My title*

My subtitle if needed

First author Another author

October 12, 2024

First sentence. Second sentence. Third sentence. Fourth sentence.

1 Introduction

Overview paragraph

Estimand paragraph

Results paragraph

Why it matters paragraph

Telegraphing paragraph: The remainder of this paper is structured as follows. Section 2….

2 Data

2.1 Overview

We use the statistical programming language R (R Core Team 2023)…. Our data (Toronto Shelter & Support Services 2024)…. Following Alexander (2023), we consider…

Overview text

2.2 Measurement

Some paragraphs about how we go from a phenomena in the world to an entry in the dataset.

*Code and data are available at: https://github.com/RohanAlexander/starter_folder.

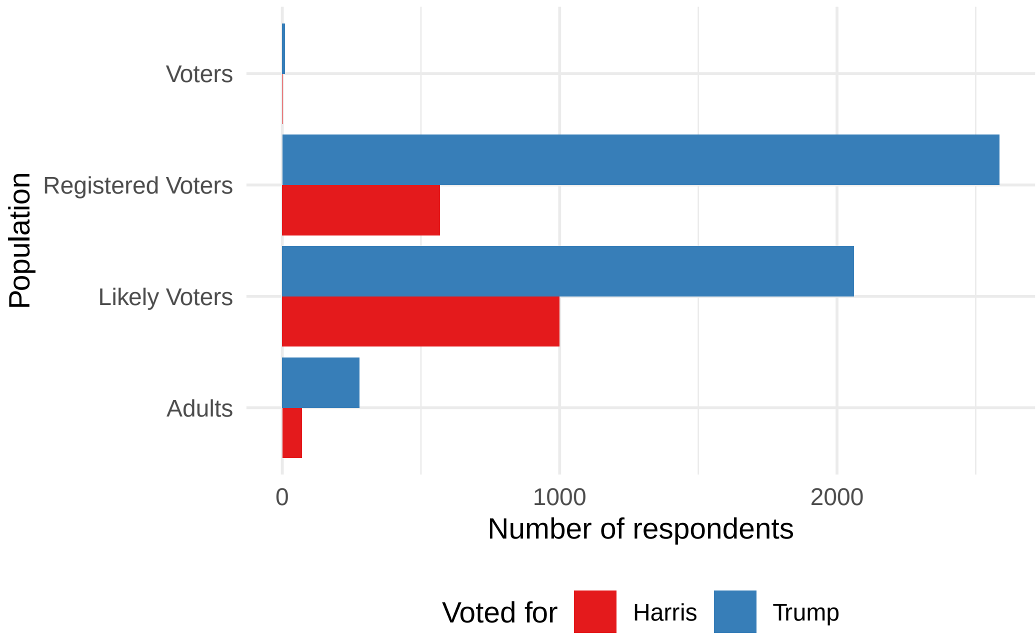
2.3 Outcome variables

Add graphs, tables and text. Use sub-sub-headings for each outcome variable or update the subheading to be singular.

Some of our data is of penguins (**?@fig-bills**), from Horst, Hill, and Gorman (2020).

```r
# Load necessary libraries
library(dplyr)
library(tidyr)

# Load the dataset
polls <- read.csv(here::here("data/02-analysis_data/analysis_data.csv"))
```

```r
polls |>
  ggplot(aes(x = population, fill = answer)) +
  stat_count(position = "dodge") +
  theme_minimal() +
  labs(
    x = "Population",
    y = "Number of respondents",
    fill = "Voted for"
  ) +
  coord_flip() +
  scale_fill_brewer(palette = "Set1") +
  theme(legend.position = "bottom")
```



