Promoting Accessibility in Duke Data Visualization

Creating a Color-Blind Friendly Duke Color Package

Aidan Gildea

4/10/23

A detailed explanation of the development and use of the duke package.

Table of contents

Preface

This is the primary documentation or the Duke Statistical Science Thesis, **"Promoting Accessibility in Duke Data Visualization: Creating a Color-Blind Friendly Duke Color Package."** It was developed and published in the Spring of 2023. A special thanks to my wonderful advisor, Dr. Mine Çetinkaya-Rundel, who carefully guided this project's creation and ensured its success. An additional thank you to the members of my thesis committee, Dr. Colin Rundel and Dr. Merlise Clyde, for offering their extensive experience in statistical computing and R package development. I hope this thesis successfully details the motivation, process, and results of the **duke** package and how it can effectively empower users to create colorblind friendly Duke visualizations.

For questions and concerns, please reach out to me at aidan.gildea@duke.edu. In addition, I encourage you to post any Issues on the package Github repository.

1 Introduction

> Accessible design is good design – it benefits people who don't have disabilities as well as people who do. Accessibility is all about removing barriers and providing the benefits for everyone.

> Steve Ballmer, former Microsoft CEO (2000-2014)

In nearly all disciplines, there is an increasing push to promote accessibility. Data visualization should be no different. The goal of this thesis is to detail the motivations, development, and use of the **duke** package. By the end of "Promoting Accessibility in Duke Data Visualization," you should be sufficiently informed on how to create colorblind-friendly visualizations with colors that conform to Duke branding guidelines.

1.1 Motivation

My interest in writing an undergraduate thesis is centered on two goals: 1) identifying a material issue and 2) addressing it with a tangible solution. R packages tailored towards organization or company branding are quite common. The **ggtech** package allows users to incorporate colors representative of the Airbnb, Facebook, and Google brands, among many others, in their visualizations. Users can also transform their visualizations to have aesthetic features aligned with The Economist and The Wall Street Journal brands via the **ggthemes** package. These are just a couple of examples of how an R package can auto-theme ggplot visualizations to meet the branding requirements of a company. Only requiring a few functions, brand styling functions are easy-to-use yet very effective, making it an advantageous resource for companies to promote their brand.

With over 17,000 students, 38,000 employees, $1 billion annually in research expenditures, and six campuses, Duke University's identity literally spans the globe (Duke Research & Innovation, n.d.). And still, there exists no package to automate the process of making ggplot visualizations representative of the Duke brand. It is an important need that has yet to be met and can only be in addressed via package development. By learning the skill of package development to create a new Duke branding package, **duke**, I successfully meet my thesis criterion of 1) identifying a significant issue and 2) solving it with a real solution.

To best represent and communicate the Duke brand; however, the **duke** package must cater to the largest possible audience. Thus, this project also centers on an important consideration

in data visualization and branding package development: colorblind accessibility. As our use of data increasingly permeates all aspects of society, it is crucial that we integrate accessibility and inclusivity into our data visualization best practices. We must continuously consider questions like: Who is our audience? How can we make our visualizations more intelligible? Who might we be leaving behind? This thesis aims to empower users right here in the Duke community to more easily answer these questions. By developing an R package that allows for the creation of Duke official visualizations in a colorblind friendly way, we can begin to make accessibility a core tenet of data visualization.

1.2 Background

There are approximately 300 million color blind people across the globe ("About Colour Blindness," n.d.a). Approximately 1 in 12 men (8%) and 1 in 200 women (0.5%) are affected ("About Colour Blindness," n.d.b). More properly known as Color Vision Deficiency, or CVD, the condition is primarily inherited genetically but can also be acquired due to other chronic conditions such as diabetes. With the data visualization industry expected to nearly double by 2026, its application will further become a mainstay in fields like marketing, research, public health, and journalism (Vissichelli 2022). More people will be interacting with data visualization than ever before – both on the creator and reader ends – therefore, it is crucial that it is increasingly accessible to those with CVD.

R packages that implement color blind friendly palettes already exist, but this project is unique in its aim to construct a package that makes Duke's color palette more accessible. Duke provides a detailed description of its official color suite and stresses its commitment to meeting the Web Content Accessibility Guidelines (WCAG). The WCAG are globally recognized as the standard for web accessibility best practices in all fields (Initiative, n.d.). Duke details which color combinations from their suite are compliant, but does not offer a simple way to integrate the information into data visualizations.

The **duke** package intends to fill this need. For Duke community members, the package will allow for easy integration of key aesthetic and geometric features that will make their visualizations clean, Duke branded, and accessible. Given that the Statistics curriculum at Duke primarily uses **ggplot2** for visualization, **duke** will be focused on being compatible with its functionality. By establishing a package that allows users to easily integrate Duke's style guide with color blind friendly adjustments, we can improve the accessibility of Duke related visualizations. The implementation of this package will be an effective step toward accessibility in the Duke community, and hopefully, will prompt future work on making data visualization more inclusive.

2 Methods

2.1 Color Sources

The office of University Communications provides an official color guide for the Duke brand. The color palette is intended to both bolster the brand of the University, as well as ensure cohesion among all official content ("Colors," n.d.a). The primary colors of the official palette are Duke Royal Blue and Duke Navy Blue. It is highly encouraged that one of these colors is present in any Duke official project, as they are reflective of the Duke brand. The color guide also details an extended palette, which includes an additional 18 secondary and tertiary colors. Colors such as Copper, Eno, Ironweed, Persimmon, and Cast Iron are in the extended palette. University Communications urges the use of this extended palette for, "graphic accents, typography, backgrounds, [and] call-to-action buttons" ("Colors," n.d.b). For each color in the palette, users can access the following information:

- Color Name
- Pantone Matching System (PMS)
- Hex Code (HEX)
- Cyan, Magenta, Yellow, and Key (CMYK) Color Model
- Red, Green, Blue Color Model

We will primarily interact with the Hex codes for these colors in this project, shown in Table 2.1. Hex codes are a hexidecimal (base 16 numbering system) format for representing the amount of RBG in a particular shade (Rouse 2021). Additionally, Hex codes are recognized as the best color system for HTML and CSS formats; therefore, their use will be advantageous for data visualization in publication and web browser contexts.

Table 2.1: **Duke Color Palette**

Name	Hex Code
Duke Royal Blue	#00539B
Duke Navy Blue	#012169
Copper	#C84E00
Persimmon	#E89923

Name	Hex Code
Dandelion	#FFD960
Piedmont	#A1B70D
Eno	#339898
Magnolia	#1D6363
Prussian Blue	#005587
Shale Blue	#0577B1
Ironweed	#993399
Hatteras	#E2E6ED
Whisper Gray	#F3F2F1
Ginger Beer	#FCF7E5
Dogwood	#988675
Shackleford	#DAD0C6
Cast Iron	#262626
Graphite	#666666
Granite	#B5B5B5
Limestone	#E5E5E5

In 2018, Duke added a color accessibility guide to their branding site outlining how users could meet the Web Content Accessibility Guidelines (WCAG) 2.0 AA guidelines. The WCAG are internationally recognized standards on how to make web content more accessible to people with disabilities - including those with color vision deficiency. Their accessibility standards are principled on four tenets: Perceivable, Operable, Understandable, and Robust ("Introduction to Understanding WCAG 2.0 | Understanding WCAG 2.0," n.d.). Duke's accessibility guide illustrates a grid of every text-background color pairing in their palette and their respective compliance to the WCAG. There are four compliance categories specified:

- **AAA:** highest compliance, text-background contrast ratio of 7+

- **AA:** mid-range compliance, text-background contrast ratio of 4.5+

- **AA18:** mid-range compliance with large text only (18pt or 14pt bold), text-background contrast ratio of 3+

- **DNP:** does not pass compliance guidelines, text-background contrast ratio of less than 3

These compliance metrics will be an important resource for identifying color combinations that are accessible and selecting them for use in the **duke** package. The ratios provided by this guideline are intended for background and foreground combinations; however, they also apply to colors near and/or adjacent to one another. Duke's inclusion of WCAG compliance information on their branding website is an important step towards accessibility in University content, but it would benefit from a built-in functionality that allows users to interact with

it more easily. By incorporating these recommendations into the **duke** package, this project aims to empower users to more deftly create accessible Duke data visualizations.

2.2 Color Metrics

In selecting the optimal color palette, several color difference metrics were considered. Euclidean distance is a commonly used measure that calculates the difference in the RGB coordinates of two colors:

$$\text{distance}_e = \sqrt{(R_2 - R_1)^2 + (G_2 - G_1)^2 + (B_2 - B_1)^2}.$$

(Iyer 2020)

However, Euclidean distance can be a flawed metric because the standard RGB (sRGB) color space is perceptually uniform. This is because of irregularity in the sRGB color space, which causes evenly spaced hues to range in their perceptual similarity or difference (Madsen, n.d.a). In turn, color pairings that appear similar can have a greater Euclidean distance than those that appear different (Madsen, n.d.b). The CIE chromaticity diagram (see Figure 2.1) exhibits the color spectrum according to human vision. Its parabolic nature leads to irregularities in linearly scaled color scales (i.e., sRGB) and therefore weakens its efficacy in this context.
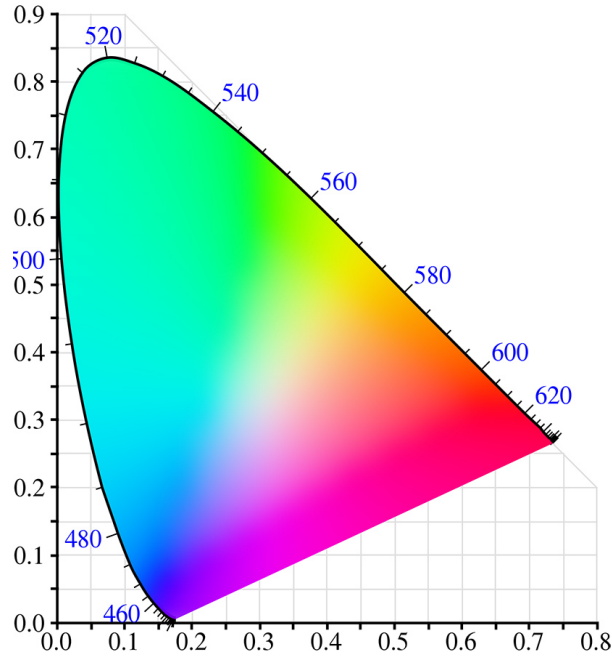
Figure 2.1: CIE chromaticity diagram.

10

To accommodate these irregularities and center our color selection on human perception, we use the color contrast ratio as our primary metric. Color contrast is computed using "relative luminance" ("How to Calculate Colour Contrast" 2019). According to WCAG, relative luminance is the "relative brightness of any point in a color space, normalized to 0 for darkest black and 1 for lightest white" ("Web Content Accessibility Guidelines (WCAG) 2.0," n.d.a). The formula for the color contrast ratio is as follows:

$$\frac{(L1 + 0.05)}{(L2 + 0.05)}$$

By dividing the the relative luminance of the lighter color (L1) by that of the darker color (L2), the color contrast ratio can range from a value of 1 (no contrast) to 21 (highest contrast) ("Web Content Accessibility Guidelines (WCAG) 2.0," n.d.b). In line with the WCAG guidelines, we plan on only including color pairings in the package that meet a compliance level of AA18 or better.

It must be noted that given a total of 22 colors in the Duke official suite, our definition of the "optimal" palette may not be fully reliant on the color contrast ratio. The package is, at its core, representative of the Duke brand and should meet certain aesthetic expectations for those with and without color vision deficiency. The development of our color scales, specifically, incorporates a somewhat ad-hoc approach in order to strike a balance between accessibility and recognition of the Duke brand. In finding a color palette successful on both fronts, users can be confident that they are producing visualizations that are identifiably Duke-branded and accessible to a greater audience.

2.3 Package Development

As previously mentioned, package development posed a new challenge to the skill set I had learned through the Statistics curriculum. The R Packages (2e) text book served as an essential guide for learning, understanding, and eventually building the **duke** package. By clearly outlining the key structures of an R package, the functionality of the **devtools** package, and the importance of testing, I was well equipped to create **duke**.

The **duke** package offers four functions for users to make Duke branded visualizations in a colorblind friendly way.

