Doing things with data frames

Doing things with data frames

Let's go back to our Australian athletes:

```
##
## -- Column specification ----------------------------
## cols(
##   Sex = col_character(),
##   Sport = col_character(),
##   RCC = col_double(),
##   WCC = col_double(),
##   Hc = col_double(),
##   Hg = col_double(),
##   Ferr = col_double(),
##   BMI = col_double(),
##   SSF = col_double(),
##   `%Bfat` = col_double(),
##   LBM = col_double(),
##   Ht = col_double(
```

Choosing a column

```
athletes %>% select(Sport)
```

Sport
Netball
Netball
Netball
Netball
Netball
Netball
Netball
Netball
Netball
Netball
Netball
Netball
Netball

Choosing several columns

```
athletes %>% select(Sport, Hg, BMI)
```

Sport	Hg	BMI
Netball	13.6	19.16
Netball	12.7	21.15
Netball	12.3	21.40
Netball	12.3	21.03
Netball	12.8	21.77
Netball	11.8	21.38
Netball	12.7	21.47
Netball	12.4	24.45
Netball	12.4	22.63
Netball	14.1	22.80
Netball	12.5	23.58
Netball	12.1	20.06
Netball	12.7	23.01

Choosing consecutive columns

```
athletes %>% select(Sex:WCC)
```

Sex	Sport	RCC	WCC
female	Netball	4.56	13.30
female	Netball	4.15	6.00
female	Netball	4.16	7.60
female	Netball	4.32	6.40
female	Netball	4.06	5.80
female	Netball	4.12	6.10
female	Netball	4.17	5.00
female	Netball	3.80	6.60
female	Netball	3.96	5.50
female	Netball	4.44	9.70
female	Netball	4.27	10.60
female	Netball	3.90	6.30
female	Netball	4.02	9.10

Choosing all-but some columns

```r
athletes %>% select(-(RCC:LBM))
```

Sex	Sport	Ht	Wt
female	Netball	176.8	59.90
female	Netball	172.6	63.00
female	Netball	176.0	66.30
female	Netball	169.9	60.70
female	Netball	183.0	72.90
female	Netball	178.2	67.90
female	Netball	177.3	67.50
female	Netball	174.1	74.10
female	Netball	173.6	68.20
female	Netball	173.7	68.80
female	Netball	178.7	75.30
female	Netball	183.3	67.40
female	Netball	174.4	70.00

Select-helpers

Other ways to select columns: those whose name:

- `starts_with` something
- `ends_with` something
- `contains` something
- `matches` a “regular expression”
- `everything()` select all the columns

Columns whose names begin with S

```
athletes %>% select(starts_with("S"))
```

Sex	Sport	SSF
female	Netball	49.0
female	Netball	110.2
female	Netball	89.0
female	Netball	98.3
female	Netball	122.1
female	Netball	90.4
female	Netball	106.9
female	Netball	156.6
female	Netball	101.1
female	Netball	126.4
female	Netball	114.0
female	Netball	70.0
female	Netball	77.0

Columns whose names end with C

either uppercase or lowercase:

```
athletes %>% select(ends_with("c"))
```

RCC	WCC	Hc
4.56	13.30	42.2
4.15	6.00	38.0
4.16	7.60	37.5
4.32	6.40	37.7
4.06	5.80	38.7
4.12	6.10	36.6
4.17	5.00	37.4
3.80	6.60	36.5
3.96	5.50	36.3
4.44	9.70	41.4
4.27	10.60	37.7
3.90	6.30	35.9

Case-sensitive

This works with any of the select-helpers:

```
athletes %>% select(ends_with("C", ignore.case=F))
```

RCC	WCC
4.56	13.30
4.15	6.00
4.16	7.60
4.32	6.40
4.06	5.80
4.12	6.10
4.17	5.00
3.80	6.60
3.96	5.50
4.44	9.70
4.27	10.60

Column names containing letter R

```r
athletes %>% select(contains("r"))
```

Sport	RCC	Ferr
Netball	4.56	20
Netball	4.15	59
Netball	4.16	22
Netball	4.32	30
Netball	4.06	78
Netball	4.12	21
Netball	4.17	109
Netball	3.80	102
Netball	3.96	71
Netball	4.44	64
Netball	4.27	68
Netball	3.90	78
Netball	4.02	107

Exactly two characters, ending with T

In regular expression terms, this is `^.t$`:

