University of Zurich[UZH]

Department of Geography

GEO 812
Getting started with R for Spatial Analysis

Session 1: Data exploration

Peter Ranacher

September 2019

Learning objectives

You are able to

- perform basic operations in R
- visualize data
- transform and explore data

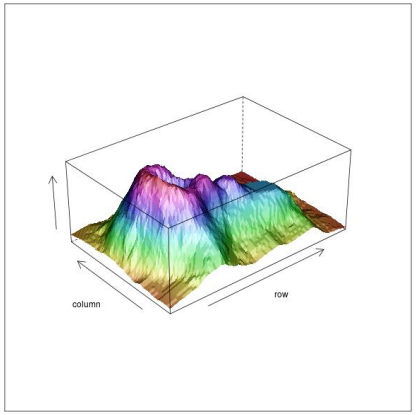
What is programming?

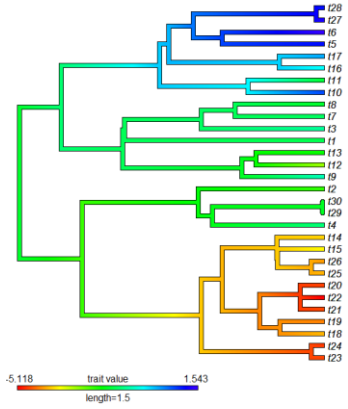
Giving a sequence of instructions to perform a specific task on a computer

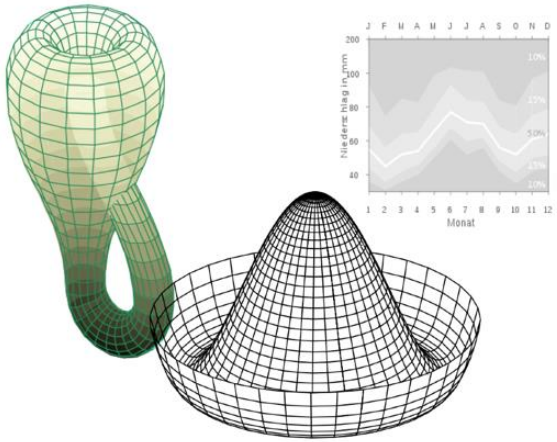
The “obedient dog” metaphor

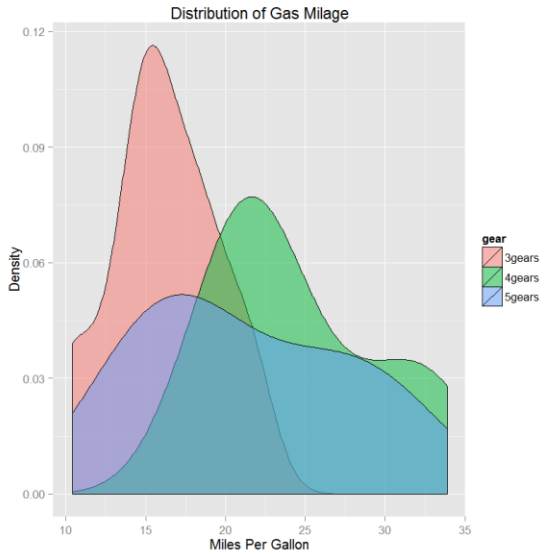
What is R?

- Programming language and software environment for statistical analysis
- Super popular in science
- Open, independent and **free**

