Poking and Prodding at data frames for the infrequent useR

Kim Cressman

Grand Bay NERR

kimberly.cressman@msstate.edu

May 19-20, 2022

Indexing

[row, column]

How to identify the location of one or more cells: **`df[row, column]`**

- Think of the index like an *address* within your data frame
- We will learn this by counting and typing numeric values.
- Real-life use cases are more sophisticated, but trust me: we need to start with these concrete cases.

Mean DO_pct on 1/25/2022? **wq_summ[2, 5]**

	1	2	3	4	5	6	7	8	9	10
	Date	Depth_m_mean	Depth_m_min	Depth_m_max	DO_pct_mean	DO_pct_min	DO_pct_max	DO_mgl_mean	DO_mgl_min	DO_mgl_max
1	2022-01-24	0.5909184	0.526	0.675	105.71224	98.6	110.6	10.258571	9.81	10.60
2	2022-01-25	0.6244688	0.462	0.770	100.71771	93.2	104.1	9.494271	8.93	10.01
3	2022-01-26	0.6558437	0.369	0.951	93.36562	82.1	105.6	9.046875	7.90	10.01
4	2022-01-27	0.6423646	0.351	0.920	90.75521	79.4	102.6	8.840938	7.93	9.77
5	2022-01-28	0.4515312	0.131	0.789	88.69167	78.9	99.6	8.752917	7.82	9.72
6	2022-01-29	0.4813125	0.180	0.835	99.10729	83.0	111.1	11.165000	8.74	16.93
7	2022-01-30	0.4861563	0.162	0.893	99.06875	87.6	107.9	9.921875	9.41	10.42
8	2022-01-31	0.5421458	0.135	1.104	103.82083	89.1	120.9	9.857083	8.79	10.99
9	2022-02-01	0.7070937	0.330	1.203	100.79063	81.3	116.1	9.140833	7.64	10.58

What does **`wq_summ[5, 2]`** give us?

Mean depth on 1/28/2022

	1	2	3	4	5	6	7	8	9	10
	Date	Depth_m_mean	Depth_m_min	Depth_m_max	DO_pct_mean	DO_pct_min	DO_pct_max	DO_mgl_mean	DO_mgl_min	DO_mgl_max
1	2022-01-24	0.5909184	0.526	0.675	105.71224	98.6	110.6	10.258571	9.81	10.60
2	2022-01-25	0.6244688	0.462	0.770	100.71771	93.2	104.1	9.494271	8.93	10.01
3	2022-01-26	0.6558437	0.369	0.951	93.36562	82.1	105.6	9.046875	7.90	10.01
4	2022-01-27	0.6423646	0.351	0.920	90.75521	79.4	102.6	8.840938	7.93	9.77
5	2022-01-28	0.4515312	0.131	0.789	88.69167	78.9	99.6	8.752917	7.82	9.72
6	2022-01-29	0.4813125	0.180	0.835	99.10729	83.0	111.1	11.165000	8.74	16.93
7	2022-01-30	0.4861563	0.162	0.893	99.06875	87.6	107.9	9.921875	9.41	10.42
8	2022-01-31	0.5421458	0.135	1.104	103.82083	89.1	120.9	9.857083	8.79	10.99
9	2022-02-01	0.7070937	0.330	1.203	100.79063	81.3	116.1	9.140833	7.64	10.58

[row, column]

`df[row, column]`

Leave one blank to select *everything*

What if we want all of the parameters from 1/28/2022?

