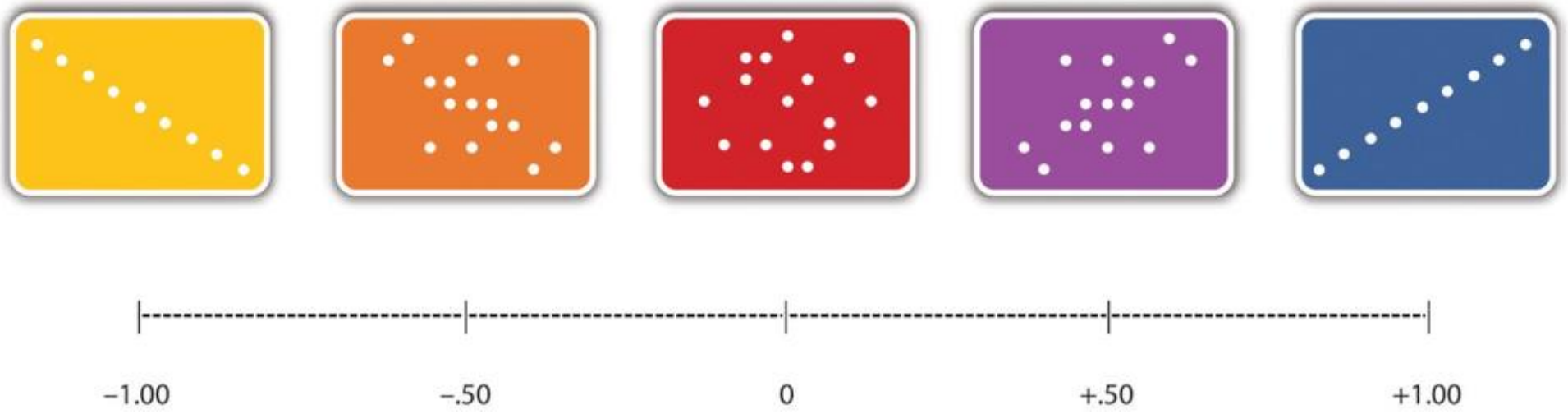




Lammers & Badia (2013)

Correlation

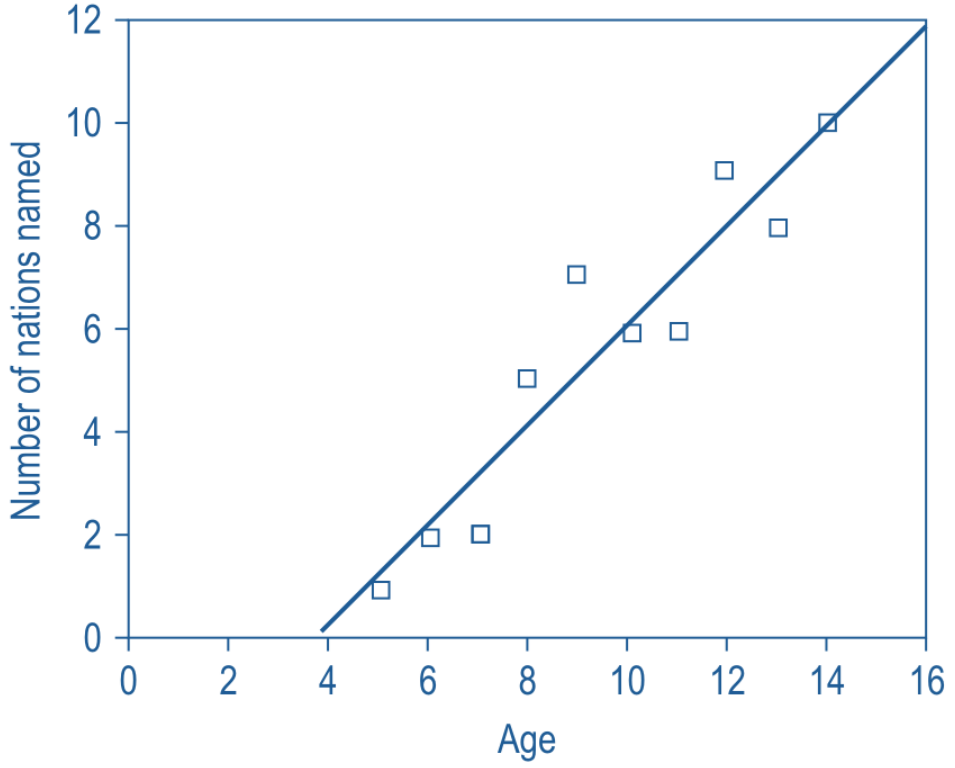
- The degree of relationship between two variables is calculated as a statistic called a correlation index (e.g., Pearson's *r*), which can vary from −1 (a perfect negative relationship), through zero (no relationship), to +1 (a perfect positive relationship).



