Introduction to R - Day 3

Lidiya Mishieva

03 February, 2026

Intro

- Day 1: basic syntax, classes, objects, functions
- Day 2: base package, tidy programming & tidyverse
- **Day 3:**
  - **Part 1: Summarizing data / tables**
  - **Part 2: Reporting with RMarkdown / Quarto**
- Day 4: Git & RStudio
- *Still missing: further operations on vectors. graphs, referencing in rmd/qmd, citations and bibliography*

Recap last week

- tidy paradigm
- working with data using base and tidyverse packages
- import and export, subsetting and filtering, rlong/wide format, variable transformations

Part 1

Summarizing data

- table()

```
1  table(penguins$species, penguins$island)
```

```
           Biscoe Dream Torgersen
Adelie         44    56        52
Chinstrap       0    68         0
Gentoo        124     0         0
```

- prop.table()

```
1  prop.table(table(penguins$species, penguins$island), margin = NULL) # margin=1 for rows, margin=2 for columns
```

```
              Biscoe     Dream Torgersen
Adelie    0.1279070 0.1627907 0.1511628
Chinstrap 0.0000000 0.1976744 0.0000000
Gentoo    0.3604651 0.0000000 0.0000000
```

- addmargins()

```
1  addmargins(prop.table(table(penguins$species, penguins$island), margin = NULL)) # margin=1 for rows, margin=2 for columns
```

```
              Biscoe     Dream Torgersen       Sum
Adelie    0.1279070 0.1627907 0.1511628 0.4418605
Chinstrap 0.0000000 0.1976744 0.0000000 0.1976744
Gentoo    0.3604651 0.0000000 0.0000000 0.3604651
Sum       0.4883721 0.3604651 0.1511628 1.0000000
```

Summarizing data

```r
table(penguins$species, penguins$island, penguins$sex)
```

```
, ,  = female

           Biscoe Dream Torgersen
  Adelie       22    27        24
  Chinstrap     0    34         0
  Gentoo       58     0         0

, ,  = male

           Biscoe Dream Torgersen
  Adelie       22    28        23
  Chinstrap     0    34         0
  Gentoo       61     0         0
```

Summarizing data

- `summary()`

```
1  summary(penguins$bill_len, useNA="always")
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
  32.10   39.23   44.45   43.92   48.50   59.60       2
```

- can be used as a general method to get a summary of different model outputs
- can be used to get a summary of an entire dataset

```
1  summary(penguins, useNA="always")
```

```
     species          island       bill_len        bill_dep
 Adelie   :152   Biscoe   :168   Min.   :32.10   Min.   :13.10
 Chinstrap: 68   Dream    :124   1st Qu.:39.23   1st Qu.:15.60
 Gentoo   :124   Torgersen: 52   Median :44.45   Median :17.30
                                 Mean   :43.92   Mean   :17.15
                                 3rd Qu.:48.50   3rd Qu.:18.70
                                 Max.   :59.60   Max.   :21.50
                                 NA's   :2       NA's   :2
  flipper_len      body_mass       sex           year
 Min.   :172.0   Min.   :2700   female:165   Min.   :2007
 1st Qu.:190.0   1st Qu.:3550   male  :168   1st Qu.:2007
 Median :197.0   Median :4050   NA's  : 11   Median :2008
 Mean   :200.9   Mean   :4202                Mean   :2008
 3rd Qu.:213.0   3rd Qu.:4750                3rd Qu.:2009
 Max.   :231.0   Max.   :6300                Max.   :2009
 NA's   :2       NA's   :2
```

