SCALES OF SUCCESS: A STUDY INTO FIRST-YEAR STUDENT OUTCOMES IN GATOR NATION

By

AL E. GATOR

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2025

Update the text in `preface/dedicationFile.tex` to change this.

It should begin with the word "To" and should not have a period at the end, e.g. "To Mom and Dad"

ACKNOWLEDGEMENTS

Update the text in `preface/acknowledgmentsFile.tex` to change this.

Keep in mind this should be written in first person, eg; “This template owes thanks to all the LaTeX specialists who have helped contribute over the years. A big thanks to Ron Smith, James Booth, Jason Nowell, December Slater, and Meg Renard for their contributions and maintenance of this document.”

TABLE OF CONTENTS

LIST OF TABLES

Tables page

LIST OF FIGURES

Figures page

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

SCALES OF SUCCESS: A STUDY INTO FIRST-YEAR STUDENT OUTCOMES IN GATOR NATION

By

Al E. Gator

August 2025

Chair: W. Kent Fuchs
Cochair: Joseph Glover
Major: Higher Education Administration

Update the text in `preface/abstractFile.tex` to change this.

This is what will appear in the place of an abstract, no formatting or other content is needed, just fill this file with your actual abstract, eg; “In this paper we give examples of the various files and configurations used in the graduate school LaTeX template for dissertations and thesis papers.”

CHAPTER 1
HEADINGS (INTRODUCTION)

1.1 Using Headings

You can use level one and two headings to structure your chapter, these will automatically create the table of contents with links as appropriate

1.1.1 A Sub-Heading

Note, you are not allowed single sub-sections, you need at least two per level & section if you use them

1.1.2 Another Sub-Heading

Don't use level 3 or higher headings, as those aren't part of the UF template

1.2 Paragraph Headings

If you need additional levels of sub-headings, use bold text as the first sentence of the paragraph (including a period).

A Paragraph Header. Don't leave a blank line between the paragraph header and the text, it should be part of the paragraph. These will not show up in the table of contents.

CHAPTER 2
FIGURES (LITERATURE REVIEW)

Figures can be included one of two ways.

2.1 As Quarto Output

For simpler plots where the default output and scaling works, you can include them from an R (or Python etc.) code block

````
```{{r}}
#| label: fig-gators
#| fig-cap: Albert and Alberta's Times


data <- tribble(~date, ~mascot, ~time,
                2020, "Albert", 300,
                2020, "Alberta", 290,
                2021, "Albert", 280,
                2021, "Alberta", 285,
                2022, "Albert", 278,
                2022, "Alberta", 275,
                2023, "Albert", 285,
                2023, "Alberta", 275,
                2024, "Albert", 270,
                2024, "Alberta", 265,
                2025, "Albert", 265,
                2025, "Alberta", 260)


ggplot(data) +
  geom_line(aes(x = date, y = time, color = mascot)) +
  scale_color_manual(values = c("#FA4616", "#0021A5"))
```
````