Lammers & Badia (2013)

Simple liner regression

- Appropriate when we are interested to make *predictions*: (i) how much one variable will change on the basis of known changes on another variable, and (ii) how a particular individual with a given score on one variable will score on another variable.



$$\#\ \text{of nations} = -3.15 + 0.97 \times \text{Age}$$

Sani & Todman (2006)

Basic stats

```
> as.matrix(acc_mean, 10, 1)
          [,1]
 [1,] 0.8076923
 [2,] 0.7125000
 [3,] 0.6875000
 [4,] 0.6493506
 [5,] 0.7750000
 [6,] 0.7250000
 [7,] 0.7662338
 [8,] 0.8000000
 [9,] 0.7500000
[10,] 0.6750000
```

```
mean(acc_mean)
sd(acc_mean)
sem(acc_mean)

t.test(acc_mean, mu = 0.5) # one sample t-test

        One Sample t-test

data:  acc_mean
t = 13.788, df = 9, p-value = 2.34e-07
alternative hypothesis: true mean is not equal to 0.5
95 percent confidence interval:
 0.6962988 0.7733565
sample estimates:
mean of x
0.7348277
```

Basic correlation

```
> descriptive
   subjID        IQ      Age    acc
1       1 123.98691 31.07218 0.8125
2       2  87.63187 30.13800 0.7125
3       3  89.39930 23.44219 0.6875
4       4  84.34607 27.44848 0.6500
5       5 134.72208 23.30624 0.7750
6       6  84.60797 25.67858 0.7250
7       7 111.10238 24.36375 0.7750
8       8 117.89599 32.74026 0.8000
9       9  96.88233 22.80211 0.7500
10     10  76.01652 30.44258 0.6750
```

```
load('_data/RL_descriptive.RData')
descriptive$acc = acc_mean
df = descriptive

cor.test(df$IQ, df$acc)

        Pearson's product-moment correlation

data:  df$IQ and df$acc
t = 4.8347, df = 8, p-value = 0.001297
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.5114810 0.9671586
sample estimates:
      cor
0.8631401
```

Exercise IV

```
…/01.R_basics/_scripts/R_basics.R
```

TASK:

Read in the descriptive data: _data/descriptive.RData

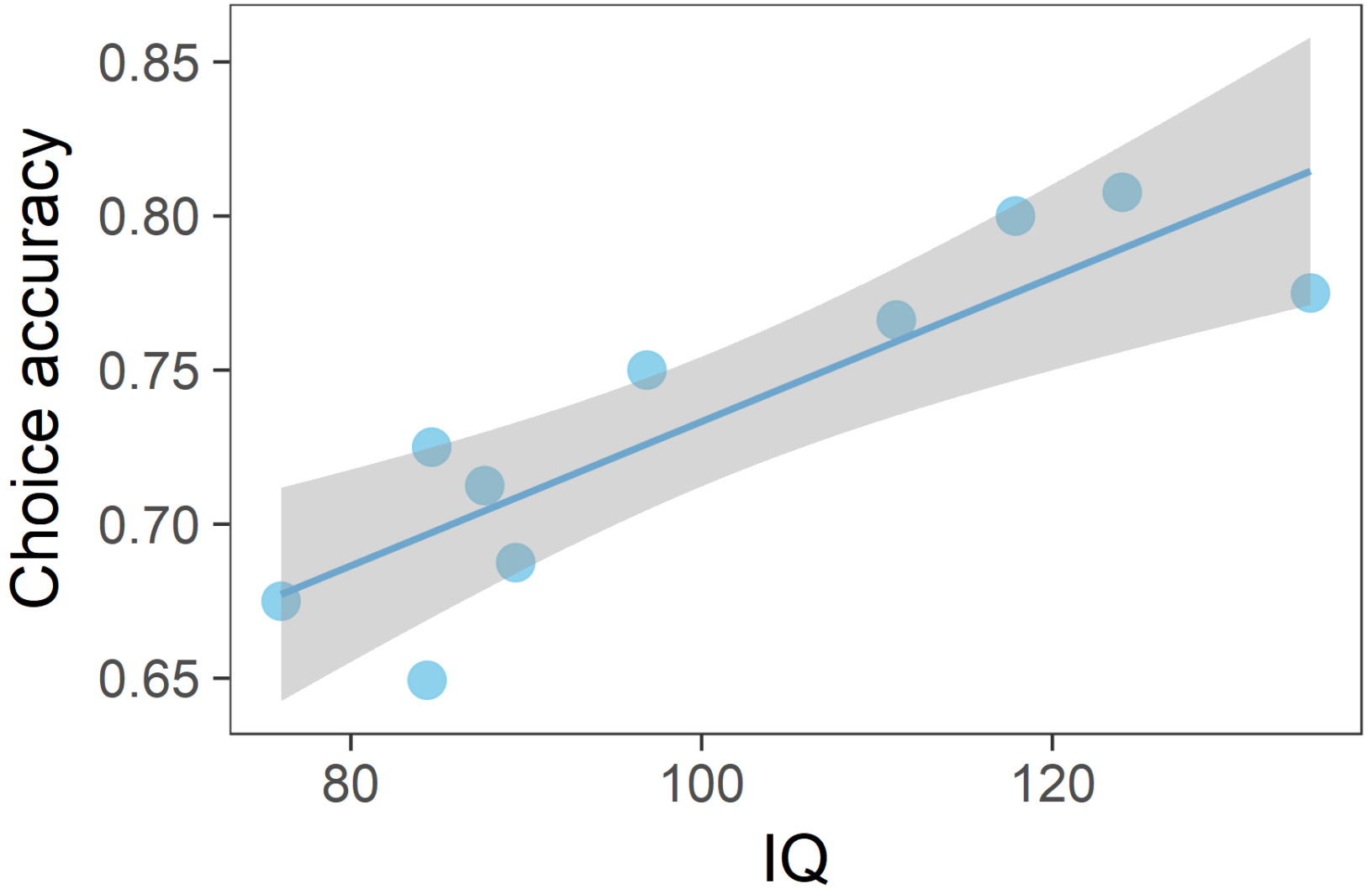
…include ‘acc_mean’ as a new column, and