Summarizing data

- dplyr package: `group_by()` and `summarize()`

```
penguins %>%
  group_by(island) %>%
  summarise(mean_bill_len = mean(bill_len, na.rm=TRUE))
```

```
# A tibble: 3 × 2
  island    mean_bill_len
  <fct>             <dbl>
1 Biscoe             45.3
2 Dream              44.2
3 Torgersen          39.0
```

```
penguins %>%
  select(-c(year)) %>%
  drop_na() %>%
  group_by(island, sex) %>%
  summarise(across(where(is.numeric), mean))
```

```
# A tibble: 6 × 6
# Groups:   island [3]
  island    sex    bill_len bill_dep flipper_len body_mass
  <fct>     <fct>     <dbl>    <dbl>       <dbl>     <dbl>
1 Biscoe    female     43.3     15.2        206.     4319.
2 Biscoe    male       47.1     16.6        213.     5105.
3 Dream     female     42.3     17.6        190.     3446.
4 Dream     male       46.1     19.1        196.     3987.
5 Torgersen female     37.6     17.6        188.     3396.
6 Torgersen male       40.6     19.4        195.     4035.
```

Summary tables for reports

- many different (not very practical) ways for producing summary tables
  - check out this guide:

    https://cran.r-project.org/web/packages/DescTools/vignettes/TablesInR.pdf
- dataframe is a table and can be used always as such

```
1  kableExtra::kable(penguins[1:4, ])
```

species	island	bill_len	bill_dep	flipper_len	body_mass	sex	year
Adelie	Torgersen	39.1	18.7	181	3750	male	2007
Adelie	Torgersen	39.5	17.4	186	3800	female	2007
Adelie	Torgersen	40.3	18.0	195	3250	female	2007
Adelie	Torgersen	NA	NA	NA	NA	NA	2007

Summary tables for reports

- programming and formatting usually with different packages
- `gtsummary` package for programming (unless it is a very specific table)
- package for formatting depends on the output file

Print Engine	Function	HTML	Word	PDF	RTF
gt	`as_gt()`	😁	😁	😁	🚧
flextable	`as_flex_table()`	😁	😁	😁	😁
huxtable	`as_hux_table()`	😁	😁	😁	😁
kableExtra	`as_kable_extra()`	😁	⛔	😁	⛔
kable	`as_kable()`	😐	😐	😐	😐
tibble	`as_tibble()`	☹	☹	☹	☹

😁 Output fully supported
😐 Missing indentation, footnotes, spanning headers
☹ No formatted output
⛔ Output not supported
🚧 Under development, missing indentation

Source: https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html

Tables with gtsummary

- summary tables from dataframes/tibbles
- regression tables from model outputs
- merging and stacking of tables
- some features for customization

Tables with gtsummary - descriptives

Variable	Treatment Received			p-value[2]
	Indometh, N = 295 (49%)[1]	**Placebo**, N = 307 (51%)[1]	**Overall**, N = 602 (100%)[1]	
Age	44 (33, 54)	46 (36, 55)	45 (35, 54)	0.15
Missing	0	0	0	
Risk				0.2
Median (Q1, Q3)	2.50 (2.00, 3.00)	2.50 (1.50, 3.00)	2.50 (1.50, 3.00)	
Min, Max	1.00, 5.50	1.00, 4.50	1.00, 5.50	
Missing	0	0	0	
type				0.6
0_no SOD	47 (16%)	60 (20%)	107 (18%)	
1_type 1	38 (13%)	43 (14%)	81 (13%)	
2_type 2	139 (47%)	135 (44%)	274 (46%)	
3_type 3	71 (24%)	69 (22%)	140 (23%)	
Missing	0	0	0	

[1] Median (Q1, Q3); n (%)

[2] Two Sample t-test; Pearson's Chi-squared test

Tables with gtsummary - descriptives

```r
# example datasets for different types of medical data
library(medicaldata)

# creating tables
library(gtsummary)


indo_rct %>%
  # filter the columns of the dataset
  select(c("rx", "age", "risk", "type")) %>%
  # select(c(rx, age, risk, type)) %>% # also works here!

  # recode the labels of the treatment variable rx
  # factor is a character vector, that allows labeling and ordering
  mutate(rx = factor(
    rx,
    levels = c("1_indomethacin", "0_placebo"),
    labels = c("Indometh", "Placebo"))
    ) %>%

  # specify a summary table
  tbl_summary(

    # stratification by treatment
    by = rx,
```