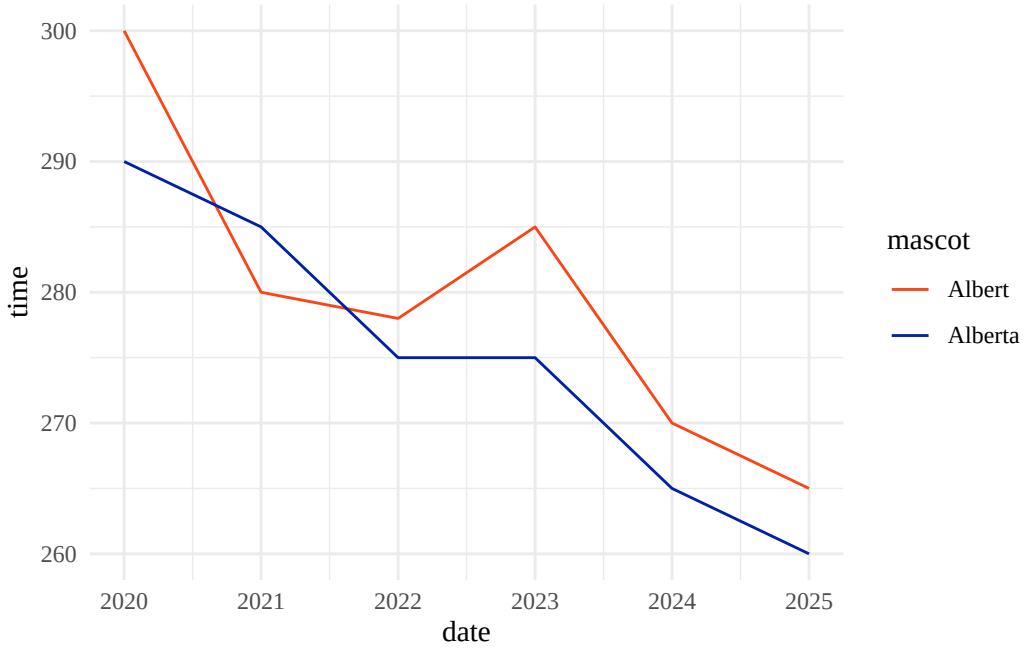

Figure 2-1. Albert and Alberta's Times

```
Saving 5.5 x 3.5 in image
```

For these you need to use Quarto's figure cross referencing. This means labeling the chunk starting with fig `#| label: fig-gators` and adding a caption `#| fig-cap: Albert and Alberta's Times`.

You can then refer to this plot with `@fig-gators` in your markdown text and it will create in text figure numbers correctly based on chapter and order and hyperlink directly to the figure.

2.2 As an Image

Sometimes you might have figures created elsewhere, or, want to have more control over the scaling/shape of the plot than you get with Quarto by default. In this case, you want to save the figures as an image (e.g., my-plot.png) and the add and cross reference it using inline markdown like `![Albert and Alberta's Times](figures/gator-plot.png){#fig-gators}`. This was how I handled most of my plots.

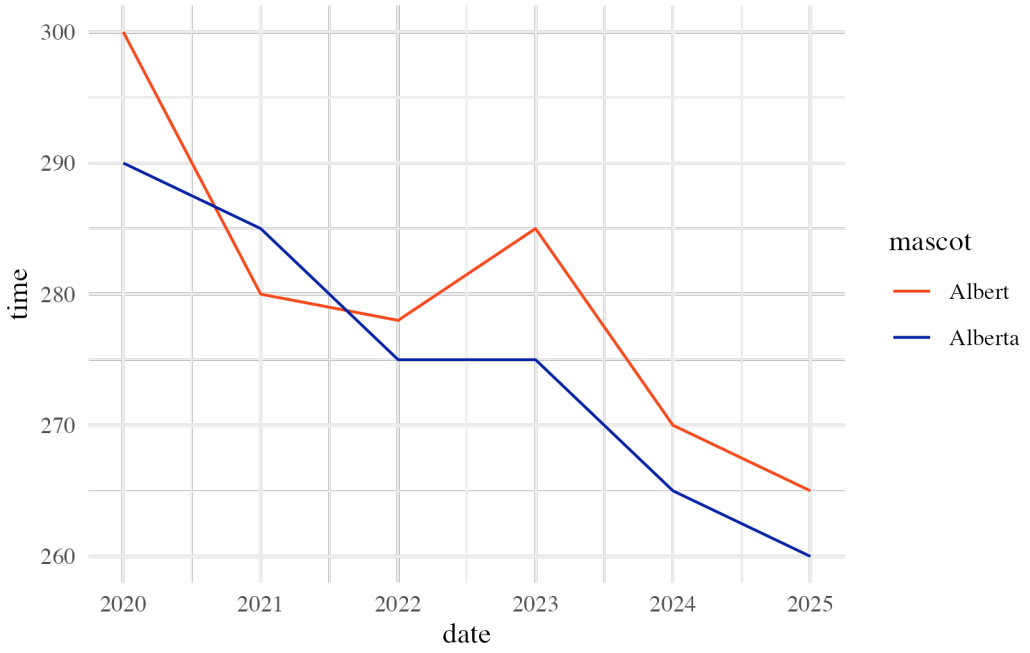

Figure 2-2. Albert and Alberta's Times

2.3 Mix n' Match

Finally, you can mix and match these methods and all the figure numbering keeps up to date, no extra steps needed!