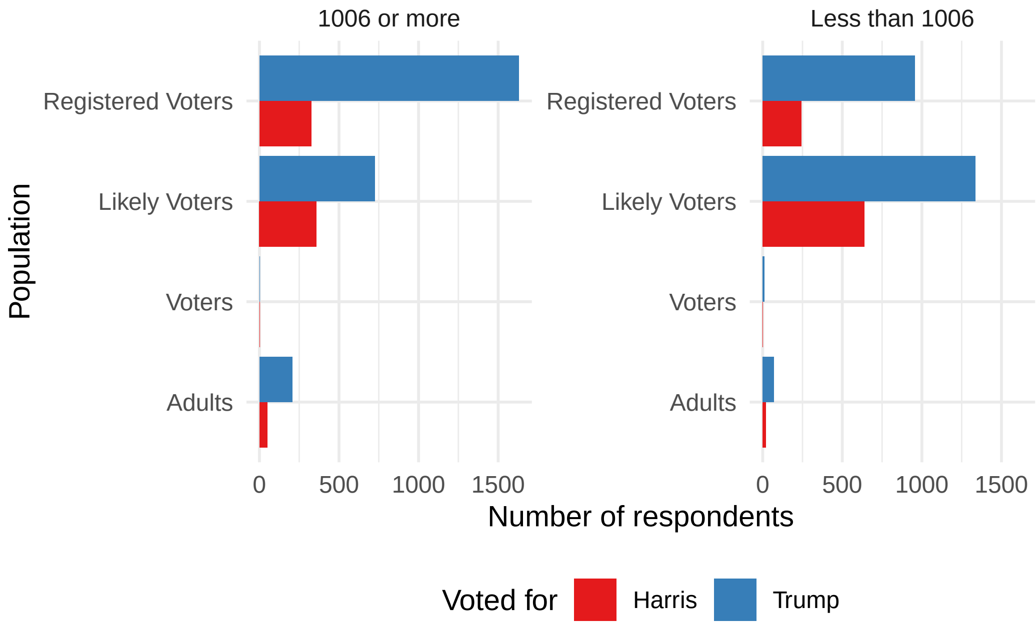

```
# Finding Median of Sample Size
median_sample_size <- median(polls$sample_size, na.rm = TRUE)
print(median_sample_size)
```

```
[1] 1006
```

```
# The U.S. uses an Electoral College system, where each state is assigned a certain number o
# Most states use a winner-takes-all system, where the candidate who wins the popular vote i
# A total of 538 electoral votes are available. To win the presidency, a candidate must secu

# Differentiating sample sizes in presidential election polls is important because larger sa
```

```
library(ggplot2)
library(dplyr)

library(ggplot2)
library(dplyr)
library(tidyr)

# Define population levels (adjust based on your actual data)
population_levels <- c("Adults", "Voters", "Likely Voters", "Registered Voters")  # Update t

# Ensure all factor levels are present and handle missing combinations
```

```r
polls_clean <- polls %>%
  mutate(
    population = factor(population, levels = population_levels),
    sample_category = ifelse(sample_size < 1006, "Less than 1006", "1006 or more"),
    answer = as.factor(answer)  # Convert answer to factor if it represents candidates
  ) %>%
  complete(population, answer, sample_category, fill = list(n = 0))  # Fill missing combinati

# Plot with facet_wrap
polls_clean %>%
  ggplot(aes(x = population, fill = answer)) +
  stat_count(position = "dodge") +
  theme_minimal() +
  labs(
    x = "Population",
    y = "Number of respondents",
    fill = "Voted for"
  ) +
  coord_flip() +
  scale_fill_brewer(palette = "Set1") +
  theme(legend.position = "bottom") +
  facet_wrap(~sample_category, scales = "free_y")
```



```r
set.seed(304)

polls_reduced <-
  polls_clean |>
  slice_sample(n = 1000)

political_preferences <-
  stan_glm(
    answer ~  + sample_category + population,
    data = polls_reduced,
    family = binomial(link = "logit"),
    prior = normal(location = 0, scale = 2.5, autoscale = TRUE),
    prior_intercept =
      normal(location = 0, scale = 2.5, autoscale = TRUE),
    seed = 853
  )
```

```
SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 0.000131 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 1.31 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 0.496 seconds (Warm-up)
Chain 1:                0.586 seconds (Sampling)
Chain 1:                1.082 seconds (Total)
Chain 1:
```

```
SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 2.1e-05 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.21 seconds.
Chain 2: Adjust your expectations accordingly!
Chain 2:
Chain 2:
Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 2:
Chain 2:  Elapsed Time: 0.559 seconds (Warm-up)
Chain 2:                0.379 seconds (Sampling)
Chain 2:                0.938 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 1.9e-05 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.19 seconds.
Chain 3: Adjust your expectations accordingly!
Chain 3:
Chain 3:
Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
```

```
Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 3:
Chain 3:  Elapsed Time: 0.56 seconds (Warm-up)
Chain 3:                0.583 seconds (Sampling)
Chain 3:                1.143 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 1.8e-05 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.18 seconds.
Chain 4: Adjust your expectations accordingly!
Chain 4:
Chain 4:
Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 4:
Chain 4:  Elapsed Time: 0.476 seconds (Warm-up)
Chain 4:                0.528 seconds (Sampling)
Chain 4:                1.004 seconds (Total)
Chain 4:
```