- `^` means "start of text"
- `.` means "exactly one character, but could be anything"
- `$` means "end of text".

```
athletes %>% select(matches("^.t$"))
```

Ht	Wt
176.8	59.90
172.6	63.00
176.0	66.30
169.9	60.70
183.0	72.90
178.2	67.90
177.3	67.50
174.1	74.10

Choosing columns by property

- Use `where` as with summarizing several columns
- eg, to choose text columns:

```r
athletes %>% select(where(is.character))
```

Sex	Sport
female	Netball
female	Netball
female	Netball
female	Netball
female	Netball
female	Netball
female	Netball
female	Netball
female	Netball
female	Netball
female	Netball

Choosing rows by number

```
athletes %>% slice(16:25)
```

Sex	Sport	RCC	WCC	Hc	Hg	Ferr	BMI	SSF	%Bfat
female	Netball	4.25	10.7	39.5	13.2	127	24.47	156.6	26.50
female	Netball	4.46	10.9	39.7	13.7	102	23.99	115.9	23.01
female	Netball	4.40	9.3	40.4	13.6	86	26.24	181.7	30.10
female	Netball	4.83	8.4	41.8	13.4	40	20.04	71.6	13.93
female	Netball	4.23	6.9	38.3	12.6	50	25.72	143.5	26.65
female	Netball	4.24	8.4	37.6	12.5	58	25.64	200.8	35.52
female	Netball	3.95	6.6	38.4	12.8	33	19.87	68.9	15.59
female	Netball	4.03	8.5	37.7	13.0	51	23.35	103.6	19.61
female	BBall	3.96	7.5	37.5	12.3	60	20.56	109.1	19.75
female	BBall	4.41	8.3	38.2	12.7	68	20.67	102.8	21.30

Non-consecutive rows

```r
athletes %>%
  slice(10,13,17,42)
```

Sex	Sport	RCC	WCC	Hc	Hg	Ferr	BMI	SSF	%Bfat
female	Netball	4.44	9.7	41.4	14.1	64	22.80	126.4	24.97
female	Netball	4.02	9.1	37.7	12.7	107	23.01	77.0	18.14
female	Netball	4.46	10.9	39.7	13.7	102	23.99	115.9	23.01
female	Row	4.37	8.1	41.8	14.3	53	23.47	98.0	21.79

A random sample of rows

```
athletes %>% slice_sample(n=8)
```

Sex	Sport	RCC	WCC	Hc	Hg	Ferr	BMI	SSF	%Bfat
female	Row	4.37	8.1	41.8	14.3	53	23.47	98.0	21.79
female	Netball	4.52	5.1	38.8	13.1	58	18.26	80.1	17.22
male	Tennis	5.66	8.3	50.2	17.7	38	23.76	56.5	10.05
male	Row	5.40	6.8	49.5	17.3	183	26.07	44.7	8.61
male	Row	5.09	10.1	44.9	14.8	118	26.79	58.3	9.79
female	Netball	4.16	7.6	37.5	12.3	22	21.40	89.0	19.39
female	Netball	4.25	10.7	39.5	13.2	127	24.47	156.6	26.50
female	Netball	4.39	9.6	38.3	12.5	39	24.64	148.9	26.78

Rows for which something is true

```
athletes %>% filter(Sport == "Tennis")
```

Sex	Sport	RCC	WCC	Hc	Hg	Ferr	BMI	SSF	%Bfat	LBM
female	Tennis	4.00	4.2	36.6	12.0	57	25.36	109.0	20.86	56.58
female	Tennis	4.40	4.0	40.8	13.9	73	22.12	98.1	19.64	56.01
female	Tennis	4.38	7.9	39.8	13.5	88	21.25	80.6	17.07	46.52
female	Tennis	4.08	6.6	37.8	12.1	182	20.53	68.3	15.31	51.75
female	Tennis	4.98	6.4	44.8	14.8	80	17.06	47.6	11.07	42.15
female	Tennis	5.16	7.2	44.3	14.5	88	18.29	61.9	12.92	48.76
female	Tennis	4.66	6.4	40.9	13.9	109	18.37	38.2	8.45	41.93
male	Tennis	5.66	8.3	50.2	17.7	38	23.76	56.5	10.05	72.00
male	Tennis	5.03	6.4	42.7	14.3	122	22.01	47.6	8.51	68.00
male	Tennis	4.97	8.8	43.0	14.9	233	22.34	60.4	11.50	63.00
male	Tennis	5.38	6.3	46.0	15.7	32	21.07	34.9	6.26	72.00