…rename ‘descriptive’ as df.

Practice all the basic stats.

```
df$new_Col = new_Col
```

A simple linear regression



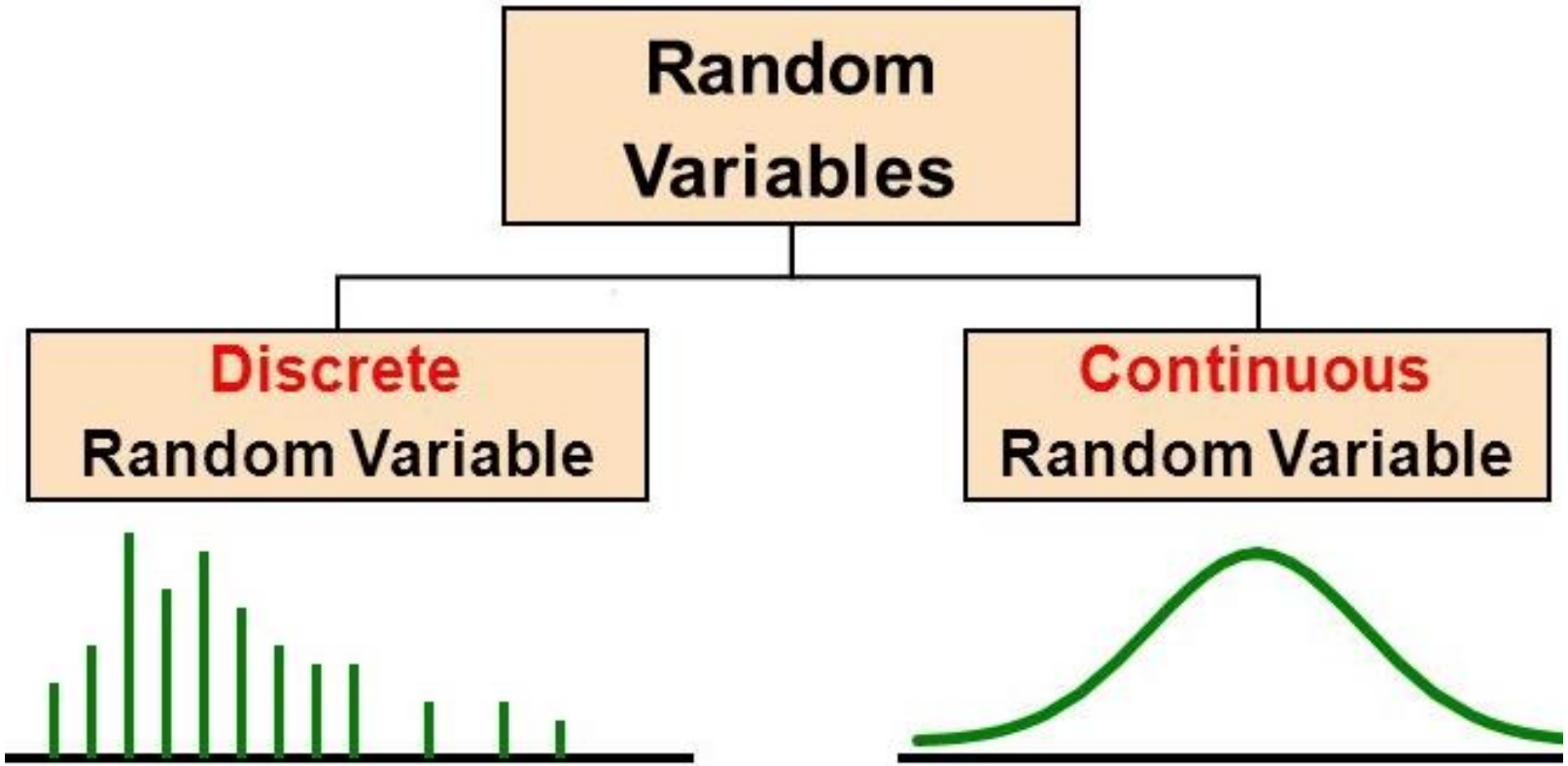
What is exactly the regression line in R?