2.3.1 `theme_duke()`

This function is largely inspired by the `theme()` function and its many derivatives. Its primary focus is to set the aesthetic elements for the visualization's background and accessory features. These include the background color, grid lines, axis ticks, plot text, etc. With accessibility at

the forefront of this project's goals, it is essential that these choices lead to not only a visually pleasing result, but to one that is comprehensible as well. To do so, `theme_duke()` implements a minimalist design with a White background and grid lines removed. This decision reduces excessive noise in the plot, and instead allows audiences to focus on the story the data tells.

This design is complemented by deliberate text specifications that ensure plot readability. The function applies uniform coloring for text across the entirety of the visualization. All possible text features of a ggplot (title, axes, caption, legend, etc.) are in in Duke Navy Blue because of its strong contrast to the White background. It has a contrast ratio of 14.76 with White, which is the third highest among all colors in the Duke extended palette. Sizing of major text elements is also handled carefully to promote legibility. Readers with color vision impairment and/or low vision benefit from larger text settings (UNC School of Medicine, n.d.). Rather than manually increasing text size, `theme_duke()` encodes text sizing relative to the scale of the visualization. In turn, text adjusts according to users' manipulations of plot dimensions, guaranteeing that text is both appropriately large and adequately fit to the visualization.

In addition, `theme_duke()` allows for the simple integration of specific fonts in visualizations. We recommend that all text is in Atkinson-Hyperlegible font. Named after the founder of the Braille Institute, J. Robert Atkinson, the font uses "unambiguous, distinctive elements" to improve readability for individuals with vision impairment ("Atkinson Hyperlegible," n.d.). Its use - both in the package and all complimentary documentation - is essential to promoting vision accessibility. Font implementation was one of the most difficult challenges throughout the package development process. R provides a limited suite of fonts, requiring users to download and load additional font families to their own system. On the individual-level, this is not too difficult. However, the process necessitates careful thought when building a package for community use. *How can users integrate quickly access the package's required fonts? What font implementation will require minimal memory? Why is being transparent with users important?* We must closely consider these questions, among others, when adding font specification to **duke**.

The ongoing aim of this theme function is to implement Atkinson-Hyperlegible font by default across all text features of visualizations. This requires that users have the necessary font files locally installed to their systems. If already available locally, we imagine `theme_duke()` automatically using Atkinson-Hyperlegible font. If not, a message would be printed directing users to where the font files can downloaded and fonts used in plots would default to the **ggplot2** default font, sans. We have constructed a single-use function that downloads the necessary Atkinson-Hyperlegible files to be used in that instance. This solution has not been added yet because it is not stable across all user systems. There are a few minor bugs in the **systemfonts** package that have roadblocked our development process and must be addressed first before we can finish this solution.

In the mean time, we recommend that users download Atkinson-Hyperlegible and then specify its use in the base family argument of `theme_duke()`. This is not a fully streamlined fix, yet it ensures that users on all systems can implement **duke** without encountering errors. If font in not specified, the function uses sans font. The application of Atkinson Hyperlegible font is

key to improving accessibility in data visualization and is of high priority in **duke**'s ongoing development.

2.3.2 `scale_duke_continuous()`

This function integrates Duke's official color palette into geometric objects (geom) that use a continuous scale. Examples of commonly used plots implementing continuous geom objects include scatter plots, heat maps, and jitter plots. The primary challenge of adopting a continuous color scale in a color blind friendly way is the lack of color distinction that naturally occurs. Continuous color scales are defined by a low and high end of a two color gradient, where the data assumes colors within this range based on its respective value. However, continuous plots do not need complete distinction to be successful. It is often most important that readers can identify overall trends in continuous plots.

To optimize `scale_duke_continuous()` for users with impaired color vision, the continuous gradient must sequence from a dark to a light color (C. Wilke, n.d.). This is crucial because the data will be plotted beginning on the darker end of the scale, which has a higher contrast with our White background. Therefore, our gradient scale defines its low end as Duke Royal Blue and its high end as Hatteras. Duke Royal Blue has a color contrast ratio of 7.75 with White, making it sufficiently distinguishable. This, in tandem with Duke's recommendation to include Duke Royal or Navy Blue in all communications, makes it the ideal choice for the low end of the scale. Hatteras is a light grey with subtle violet undertones, making it a pleasing complement to Duke Royal Blue ("Hatteras Gray," n.d.). It is not highly distinct from the White background, but this is typical for the light extreme of continuous scales.

2.3.3 `scale_duke_color_discrete()` and `scale_duke_fill_discrete()`

To implement a discrete scale from Duke's official color palette, users can add `scale_duke_color_discrete()` and `scale_duke_fill_discrete()` layers to their visualizations. These functions provide a discrete palette that can be used for the color and fill parameters of ggplot geometric objects. An appropriate discrete palette is essential to creating user-friendly visualizations because its colors indicate distinct categories that exist in the data. Whereas continuous plots typically exhibit overall trends, discrete plots must be able to delineate effectively between specific groups. A majority of geom objects can incorporate a discrete scale, including bar plots, histograms, and density plots.

As previously mentioned, this project observes color contrast ratio as the primary metric for determining colorblind friendly combinations. To identify visibly distinct colors for our discrete palette, we generate the contrast ratio between all pairing from the Duke color guide, with the addition of White since it is the background color. Figure 2.2 shows a heat map of color contrasts of all pairings. Lighter tiles in the heat map indicate pairings with higher contrast ratios, which are preferred.
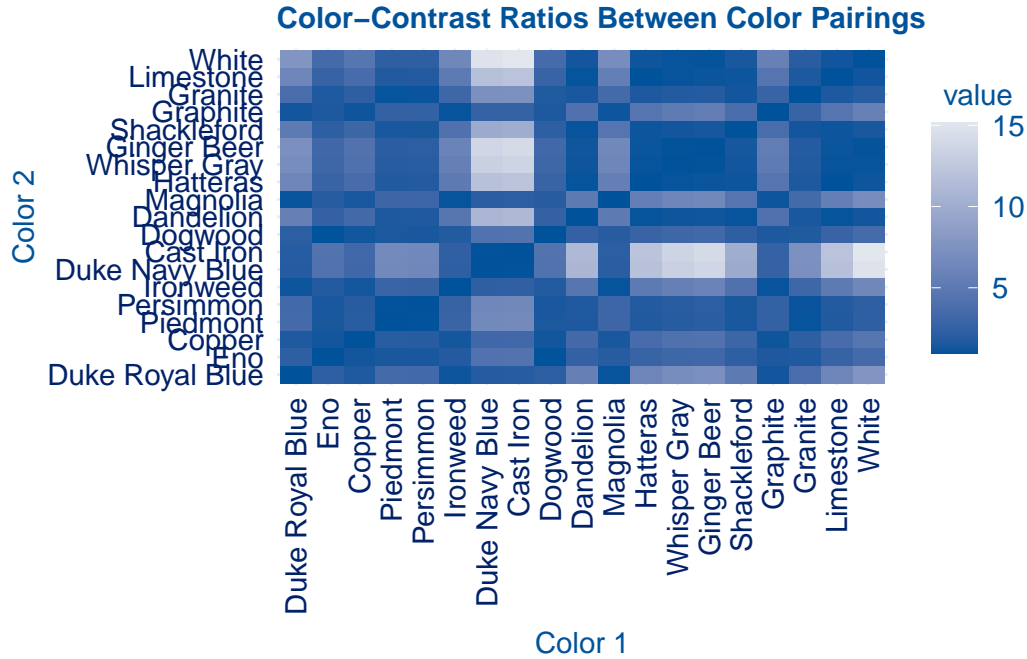
Figure 2.2: Heatmap of color contrast ratio between color pairings.

Then, using these ratios, we craft a palette that promotes accessibility and the Duke aesthetic. A rule of thumb for discrete color scales is limiting to three to five colors. Additionally, once eight to ten categories or more are reached, matching colors to categories becomes too burdensome to be useful, even if the colors are distinguishable in principle (C. O. Wilke, n.d.). This lends to less busy plots that are more easily understandable. Thus, the goal of the our discrete palette is to carefully order colors so that those most used (those at beginning of palette) are most compliant. The result is the palette shown in Figure 2.3.
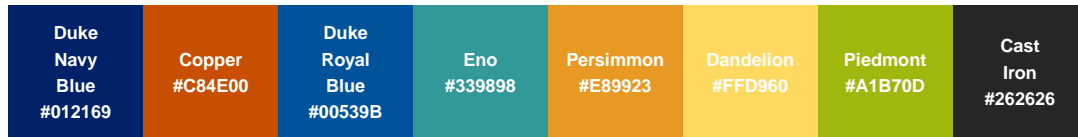
Figure 2.3: Discrete palette. Colors (from left to right): Duke Navy Blue, Piedmont, Copper, Duke Royal Blue, Eno, Persimmon, Dandelion, Cast Iron.

While not perfectly compliant with WCAG guidelines, the palette achieves our goal of optimizing colorblind accessibility and representing the Duke brand. As seen in Figure 2.4, a majority of our color pairings meet at minimum the AA18 rating, as indicated by lighter tiles.

The final palette includes the following colors: Duke Navy Blue, Piedmont, Copper, Duke Royal Blue, Eno, Persimmon, Dandelion and Cast Iron. Out the 28 distinct colors pairings
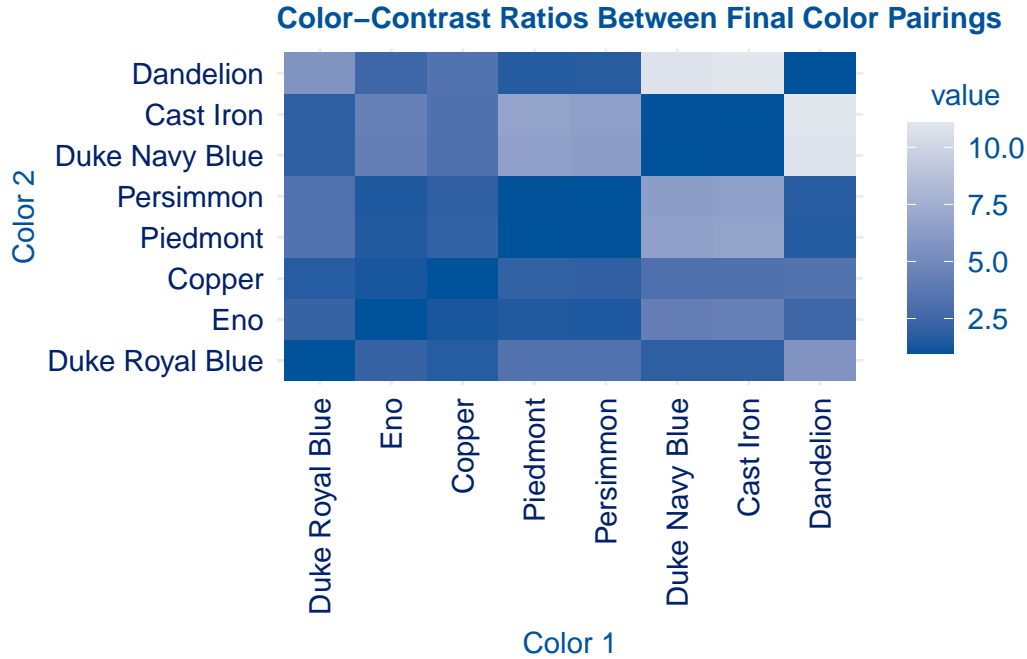
Figure 2.4: Heatmap of color contrast ratio between final color pairings.

that exist from this palette (excluding pairings of identical colors), 14 have a color contrast ratio of 3 or greater. This is compliant with WCAG's AA18 rating, which specifies the use of larger text to be accessible. The colors from this palette will primarily be used to fill or color geometric objects in ggplot - rather than text - which offers complimentary features (ie. shape, larger areas, borders) that promote accessibility. Thus, we proceed under the assumption that an AA18 categorization is sufficient.

To ensure that this palette is accessible, we use the `colorblindr` package to visualize sample plots from the perspective of individuals with different types of color vision deficiency. There are several categorizations of CVD, yet the most common is red-green color blindness. As the name suggests, it makes it difficult for individuals to distinguish between red and green. Among this population, approximately 62.5% are deuteranomalous - meaning they have green-weak vision that makes green appear more red (Woods 2021a). Protonopia and deuteranopia prevent individuals from distinguishing between red and green at all, yet are far less prevalent at only 12.5% of the red-green colorblind population each (Woods 2021b).

Red-green color blindness is by far the most common type, accounting for approximately 99% of those with CVD. For the purpose of creating a feasible project scope, we focus primarily on accessibility for those with red-green colorblindness. Future work on this project - or one with a similar goal - should investigate the feasibility of visualization accessibility for those with blue - yellow color blindness, monochromacy, etc.