Tables with gtsummary - survival data

	Survival				
Variable	**6 Month**	**12 Month**	**18 Month**	**24 Month**	**p-value**[1]
Chemotherapy Treatment					0.2
Drug A	99% (97%, 100%)	91% (85%, 97%)	70% (62%, 80%)	47% (38%, 58%)	
Drug B	99% (97%, 100%)	86% (80%, 93%)	60% (51%, 70%)	41% (33%, 52%)	

[1] Log-rank test

Tables with gtsummary - survival data

```r
library(survival)

# survival
tbl_survfit(
  # compute survival curves
  survfit(
    Surv(ttdeath, death) ~ trt, trial
    # default is kaplan-meier
    # type=c("kaplan-meier","fleming-harrington", "fh2")
  ),

  # specify timepoints for estimating survival probabilities
  times = c(6, 12, 18, 24),

  # change the header label
  label_header = "**{time} Month**"
  ) %>%

  # render variable names in bold
  bold_labels() %>%

  # add log-rank test
  add_p() %>%

  # print the p-value in bold if below a threshhold
```

Tables with gtsummary - regression tables

Tables with gtsummary - regression tables

Characteristic	OR	95% CI	p-value
Chemotherapy Treatment			
Drug A	—	—	
Drug B	1.13	0.60, 2.13	0.7
Age	1.02	1.00, 1.04	0.10
Grade			
I	—	—	
II	0.85	0.39, 1.85	0.7
III	1.01	0.47, 2.15	>0.9

Abbreviations: CI = Confidence Interval, OR = Odds Ratio

Tables with gtsummary - regression tables

Characteristic	HR	95% CI	p-value
Chemotherapy Treatment			
Drug A	—	—	
Drug B	1.30	0.88, 1.92	0.2
Grade			
I	—	—	
II	1.21	0.73, 1.99	0.5
III	1.79	1.12, 2.86	0.014
Age	1.01	0.99, 1.02	0.3

Abbreviations: CI = Confidence Interval, HR = Hazard Ratio

Tables with gtsummary - regression tables

	Tumor Response			Time to Death		
Characteristic	**OR**	**95% CI**	**p-value**	**HR**	**95% CI**	**p-value**
Chemotherapy Treatment						
Drug A	—	—		—	—	
Drug B	1.13	0.60, 2.13	0.7	1.30	0.88, 1.92	0.2
Age	1.02	1.00, 1.04	0.10	1.01	0.99, 1.02	0.3
Grade						
I	—	—		—	—	
II	0.85	0.39, 1.85	0.7	1.21	0.73, 1.99	0.5
III	1.01	0.47, 2.15	>0.9	1.79	1.12, 2.86	0.014

Abbreviations: CI = Confidence Interval, HR = Hazard Ratio, OR = Odds Ratio

Tables with gtsummary - regression tables

```r
# specify a logistic model
mod1 <- glm(response ~ trt + age + grade, trial, family = binomial)
# build a regression table
t1 <- tbl_regression(mod1, exponentiate = TRUE)

# specify a cox model
mod2 <- coxph(Surv(ttdeath, death) ~ trt + grade + age, trial)
# build a regression table
t2 <-  tbl_regression(mod2, exponentiate = TRUE)

# merge tables
t3 <- tbl_merge(
  tbls = list(t1, t2),
  tab_spanner = c("**Tumor Response**", "**Time to Death**")
  )

t1
t2
t3
```

Tables with gtsummary - regression tables

```r
# table containing a set of univariable regressions
tbl_uvregression(
  trial,
  method = coxph,
  y = Surv(ttdeath, death),
  exponentiate = TRUE,
  include = c("age", "grade", "response"),
  pvalue_fun = label_style_pvalue(digits = 2)
)
```