`wq_summ[5, ]`

	1	2	3	4	5	6	7	8	9	10
	Date	Depth_m_mean	Depth_m_min	Depth_m_max	DO_pct_mean	DO_pct_min	DO_pct_max	DO_mgl_mean	DO_mgl_min	DO_mgl_max
1	2022-01-24	0.5909184	0.526	0.675	105.71224	98.6	110.6	10.258571	9.81	10.60
2	2022-01-25	0.6244688	0.462	0.770	100.71771	93.2	104.1	9.494271	8.93	10.01
3	2022-01-26	0.6558437	0.369	0.951	93.36562	82.1	105.6	9.046875	7.90	10.01
4	2022-01-27	0.6423646	0.351	0.920	90.75521	79.4	102.6	8.840938	7.93	9.77
5	2022-01-28	0.4515312	0.131	0.789	88.69167	78.9	99.6	8.752917	7.82	9.72
6	2022-01-29	0.4813125	0.180	0.835	99.10729	83.0	111.1	11.165000	8.74	16.93
7	2022-01-30	0.4861563	0.162	0.893	99.06875	87.6	107.9	9.921875	9.41	10.42
8	2022-01-31	0.5421458	0.135	1.104	103.82083	89.1	120.9	9.857083	8.79	10.99
9	2022-02-01	0.7070937	0.330	1.203	100.79063	81.3	116.1	9.140833	7.64	10.58

What if we want all of the DO_pct_mean values?

`wq_summ[ , 5]`

	1	2	3	4	5	6	7	8	9	10
	Date	Depth_m_mean	Depth_m_min	Depth_m_max	DO_pct_mean	DO_pct_min	DO_pct_max	DO_mgl_mean	DO_mgl_min	DO_mgl_max
1	2022-01-24	0.5909184	0.526	0.675	105.71224	98.6	110.6	10.258571	9.81	10.60
2	2022-01-25	0.6244688	0.462	0.770	100.71771	93.2	104.1	9.494271	8.93	10.01
3	2022-01-26	0.6558437	0.369	0.951	93.36562	82.1	105.6	9.046875	7.90	10.01
4	2022-01-27	0.6423646	0.351	0.920	90.75521	79.4	102.6	8.840938	7.93	9.77
5	2022-01-28	0.4515312	0.131	0.789	88.69167	78.9	99.6	8.752917	7.82	9.72
6	2022-01-29	0.4813125	0.180	0.835	99.10729	83.0	111.1	11.165000	8.74	16.93
7	2022-01-30	0.4861563	0.162	0.893	99.06875	87.6	107.9	9.921875	9.41	10.42
8	2022-01-31	0.5421458	0.135	1.104	103.82083	89.1	120.9	9.857083	8.79	10.99
9	2022-02-01	0.7070937	0.330	1.203	100.79063	81.3	116.1	9.140833	7.64	10.58

How would you be more likely to want to identify a single cell, row, or column?

- By name
- By a logical (TRUE/FALSE) condition

Selecting Columns

Names

- Names require quotation marks (‘single’ or “double” – both are okay)
- Or a $

```
wq_summ[ , 5]        wq_summ[5]

wq_summ["DO_pct_mean"]
```

RStudio will help auto-complete after a $!

```
wq_summ$DO_pct_mean
```

	Date	Depth_m_mean	Depth_m_min	Depth_m_max	DO_pct_mean	DO_pct_min	DO_pct_max	DO_mgl_mean	DO_mgl_min	DO_mgl_max
1	2022-01-24	0.5909184	0.526	0.675	105.71224	98.6	110.6	10.258571	9.81	10.60
2	2022-01-25	0.6244688	0.462	0.770	100.71771	93.2	104.1	9.494271	8.93	10.01
3	2022-01-26	0.6558437	0.369	0.951	93.36562	82.1	105.6	9.046875	7.90	10.01
4	2022-01-27	0.6423646	0.351	0.920	90.75521	79.4	102.6	8.840938	7.93	9.77
5	2022-01-28	0.4515312	0.131	0.789	88.69167	78.9	99.6	8.752917	7.82	9.72
6	2022-01-29	0.4813125	0.180	0.835	99.10729	83.0	111.1	11.165000	8.74	16.93
7	2022-01-30	0.4861563	0.162	0.893	99.06875	87.6	107.9	9.921875	9.41	10.42
8	2022-01-31	0.5421458	0.135	1.104	103.82083	89.1	120.9	9.857083	8.79	10.99
9	2022-02-01	0.7070937	0.330	1.202	100.79063	81.3	116.1	9.140833	7.64	10.58

Tidyverse: **dplyr::select()**

```
wq_summ %>%
    select(DO_pct_mean)
```

Or multiple columns!

```
wq_summ %>%
    select(DO_pct_mean,
           Sal_psu_mean)
```

Order matters.

```
# A tibble: 60 x 2
   DO_pct_mean Sal_psu_mean
         <dbl>        <dbl>
 1        106.         22.0
 2        101.         24.2
 3         93.4        22.4
 4         90.8        22.5
 5         88.7        21.4
 6         99.1        17.2
 7         99.1        21.9
 8        104.         22.4
 9        101.         23.7
10         90.2        23.6
# ... with 50 more rows
```

Selecting Columns

Base: df$column_name

Tidyverse: df %>% select(column_name)

Selecting Rows

How to identify which rows?