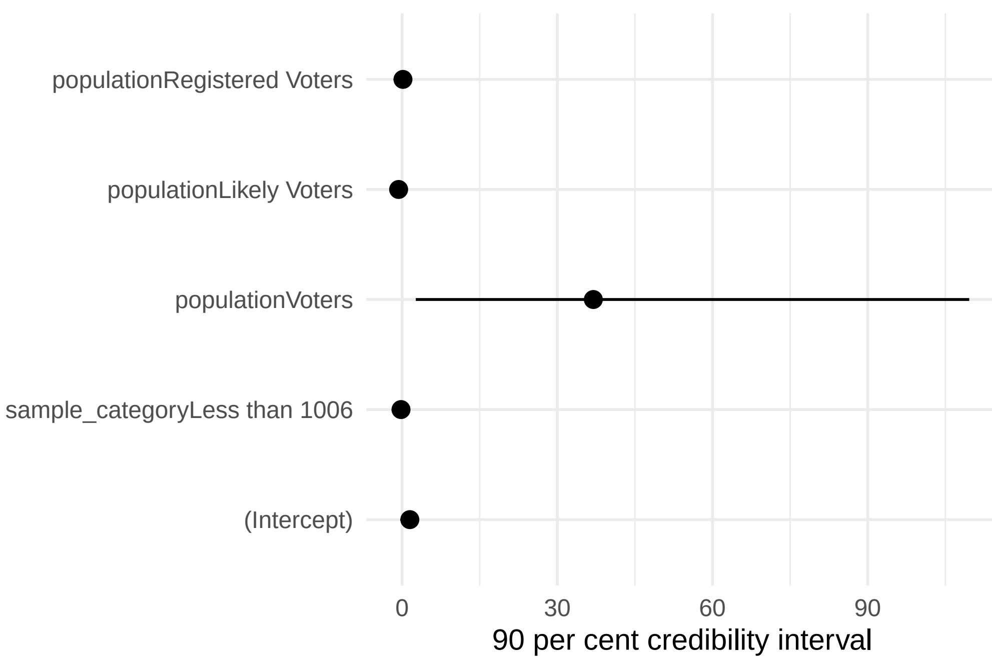
```r
saveRDS(
  political_preferences,
  file = "political_preferences.rds"
)
```

```r
political_preferences <-
  readRDS(file = "political_preferences.rds")
```

```
modelsummary(
  list(
    "Support Biden" = political_preferences
  ),
  statistic = "mad"
  )
```

```
Warning:
`modelsummary` uses the `performance` package to extract goodness-of-fit
statistics from models of this class. You can specify the statistics you wish
to compute by supplying a `metrics` argument to `modelsummary`, which will then
push it forward to `performance`. Acceptable values are: "all", "common",
"none", or a character vector of metrics names. For example: `modelsummary(mod,
metrics = c("RMSE", "R2")` Note that some metrics are computationally
expensive. See `?performance::performance` for details.
 This warning appears once per session.
```

```
modelplot(political_preferences, conf_level = 0.9) +
  labs(x = "90 per cent credibility interval")
```



Talk way more about it.