More complicated selections

```
athletes %>% filter(Sport == "Tennis", RCC < 5)
```

Sex	Sport	RCC	WCC	Hc	Hg	Ferr	BMI	SSF	%Bfat
female	Tennis	4.00	4.2	36.6	12.0	57	25.36	109.0	20.86
female	Tennis	4.40	4.0	40.8	13.9	73	22.12	98.1	19.64
female	Tennis	4.38	7.9	39.8	13.5	88	21.25	80.6	17.07
female	Tennis	4.08	6.6	37.8	12.1	182	20.53	68.3	15.31
female	Tennis	4.98	6.4	44.8	14.8	80	17.06	47.6	11.07
female	Tennis	4.66	6.4	40.9	13.9	109	18.37	38.2	8.45
male	Tennis	4.97	8.8	43.0	14.9	233	22.34	60.4	11.50

Another way to do "and"

```
athletes %>% filter(Sport == "Tennis") %>%
  filter(RCC < 5)
```

Sex	Sport	RCC	WCC	Hc	Hg	Ferr	BMI	SSF	%Bfat
female	Tennis	4.00	4.2	36.6	12.0	57	25.36	109.0	20.86
female	Tennis	4.40	4.0	40.8	13.9	73	22.12	98.1	19.64
female	Tennis	4.38	7.9	39.8	13.5	88	21.25	80.6	17.07
female	Tennis	4.08	6.6	37.8	12.1	182	20.53	68.3	15.31
female	Tennis	4.98	6.4	44.8	14.8	80	17.06	47.6	11.07
female	Tennis	4.66	6.4	40.9	13.9	109	18.37	38.2	8.45
male	Tennis	4.97	8.8	43.0	14.9	233	22.34	60.4	11.50

Either/Or

```
athletes %>% filter(Sport == "Tennis" | RCC > 5)
```

Sex	Sport	RCC	WCC	Hc	Hg	Ferr	BMI	SSF	%Bfat
female	Row	5.02	6.4	44.8	15.2	48	19.76	91.0	19.20
female	T400m	5.31	9.5	47.1	15.9	29	21.35	57.9	11.07
female	Field	5.33	9.3	47.0	15.0	62	25.27	102.8	19.51
female	TSprnt	5.16	8.2	45.3	14.7	34	20.30	46.1	10.15
female	Tennis	4.00	4.2	36.6	12.0	57	25.36	109.0	20.86
female	Tennis	4.40	4.0	40.8	13.9	73	22.12	98.1	19.64
female	Tennis	4.38	7.9	39.8	13.5	88	21.25	80.6	17.07
female	Tennis	4.08	6.6	37.8	12.1	182	20.53	68.3	15.31
female	Tennis	4.98	6.4	44.8	14.8	80	17.06	47.6	11.07
female	Tennis	5.16	7.2	44.3	14.5	88	18.29	61.9	12.92
female	Tennis	4.66	6.4	40.9	13.9	109	18.37	38.2	8.45
male	Swim	5.13	7.1	46.8	15.9	34	22.46	44.5	8.47
male	Swim	5.09	4.7	46.6	15.9	55	23.68	33.7	6.16

Sorting into order

```
athletes %>% arrange(RCC)
```

Sex	Sport	RCC	WCC	Hc	Hg	Ferr	BMI	SSF	%Bfat
female	Netball	3.80	6.60	36.5	12.4	102	24.45	156.6	26.57
female	Netball	3.90	6.30	35.9	12.1	78	20.06	70.0	15.01
female	T400m	3.90	6.00	38.9	13.5	16	19.37	48.4	10.48
female	Row	3.91	7.30	37.6	12.9	43	22.27	125.9	25.16
female	Netball	3.95	6.60	38.4	12.8	33	19.87	68.9	15.59
female	Row	3.95	3.30	36.9	12.5	40	24.54	74.9	16.38
female	Netball	3.96	5.50	36.3	12.4	71	22.63	101.1	17.93
female	BBall	3.96	7.50	37.5	12.3	60	20.56	109.1	19.75
female	Tennis	4.00	4.20	36.6	12.0	57	25.36	109.0	20.86
female	Netball	4.02	9.10	37.7	12.7	107	23.01	77.0	18.14
female	Netball	4.03	8.50	37.7	13.0	51	23.35	103.6	19.61
female	Netball	4.06	5.80	38.7	12.8	78	21.77	122.1	23.11
female	Swim	4.07	5.90	39.5	13.3	25	20.42	54.6	11.47