- Come up with a logical (TRUE/FALSE) statement that can be applied to each row
- Only rows that meet the condition “TRUE” are kept

Logical conditions

- We use `==` to specify an exact condition (is equal to)
  - < less than
  - <= less than or equal to
  - == equals
  - != *not* equal to
  - >= greater than or equal to
  - > greater than

How to get all the rows where the daily mean salinity was < 15 psu?

```
# A tibble: 5 x 28
  Date       Depth_m_mean Depth_m_min Depth_m_max DO_pct_mean DO_pct_min
  <date>            <dbl>       <dbl>       <dbl>       <dbl>      <dbl>
1 2022-03-12        0.290       0.015       0.529        95.9       66.6
2 2022-03-19        0.485       0.35        0.663        81.3       66.9
3 2022-03-20        0.545       0.345       0.744        86.2       66.7
4 2022-03-21        0.623       0.401       0.869        91.1       68
5 2022-03-24        0.435       0.242       0.767        62.0       56.3
# ... with 22 more variables: DO_pct_max <dbl>, DO_mgl_mean <dbl>,
#   DO_mgl_min <dbl>, DO_mgl_max <dbl>, Sal_psu_mean <dbl>,
```

`wq_summ[which("Sal_psu_mean" < 15), ]`

Pulls out the indices of "True"s

True or false condition

Keeps all columns

Tidyverse: **dplyr::filter()**

```
wq_summ %>%
    filter(Sal_psu_mean < 15)
```

```
# A tibble: 5 x 28
  Date       Depth_m_mean Depth_m_min Depth_m_max DO_pct_mean DO_pct_min
  <date>            <dbl>       <dbl>       <dbl>       <dbl>      <dbl>
1 2022-03-12        0.290       0.015       0.529        95.9       66.6
2 2022-03-19        0.485       0.35        0.663        81.3       66.9
3 2022-03-20        0.545       0.345       0.744        86.2       66.7
4 2022-03-21        0.623       0.401       0.869        91.1       68
5 2022-03-24        0.435       0.242       0.767        62.0       56.3
# ... with 22 more variables: DO_pct_max <dbl>, DO_mgl_mean <dbl>,
#   DO_mgl_min <dbl>, DO_mgl_max <dbl>, Sal_psu_mean <dbl>,
```

Columns *and* rows

Selecting Rows

Tidyverse: df %>% filter(logical_condition)