CHAPTER 3
TABLES (METHODOLOGY)

Tables are both the most involved and most beneficial part of using this template. Tables are added to the directly from R output. This means that if you add a new variable to your model, if you re-run the code to make the table, your results will update automatically, no more hand copying etc.! However, you will need to a little bit of tweaking to get your tables exactly as needed. Depending on how much you need to customize a table, you might need to work out a little bit of LaTeX, but ChatGPT etc. are extremely helpful here. I have included examples of all the ways I created tables in my dissertation, but, this may not cover everything you need, but they should be a good starting point.

3.1 Simple Tables Added Hand

The first way you can add a table is simply creating it in a code chunk, turning it into a `kable()`, and doing some simple reformatting. This is super-flexible, and you could pretty much create any table you need this way (if you know enough LaTeX). That said, there are more efficient ways for common tables using the `gtsummary` package below.

You will see all chunks that create tables are preceeded by `\realSingleSpace` and proceeded by `\doublespacing`. This ensures the text in the table is single spaced, as required for UF formatting.

For example, let's create a summary table the Gators data from earlier and include it as a simple `kable()` with a little extra formatting

```
data |>
  group_by(mascot) |>
  summarize(mean = mean(time), sd = sd(time), min = min(time), max = max(time)) |>
  kable(col.names = c("Mascot", "Mean", "SD", "Min", "Max"),
      format = "latex",
      booktabs = TRUE,
      digits = 2,
      escape = TRUE,
      linesep = "",
      align = c("l", "r", "r", "r", "r")) |>
  kable_styling(full_width = TRUE) |>
  column_spec(1, width = "3in") |>  # Adjust width as needed
  row_spec(0, align = "l")
```

Table 3-1. Summary of Mascot Times

Mascot	Mean	SD	Min	Max
Albert	279.67	12.27	265	300
Alberta	275.00	11.40	260	290

That formatting code will be a good starting point for any `kable()`s. If you

3.1.1 Multi-page Kables

If you need a table that goes over a single page, it's a little trickier, as we can't use `kable_styling(full_width = TRUE)` (as we need to use `longtable = TRUE` and they aren't compatible for some reason). So, instead you need to manually set all the column widths so the table is turns out the width of a page. This is done with `column_spec(1, width = "3in")` lines in the below code. When I needed to do this, I started with values that total 6 inches and tweaked it from there until it looked good. If you want to check it's correct, the edges should line up with a single page table that used `kable_styling(full_width = TRUE)`, so you could always add a dummy one of those above/below to check against.

```
long_gator_table <- bind_rows(data, data, data, data)

long_gator_table |>
  kable(col.names = c("Date", "Mascot", "Time"),
      format = "latex",
                  booktabs = TRUE,
                  linesep = "",
                  align = c("l", "r", "r", "r"),
                  longtable = TRUE) |>
    kable_styling(full_width = FALSE,
                  latex_options = c("repeat_header"),
                  repeat_header_text = "Table A-1. Continued. (UPDATE THIS)") |>
  column_spec(1, width = "3in") |>  # Adjust width as needed
  column_spec(2, width = "1.5in") |>  # Adjust width as needed
  column_spec(3, width = "1.5in") |>  # Adjust width as needed
  row_spec(0, align = "l")
```

Table 3-2. Summary of Mascot Times

Date	Mascot	Time
2020	Albert	300
2020	Alberta	290
2021	Albert	280
2021	Alberta	285
2022	Albert	278
2022	Alberta	275
2023	Albert	285
2023	Alberta	275
2024	Albert	270
2024	Alberta	265
2025	Albert	265
2025	Alberta	260
2020	Albert	300
2020	Alberta	290
2021	Albert	280
2021	Alberta	285
2022	Albert	278
2022	Alberta	275
2023	Albert	285
2023	Alberta	275
2024	Albert	270
2024	Alberta	265
2025	Albert	265
2025	Alberta	260
2020	Albert	300
2020	Alberta	290
2021	Albert	280
2021	Alberta	285
2022	Albert	278
2022	Alberta	275
2023	Albert	285
2023	Alberta	275
2024	Albert	270
2024	Alberta	265
2025	Albert	265
2025	Alberta	260
2020	Albert	300
2020	Alberta	290
2021	Albert	280
2021	Alberta	285
2022	Albert	278
2022	Alberta	275