Characteristic	N	HR	95% CI	p-value
Age	189	1.01	0.99, 1.02	0.33
Grade	200			
I		—	—	
II		1.28	0.80, 2.05	0.31
III		1.69	1.07, 2.66	0.024
Tumor Response	193	0.50	0.31, 0.78	0.003

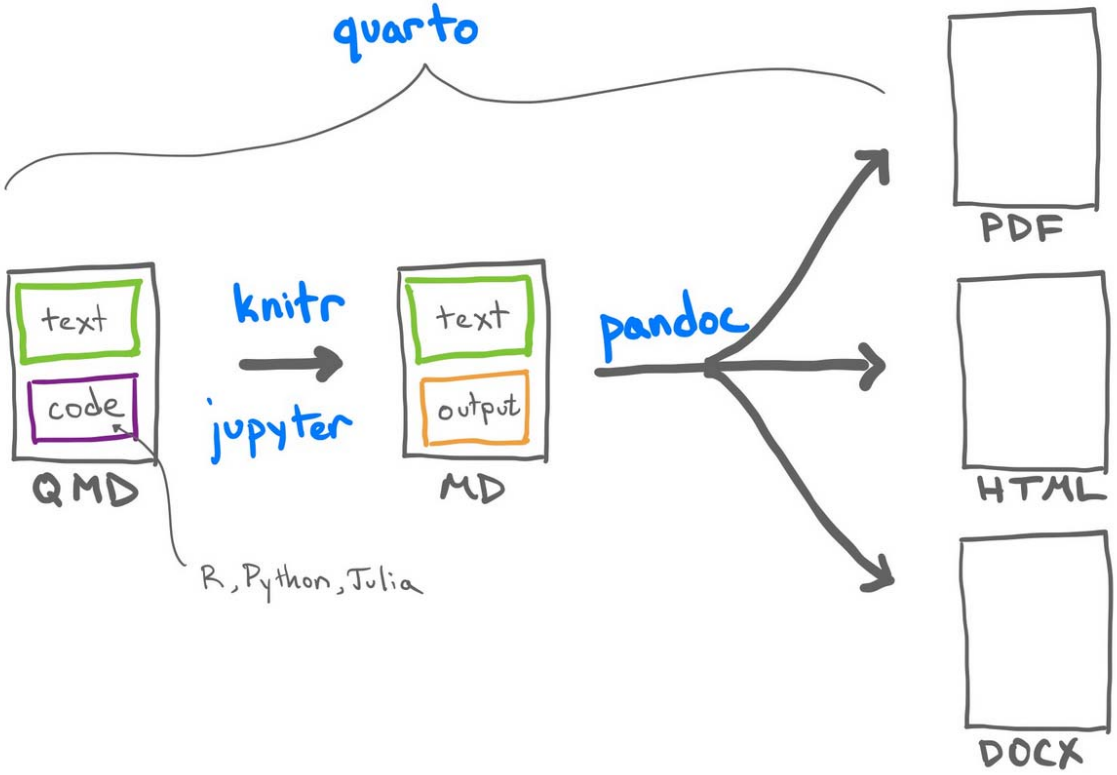
Abbreviations: CI = Confidence Interval, HR = Hazard Ratio

Part 2

Reporting with RStudio

Source: https://allisonhorst.com

A quarto document



Source: https://ubc-mds.github.io/DSCI_521_platforms-dsci_book/lectures/6-rmarkdown-quarto-slides-ghpages.html

Markup languages

- text-encoding systems that specifiy the structure & formatting of a document
- well-known examples: TeX, HTML, XML …
- combine plain text & code used for formatting
- markdown (md) is a simplified markup language that uses minimal code

Markup languages - HTML example

```html
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

My First Heading

My first paragraph.

Source: https://www.w3schools.com/html/tryit.asp?filename=tryhtml_basic_document