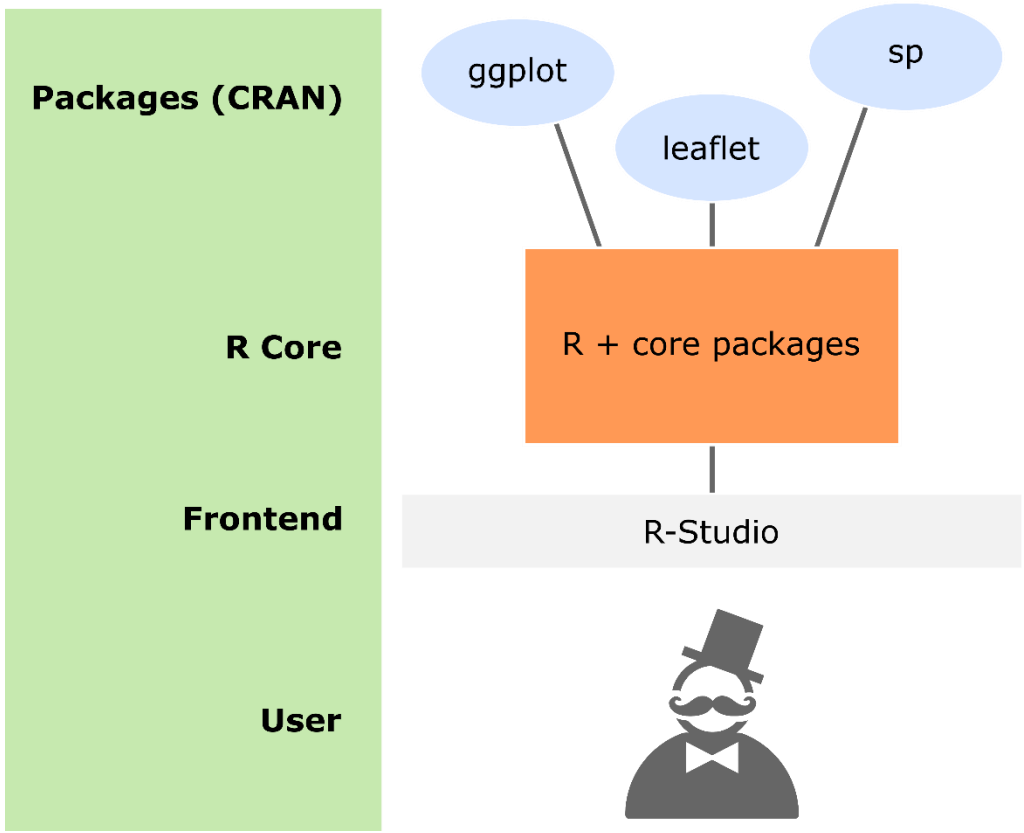






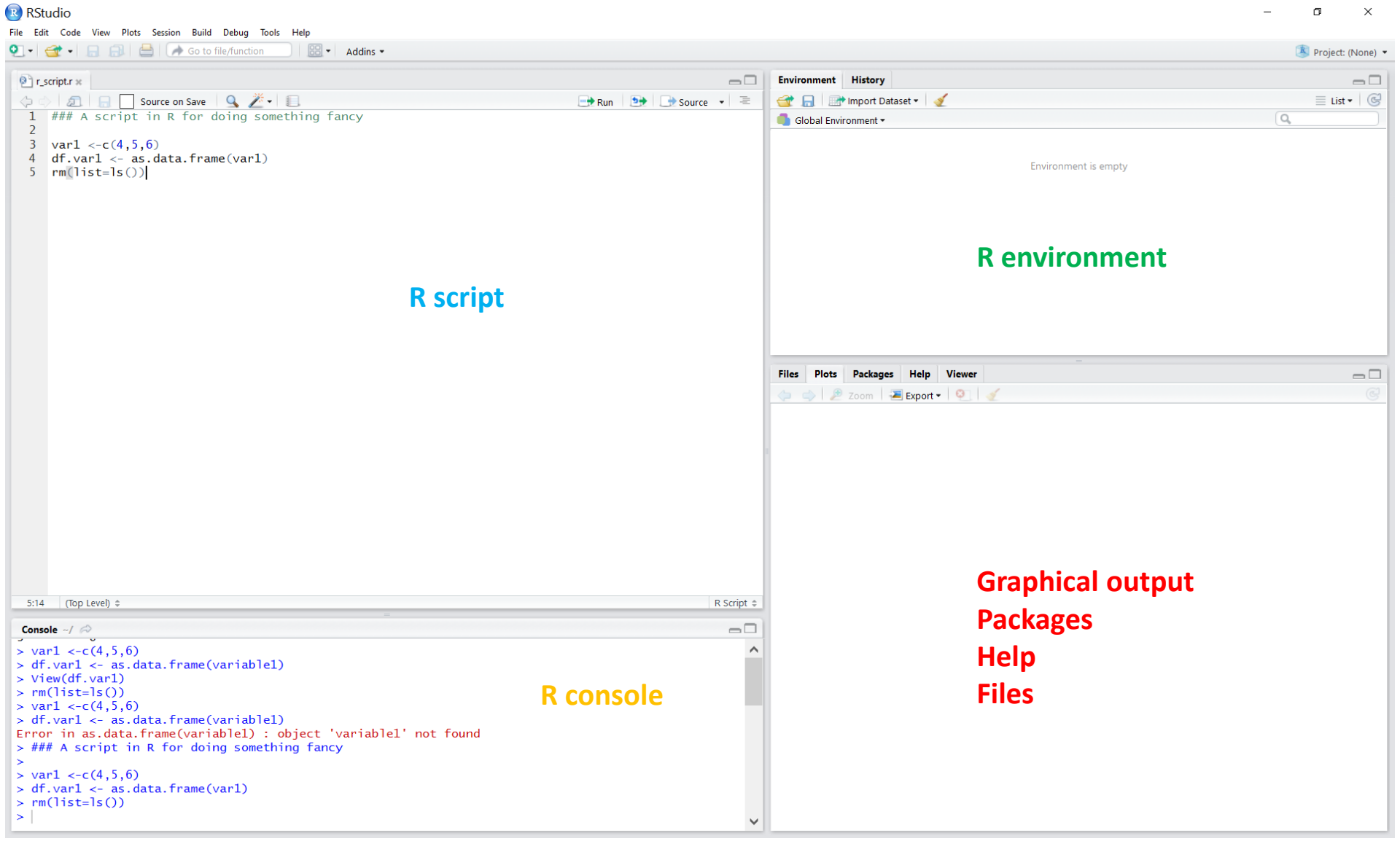


The architecture of R



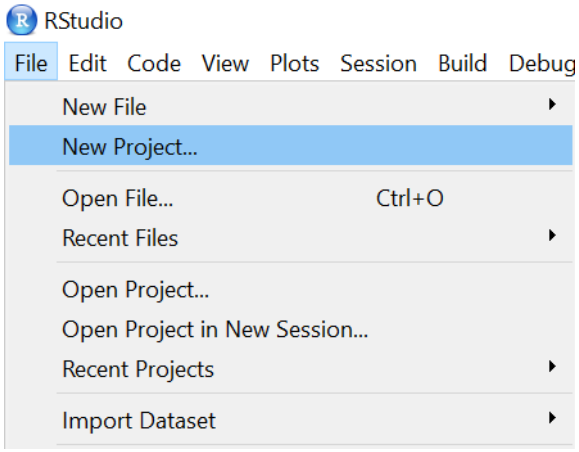
http://www.ats.ucla.edu/stat/r/seminars/intro.htm

Your interface to R: Welcome to R studio!

