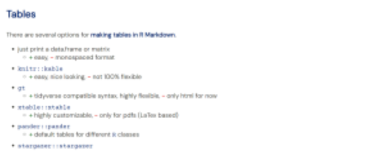
Module 4 Pre-Read - Parameterized Reports

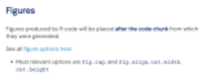
Sunday, February 11, 2024 15:51



Tables

There are several options for **making tables in R Markdown**.

- just print a data.frame or matrix
  - + easy, – monospaced format
- `knitr::kable`
  - + easy, nice looking, – not 100% flexible
- `gt`
  - + tidyverse compatible syntax, highly flexible, – only html for now
- `xtable::xtable`
  - + highly customizable, – only for pdfs (LaTex based)
- `pander::pander`
  - + default tables for different R classes
- `stargazer::stargazer`



Figures

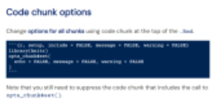
Figures produced by R code will be placed **after the code chunk** from which they were generated.

See all figure options here

- Most relevant options are `fig.cap` and `fig.align`, `out.width`, `out.height`

~~~
---
title: VRC01 neutralization report
author: David Benkeser
date: "`r format(Sys.Date(), '%Y-%m-%d')`"
output: html_document
---
```{r, load-data, echo = FALSE}
# replace with your own
abs_file_path <-
"~/Dropbox/Emory/Teaching/DSTK/intro_repro_workflo/lectures/04
_rmarkdown/"
full_file_path <- paste0(
  abs_file_path, "hiv_project/data/vrc01_data.csv"
)
data <- read.csv(full_file_path, header = TRUE)
```
# Descriptive Analysis
## Tabular analysis
Characteristics of the `r nrow(data)` viruses from the CATNAP
database are displayed in the table below.
```{r, table1}
# labelled + gtsummary package for making pretty html tables
library(labelled)
library(gtsummary)
# label variables
var_label(data) <- list(
  id = "ID",
  ab_resistance = "Antibody resistance",
  shield_glycans = "Shield glycans",
  region = "Region",
  env_length = "Length of Env protein"
)
data$number_glycans <- ifelse(data$shield_glycans < 4, "< 4", ">=
4")
# make the table
data |>
  select("region", "env_length", "ab_resistance",
"number_glycans") |>
  tbl_summary(by = number_glycans) |>
  modify_spanning_header(c("stat_1", "stat_2") ~ "**PNGs in
Glycan Shield**") |>
  add_overall() |>
  add_p()
```
## Graphical analysis
Below we report a scatter plot of resistance as a function of
PNGs in the glycan shield.
```{r, plot-data}
#| fig.cap = "Scatter plot with linear model smoother",
#| fig.align = "center",
#| out.width = "600px"
library(ggplot2)
ggplot(data, aes(x = shield_glycans, y = ab_resistance)) +
  geom_point() +
  geom_smooth(method = lm) +
  theme_bw()
```
# Regression Analysis
## Primary model
```{r, fit-primary-model}
mod <- glm(
  ab_resistance ~ shield_glycans + region + env_length,
  data = data
)
tbl_regression(mod) |>
  add_global_p()
```
## Secondary model
In this analysis, we analyzed a binary endpoint where level 1
means antibody resistance score was greater than 1.
```{r, binary-model}
binary_mod <- glm(
  I(ab_resistance > 1) ~ shield_glycans + region + env_length,
  data = data,
  family = binomial()
)
tbl_regression(binary_mod, exponentiate = TRUE) |>
  add_global_p()
```
~~~

From <https://raw.githubusercontent.com/benkeser/intro_repro_workflo/main/lectures/04_rmarkdown/hiv_project/tmp_reports/03_hiv_report.Rmd>



Code chunk options

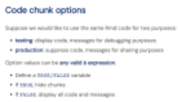
Change **options for all chunks** using code chunk at the top of the `.Rmd`.

~~~
```{r, setup, include = FALSE, message = FALSE, warning = FALSE}
library(knitr)
opts_chunk$set(
  echo = FALSE, message = FALSE, warning = FALSE
)
```
~~~

Note that you still need to suppress the code chunk that includes the call to `opts_chunk$set()`.

~~~
```{r, chunk-opts, include = FALSE, message =
FALSE, warning = FALSE}
library(knitr)
opts_chunk$set(
  echo = FALSE, warning = FALSE, message = FALSE
)
```
~~~

From
<https://raw.githubusercontent.com/benkeser/intro_repro_workflo/main/lectures/04_rmarkdown/hiv_project/tmp_reports/04_hiv_report.Rmd>



Code chunk options

Suppose we would like to use the same Rmd code for two purposes:

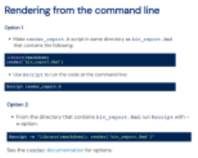
- **testing**: display code, messages for debugging purposes
- **production**: suppress code, messages for sharing purposes

Option values can be **any valid R expression**.