Breaking ties by another variable

```
athletes %>% arrange(RCC, BMI)
```

Sex	Sport	RCC	WCC	Hc	Hg	Ferr	BMI	SSF	%Bfat
female	Netball	3.80	6.60	36.5	12.4	102	24.45	156.6	26.57
female	T400m	3.90	6.00	38.9	13.5	16	19.37	48.4	10.48
female	Netball	3.90	6.30	35.9	12.1	78	20.06	70.0	15.01
female	Row	3.91	7.30	37.6	12.9	43	22.27	125.9	25.16
female	Netball	3.95	6.60	38.4	12.8	33	19.87	68.9	15.59
female	Row	3.95	3.30	36.9	12.5	40	24.54	74.9	16.38
female	BBall	3.96	7.50	37.5	12.3	60	20.56	109.1	19.75
female	Netball	3.96	5.50	36.3	12.4	71	22.63	101.1	17.93
female	Tennis	4.00	4.20	36.6	12.0	57	25.36	109.0	20.86
female	Netball	4.02	9.10	37.7	12.7	107	23.01	77.0	18.14
female	Netball	4.03	8.50	37.7	13.0	51	23.35	103.6	19.61
female	Netball	4.06	5.80	38.7	12.8	78	21.77	122.1	23.11
female	Swim	4.07	5.90	39.5	13.3	25	20.42	54.6	11.47

Descending order

```
athletes %>% arrange(desc(BMI))
```

Sex	Sport	RCC	WCC	Hc	Hg	Ferr	BMI	SSF	%Bfat
male	Field	5.48	6.20	48.2	16.3	94	34.42	82.7	13.91
male	Field	4.96	8.30	45.3	15.7	141	33.73	113.5	17.41
male	Field	5.48	4.60	49.4	18.0	132	32.52	55.7	8.51
female	Field	4.75	7.50	43.8	15.2	90	31.93	131.9	23.01
male	Field	5.01	8.90	46.0	15.9	212	30.18	112.5	19.94
male	Field	5.01	8.90	46.0	15.9	212	30.18	96.9	18.08
male	Field	5.09	8.90	46.3	15.4	44	29.97	71.1	13.97
female	Field	4.58	5.80	42.1	14.7	164	28.57	109.6	21.30
female	Field	4.51	9.00	39.7	14.3	36	28.13	136.3	24.88
male	WPolo	5.34	6.20	49.8	17.2	143	27.79	75.7	13.49
male	WPolo	4.90	7.60	45.6	16.0	90	27.56	67.2	11.79
male	Field	5.11	9.60	48.2	16.7	103	27.39	65.9	11.66
female	Field	4.81	6.80	42.7	15.3	50	26.95	98.5	20.10

“The top ones”

```
athletes %>%
  arrange(desc(Wt)) %>%
  slice(1:7) %>%
  select(Sport, Wt)
```

Sport	Wt
Field	123.2
BBall	113.7
Field	111.3
Field	108.2
Field	102.7
WPolo	101.0
BBall	100.2

Another way

```
athletes %>%
  slice_max(order_by = Wt, n=7) %>%
  select(Sport, Wt)
```

Sport	Wt
Field	123.2
BBall	113.7
Field	111.3
Field	108.2
Field	102.7
WPolo	101.0
BBall	100.2

Create new variables from old ones

```
athletes %>%
  mutate(wt_lb = Wt * 2.2) %>%
  select(Sport, Sex, Wt, wt_lb) %>%
  arrange(Wt)
```

Sport	Sex	Wt	wt_lb
Gym	female	37.80	83.16
Gym	female	43.80	96.36
Gym	female	45.10	99.22
Tennis	female	45.80	100.76
Tennis	female	47.40	104.28
Gym	female	47.80	105.16
T400m	female	49.20	108.24
Row	female	49.80	109.56
T400m	female	50.90	111.98
Netball	female	51.90	114.18

Turning the result into a number

Output is always data frame unless you explicitly turn it into something else, eg. the weight of the heaviest athlete, as a number:

```
athletes %>% arrange(desc(Wt)) %>% pluck("Wt", 1)
```

```
## [1] 123.2
```