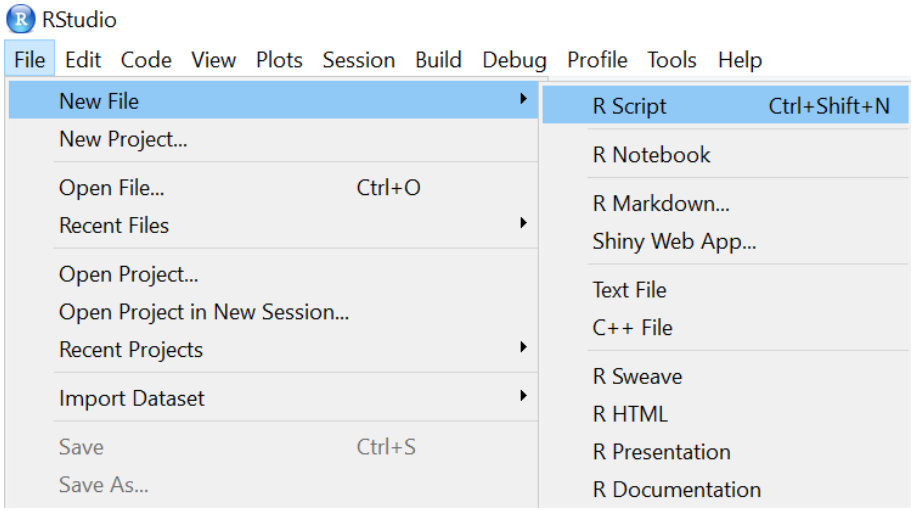


Before we start…

Create a new project



Open a new R script

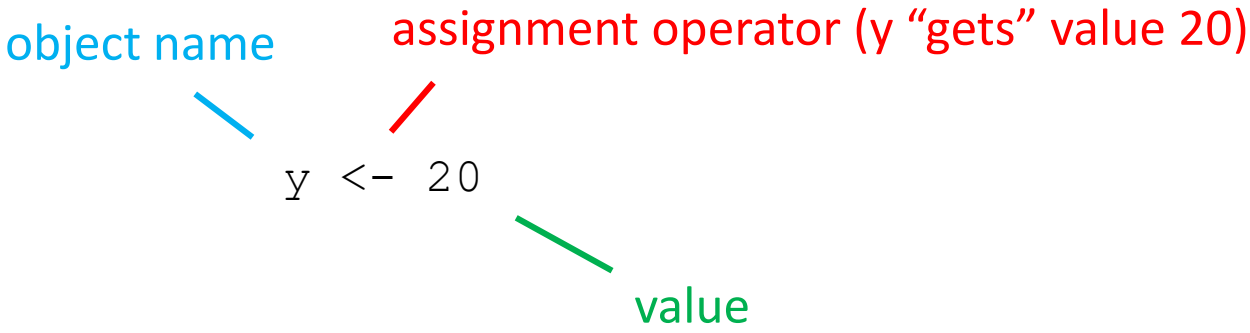


Coding Basics

R can be used as a calculator

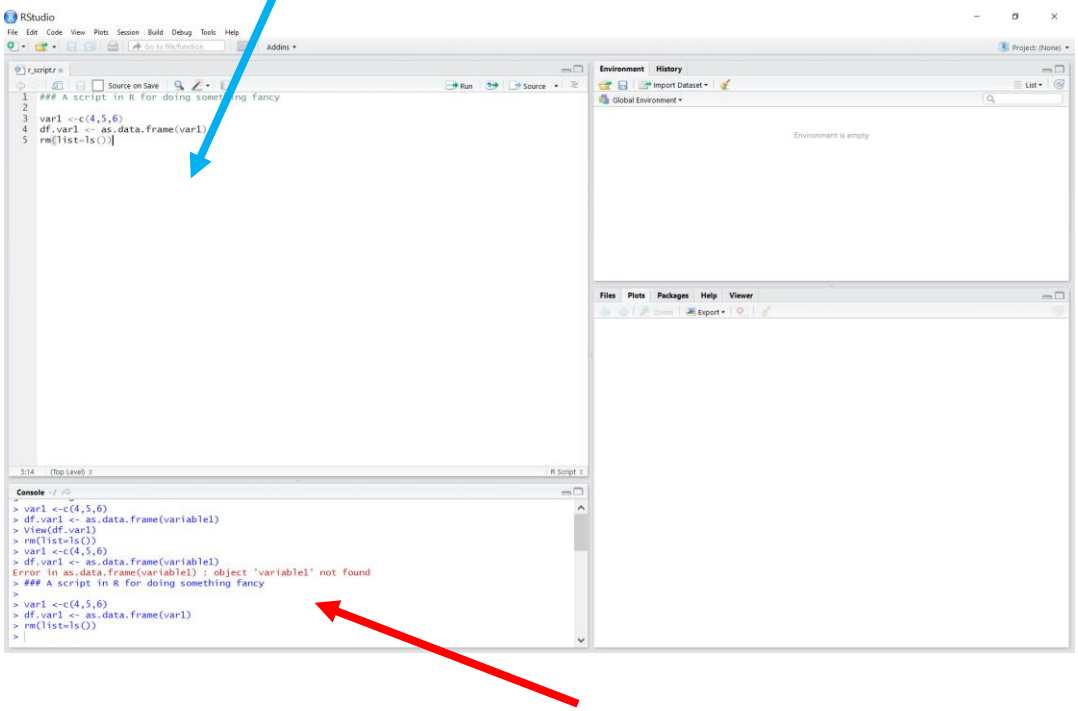
```
(3 + 15)/6
sin(pi / 2)
```

R can create new objects



Where do I write code?

Here.



Not here!

Write code in R script.

Highlight.

Click 

(or Ctrl + Enter).