Lists

First, back to Data Frames

Credit: Hands-on Programming with R,

By Garrett Grolemund

https://rstudio-education.github.io/hopr/r-objects.html#data-frames

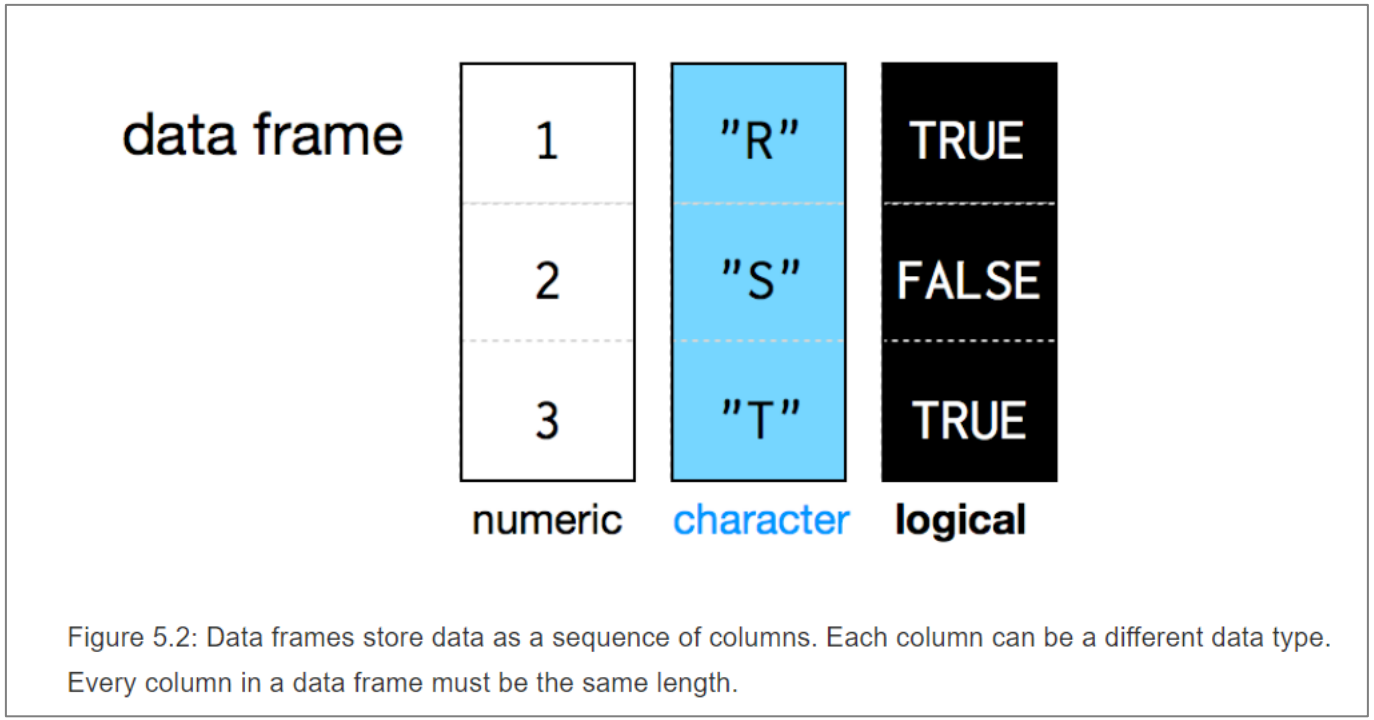


Figure 5.2: Data frames store data as a sequence of columns. Each column can be a different data type. Every column in a data frame must be the same length.

The train analogy

Credit: Hands-on Programming with R,

By Garrett Grolemund

https://rstudio-education.github.io/hopr/r-notation.html#dollar-signs-and-double-brackets