- Define a `TRUE/FALSE` variable
- If `TRUE`, hide chunks
- If `FALSE`, display all code and messages

~~~
```{r, production-or-test, include = FALSE}
# we will learn a smarter way to do this
production <- TRUE
```
```{r, chunk-opts, include = FALSE, message = FALSE, warning = FALSE,
eval = production}
# if production == FALSE, then this code will not
# be evaluated and thus code chunks will appear in
# the rendered document
library(knitr)
opts_chunk$set(
  echo = FALSE, warning = FALSE, message = FALSE
)
```
~~~

From <https://raw.githubusercontent.com/benkeser/intro_repro_workflo/main/lectures/04_rmarkdown/hiv_project/tmp_reports/05_hiv_report.Rmd>



Rendering from the command line

Option 1:

- Make `render_report.R` script in same directory as `hiv_report.Rmd` that contains the following:

```
library(rmarkdown)
render('hiv_report.Rmd')
```

- Use `Rscript` to run the code at the command line:

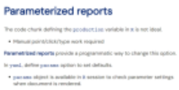
```
Rscript render_report.R
```

Option 2:

- From the directory that contains `hiv_report.Rmd`, run `Rscript` with `-e` option:

```
Rscript -e "library(rmarkdown); render('hiv_report.Rmd')"
```

See the `render` documentation for options.



Parameterized reports

The code chunk defining the `production` variable in R is not ideal.

- Manual point/click/type work required

Parameterized reports provide a programmatic way to change this option.

In `yaml`, define `params` option to set defaults.

- `params` object is available in R session to check parameter settings when document is rendered.

~~~
---
title: VRC01 neutralization report
author: David Benkeser
date: "`r format(Sys.Date(), '%Y-%m-%d')`"
output: html_document
params:
  production: FALSE
  cutpoint: 1
---
```{r, chunk-opts, include = FALSE, message = FALSE, warning = FALSE,
eval = params$production}
# if production == FALSE, then this code will not
# be evaluated and thus code chunks will appear in
# the rendered document
library(knitr)
opts_chunk$set(
  echo = FALSE, warning = FALSE, message = FALSE
)
```
```{r, load-data, echo = FALSE}
# replace with your own
abs_file_path <-
"~/Dropbox/Emory/Teaching/DSTK/intro_repro_workflo/lectures/04
_rmarkdown/"
full_file_path <- paste0(
  abs_file_path, "hiv_project/data/vrc01_data.csv"
)
data <- read.csv(full_file_path, header = TRUE)
```
# Descriptive Analysis
## Tabular analysis
Characteristics of the `r nrow(data)` viruses from the CATNAP database
are displayed in the table below.
```{r, table1}
# labelled + gtsummary package for making pretty html tables
library(labelled)
library(gtsummary)
# label variables
var_label(data) <- list(
  id = "ID",
  ab_resistance = "Antibody resistance",
  shield_glycans = "Shield glycans",
  region = "Region",
  env_length = "Length of Env protein"
)
data$number_glycans <- ifelse(data$shield_glycans < 4, "< 4", ">= 4")
# make the table
data |>
  select("region", "env_length", "ab_resistance", "number_glycans") |>
  tbl_summary(by = number_glycans) |>
  modify_spanning_header(c("stat_1", "stat_2") ~ "**PNGs in Glycan
Shield**") |>
  add_overall() |>
  add_p()
```
## Graphical analysis
Below we report a scatter plot of resistance as a function of PNGs in
the glycan shield.
```{r, plot-data}
#| fig.cap = "Scatter plot with linear model smoother",
#| fig.align = "center",
#| out.width = "600px"
library(ggplot2)
ggplot(data, aes(x = shield_glycans, y = ab_resistance)) +
  geom_point() +
  geom_smooth(method = lm) +
  theme_bw()
```
# Regression Analysis
## Primary model
```{r, fit-primary-model}
mod <- glm(
  ab_resistance ~ shield_glycans + region + env_length,
  data = data
)
tbl_regression(mod) |>
  add_global_p()
```
## Secondary model
In this analysis, we analyzed a binary endpoint where level 1 means
antibody resistance score was greater than `r params$cutpoint`.
```{r, binary-model}
binary_mod <- glm(
  I(ab_resistance > params$cutpoint) ~ shield_glycans + region +
env_length,
  data = data,
  family = binomial()
)
tbl_regression(binary_mod, exponentiate = TRUE) |>
  add_global_p()
```
~~~

From <https://raw.githubusercontent.com/benkeser/intro_repro_workflo/main/lectures/04_rmarkdown/hiv_project/tmp_reports/06_hiv_report.Rmd>

```
library(rmarkdown)
render('hiv_report.Rmd')
# now in production mode
render('hiv_report.Rmd',
  params = list(production = TRUE),
  output_file = 'hiv_report_production.html'
)
# in production mode with different cutpoint
render('hiv_report.Rmd',
  params = list(production = TRUE,
                cutpoint = 0.5),
  output_file = 'hiv_report_production_cutpoint.html'
)
```

From <https://raw.githubusercontent.com/benkeser/intro_repro_workflo/main/lectures/04_r_render_report.R>

markdown/hiv_project/tmp_reports/02