Table 2.2: Color Contrast Ratios

Color 1	Color 2	Contrast Ratio
Dandelion	Cast Iron	11.059041
Dandelion	Duke Navy Blue	10.785676
Cast Iron	Piedmont	6.698114
Duke Navy Blue	Piedmont	6.532546
Cast Iron	Persimmon	6.494391
Duke Navy Blue	Persimmon	6.333858
Dandelion	Duke Royal Blue	5.661465
Cast Iron	Eno	4.385449
Duke Navy Blue	Eno	4.277046
Piedmont	Duke Royal Blue	3.428972
Dandelion	Copper	3.376002
Persimmon	Duke Royal Blue	3.324680
Cast Iron	Copper	3.275781
Duke Navy Blue	Copper	3.194808
Dandelion	Eno	2.521758
Eno	Duke Royal Blue	2.245047
Piedmont	Copper	2.044738
Persimmon	Copper	1.982548
Cast Iron	Duke Royal Blue	1.953388
Duke Navy Blue	Duke Royal Blue	1.905103
Dandelion	Persimmon	1.702860
Copper	Duke Royal Blue	1.676973
Dandelion	Piedmont	1.651068
Piedmont	Eno	1.527350
Persimmon	Eno	1.480895
Copper	Eno	1.338749
Persimmon	Piedmont	1.031369
Cast Iron	Duke Navy Blue	1.025345

```
splot <- ggplot(penguins, aes(
  x = bill_length_mm, y = bill_depth_mm,
  color = species
)) +
  geom_point() +
  labs(title = "Flipper Length v. Depth", x = "Bill Length (mm)", y = "Bill Depth (mm)") +
  theme_duke() +
  scale_duke_color_discrete()
splot
```

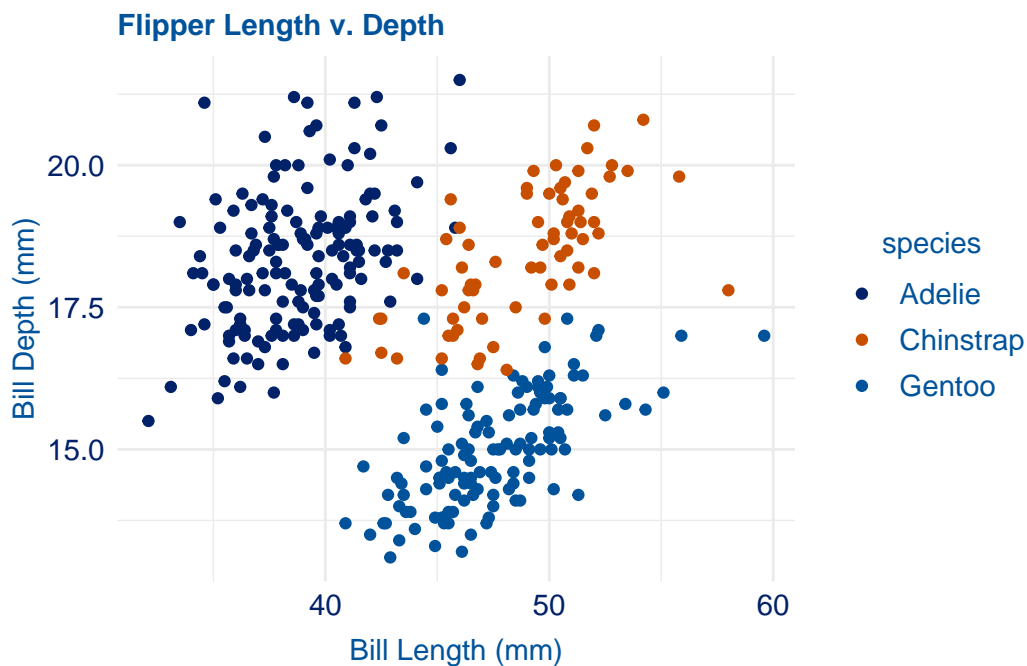Warning: Removed 2 rows containing missing values (`geom_point()`).

Figure 2.5: Sample scatter plot from non-deuteronamolous perspective.

```
#|
comp <- readRDS("~/Desktop/dukethesis/data/deutan.rds.gzip")
comp
```
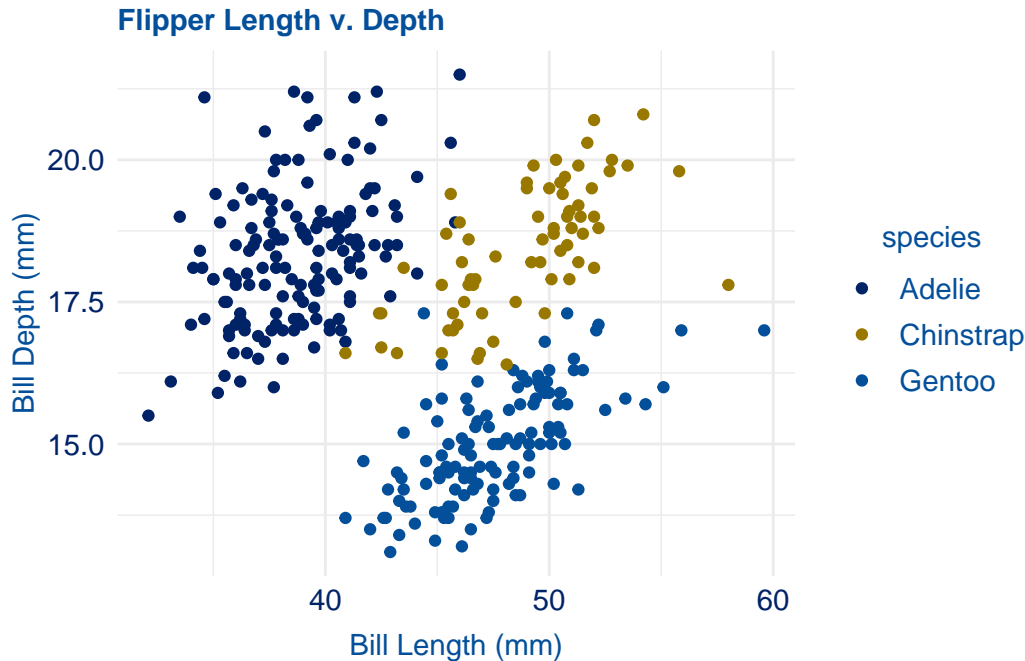
Flipper Length v. Depth

Figure 2.6: Sample scatter plot from deuteranomalous perspective.

2.4 Lessons Learned

The package development process can be daunting; I know it was for me. The excess of package dependencies, function documentation, and test scripts quickly becomes complex and overwhelming, making it all the more important to approach the process properly informed. Here are some lessons I learned throughout the process and takeaways for success in package development:

1. Package development is an iterative process, take your time. It is crucial that you approach each step in building your package with care - even those that are mundane. I often found myself skipping ahead to more attractive tasks like constructing a new function, leaving tedious but important tasks like documentation behind. This typically resulted in errors when checking my package. And they were increasingly harder to address when time had passed since I last worked on it. The process is far smoother when you continuously check for issues and address them immediately, rather than put them on the back-burner. It will save you from wasted time and frustration, so make sure to lean into the iterative style of package development.

2. The devtools package is your best friend. The `devtools` package is inherent to the package development process, but the way you use it can make all the difference. The R Package (2e) textbook was an integral resource for learning when and how to use the various

functions of `devtools`. It details the order you should use its functions to load your package, create necessary documentation and vignettes, check for existing errors, and eventually build your package. While it may seem minor, memorize the shortcuts for the primary functions of `devtools` (load_all(), check(), document()). Over time, they will save you a significant amount of time and make your package development process more efficient.

3. Engage with existing resources online and collaborate with others. A common misconception in building new packages is that it is an entirely individual undertaking. I identified colorblind accessibility in the Duke data visualization as a unique and necessary problem to address, in turn, inspiring the **duke** package. At first, I thought that this meant I would and *should* construct the package on my own. This could not be further from the truth. As a package intended to be integrated into ggplot2 visualizations, **duke** shares many of the same functionalities as those in `theme()` and `scale()`. These existing resources demonstrated how to create the general structures of my functions, which then I could adapt to implement Duke's official palette, a minimalist background aesthetic, etc. The "worth" of your package is not defined by its lack in crossover with other packages existing; therefore, make sure to engage with the resources at your disposal. I recommend you explore existing packages on GitHub, troubleshoot package issues by asking questions in the Posit Community, and find answers to difficult questions already asked on StackOverflow.

3 Function Descriptions

3.1 `theme_duke()`

Implements theme components.

Defines the overall aesthetic and thematic features of the plot. This function specifies simple background, grid line, text, and legend arguments to create minimalist design. Its use is intended for ggplot objects. See https://aidangildea.github.io/duke/reference/theme_duke.html for more information and examples.

3.2 `scale_duke_continuous()`

Continuous scale constructor.

Applies a Duke branded and accessible continuous color scale to ggplot geometric objects. It is applicable for both fill and color arguments. Defines gradient scale from dark to light to improve visibility and contrast for readers. See https://aidangildea.github.io/duke/reference/scale_duke_continuous.html for more information and examples.

3.3 `scale_duke_fill_discrete()`

Discrete scale constructor for fill argument.

Applies a Duke branded and accessible discrete color palette to ggplot geometric objects using fill argument. It prioritizes high contrast colors aligned with Web Content Accessibility Guidelines (WCAG). See https://aidangildea.github.io/duke/reference/scale_duke_fill_discrete.html for more information and examples.

3.4 `scale_duke_color_discrete()`

Discrete scale constructor for color argument.

Applies a Duke branded and accessible discrete color palette to ggplot geometric objects using color argument. It prioritizes high contrast colors aligned with Web Content Accessibility Guidelines (WCAG). See https://aidangildea.github.io/duke/reference/scale_duke_color_discrete.html for more information and examples.

3.5 `duke_pal()`

Discrete palette constructor.

Defines 8-color palette to be used by discrete scale function. It is intended for internal package use and not to be directly called. See https://aidangildea.github.io/duke/reference/duke_pal.html for more information and examples.

4 Package Use

```
library(duke)
library(palmerpenguins)
library(ggmosaic)
library(ggplot2)
library(dplyr)
```

This vignette aims to comprehensively demonstrate the use and functionality of the package **duke**. **duke** is fully integrated with the **ggplot2** and allows for the creation of Duke official branded visualizations that are color blind friendly.

For this purposes of this vignette, we will use the palmerspenguins package, which provides a simple dataset on Antarctic penguins and their characteristics: `penguins` (Horst, Hill, and Gorman, n.d.). The dataset has eight variables - some numeric and some categorical - and 344 observations, each representing a unique penguin.