	Support Biden
(Intercept)	1.488
	(0.379)
sample_categoryLess than 1006	−0.218
	(0.148)
populationVoters	36.961
	(33.253)
populationLikely Voters	−0.678
	(0.387)
populationRegistered Voters	0.160
	(0.392)
Num.Obs.	1000
R2	0.040
Log.Lik.	−550.459
ELPD	−554.6
ELPD s.e.	15.4
LOOIC	1109.2
LOOIC s.e.	30.8
WAIC	1109.2
RMSE	0.43

2.4 Predictor variables

Add graphs, tables and text.

Use sub-sub-headings for each outcome variable and feel free to combine a few into one if they go together naturally.

3 Model

The goal of our modelling strategy is twofold. Firstly,…

Here we briefly describe the Bayesian analysis model used to investigate… Background details and diagnostics are included in Appendix B.

3.1 Model set-up

Define y_i as the number of seconds that the plane remained aloft. Then β_i is the wing width and γ_i is the wing length, both measured in millimeters.

$$
\begin{aligned}
y_i|\mu_i, \sigma &\sim \text{Normal}(\mu_i, \sigma) && (1)\\
\mu_i &= \alpha + \beta_i + \gamma_i && (2)\\
\alpha &\sim \text{Normal}(0, 2.5) && (3)\\
\beta &\sim \text{Normal}(0, 2.5) && (4)\\
\gamma &\sim \text{Normal}(0, 2.5) && (5)\\
\sigma &\sim \text{Exponential}(1) && (6)
\end{aligned}
$$

We run the model in R (R Core Team 2023) using the `rstanarm` package of Goodrich et al. (2022). We use the default priors from `rstanarm`.

3.1.1 Model justification

We expect a positive relationship between the size of the wings and time spent aloft. In particular…

We can use maths by including latex between dollar signs, for instance θ.

Table 1: Explanatory models of flight time based on wing width and wing length

	First model
(Intercept)	1.12
	(1.70)
length	0.01
	(0.01)
width	−0.01
	(0.02)
Num.Obs.	19
R2	0.320
R2 Adj.	0.019
Log.Lik.	−18.128
ELPD	−21.6
ELPD s.e.	2.1
LOOIC	43.2
LOOIC s.e.	4.3
WAIC	42.7
RMSE	0.60

4 Results

Our results are summarized in Table 1.

5 Discussion

5.1 First discussion point

If my paper were 10 pages, then should be be at least 2.5 pages. The discussion is a chance to show off what you know and what you learnt from all this.

5.2 Second discussion point

Please don't use these as sub-heading labels - change them to be what your point actually is.

5.3 Third discussion point

5.4 Weaknesses and next steps

Weaknesses and next steps should also be included.

Appendix

A Additional data details

B Model details

B.1 Posterior predictive check

In **?@fig-ppcheckandposteriorvsprior-1** we implement a posterior predictive check. This shows...

In **?@fig-ppcheckandposteriorvsprior-2** we compare the posterior with the prior. This shows...

Examining how the model fits, and is affected by, the data

B.2 Diagnostics

?@fig-stanareyouokay-1 is a trace plot. It shows... This suggests...

?@fig-stanareyouokay-2 is a Rhat plot. It shows... This suggests...

Checking the convergence of the MCMC algorithm

References

Alexander, Rohan. 2023. *Telling Stories with Data*. Chapman; Hall/CRC. https://tellingstorieswithdata.com/.

Goodrich, Ben, Jonah Gabry, Imad Ali, and Sam Brilleman. 2022. "rstanarm: Bayesian applied regression modeling via Stan." https://mc-stan.org/rstanarm/.

Horst, Allison Marie, Alison Presmanes Hill, and Kristen B Gorman. 2020. *palmerpenguins: Palmer Archipelago (Antarctica) penguin data*. https://doi.org/10.5281/zenodo.3960218.

R Core Team. 2023. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Toronto Shelter & Support Services. 2024. *Deaths of Shelter Residents*. https://open.toronto.ca/dataset/deaths-of-shelter-residents/.