A quick look at the R environment

Includes all the user-defined objects

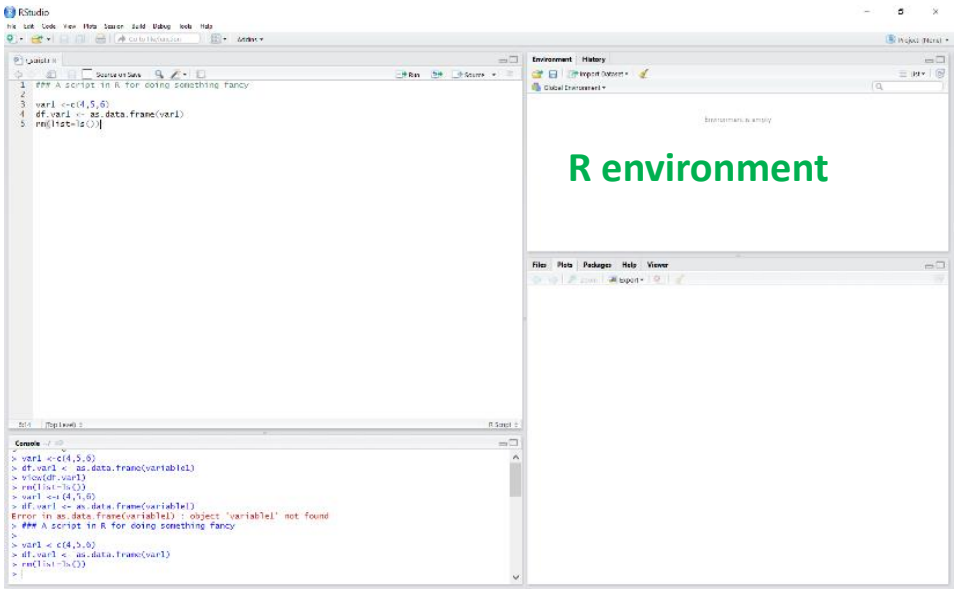
(functions, data, values, …)

List all the objects in the workspace

`ls()`

Remove a particular object

`rm(y)`



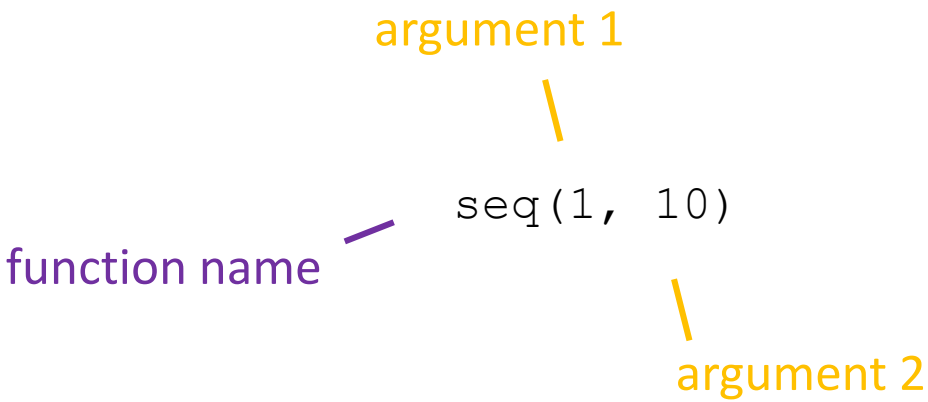
Naming things in R

Names

- consist of a combination of letters, numbers and _ and .
- must start with a letter
- are case sensitive

`some_name` ≠ `Some_Name` ≠ `SOME_NAME`

Calling functions



Specify arguments by complete name

```
seq(from = 1, to = 10)
```

Assign output of a function to an object

```
x <- seq(1, 10, by = 2)
```

How are arguments specified in `seq(1, 10)`?

The tidyverse

Collection of R packages for data science

Common design philosophy, grammar, and data structure

Tidy? → work in harmony!

Install and load the tidyverse packages

```
install.packages("tidyverse")
```

Only run once

```
library(tidyverse)
```

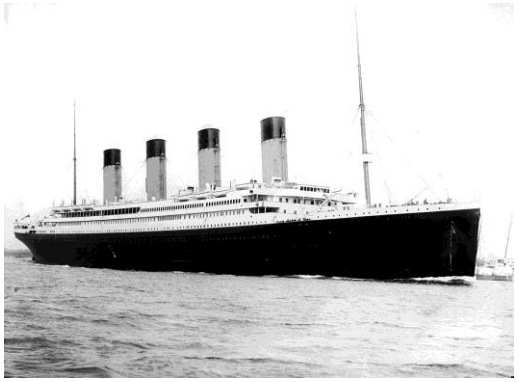
Reload in every new session

The msleep and TitanicSurvival data

`msleep`

Sleep times and weights of 83 mammals

part of the tidyverse data



```
install.packages("carData")
library(carData)
TitanicSurvival
```