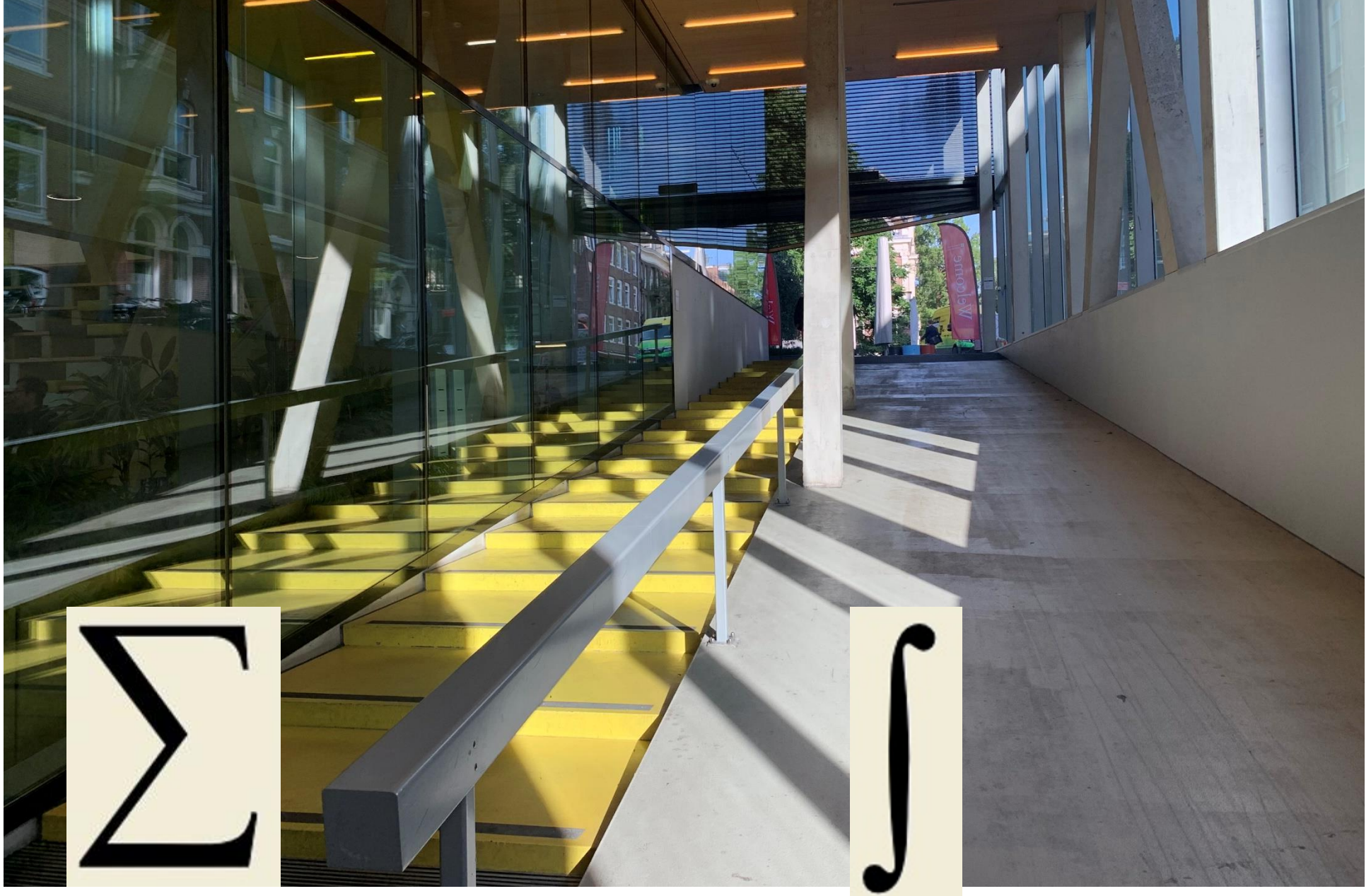
$$\mu_i = \alpha + \beta x_i$$

$$y_i = \mu_i + \varepsilon$$

```
fit1 = lm(acc ~ IQ, data = df)
summary(fit1)

Call:
lm(formula = acc ~ IQ, data = df)

Residuals:
      Min        1Q    Median        3Q       Max
-0.047305 -0.016277  0.007562  0.022577  0.027731

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.499292   0.049565  10.073 8.04e-06 ***
IQ          0.002340   0.000484   4.835   0.0013 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.02885 on 8 degrees of freedom
Multiple R-squared:  0.745,  Adjusted R-squared:  0.7131
F-statistic: 23.37 on 1 and 8 DF,  p-value: 0.001297
```