4.1 Scatter Plot - Continuous Color

```
scatterplot_c <- ggplot(
  penguins,
  aes(x = bill_length_mm, y = bill_depth_mm)
) +
  geom_point(aes(color = body_mass_g)) +
  labs(
    title = "Bill Length vs. Bill Depth",
    x = "Bill Length (mm)", y = "Bill Depth (mm)"
  )

scatterplot_c +
  scale_duke_continuous() +
  theme_duke()
```

Bill Length vs. Bill Depth
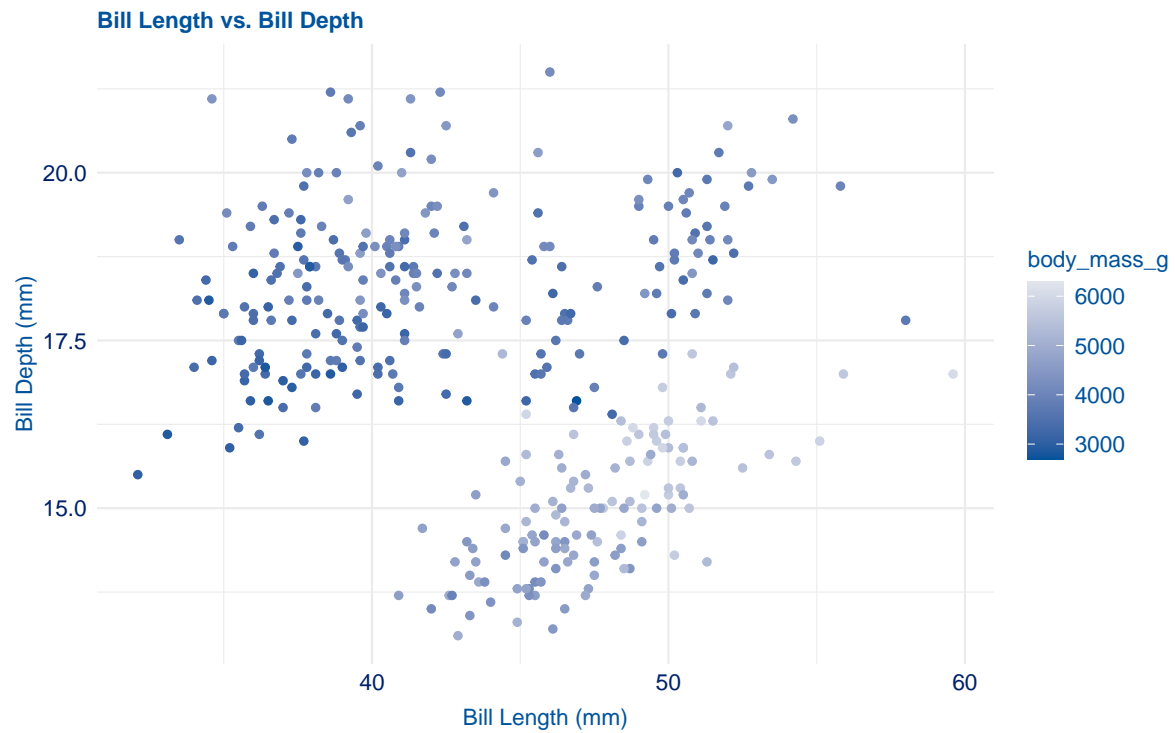

```
scatterplot_c +
  geom_point(aes(shape = species)) +
  scale_duke_continuous() +
  theme_duke()
```

Bill Length vs. Bill Depth
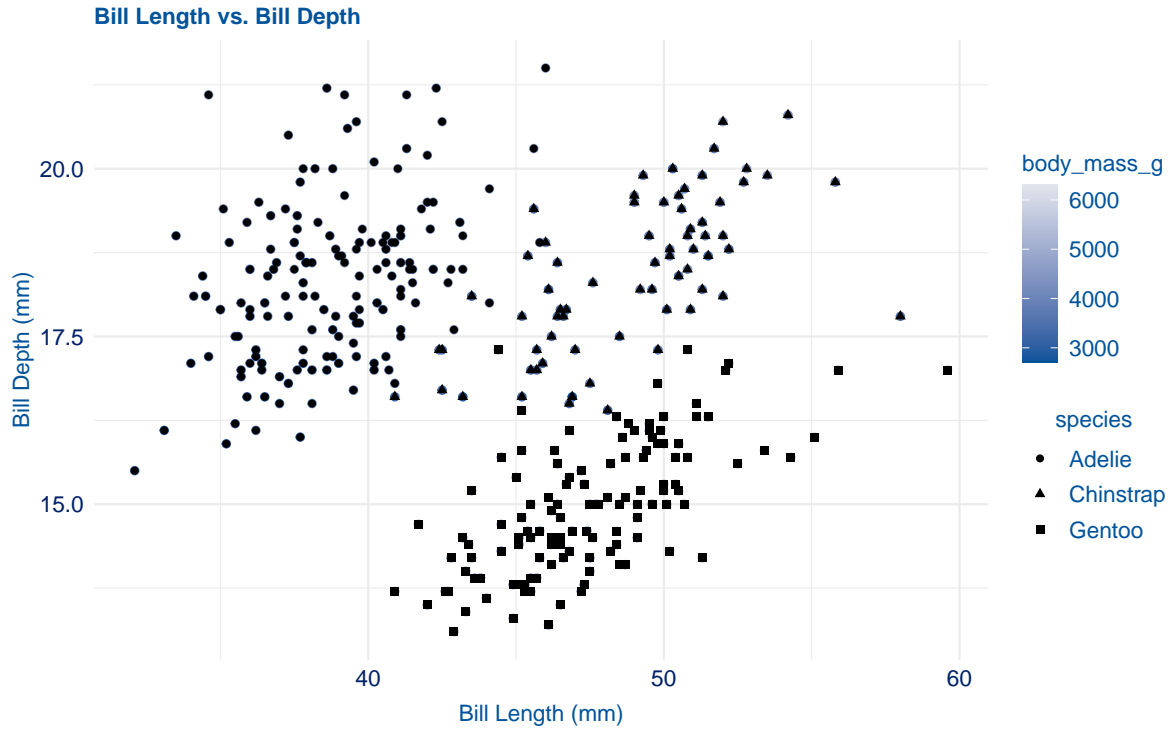
## 4.2 Scatter Plot - Discrete Color

```
scatterplot_d <- ggplot(
  penguins,
  aes(x = bill_length_mm, y = bill_depth_mm, color = species)
) +
  geom_point(size = 3) +
  labs(
    title = "Bill Length vs. Bill Depth",
    caption = "(Colors used) \n Duke Royal Blue, Eno, Copper",
    x = "Bill Length (mm)",
    y = "Bill Depth (mm)"
  )

scatterplot_d +
  theme_duke() +
  scale_duke_color_discrete()
```

Bill Length vs. Bill Depth

Bill Depth (mm)

20.0

17.5

15.0

40   50   60

Bill Length (mm)

species

● Adelie
● Chinstrap
● Gentoo

(Colors used)
Duke Royal Blue, Eno, Copper
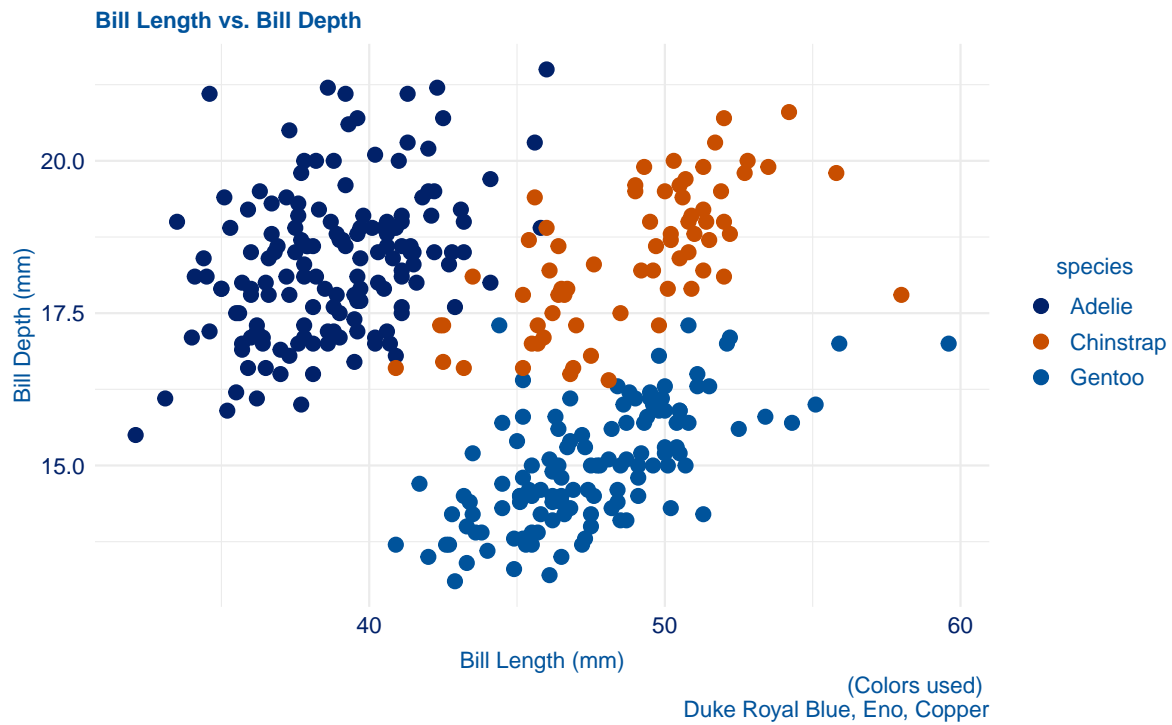
```
scatterplot_d +
  facet_wrap(~species) +
  theme_duke() +
  scale_duke_color_discrete()
```

Bill Length vs. Bill Depth

(Colors used)
Duke Royal Blue, Eno, Copper
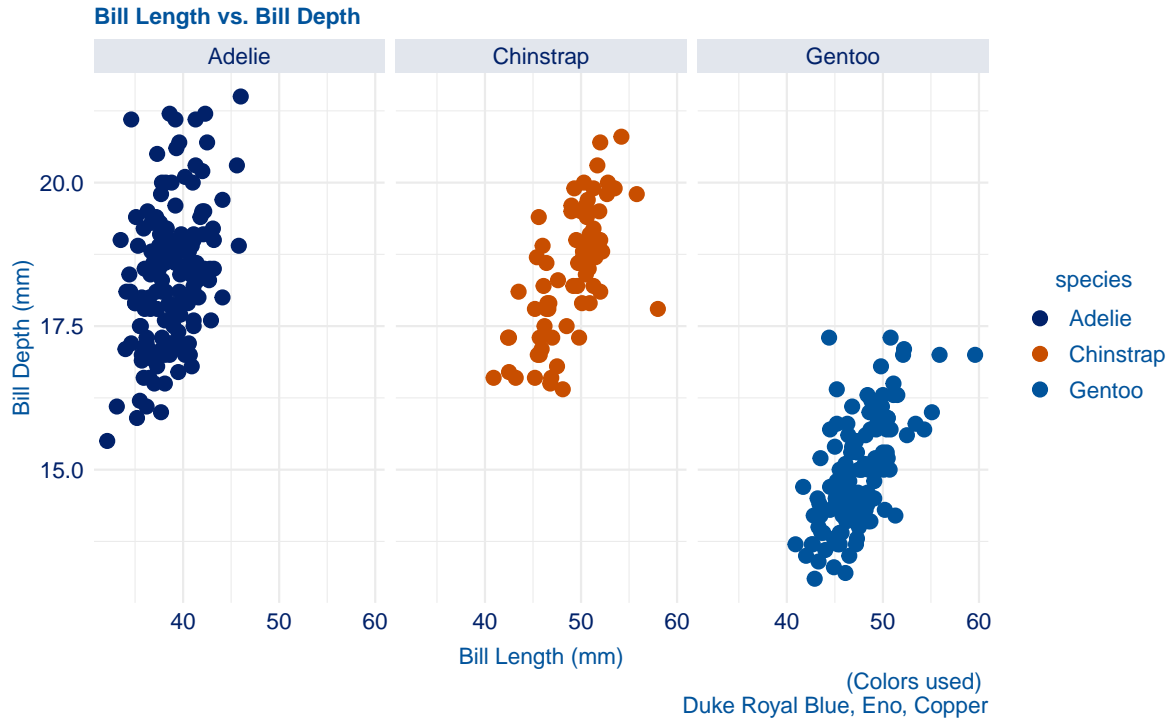
4.3 Bar Plot

```
barplot <-
  ggplot(penguins, aes(x = species, fill = species)) +
  geom_bar() +
  labs(
    title = "Distribution of Penguin Species",
    caption = "(Colors from left to right) \n Duke Royal Blue, Duke Navy Blue, Copper",
    x = "Species",
    y = "Count"
  )

barplot +
  scale_duke_fill_discrete() +
  theme_duke()
```