Survival status of the 1309 passengers of the Titanic

Try `?msleep` and `?TitanicSurvival`

Data frames

variable

column names

row names

observation

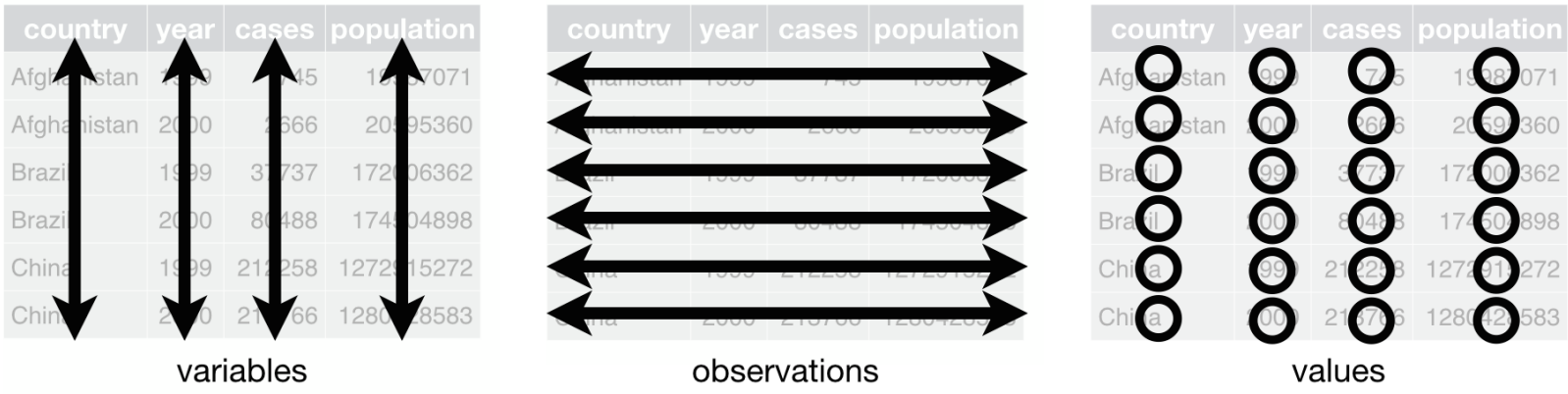
	survived	sex	age	passengerClass
Allen, Miss. Elisabeth	yes	female	29	1st
Allison, Master. Hudson	yes	male	1	1st
Allison, Miss. Helen	no	female	2	1st
Allison, Mr. Hudson	no	male	30	1st
Allison, Mrs. Hudson	no	female	25	1st

Rectangular collection of observations (rows) and variables (columns)

Entries in one column are of the same type (e.g. integer, character)

Keep your data frames tidy!

1. Each variable must have its own column.
2. Each observation must have its own row.
3. Each value must have its own cell.



Tibbles

Equivalent of data frames in the tidyverse

Only small differences, eg.:

- no row names in tibbles,
- better suited for previewing large data sets

From data frame to tibble

```
titanic_survival <- as_tibble(TitanicSurvival)
```

From tibble to data frame

```
as.data.frame(msleep)
```

Plotting tibbles with

creates a coordinate system that you can add layers to

data set to use in the graph

specifies which variables to map to the x and y axis

```
ggplot(data = msleep) +
  geom_point(mapping = aes(x = sleep_total, y = sleep_rem))
```

columns in the data set mapped to the x and y axis

adds a layer of points to the plot (scatterplot)

Defines how variables in the data set are mapped to visual properties

What does the warning message mean and why does it appear?

Aesthetic mappings

Aesthetic = a visual property of the objects in the plot, for example

color, size, alpha

column in the data set mapped to color

Mapping to color

```
ggplot(data = msleep) +
  geom_point(mapping = aes(sleep_total, sleep_rem, color = vore))
```

column in the data set mapped to size

Mapping to size

```
ggplot(data = msleep) +
  geom_point(mapping = aes(sleep_total, sleep_rem, size = bodywt))
```

Bar charts

adds a bar chart to the plot

```
ggplot(data = titanic_survival) +
  geom_bar(aes(x = passengerClass))
```