Or the 20 heaviest weights in descending order:

```
athletes %>%
  arrange(desc(Wt)) %>%
  slice(1:20) %>%
  pluck("Wt")
```

```
##  [1] 123.20 113.70 111.30 108.20 102.70 101.00
##  [7] 100.20  98.00  97.90  97.90  97.00  96.90
## [13]  96.30  94.80  94.80  94.70  94.70  94.60
## [19]  94.25  94.20
```

Another way to do the last one

```r
athletes %>%
  arrange(desc(Wt)) %>%
  slice(1:20) %>%
  pull("Wt")
```

```
##  [1] 123.20 113.70 111.30 108.20 102.70 101.00
##  [7] 100.20  98.00  97.90  97.90  97.00  96.90
## [13]  96.30  94.80  94.80  94.70  94.70  94.60
## [19]  94.25  94.20
```

`pull` grabs the column you name *as a vector* (of whatever it contains).

To find the mean height of the women athletes

Two ways:

```
athletes %>% group_by(Sex) %>% summarize(m = mean(Ht))
```

Sex	m
female	174.5940
male	185.5059

```
athletes %>%
  filter(Sex == "female") %>%
  summarize(m = mean(Ht))
```

m
174.594

Summary of data selection/arrangement “verbs”

Verb	Purpose
`select`	Choose columns
`print`	Display non-default # of rows/columns
`slice`	Choose rows by number
`sample_n`	Choose random rows
`filter`	Choose rows satisfying conditions
`arrange`	Sort in order by column(s)
`mutate`	Create new variables
`group_by`	Create groups to summarize by
`summarize`	Calculate summary statistics (by groups if defined)
`pluck`	Extract items from data frame
`pull`	Extract a single column from a data frame as a vector

Looking things up in another data frame

Recall the tuberculosis data set, tidied:

```
tb3
```

iso2	year	gender	age	freq
AD	1996	m	014	0
AD	1996	m	1524	0
AD	1996	m	2534	0
AD	1996	m	3544	4
AD	1996	m	4554	1
AD	1996	m	5564	0
AD	1996	m	65	0
AD	1996	f	014	0
AD	1996	f	1524	1
AD	1996	f	2534	1
AD	1996	f	3544	0
AD	1996	f	4554	0

Actual country names

Found actual country names to go with those abbreviations, in spreadsheet:

```
my_url <-
  "http://www.utsc.utoronto.ca/~butler/c32/ISOCountryCodes081507.xlsx"
```

Note trick for reading in `.xlsx` from URL:

```
f <- tempfile()
download.file(my_url, f)
country_names <- read_excel(f)
```

- set up temporary file
- download spreadsheet to there
- read it from temporary file (which is “local”)

The country names

```
country_names
```

Code	Code_UC	Country
ad	AD	Andorra
ae	AE	United Arab Emirates
af	AF	Afghanistan
ag	AG	Antigua and Barbuda
ai	AI	Anguilla
al	AL	Albania
am	AM	Armenia
an	AN	Netherlands Antilles
ao	AO	Angola
aq	AQ	Antarctica
ar	AR	Argentina
arpa	ARPA	Old style Arpanet
as	AS	American Samoa

Looking up country codes

Matching a variable in one data frame to one in another is called a **join** (database terminology):

```
tb3 %>% left_join(country_names, by = c("iso2" = "Code_UC"))
```

iso2	year	gender	age	freq	Code	Country
AD	1996	m	014	0	ad	Andorra
AD	1996	m	1524	0	ad	Andorra
AD	1996	m	2534	0	ad	Andorra
AD	1996	m	3544	4	ad	Andorra
AD	1996	m	4554	1	ad	Andorra
AD	1996	m	5564	0	ad	Andorra
AD	1996	m	65	0	ad	Andorra
AD	1996	f	014	0	ad	Andorra
AD	1996	f	1524	1	ad	Andorra
AD	1996	f	2534	1	ad	Andorra
AD	1996	f	3544	0	ad	Andorra

Total cases by country

```
options(dplyr.summarise.inform=FALSE)
```

```
tb3 %>%
  group_by(iso2) %>%
  summarize(cases = sum(freq)) %>%
  left_join(country_names, by = c("iso2" = "Code_UC")) %>%
  select(Country, cases)
```