26

Distribution of Penguin Species

(Colors from left to right)
Duke Royal Blue, Duke Navy Blue, Copper
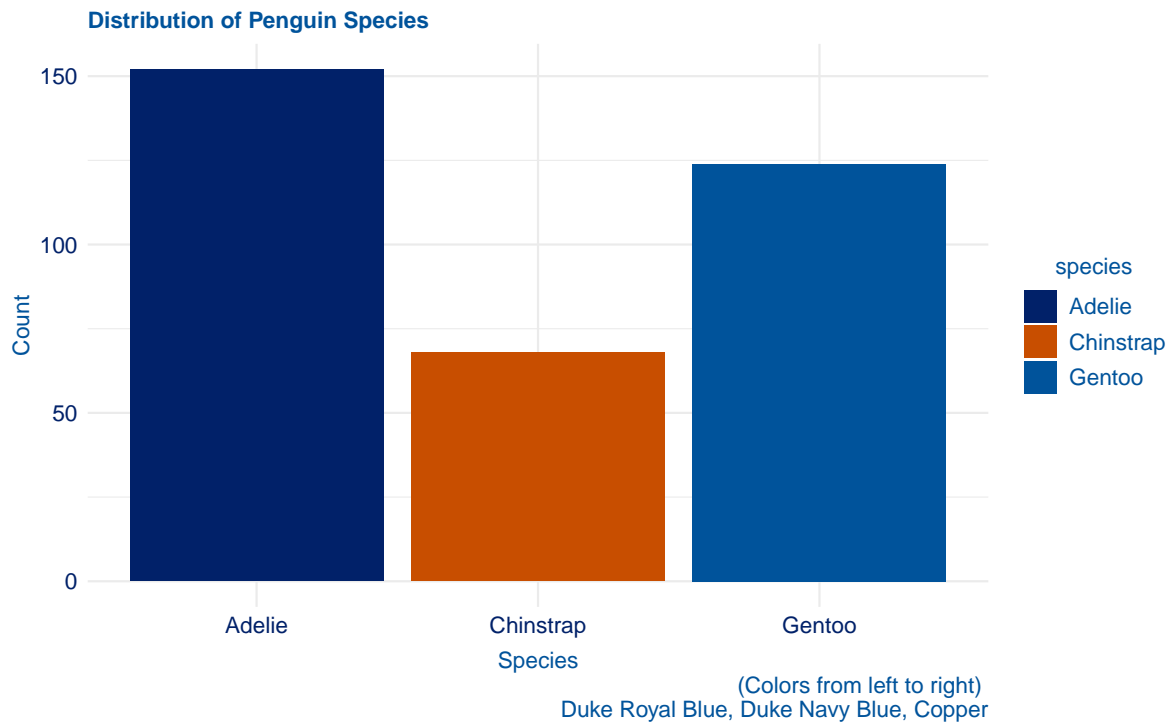
```r
m_penguins <- penguins %>%
  dplyr::filter(sex == "male")

barplot2 <- ggplot(m_penguins, aes(x = sex, fill = sex)) +
  geom_bar()

barplot2 +
  scale_duke_fill_discrete() +
  theme_duke()
```

```
# 8-category plot
barplot8 <- ggplot(students, aes(x = grade, fill = grade)) +
  geom_bar()

barplot8 +
  scale_duke_fill_discrete() +
  theme_duke()
```
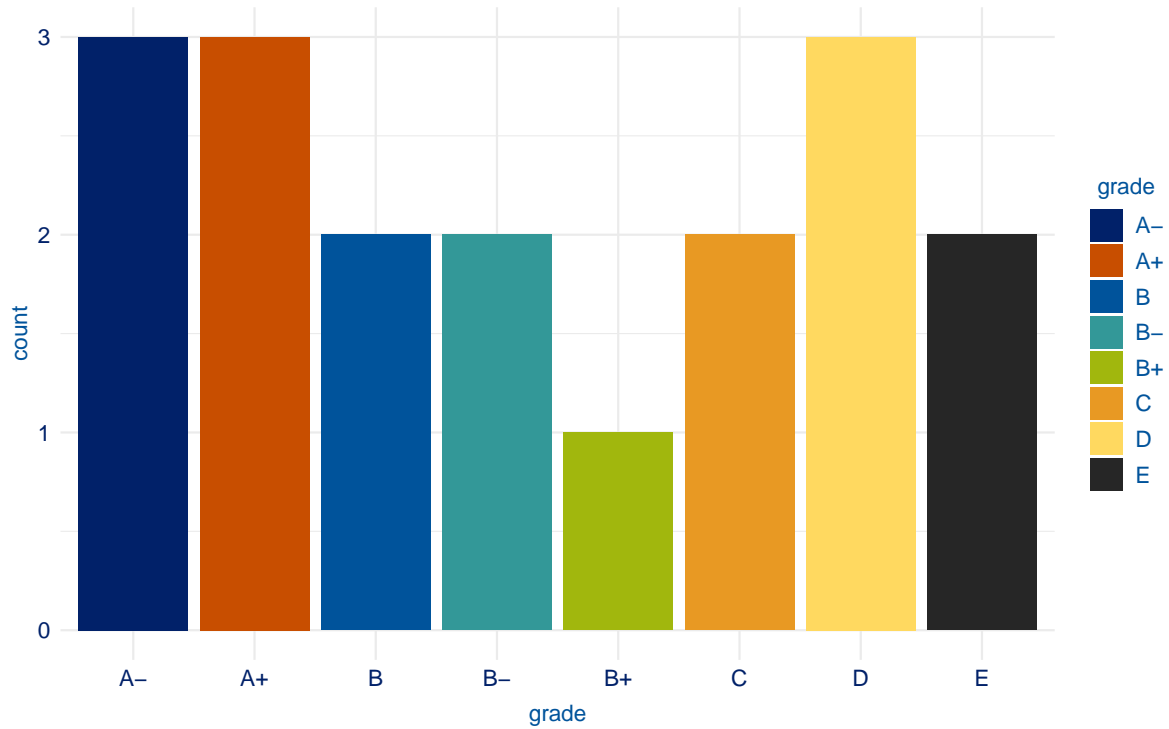
```r
# 7-category plot
barplot7 <- students %>%
  slice(-13) %>%
  ggplot(aes(x = grade, fill = grade)) +
  geom_bar() +
  scale_duke_fill_discrete() +
  theme_duke()

# 6-category plot
barplot6 <- students %>%
  slice(-c(9, 10, 13)) %>%
  ggplot(aes(x = grade, fill = grade)) +
  geom_bar() +
  scale_duke_fill_discrete() +
  theme_duke()

# 5-category plot
barplot5 <- students %>%
  slice(-c(9, 10, 13, 7, 18)) %>%
```

```
  ggplot(aes(x = grade, fill = grade)) +
  geom_bar() +
  scale_duke_fill_discrete() +
  theme_duke()

# 4-category plot
barplot4 <- students %>%
  slice(-c(9, 10, 13, 7, 18, 4, 8)) %>%
  ggplot(aes(x = grade, fill = grade)) +
  geom_bar() +
  scale_duke_fill_discrete() +
  theme_duke()
```

4.4 Histogram

```
histplot <- ggplot(penguins, aes(body_mass_g)) +
  geom_histogram(aes(fill = species), alpha = 0.8) +
  labs(title = "Distribution of Penguin Body Mass",
       caption = "(Colors used) \n Duke Royal Blue, Duke Navy Blue, Copper",
       x = "Body Mass (g)",
       y = "Count")

histplot +
  scale_duke_fill_discrete() +
  theme_duke()
```

Distribution of Penguin Body Mass

(Colors used)
Duke Royal Blue, Duke Navy Blue, Copper
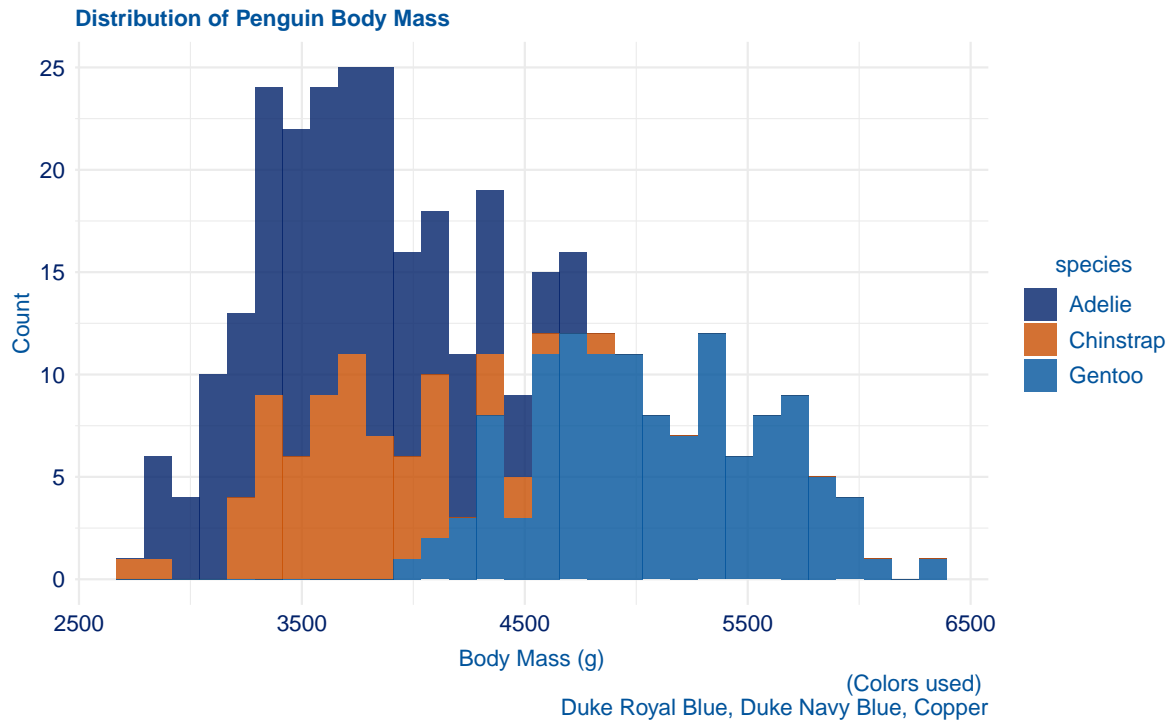
4.5 Box Plot

```r
boxplot <- ggplot(penguins, aes(sex, body_mass_g)) +
  geom_boxplot() +
  labs(
    title = "Comparison of Body Mass By Sex",
    x = "Sex",
    y = "Body Mass (g)"
  )

boxplot +
  theme_duke()
```

Comparison of Body Mass By Sex
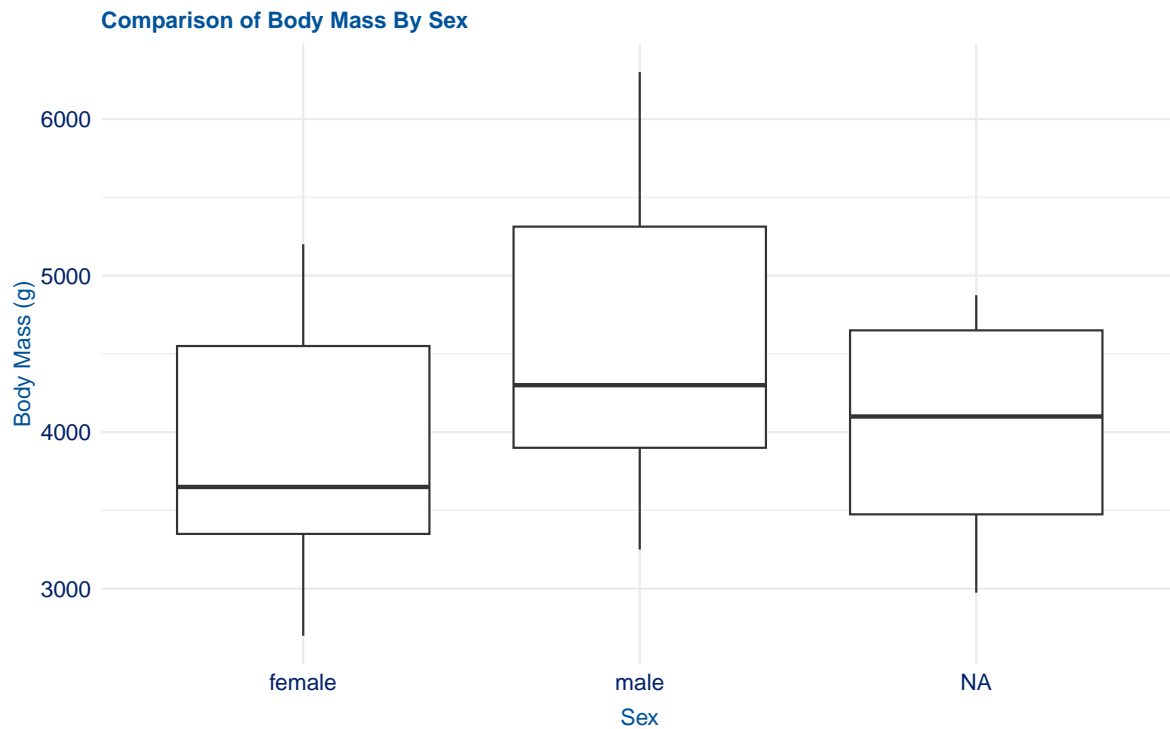
4.6 Density Plot

```
densityplot <- ggplot(penguins, aes(bill_depth_mm)) +
  geom_density(aes(fill = species)) +
  labs(
    title = "Density of Penguin Bill Depth",
    x = "Bill Depth (mm)",
    y = "Densiy"
  )

densityplot +
  scale_duke_fill_discrete() +
  theme_duke()
```