Table A-1. Continued. (UPDATE THIS)

Date	Mascot	Time
2023	Albert	285
2023	Alberta	275
2024	Albert	270
2024	Alberta	265
2025	Albert	265
2025	Alberta	260

3.2 `gtsummary` Tables

The kable option is flexible, but, one of the great advantages of including code from R is all the table generation packages that create nicely formatted tables for common situations. There are countless table packages in R, and I'm sure most could be adapted to this template, but my personal favorite is `gtsummary`, so that is what I have written some helper functions for. These functions are included in the `table-functions.R` script.

Specifically, there are pre-written functions to create descriptive statistics tables and regression (both lm and glm) results tables.

3.2.1 Variable Labels

The first step of creating gtsummary tables is setting your variable labels. This should be a named list like below and will be used to replace variable names in the formatted tables. This is pretty intuitivie, the only trick is for regression tables, you need labels modified terms to match their R regression output, such as `log(time)` in the example below

3.2.2 Descriptive Statitistics

If you look in the inside `table-functions.R` you will see two functions. `create_desc_table()` takes a full dataset and creates a gtsummary descriptive statistics table as I have them in my dissertation. If you want to change the type of summary statistics in the table, you will want to edit this and then edit the formatting function to match. `format_desc_table()` takes a gtsummary table, and spits out a LaTeX table that formats correctly in the template.

The way it is currently set up requires you add a `treatment =` to get descriptives for each group and overall.

The decision as to what gtsummary considers continuous is usually pretty good. However, I have seen with small datasets, double columns can be treated as categorical.

```
gtsummary_table <- create_desc_table(data, treatment = "mascot", labels = label_list)

format_desc_table(gtsummary_table)
```

Table 3-3. Summary of Mascot Times

Variable	Overall *N = 12*	Albert *N = 6*	Alberta *N = 6*
Date of Race			
2020	2 (17%)	1 (17%)	1 (17%)
2021	2 (17%)	1 (17%)	1 (17%)
2022	2 (17%)	1 (17%)	1 (17%)
2023	2 (17%)	1 (17%)	1 (17%)
2024	2 (17%)	1 (17%)	1 (17%)
2025	2 (17%)	1 (17%)	1 (17%)
Time to Finish			
260	1 (8.3%)	0 (0%)	1 (17%)
265	2 (17%)	1 (17%)	1 (17%)
270	1 (8.3%)	1 (17%)	0 (0%)
275	2 (17%)	0 (0%)	2 (33%)
278	1 (8.3%)	1 (17%)	0 (0%)
280	1 (8.3%)	1 (17%)	0 (0%)
285	2 (17%)	1 (17%)	1 (17%)
290	1 (8.3%)	0 (0%)	1 (17%)
300	1 (8.3%)	1 (17%)	0 (0%)

By default, this will use all variables in your data, but you can specify `variables = c()` to make a smaller table.

3.2.3 Regression Tables

CHAPTER 4
RESULTS

jgkfjkl

CHAPTER 5
DISCUSSION AND CONCLUSION

fjkdjaklf

Capaldi (2023)

APPENDIX A
DESCRIPTIVE STATISTICS TABLES

APPENDIX B
DIAGNOSTIC DETAILS

REFERENCES

Capaldi, M. J. (2023). The route to graduation: an exploration of the association between transit stop proximity and pell recipient completion rates at us commuter colleges. *Journal of College Student Retention: Research, Theory & Practice*. https://doi.org/10.1177/15210251231156421

BIOGRAPHICAL SKETCH

Update the text in `preface/biographyFile.tex` to change this.

Keep in mind this should be written in third person and should assume you have already completed your degree that you are writing this thesis or dissertation for.