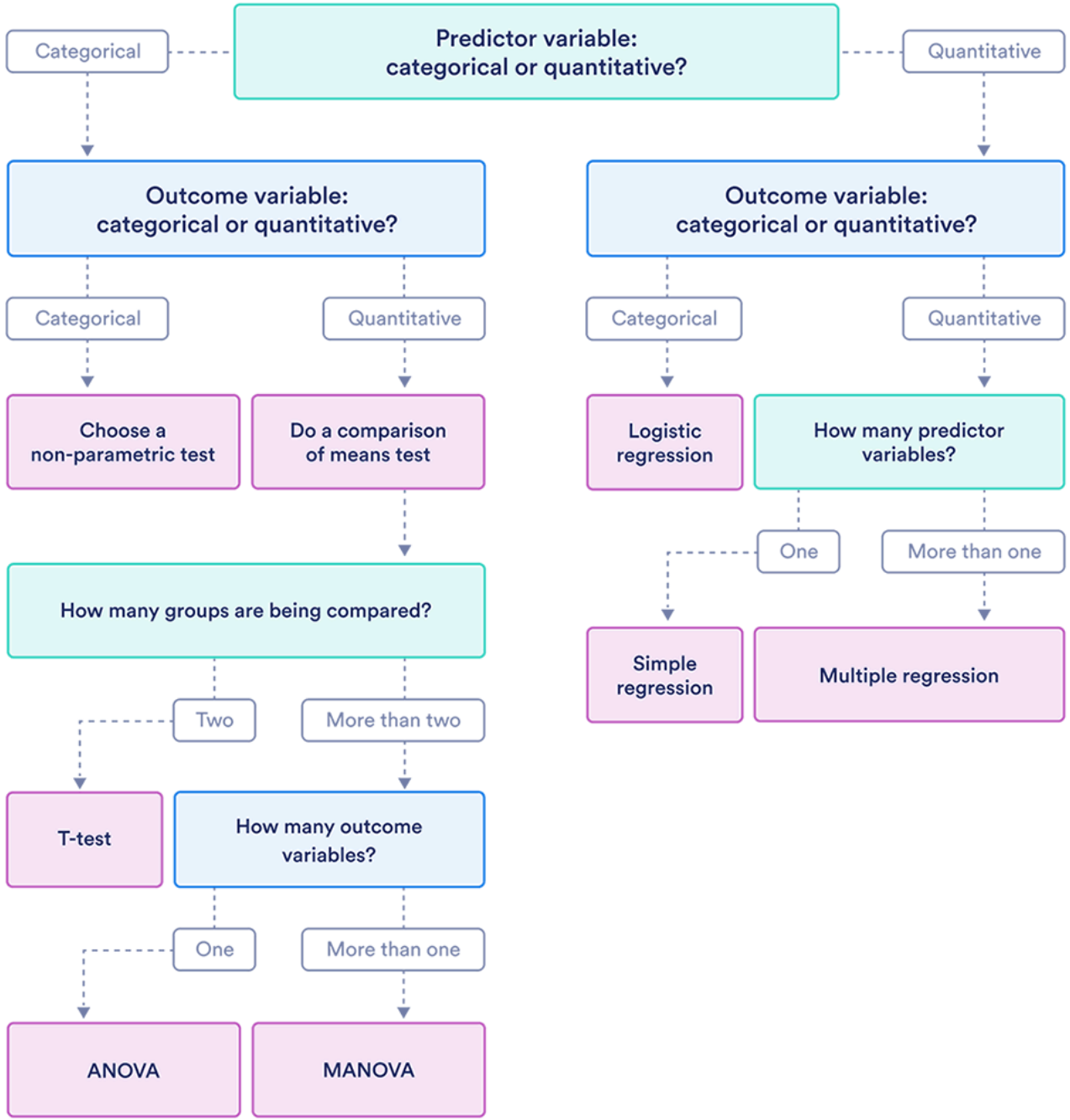
Probability Functions





$\sum$

$\int$


Predictor variable:
categorical or quantitative?
Categorical
Quantitative
Outcome variable:
categorical or quantitative?
Categorical
Quantitative
Choose a
non-parametric test
Do a comparison
of means test
How many groups are being compared?
Two
More than two
T-test
How many outcome
variables?
One
More than one
ANOVA
MANOVA
Outcome variable:
categorical or quantitative?
Categorical
Quantitative
Logistic
regression
How many predictor
variables?
One
More than one
Simple
regression
Multiple regression