```
225  <li class="fragment"><p>Packages</p></li>
226  </ul>
227  </section>
228  <section id="tidy-paradigm---intuitive-readable-syntax" class="slide level2">
229  <h2>Tidy paradigm - intuitive ‘readable’ syntax</h2>
230  <ul>
231  <li class="fragment"><p>Consider this story:</p>
232  <ol type="1">
233  <li class="fragment"><p><em>Little bunny Foo Foo</em></p></li>
234  <li class="fragment"><p><em>Went hopping through the forest</em></p></li>
235  <li class="fragment"><p><em>Scooping up the field mice</em></p></li>
236  <li class="fragment"><p><em>And bopping them on the head</em></p></li>
237  </ol></li>
238  <li class="fragment"><p>Translate into code:</p></li>
239  <li class="fragment"><div class="cell">
240  <div class="sourceCode cell-code" id="cb1"><pre class="sourceCode numberSource r number-lines code-wi
241  <span id="cb1-2"><a href=""></a>foo_foo <span class="ot">&lt;-</span> <span class="fu">little_bunny</
242  <span id="cb1-3"><a href=""></a><span class="co"># define the first operation</span></span>
243  <span id="cb1-4"><a href=""></a>foo_foo_1 <span class="ot">&lt;-</span> <span class="fu">hop</span>(f
244  <span id="cb1-5"><a href=""></a><span class="co"># define the second operation</span></span>
245  <span id="cb1-6"><a href=""></a>foo_foo_2 <span class="ot">&lt;-</span> <span class="fu">scoop</span>
246  <span id="cb1-7"><a href=""></a><span class="co"># define the thirs operation</span></span>
247  <span id="cb1-8"><a href=""></a>foo_foo_3 <span class="ot">&lt;-</span> <span class="fu">bop</span>(f
248  </div></li>
249  </ul>
```

HTML behind our Day 2 presentation

Markdown basics

Markdown basics - text

Markdown Syntax	Output
`*italics*, **bold**, ***bold italics***`	*italics*, **bold**, ***bold italics***
`superscript^2^ / subscript~2~`	superscript^2 / subscript$_2$
`~~strikethrough~~`	~~strikethrough~~
`` `verbatim code` ``	`verbatim code`

Markdown basics - headings

Markdown syntax	Output
`# Heading 1`	# Heading 1
`## Heading 2`	## Heading 2
`### Heading 3`	### Heading 3
`#### Heading 4`	#### Heading 4

Markdown basics - footnotes

- Footnote reference

```
Here is a footnote reference[^1].

[^1]: Here is the footnote.
```

- Inline note

```
Here is an inline note.^[Inlines notes are easier to write, since you don't have to pick an identifier and move down to type the note.]
```

Markdown basics - inserting images

```
![*Source: https://allisonhorst.com*](images/clipboard-4033369980.png){fig-align="left"}
```

Source: https://allisonhorst.com

Markdown basics

- There are many more formatting options such as formatting tables and diagrams, adding page breaks etc.
- Checkout the quarto website for details:
  - https://quarto.org/docs/authoring/markdown-basics.html

Maths using TeX

(1) inline maths

```
In simple linear regression, the model is $y = \beta_0 + \beta_1 x + \varepsilon$.
```

In simple linear regression, the model is $y = \beta_0 + \beta_1 x + \varepsilon$.

(2) standalone equation

```
In simple linear regression, the model is

$$
\begin{equation}
y = \beta_0 + \beta_1 x + \varepsilon.
\end{equation}
$$
```

In simple linear regression, the model is

$$y = \beta_0 + \beta_1 x + \varepsilon.$$

Cheatsheet: https://tug.ctan.org/info/undergradmath/undergradmath.pdf

Adding code to RMD / QMD

- **Option 1**: code chunk
  - ```` ```{r}
    #| eval: false
    #| include: false
    #| output: false

    round(
      mean(
        penguins$bill_len, na.rm = TRUE
      ), 2
)
    ``` ````