Country	cases
Andorra	64
United Arab Emirates	487
Afghanistan	80005
Antigua and Barbuda	21
Anguilla	1
Albania	2467
Armenia	6757

or even sorted in order

```r
tb3 %>%
  group_by(iso2) %>%
  summarize(cases = sum(freq)) %>%
  left_join(country_names, by = c("iso2" = "Code_UC")) %>%
  select(Country, cases) %>%
  arrange(desc(cases))
```

Country	cases
China	4065174
India	3966169
Indonesia	1129015
South Africa	900349
Bangladesh	758008
Vietnam	709695
NA	603095
Philippines	490040

Comments

- This is probably not quite right because of:
    - the 1994-1995 thing
    - there is at least one country in `tb3` that was not in `country_names` (the NA above). Which?

```
tb3 %>%
  anti_join(country_names, by = c("iso2" = "Code_UC")) %>%
  distinct(iso2)
```

iso2
CD
ME
NA
PS
RS
TL

Doing things one row at a time

A data frame `d`:

x1	x2
10	13
11	8
3	4

Want largest value in each row.

Try number 1

```
d %>% mutate(mx = max(x1, x2))
```

x1	x2	mx
10	13	13
11	8	13
3	4	13

- Fails because `max` finds the largest of *all* values in `x1` and `x2` (and repeats answer 3 times), rather than max of the two values in each row.

Try number 2

```
d %>% rowwise() %>%
  mutate(mx = max(x1, x2))
```

x1	x2	mx
10	13	13
11	8	11
3	4	4

- Works because rowwise works one row at a time: “find max of all numbers in 1st row”, then “all in 2nd” etc.

Calculations for groups

- Back to Australian athletes data: suppose we want the correlation between height and weight for athletes of each sport separately:

```
athletes %>% group_by(Sport) %>%
  summarize(correl = cor(Ht, Wt))
```

Sport	correl
BBall	0.8795746
Field	0.4713572
Gym	0.6528610
Netball	0.3630237
Row	0.8729486
Swim	0.8600633
T400m	0.8281211
Tennis	0.8259479
TSprnt	0.8057847
WPolo	0.5947771

Another way

- Break the data set into groups first:

```
athletes %>% nest_by(Sport)
```

Sport	data
BBall	female, female, female, female, female, female, female, female, female, female, female, female, female, male , male , male , male , male , male , male , male , male , male , male , male , 3.96 , 4.41 , 4.14 , 4.11 , 4.45 , 4.1 , 4.31 , 4.42 , 4.3 , 4.51 , 4.71 , 4.62 , 4.35 , 5.24 , 4.54 , 5.13 , 5 , 5.17 , 4.89 , 4.5 , 4.84 , 4.13 , 4.87 , 4.82 , 4.73 , 7.5 , 8.3 , 5 , 5.3 , 6.8 , 4.4 , 5.3 , 5.7 , 8.9 , 4.4 , 5.3 , 7.3 , 7.8 , 7.2 , 5.9 , 5.8 , 6.7 , 8 , 7.5 , 9.2 , 8.3 , 8.9 , 7.4 , 6.4 , 6.7 , 37.5 , 38.2 , 36.4 , 37.3 , 41.5 , 37.4 , 39.6 , 39.9 , 41.1 , 41.6 , 41.4 , 43.8 , 41.4 , 46.6 , 44.4 , 46.1 , 45.3 , 47.9 , 41.6 , 40.7 , 46.3 , 40.3 , 43.5 , 44.3 , 42.8 , 12.3 , 12.7 , 11.6 , 12.6 , 14 , 12.5 , 12.8 , 13.2 , 13.5 , 12.7 , 14 , 14.7 , 14.1 , 15.9 , 15.6 , 15.9 , 15.7 , 16.4 , 14.4 , 13.7 , 15.9 , 13.5 , 15 , 14.8 , 14.9 , 60 , 68 , 21 , 69 , 29 , 42 , 73 , 44 , 41 , 44 , 38 , 26 , 30 , 58 , 97 , 110 , 72 , 36 , 53 , 72 , 39 , 61 , 49 , 35 , 8 , 20.56 , 20.67 , 21.86 , 21.88 , 18.96 , 21.04 , 21.69 , 20.62 , 22.64 , 19.44 , 25.75 , 21.2 , 22.03 , 22.96 ,

Comments

- It looks as if all the other variables have disappeared, but they are all hiding in the column `data`.
- each thing in `data` is a *dataframe* (inside a dataframe!)
- when a column consists of things that are *not* single numbers, pieces of text, etc., it's called a **list-column** (see `list` in column header).