Exercise V

```
…/01.R_basics/_scripts/R_basics.R
```

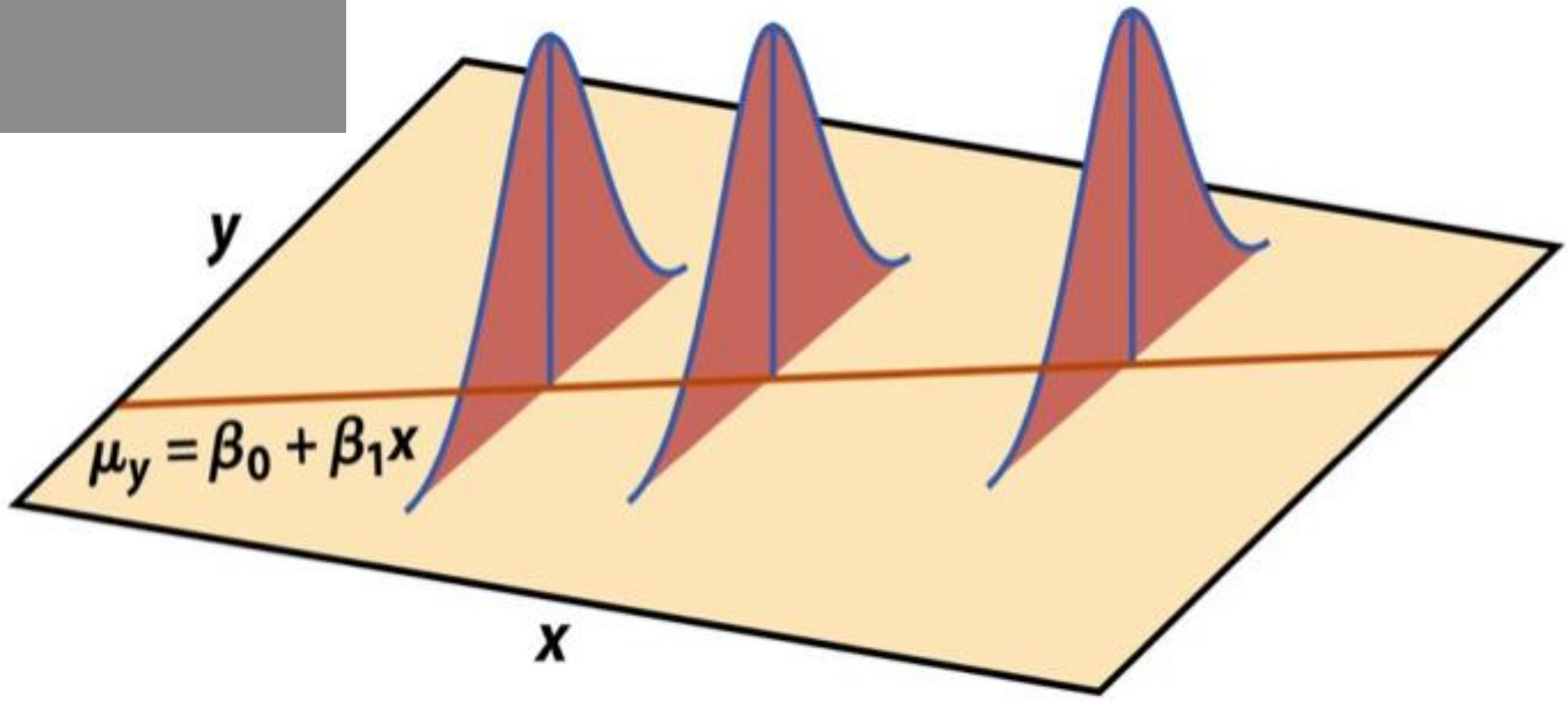
TASK:

Read and make sense of the ggplot functions,

… experiment make some adjustments (color marker size etc.), and

… run the `lm(acc ~ IQ)`

LINEAR REGRESSION

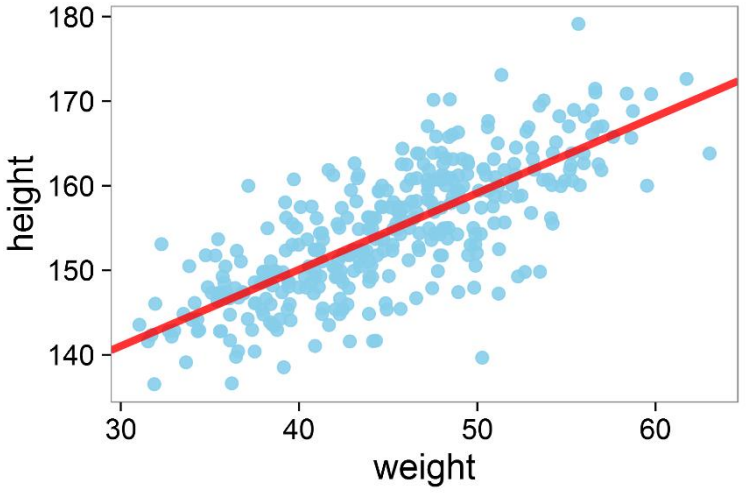


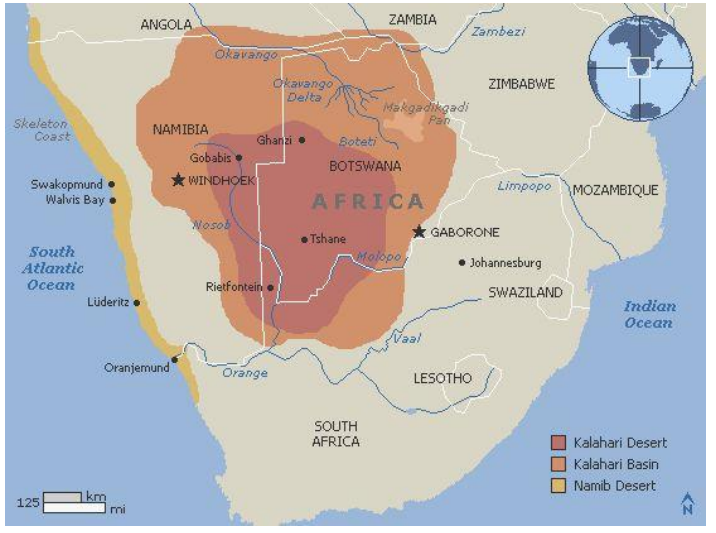
Linear Regression: height ~ weight

`…/04.regression_height/_scripts/regression_height_main.R`

make scatter plot and fit the model with `lm()`

```
>load('_data/height.RData')
>d <- Howell1
>d <- d[ d$age >= 18 , ]
>head(d)
height   weight age male
1 151.765 47.82561  63    1
2 139.700 36.48581  63    0
3 136.525 31.86484  65    0
4 156.845 53.04191  41    1
5 145.415 41.27687  51    0
6 163.830 62.99259  35    1
```