- **Option 2**: inline code
  - Add output of the code within text
    - ```` The mean in group XY was `r round(mean(penguins$bill_len, na.rm=TRUE), 2)`. ````
    - The mean in group XY was 43.92.
    ```

YAML - Yet Another Markup Language

- In Quarto it is used for overall document configuration
- A YAML section is placed in the beginning of the document
  - ```yaml
    ---
    title: "Statistical Analyses - ADAPT HER2 IV Study"
    subtitle: "Efficacy Analyses"
    author: "Authors: Christine zu Eulenburg, Lidiya Mishieva, Ona Sauliene"
    date:  "`r Sys.Date()`"
    output:
      officedown::rdocx_document:
        toc: yes
        page_margins:
          bottom: 1.27
          top: 0
          right: 2.54
          left: 2.54
          gutter: 0
        reference_docx: template.docx
    ---
    ```
- A code chunk can also contain a YAML section
  - ```r
    {r chunk1, echo=TRUE = FALSE, message = FALSE, warning = FALSE}

    summary(iris)
    ```

    ```r
    {r}
    #| label: chunk1
    #| echo: true
    #| message: false
    #| warning: false

    summary(iris)
    ```

Code chunks - execution options

Option	Description
`eval: true`	whether to evaluate the code
`echo: true`	whether to show the code
`include: true`	whether to include the code
`cache: true`	whether to cache the resutls

Code chunks - figure control

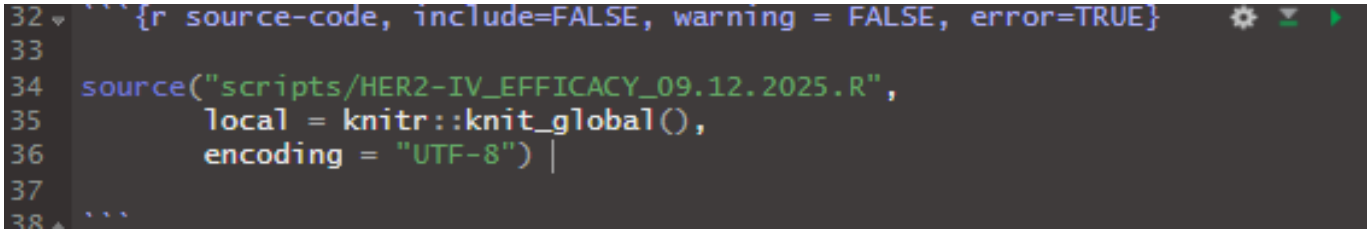
Option	Description
`fig-cap: "My figure"`	figure caption
`fig-width: 8`	whether to show the code
`fig-height: 6`	width in inches
`fig-align: "center"`	height in inches
`fig-dpi: 300`	resolution
`out-width: "80%"`	output width
`layout-ncol: 2`	number of columns for plots

Code chunks - output control

Option	Description
`warning: false`	hide warnings
`message: false`	hide messages
`error: false`	hide error messages
`output: false`	hide all output

Sourcing code

- **Option 1:** put all code into the code chunks or inline code
- **Option 2:** source your code from an external script
- You can do both (use results produced by the sourced code in the code chunks or inline code)
-