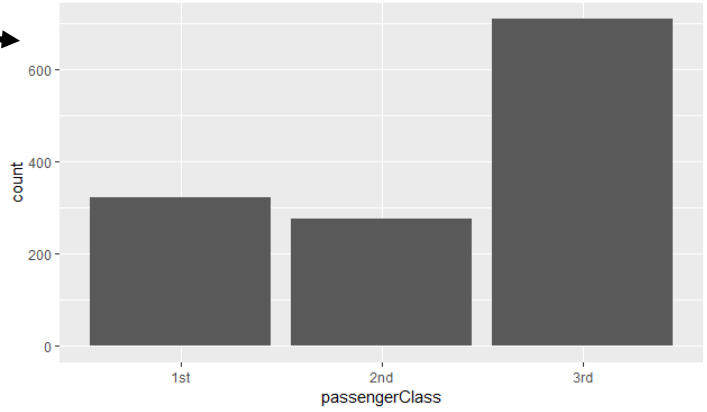
`geom_bar`

- performs a statistical transformation and counts the number of entries per `passengerClass`,
- and then plots these counts

survived	sex	age	passengerClass
yes	female	29	1st
yes	male	1	1st
no	female	2	1st
no	male	30	1st
no	female	25	1st
…	…	…	…

transformation

passengerClass	count
1st	323
2nd	277
3rd	709



Position adjustments

Map the same variable to `fill` → different colors for each bar

```
ggplot(data = titanic_survival) +
  geom_bar(mapping = aes(passengerClass, fill = passengerClass))
```

Map another variable to `fill` → stacked color bars!

column in the data set mapped to fill

```
ggplot(data = titanic_survival) +
  geom_bar(mapping = aes(passengerClass, fill = survived))
```

Add `position = "dodge"` as an argument to `geom_bar`

Exercise 1

1. Show the relationship between body and brain weight in the `msleep` data. Use different symbols (`shape`) to reflect the diet (`vore`). What do you observe?

2. What is wrong (or at least unexpected) with this code? Try to fix it!
```
ggplot(data = msleep) + geom_point(mapping = aes(sleep_total, sleep_rem, color = "blue"))
```

3. Plot the variation in the column `sleep_total` for each `vore` using the function `geom_boxplot`. Check out `?geom_boxplot` if you need to understand the function. Which eating habit corresponds to the lowest variation in total sleep?

Transforming tibbles with

We

- `filter()` observations by their values
- `arrange()` the rows and order the observations
- `select()` variables by their names
- `mutate()` the data and create new variables from existing ones
- collapse many values down to a single value and `summarise()` the data

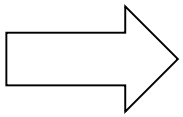
We do the above either on the entire dataset or `group_by()`-group.

`filter()` observations

Select rows by value

```
filter(titanic_survival, age >= 25)
```

survived	sex	age	passengerClass
yes	female	29	1st
yes	male	1	1st
no	female	2	1st
no	male	30	1st
no	female	25	1st

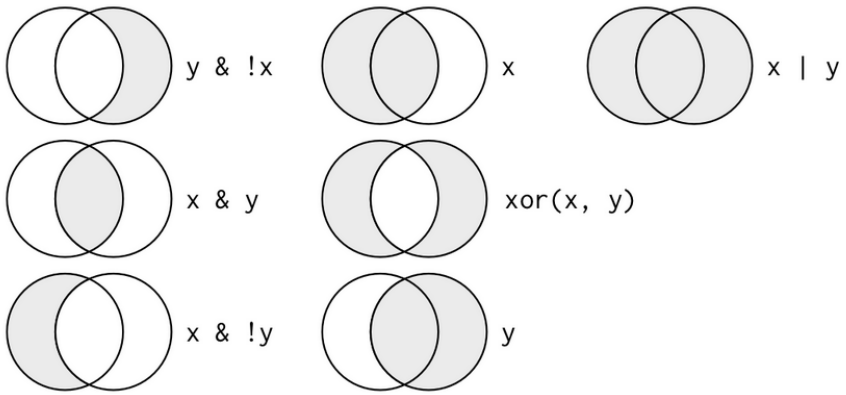


survived	sex	age	passengerClass
yes	female	29	1st
no	male	30	1st
no	female	25	1st

Operators for comparison and logical operators

Operator for comparison	description
==	equal
!=	not equal
<	less than
<=	less than or equal
>	greater than
>=	greater than or equal

Logical operators	description
&	logical AND
|	logical OR
!	logical NOT



Combining filters with logical operators

logical OR

```
filter(msleep, vore == "carni" | vore == "omni")
```

logical AND

```
filter(msleep, vore == "carni" &  sleep_total > 11)
filter(msleep, vore == "carni" ,  sleep_total > 11)
```

Remove all rows with missing values (for one variable!)

```
filter(msleep, !is.na(sleep_rem))
```