McElreath (2016)

Results with lm()

```
> L <- lm( height ~ weight, d) # estimate model by minimizing least squares errors
> summary(L)

Call:
lm(formula = height ~ weight, data = d)

Residuals:
     Min       1Q   Median       3Q      Max
-19.7464  -2.8835   0.0222   3.1424  14.7744

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 113.87939    1.91107   59.59   <2e-16 ***
weight        0.90503    0.04205   21.52   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.086 on 350 degrees of freedom
Multiple R-squared:  0.5696, Adjusted R-squared:  0.5684
F-statistic: 463.3 on 1 and 350 DF,  p-value: < 2.2e-16
```

Rethinking Regression Model

$$\mu_i = \alpha + \beta x_i$$

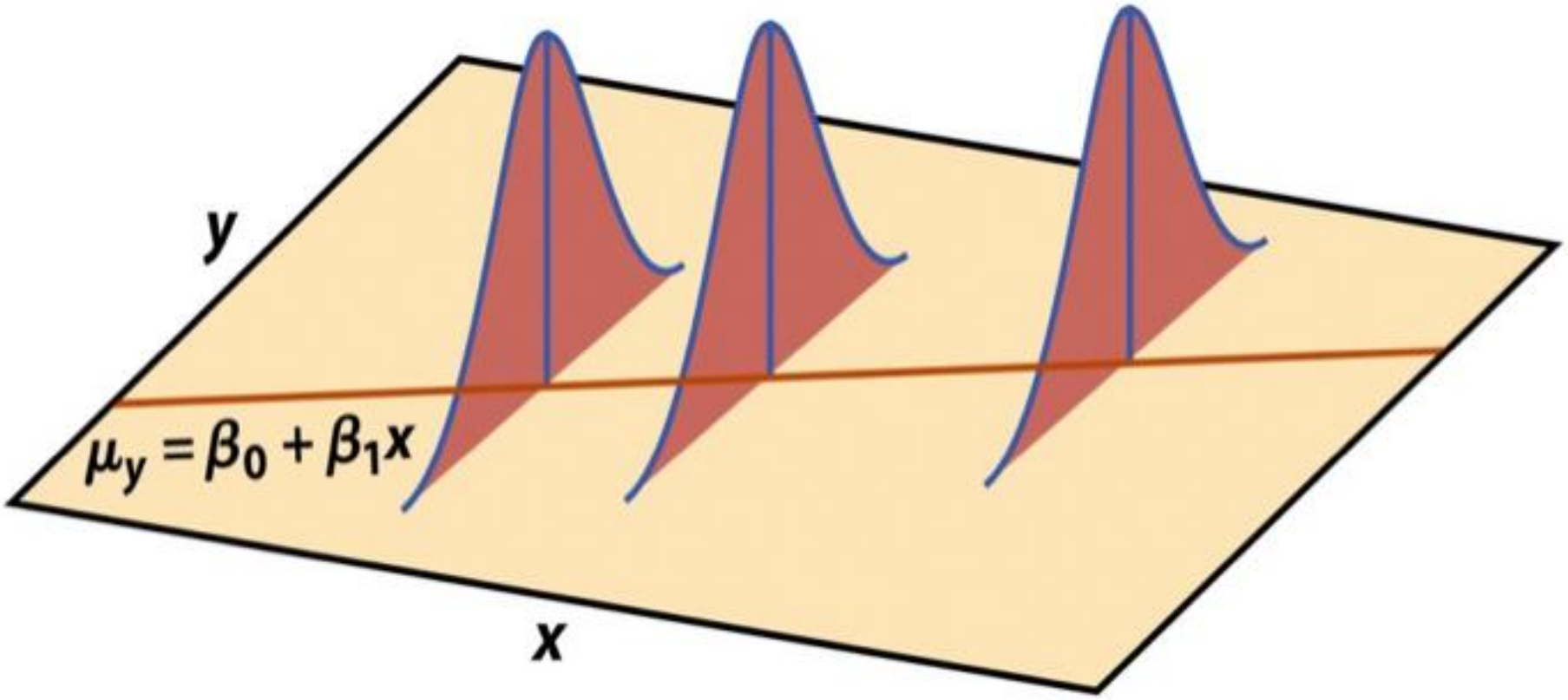
~~$$y_i = \mu_i + \varepsilon$$~~

~~$$\varepsilon \sim Normal(0, \sigma)$$~~

$$y_i \sim Normal(\mu_i, \sigma)$$

Rethinking Regression Model

$$\mu_i = \alpha + \beta x_i$$
$$y_i \sim Normal(\mu_i, \sigma)$$



Bishop (2006)