Density of Penguin Bill Depth
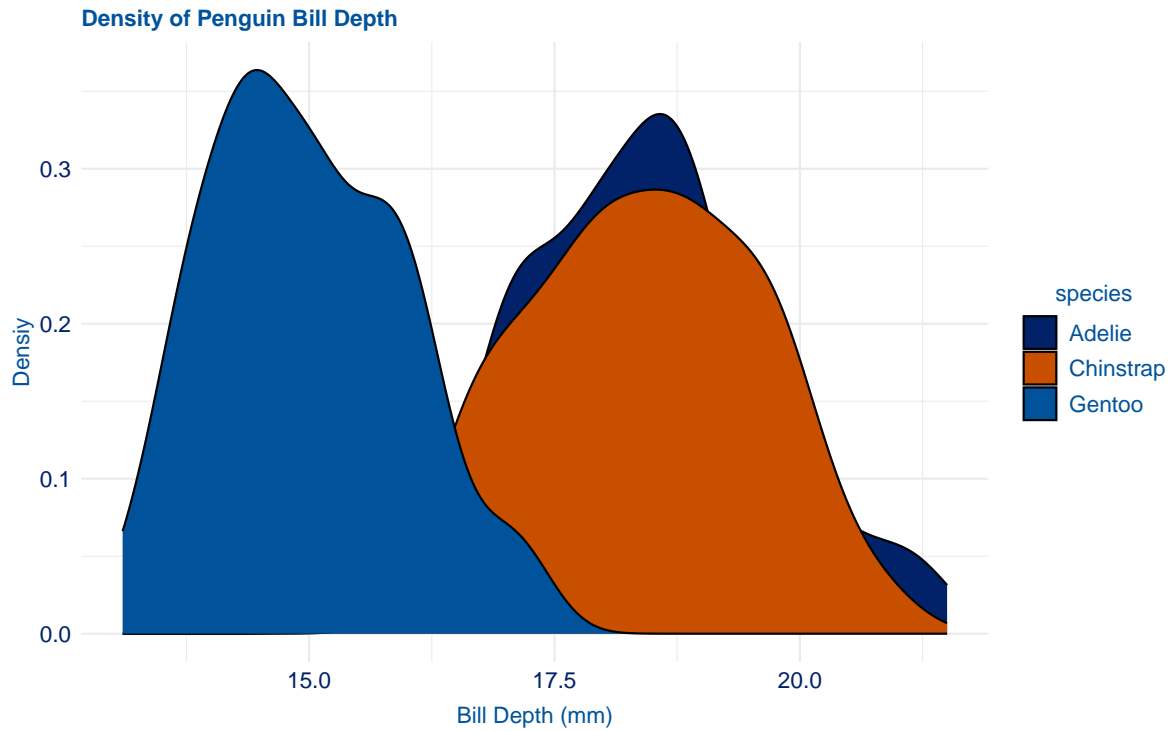
4.7 Jitter Plot - Discrete Color

```
jitterplot_d <- ggplot(penguins, aes(as.factor(year), body_mass_g)) +
  geom_jitter(aes(color = species)) +
  labs(
    title = "Comparison of Body Mass By Year",
    x = "Year",
    y = "Body Mass (g)"
  )

jitterplot_d +
  scale_duke_color_discrete() +
  theme_duke()
```

Comparison of Body Mass By Year
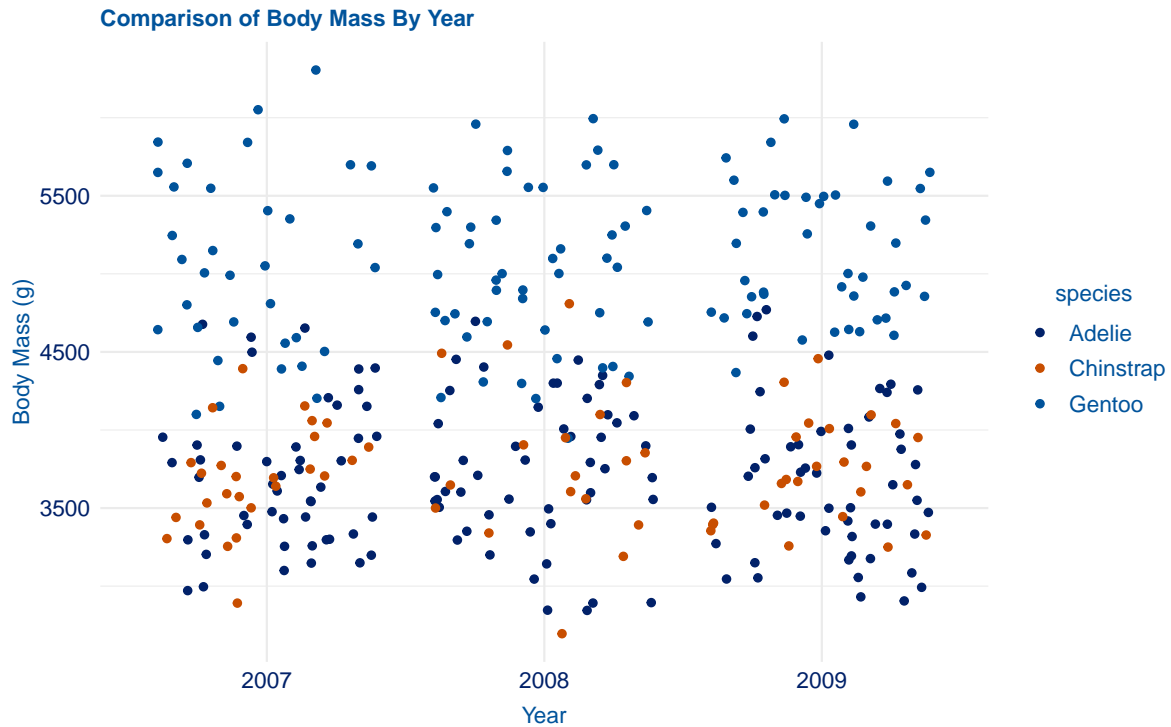
4.8 Jitter Plot - Continuous Color

```r
jitterplot_c <- ggplot(penguins, aes(as.factor(year), body_mass_g)) +
  geom_jitter(aes(color = bill_length_mm)) +
  labs(
    title = "Comparison of Body Mass By Year",
    x = "Year",
    y = "Body Mass (g)"
  )

jitterplot_c +
  scale_duke_continuous() +
  theme_duke()
```

Comparison of Body Mass By Year
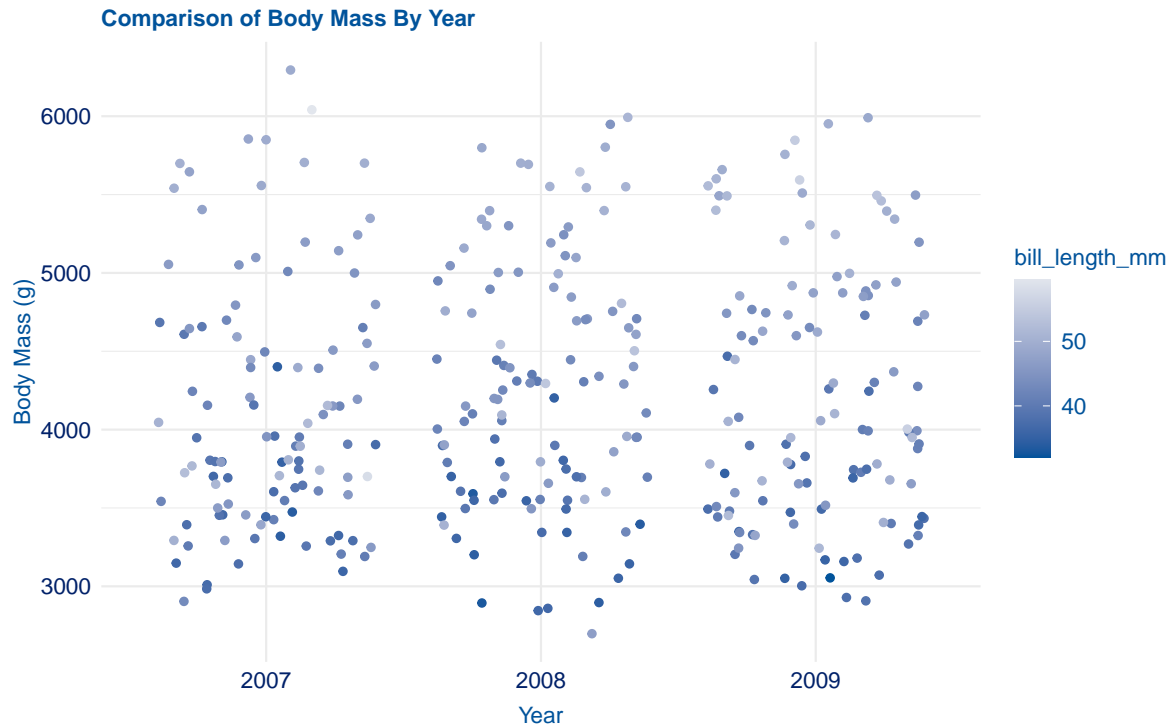
4.9 Mosaic Plot

```
mosaicplot <- ggplot(data = penguins) +
  ggmosaic::geom_mosaic(aes(x = ggmosaic::product(sex, island), fill = sex)) +
  labs(
    title = "Penguin Sex vs. Island",
    x = "Island",
    y = "Penguin Sex",
    caption = "(Color from top to bottom) \n Granite, Duke Navy Blue, Duke Royal Blue"
  )

mosaicplot +
  scale_duke_fill_discrete() +
  theme_duke()
```

Penguin Sex vs. Island

(Color from top to bottom)
Granite, Duke Navy Blue, Duke Royal Blue
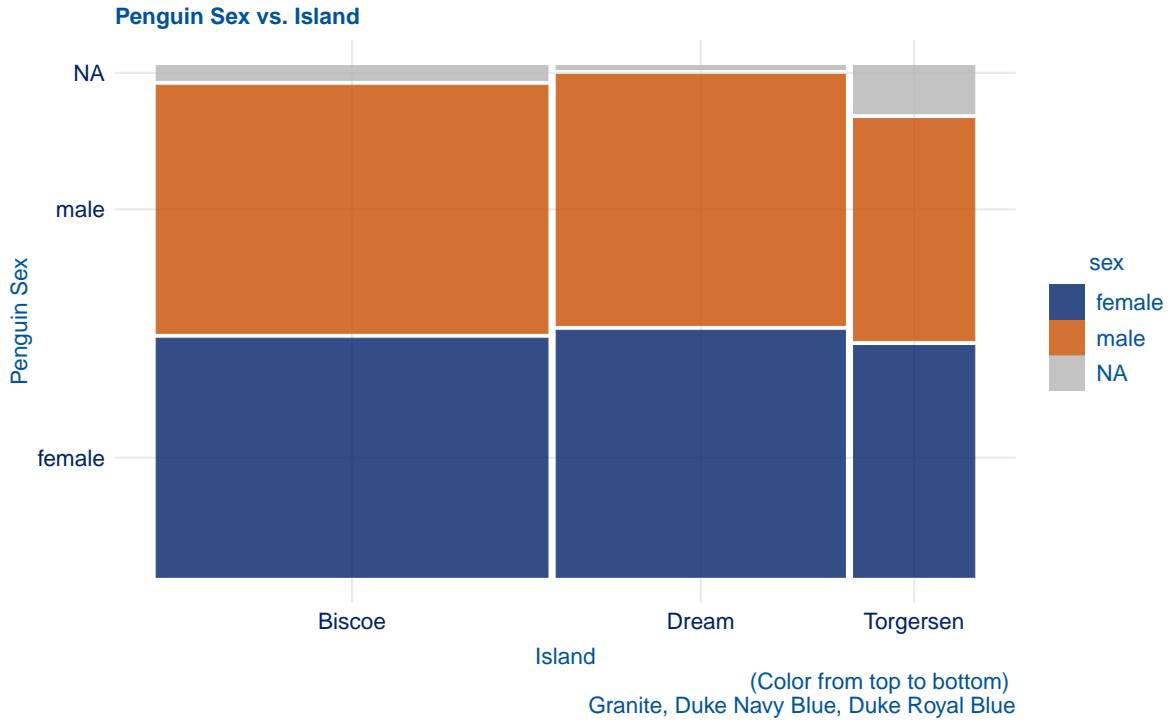
4.10 Label Plot

```
labelplot <- ggplot(
    penguins,
    aes(bill_length_mm, bill_depth_mm,
      fill = island
    )
) +
geom_label(aes(label = island)) +
labs(
  title = "Bill Length/Depth by Island",
  x = "Bill Length (mm)",
  y = "Bill Depth (mm)"
)

labelplot +
  scale_duke_fill_discrete() +
  theme_duke()
```