Rendering tables

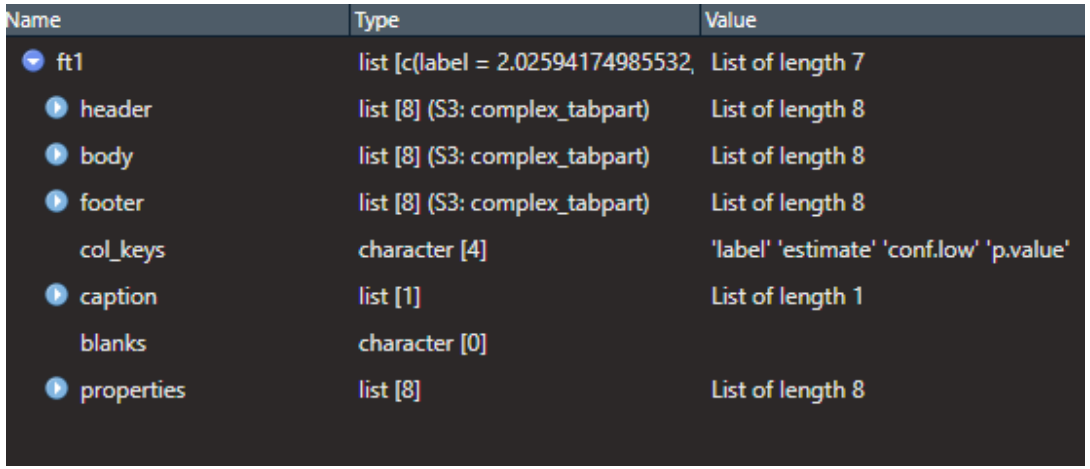
Print Engine	Function	HTML	Word	PDF	RTF
gt	`as_gt()`	😁	😁	😁	🚧
flextable	`as_flex_table()`	😁	😁	😁	😁
huxtable	`as_hux_table()`	😁	😁	😁	😁
kableExtra	`as_kable_extra()`	😁	⛔	😁	⛔
kable	`as_kable()`	😐	😐	😐	😐
tibble	`as_tibble()`	☹️	☹️	☹️	☹️

😁 Output fully supported
😐 Missing indentation, footnotes, spanning headers
☹️ No formatted output
⛔ Output not supported
🚧 Under development, missing indentation

Source: https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html

Rendering tables - flextable

- A package for creating and formatting tables
- Own class: flextable
- You can transform a gtsummary object to a flextable object
  - Add additional formatting
    - check out this webpage: https://ardata-fr.github.io/flextable-book/index.html
  - Else the format of gtsummary table is taken over to a flextable object
  - You can export it to .docx
-



Flextable to docx - fitting to page

```r
# ============================================================
# Fit a flextable to a LANDSCAPE page
# ============================================================

FitFlextableToPageLanscape <- function(

  # select a flextable object
  ft,

  # usable page width in inches:
  # A4 landscape width (29.7 cm)
  # minus 5 cm margins, converted to inches
  pgwidth = (29.7 - 5) / 2.54) {

  # apply consistent formatting
  ft_out <- ft %>%
    fontsize(size = 9, part = "body") %>%     # body text size
    fontsize(size = 10, part = "header") %>%  # header text size
    autofit() %>%                             # initial auto column widths
    height_all(0.25, part = "body") %>%       # compact row height
    hrule(rule = "exact", part = "body")      # fixed row height (vs auto)

  # rescale column widths to exactly fill the available page width
  ft_out <- width(
    ft_out,
```