Demetri
@PhDemetri

I wish linear regression was never taught as

$y = a + bx + e$

and was instead taught as

$y \sim Normal(Xb, sigma)$

Because then explaining and learning GLM would be easier

1:19 AM · Apr 12, 2021 · Twitter Web App

SCIENCE FORUM

Ten common statistical mistakes to watch out for when writing or reviewing a manuscript

Abstract Inspired by broader efforts to make the conclusions of scientific research more robust, we have compiled a list of some of the most common statistical mistakes that appear in the scientific literature. The mistakes have their origins in ineffective experimental designs, inappropriate analyses and/or flawed reasoning. We provide advice on how authors, reviewers and readers can identify and resolve these mistakes and, we hope, avoid them in the future.

TAMAR R MAKIN* AND JEAN-JACQUES ORBAN DE XIVRY

https://elifesciences.org/articles/48175

After the Workshop, you… (aka. learning outcome)

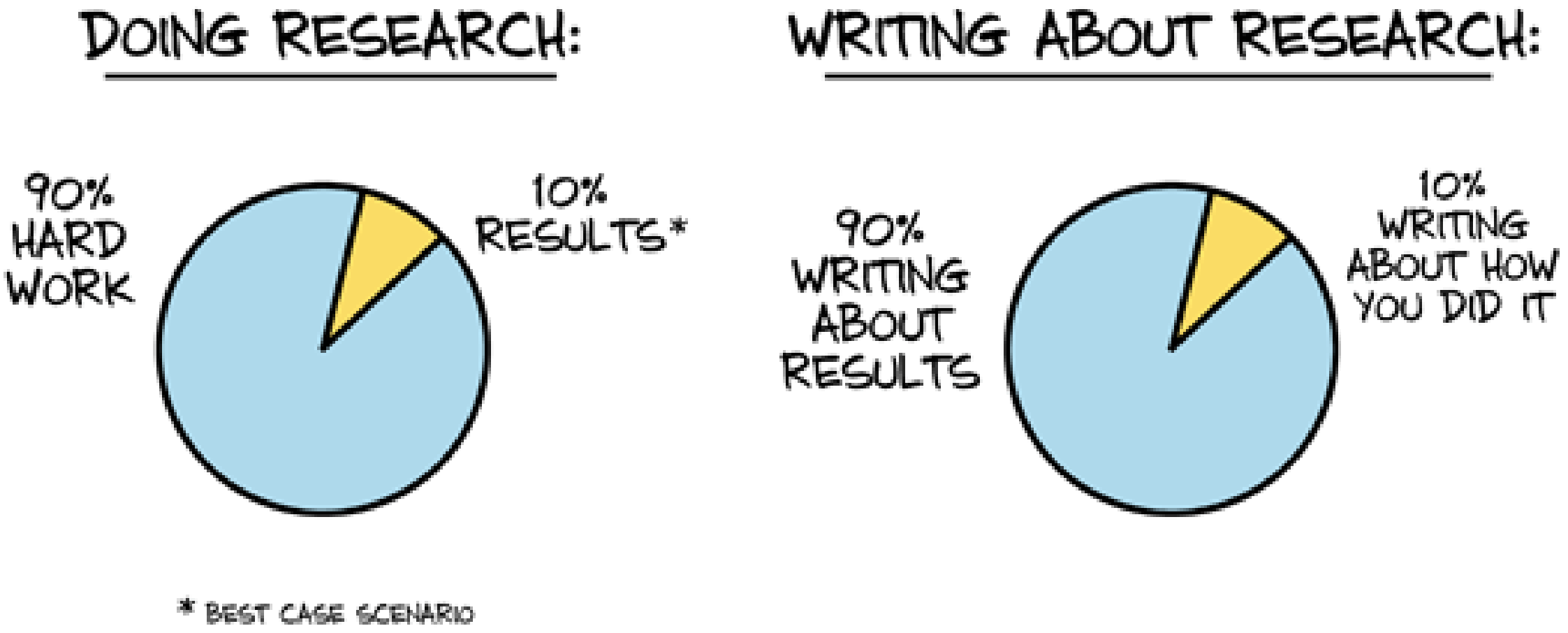
…are able to implement your own analysis

…consider the implementation of the “Statistical analysis” section

…feel comfortable with reading statistical methods

…gain insightful understanding of stats and modeling

…take it as a good start and work on it later


DOING RESEARCH:
90% HARD WORK
10% RESULTS*
* BEST CASE SCENARIO
WRITING ABOUT RESEARCH:
90% WRITING ABOUT RESULTS
10% WRITING ABOUT HOW YOU DID IT
JORGE CHAM © 2016
WWW.PHDCOMICS.COM

Richard McElreath
@rlmcelreath

I say this a lot, bc I am also confused quite often.

> 
> **Anna Jacobson** @AnnaChingChing · Feb 21
>
> "If you are confused, it is only because you are trying to understand." - @rlmcelreath in Statistical Rethinking

Last, learning can be fun!

https://www.youtube.com/@statquest


ANY
QUESTIONS
?

Happy Computing!