Bill Length/Depth by Island
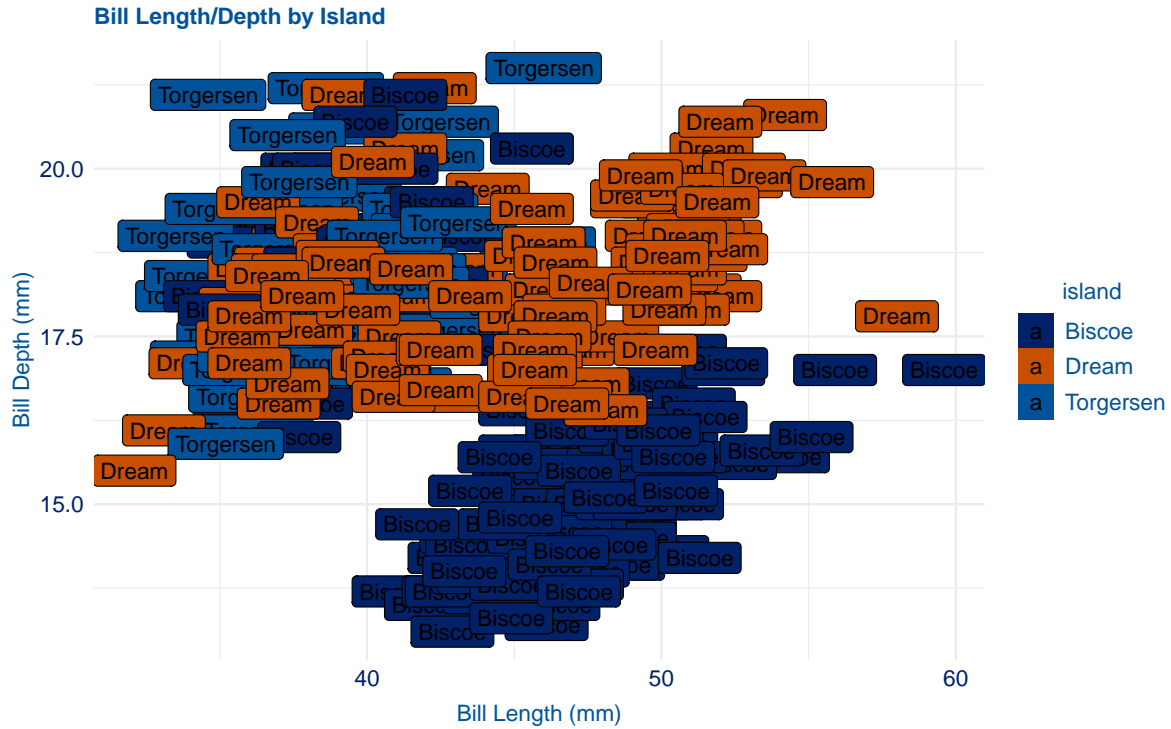
4.11 Quantile Plot

```
quantileplot <-
  ggplot(
    penguins,
    aes(bill_length_mm, bill_depth_mm, color = species)
  ) +
  geom_quantile() +
  labs(title = "Bill Length/Depth Quantiles",
              x = "Bill Length (mm)",
              y = "Bill Depth (mm)")

quantileplot +
  scale_duke_color_discrete() +
  theme_duke()
```

Bill Length/Depth Quantiles
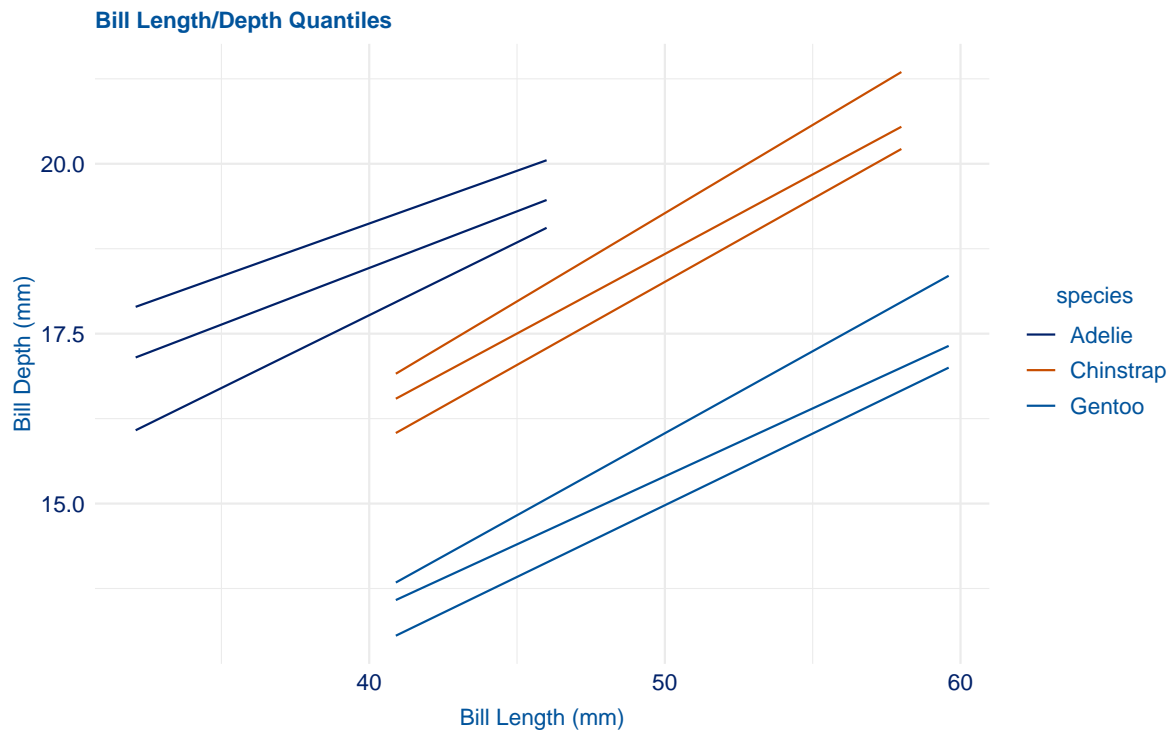
4.12 Area Plot

```r
areaplot <- ggplot(penguins, aes(body_mass_g, fill = species)) +
  geom_area(stat = "bin") +
  labs(
    title = "Area of Body Mass Index",
    x = "Body Mass (g)"
  )

areaplot +
  scale_duke_fill_discrete() +
  theme_duke()
```

Area of Body Mass Index
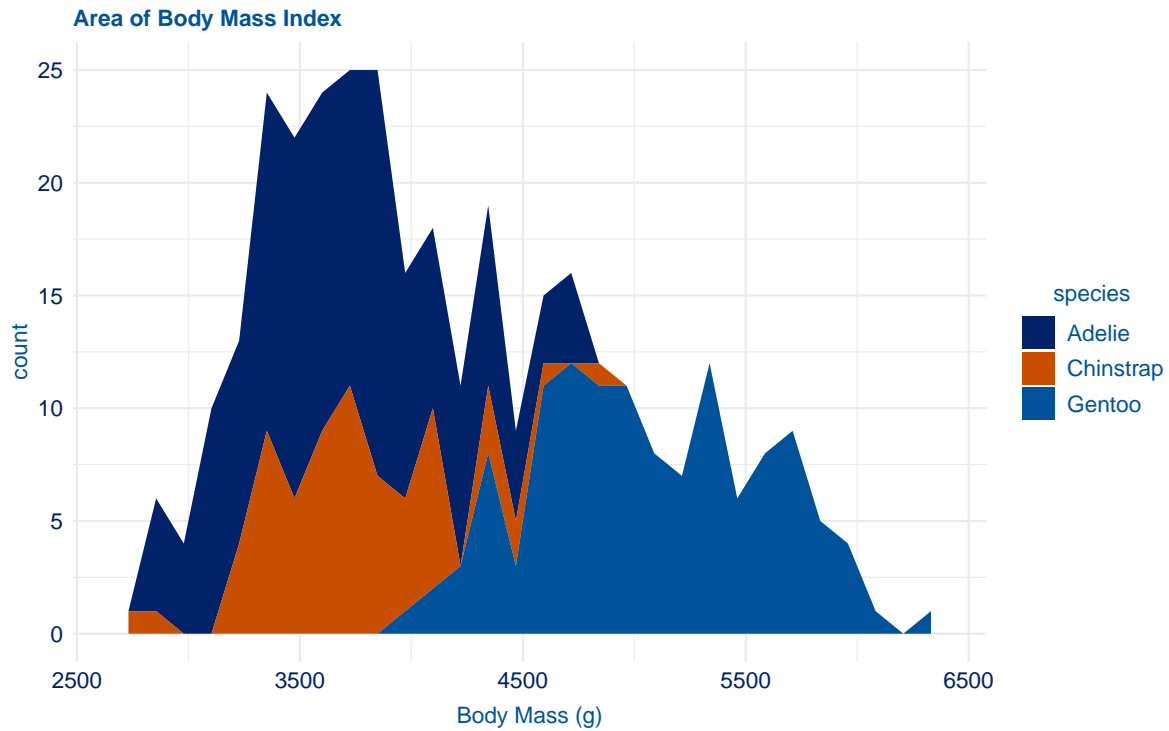
4.13 Dot Plot

```r
dotplot <- ggplot(penguins, aes(body_mass_g)) +
  geom_dotplot(aes(fill = species)) +
  labs(
    title = "Dotplot of Body Mass (g)",
    x = "Body Mass (g)"
  )

dotplot +
  scale_duke_fill_discrete() +
  theme_duke()
```

Dotplot of Body Mass (g)
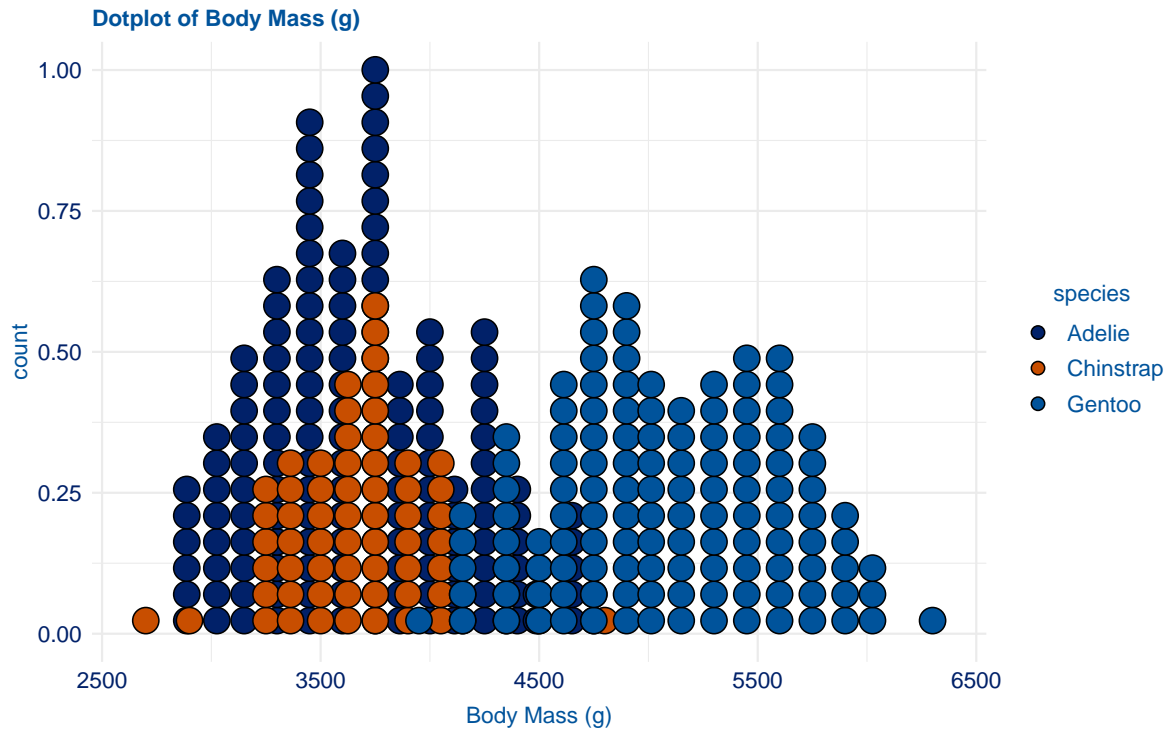
4.14 Frequency Polynomial Plot

```
freqplot <-
  ggplot(penguins, aes(body_mass_g)) +
  geom_freqpoly(aes(color = species)) +
  labs(title = "Frequency Polynomial Plot of Body Mass (g)",
              x = "Body Mass (g)")

freqplot +
  scale_duke_color_discrete() +
  theme_duke()
```