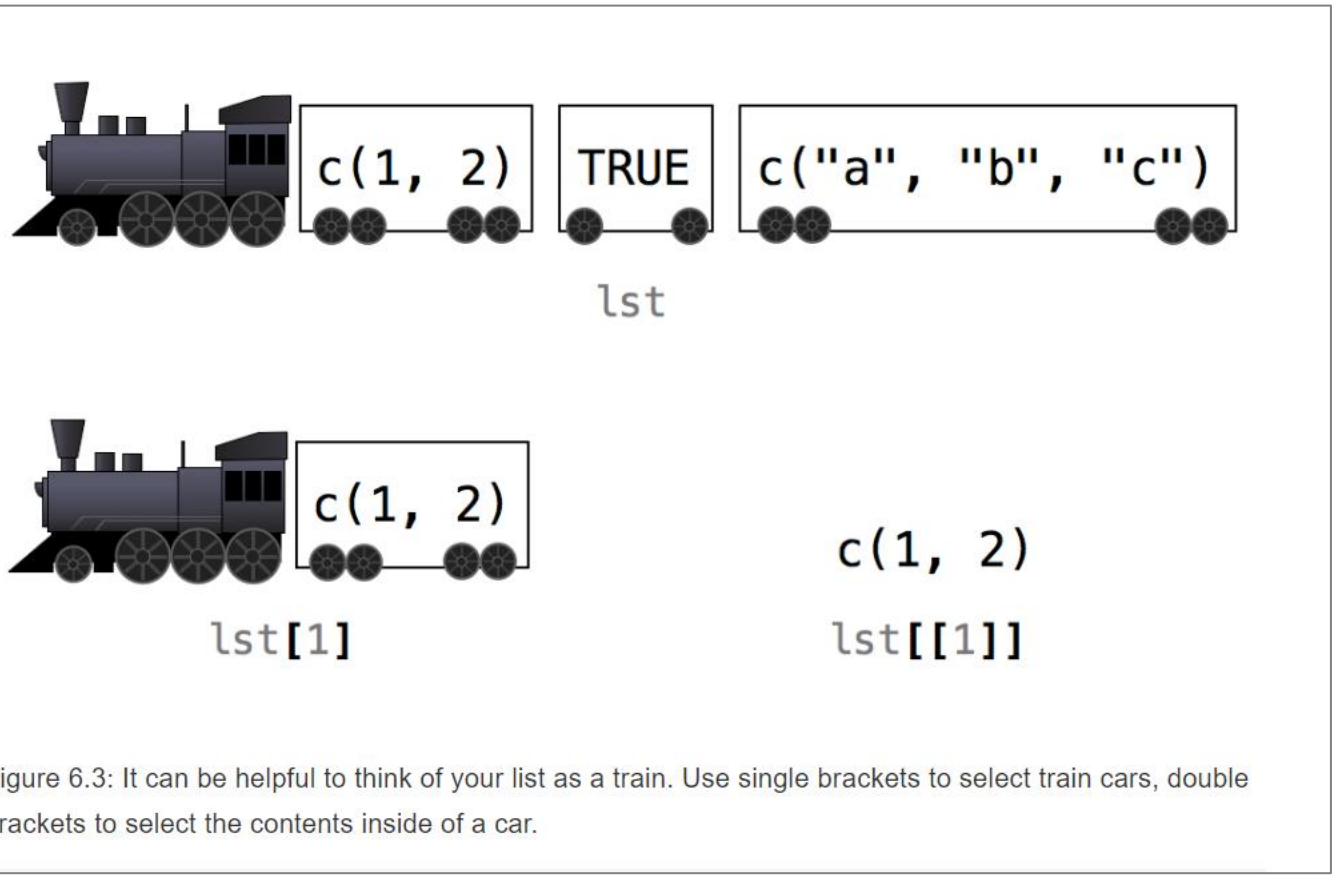
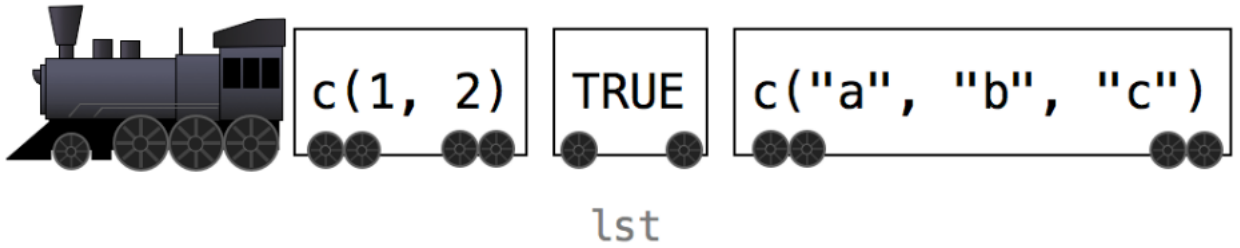
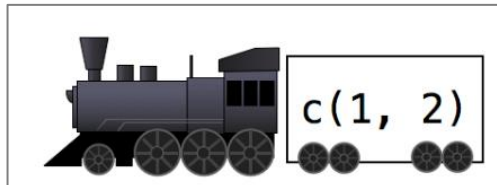


Figure 6.3: It can be helpful to think of your list as a train. Use single brackets to select train cars, double brackets to select the contents inside of a car.

The pepper-shaker analogy





- Single bracket: [ ] returns the same type of structure plus contents

c(1, 2)

lst[[1]]

- Double bracket: [[ ]] returns only the contents

1

lst[[1]][[1]]

- Double bracket followed by more brackets: [[ ]][[ ]] returns contents of the contents….. You could keep going