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

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

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

Choosing several columns

```r
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

```
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:

```
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	T400m	4.32	6.8	40.6	13.7	46	17.54	54.6	12.16
male	Swim	4.32	4.3	41.6	14.0	177	26.73	35.2	6.46
female	Tennis	4.00	4.2	36.6	12.0	57	25.36	109.0	20.86
female	Field	5.33	9.3	47.0	15.0	62	25.27	102.8	19.51
male	T400m	5.03	6.6	44.7	15.9	191	19.85	30.9	6.53
male	T400m	4.20	4.5	41.2	14.3	58	20.15	31.5	6.82
female	Netball	4.46	10.9	39.7	13.7	102	23.99	115.9	23.01
male	T400m	5.34	7.6	48.3	16.2	91	21.04	44.0	7.35

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

```r
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

```r
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

```r
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

- Suppose you are working in the nails department of a hardware store and you find that you have sold these items:

```
my_url <- "http://ritsokiguess.site/datafiles/nail_sales.csv"
sales <- read_csv(my_url)
sales
```

product_code	sales
061-5344-6	10
161-0090-0	6
061-5388-2	2
161-0199-4	8
061-5375-2	5
061-4525-2	3

Product descriptions and prices

- but you don’t remember what these product codes are, and you would like to know the total revenue from these sales.

- Fortunately you found a list of product descriptions and prices:

```
my_url <- "http://ritsokiguess.site/datafiles/nail_desc.csv"
desc <- read_csv(my_url)
desc
```

product_code	description	size	qty	price
061-4525-2	spike nail	10”	1	1.49
061-5329-4	masonry nail	1.5”	112	8.19
061-5344-6	finishing nail	1”	1298	6.99
061-5375-2	roofing nail	1.25”	192	6.99
061-5388-2	framing nail	4”	25	8.19
161-0090-0	wood nail	1”	25	2.39
161-0199-4	panel nail	1-5/8”	20	4.69

The lookup

- How do you “look up” the product codes to find the product descriptions and prices?
- `left_join`.

```
sales %>% left_join(desc)
```

```
## Joining, by = "product_code"
```

product_code	sales	description	size	qty	price
061-5344-6	10	finishing nail	1”	1298	6.99
161-0090-0	6	wood nail	1”	25	2.39
061-5388-2	2	framing nail	4”	25	8.19
161-0199-4	8	panel nail	1-5/8”	20	4.69
061-5375-2	5	roofing nail	1.25”	192	6.99
061-4525-2	3	spike nail	10”	1	1.49

What we have

- this looks up all the rows in the *first* dataframe that are also in the *second*.
- by default matches all columns with same name in two dataframes (product_code here)
- get *all* columns in *both* dataframes. The rows are the ones for that product_code.

So now can work out how much the total revenue was:

```
sales %>% left_join(desc) %>%
  mutate(product_revenue = sales*price) %>%
  summarize(total_revenue = sum(product_revenue))
```

```
## Joining, by = "product_code"
```

total_revenue
177.56

More comments

- if any product codes are not matched, you get NA in the added columns
- anything in the *second* dataframe that was not in the first does not appear (here, any products that were not sold)
- other variations (examples follow):
  - if there are two columns with the same name in the two dataframes, and you only want to match on one, use `by` with one column name
  - if the columns you want to look up have different names in the two dataframes, use `by` with a "named list"

Matching on only some matching names

- Suppose the `sales` dataframe *also* had a column `qty` (which was the quantity sold):

```
sales %>% rename("qty"="sales") -> sales1
sales1
```

product_code	qty
061-5344-6	10
161-0090-0	6
061-5388-2	2
161-0199-4	8
061-5375-2	5
061-4525-2	3

- The `qty` in `sales1` is the quantity sold, but the `qty` in `desc` is the number of nails in a package. These should *not* be matched: they are different things.

Matching only on product code

```r
sales1 %>%
  left_join(desc, by = "product_code")
```

product_code	qty.x	description	size	qty.y	price
061-5344-6	10	finishing nail	1"	1298	6.99
161-0090-0	6	wood nail	1"	25	2.39
061-5388-2	2	framing nail	4"	25	8.19
161-0199-4	8	panel nail	1-5/8"	20	4.69
061-5375-2	5	roofing nail	1.25"	192	6.99
061-4525-2	3	spike nail	10"	1	1.49

- Get `qty.x` (from `sales1`) and `qty.y` (from `desc`).

Matching on different names 1/2

- Suppose the product code in `sales` was just code:

```
sales %>% rename("code" = "product_code") -> sales2
sales2
```

code	sales
061-5344-6	10
161-0090-0	6
061-5388-2	2
161-0199-4	8
061-5375-2	5
061-4525-2	3

- How to match the two product codes that have different names?

Matching on different names 2/2

- Use by, but like this:

```
sales2 %>%
  left_join(desc, by = c("code"="product_code"))
```

code	sales	description	size	qty	price
061-5344-6	10	finishing nail	1”	1298	6.99
161-0090-0	6	wood nail	1”	25	2.39
061-5388-2	2	framing nail	4”	25	8.19
161-0199-4	8	panel nail	1-5/8”	20	4.69
061-5375-2	5	roofing nail	1.25”	192	6.99
061-4525-2	3	spike nail	10”	1	1.49

Other types of join

- `right_join`: interchanges roles, looking up keys from second dataframe in first.
- `anti_join`: give me all the rows in the first dataframe that are *not* in the second. (Use this eg. to see whether the product descriptions are incomplete.)
- `full_join`: give me all the rows in both dataframes, with missings as needed.

Full join here

```
sales %>% full_join(desc)
```

```
## Joining, by = "product_code"
```

product_code	sales	description	size	qty	price
061-5344-6	10	finishing nail	1”	1298	6.99
161-0090-0	6	wood nail	1”	25	2.39
061-5388-2	2	framing nail	4”	25	8.19
161-0199-4	8	panel nail	1-5/8”	20	4.69
061-5375-2	5	roofing nail	1.25”	192	6.99
061-4525-2	3	spike nail	10”	1	1.49
061-5329-4	NA	masonry nail	1.5”	112	8.19

- The missing `sales` for masonry nail says that it was in the lookup table, but we didn’t sell any.