Gtsummary to flextable to docx - workflow

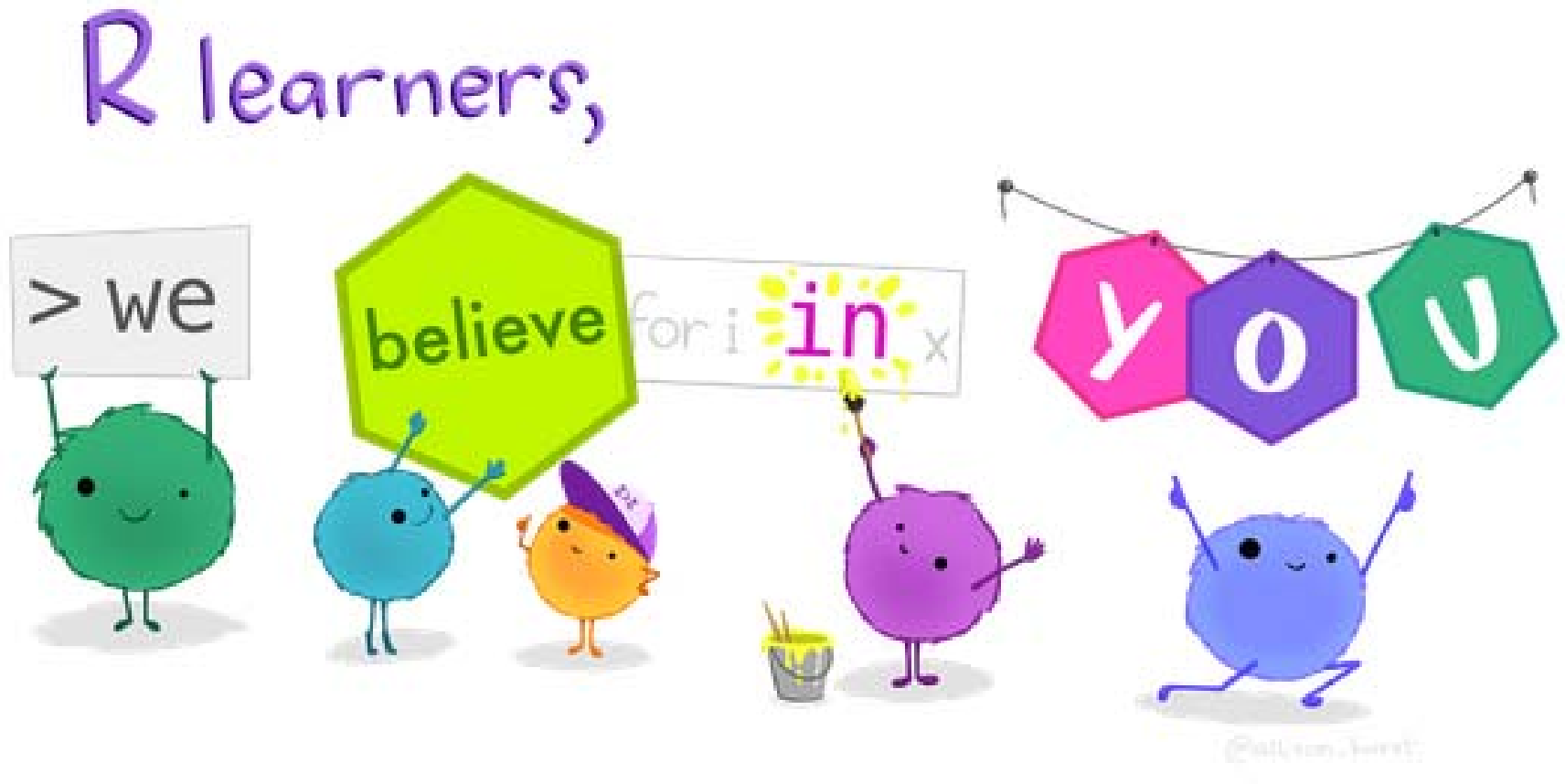
```
# specify a logistic model
mod1 <- glm(response ~ trt + age + grade, trial, family = binomial)

# build a regression table
t1 <- tbl_regression(mod1, exponentiate = TRUE)

# transform to a flextable object and fit to page
FitFlextableToPageLandscape(as_flex_table(t1))
```

Debugging with RMD/QMD

- Generally more difficult to debug code in RMarkdown/Quarto
  - Errors are indicated in the output windows of the code chunks
  - When rendering the document, RMD/QMD will just stop in case of error
  - To debug, you need to run each code chunk separately (considering the order!)
- In a script, debugging is more straight forward, an an error will be shown in the console directly after the corresponding code line
- RMD/QMD will not render, if the sourced Rscript contains errors

Reporting with RStudio - example file

The end



Source: https://allisonhorst.com

Move to the next session

Further operations on vectors

- sum, length, which

Graphs

- base R plots
- bese plots into pane with par(mfrow = c(2,2)) and dev.off()
- ggplot concept
- arranging plots with ggplot
- plots from models

Referencing with RMD/QMD

Citations and bibliography with RMD/ QMD