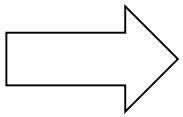
Try `na.omit(msleep)`

`arrange()` the observations

Change the order of the rows

```
arrange(titanic_survival, age)
```

survived	sex	age	passengerClass
yes	female	29	1st
yes	male	1	1st
no	female	2	1st
no	male	30	1st
no	female	25	1st



survived	sex	age	passengerClass
yes	male	1	1st
no	female	2	1st
no	female	25	1st
yes	female	29	1st
no	male	30	1st

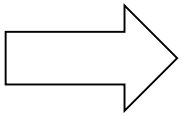
Try `arrange(titanic_survival, desc(age))`

`select()` columns

Pick variables by their names

```
select(titanic_survival, survived, age)
```

survived	sex	age	passengerClass
yes	female	29	1st
yes	male	1	1st
no	female	2	1st
no	male	30	1st
no	female	25	1st



survived	age
yes	29
yes	1
no	2
no	30
no	25

The pipe operator %>%

take the value of that which is to the left

pass it to the right as an argument

First select then filter

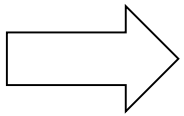
```
select(titanic_survival, survived, passengerClass) %>%
  filter(passengerClass=="2nd")
```

`mutate()` the data

Combine existing variables to create new ones

```
mutate(msleep, rem_ratio = sleep_rem / sleep_total)
```

name	….	sleep_total	sleep_rem
Owl monkey	…	17	1.8
Cow	…	4	0.7
Dog	…	10.1	2.9



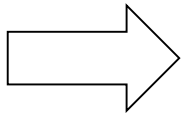
name	….	sleep_total	sleep_rem	rem_ratio
Owl monkey	…	17	1.8	0.106
Cow	…	4	0.7	0.175
Dog	…	10.1	2.9	0.287

`summarize()` the data

Collapse a tibble into a single row

```
summarise(titanic_survival, mean_age = mean(age, na.rm = TRUE))
```

survived	sex	age	passengerClass
yes	female	29	1st
yes	male	1	1st
no	female	2	1st
no	male	30	1st
no	female	25	1st



mean_age
29.9

Why does the following code return `NA`?

```
summarise(titanic_survival, mean_age = mean(age))
```

What does the argument `na.rm` do? Check `?mean` to find out!

`summarize()` the data and `group_by()`

Group the data by a variable and then collapse

```
group_by(titanic_survival, passengerClass) %>%
  summarise(mean_age = mean(age, na.rm = TRUE))
```

survived	sex	age	passengerClass
yes	female	29	1st
yes	male	1	1st
no	female	2	1st
no	male	30	1st
no	female	25	1st

passengerClass	mean_age
1st	39.2
2nd	29.5
3rd	24.8

Useful summary functions

- Measures of location

```
mean(x), median(x)
```

- Measures of spread

```
sd(x), IQR(x), mad(x)
```

- Measures of rank

```
min(x),quantile(x, 0.95), max(x)
```

- Measures of position

```
first(x), nth(x, 2), last(x)
```

- Counts and proportions of logical values

```
n(), n_distinct(x), sum(x>10)
```

Why does `n()` not have an argument?
Use the help function to find out!

Transform and plot in one go using pipes!

data transformation

```
group_by(msleep, order) %>%
  summarise(mean_sleep = mean(sleep_total, na.rm = TRUE),
            mean_bodywt = mean(bodywt, na.rm = TRUE)) %>%
  ggplot() +
  geom_point(mapping = aes(x = mean_sleep, y = mean_bodywt))
```

plotting

Exercise 2

1. What does `%in%` in the following code do? And what is `c()`?

   ```
   filter(titanic_survival, passengerClass %in% (c("2nd","3rd")))
   ```

   How could you rewrite this code using logical operators instead of `%in%` ?

2. Plot the REM ratio (`sleep_rem / sleep_total`) against the body weight, but only for omnivores between 5 and 1000 kilograms. What do you observe?

3. Run the following code:

   ```
   group_by(msleep, vore) %>%
     summarise(mean_brainwt = mean(brainwt, na.rm = TRUE))
   ```

   Why does it still return a value for `NA` even though we set `na.rm = TRUE`?
   Change the code such that `NA` does not appear in the output.

Revisiting the learning objectives

You are able to

- perform basic operations in R
- visualize data
- transform and explore data

We leave the tidyverse for the moment and head off to the jungle, that is R!