Frequency Polynomial Plot of Body Mass (g)
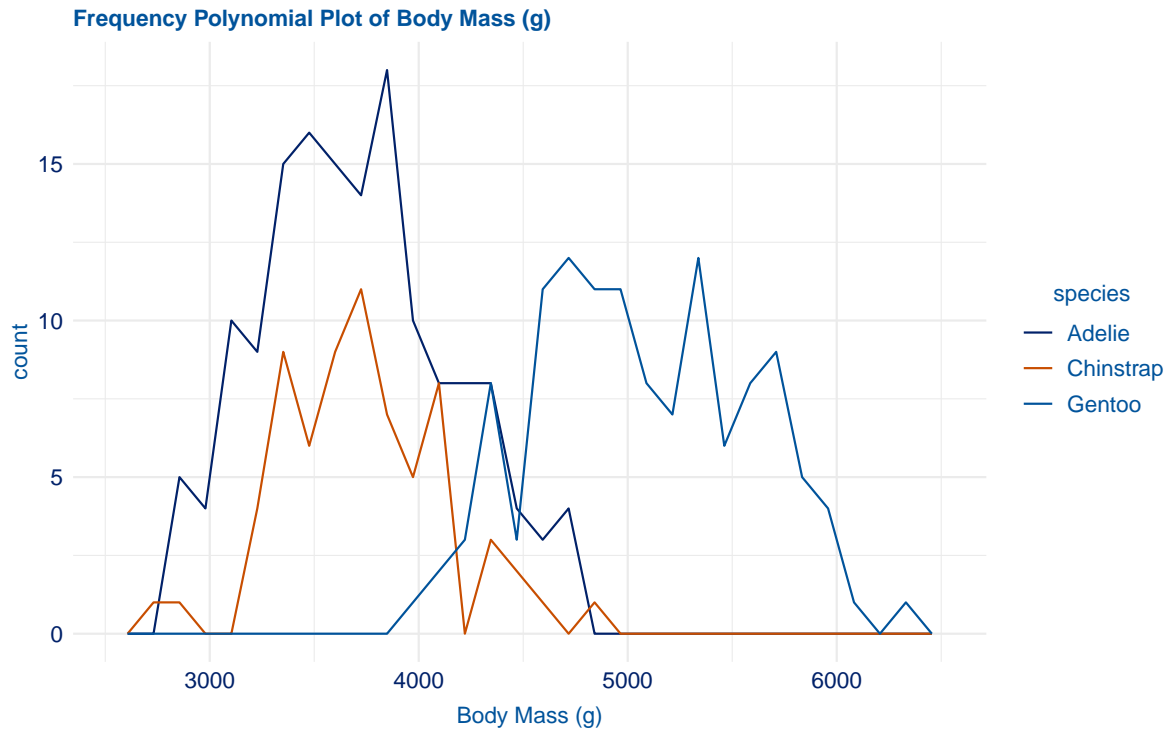
4.15 Column Plot

```
colplot <- ggplot(penguins, aes(species, body_mass_g, color = species)) +
  geom_col() +
  labs(
    title = "Body Mass (g) By Species",
    x = "Species",
    y = "Body Mass (g)"
  )

colplot +
  scale_duke_color_discrete() +
  theme_duke()
```
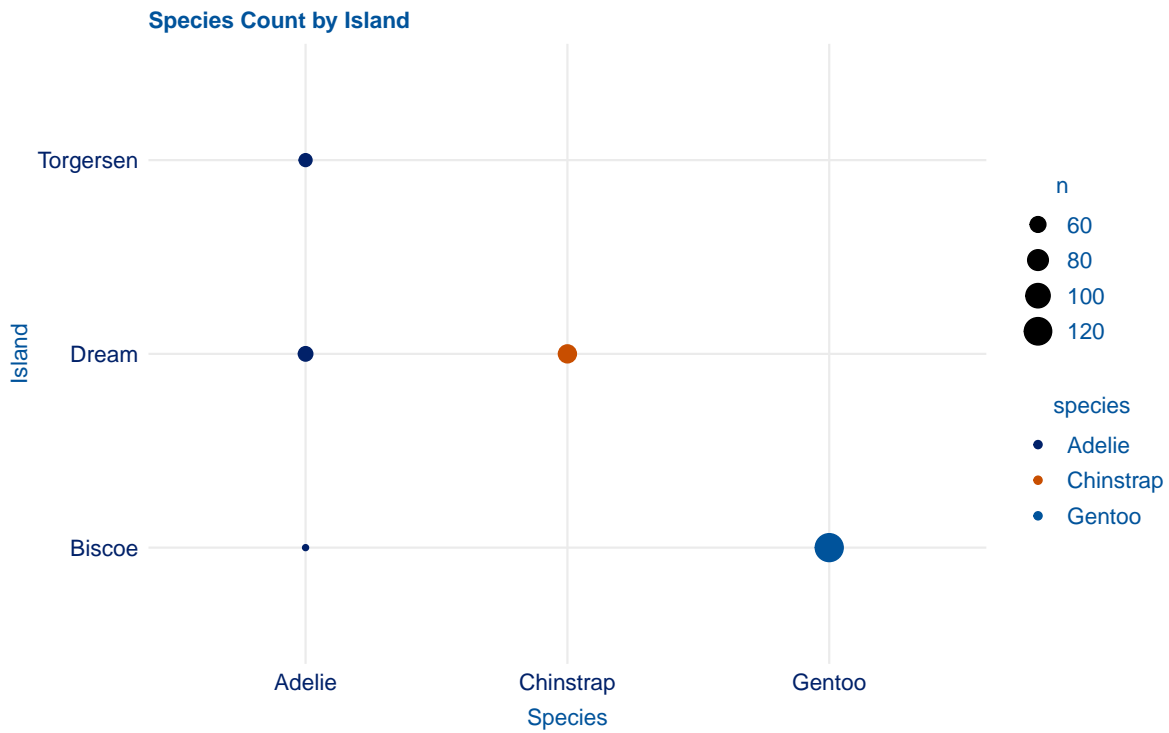
Body Mass (g) By Species
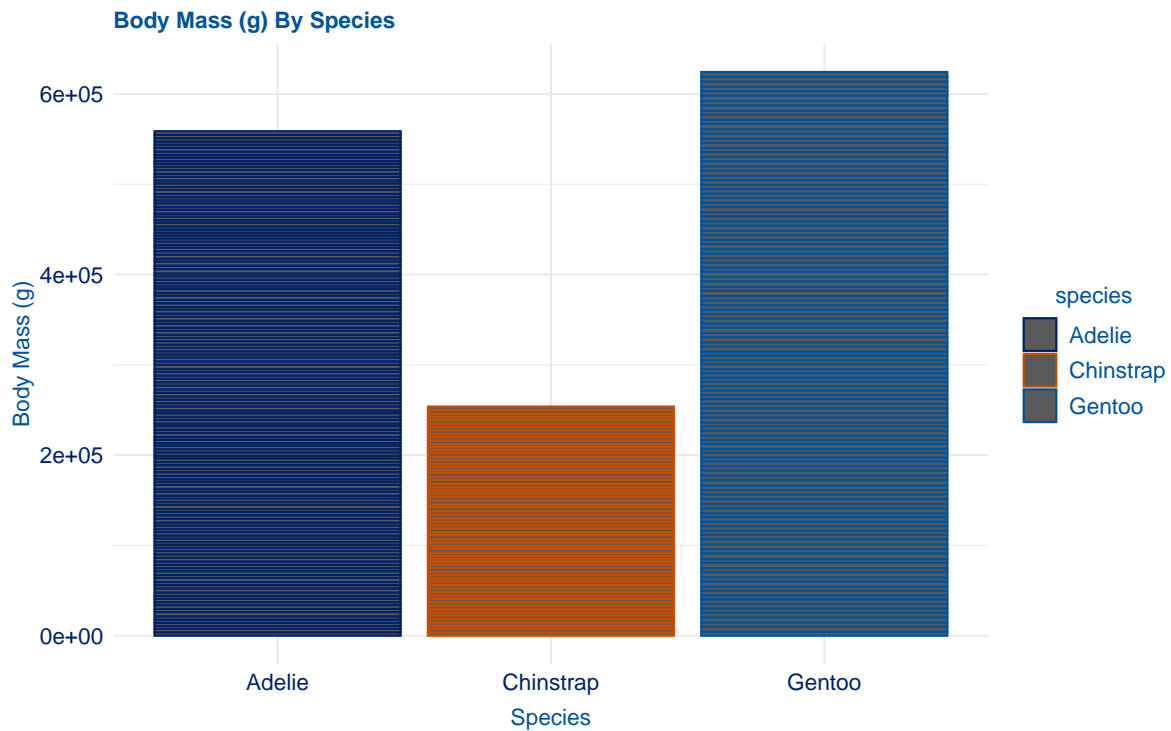
4.16 Violin Plot

```r
violinplot <- ggplot(penguins, aes(species, body_mass_g, fill = species)) +
  geom_violin(scale = "area") +
  labs(
    title = "Body Mass (g) By Species",
    x = "Species",
    y = "Body Mass (g)"
  )

violinplot +
  scale_duke_fill_discrete() +
  theme_duke()
```
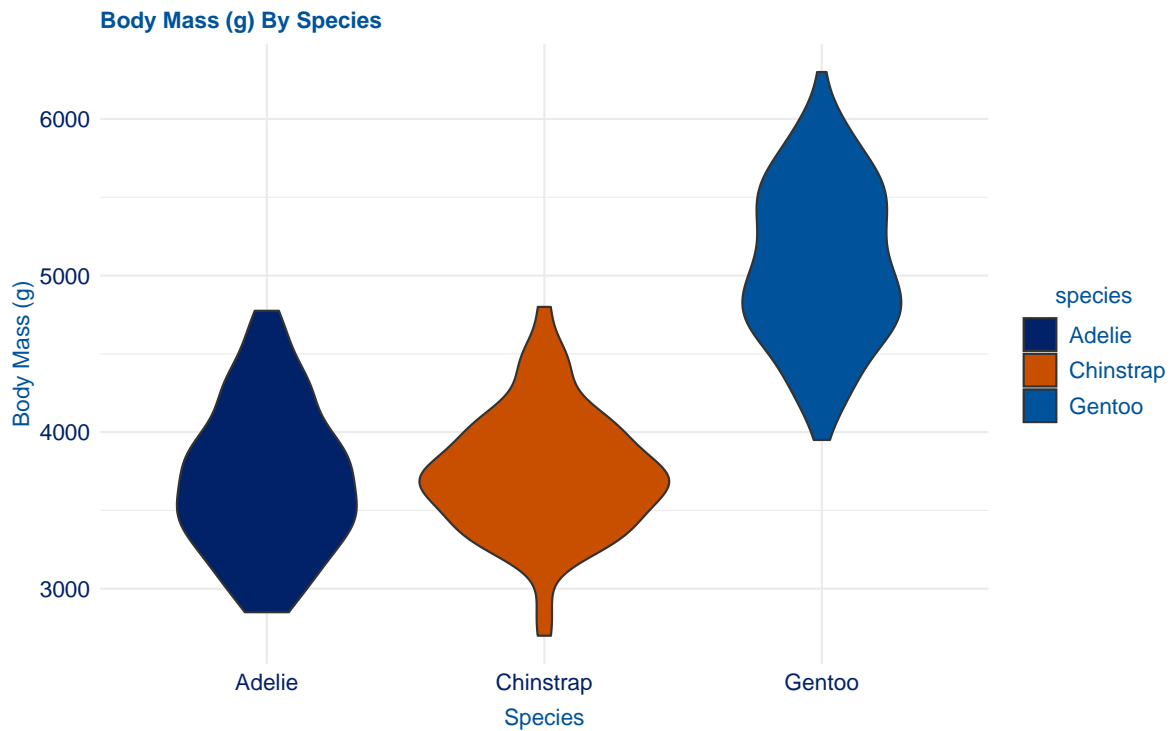
4.17 Count Plot

```
countplot <- ggplot(penguins, aes(species, island, color = species)) +
  geom_count() +
  labs(
    title = "Species Count by Island",
    x = "Species",
    y = "Island"
  )

countplot +
  scale_duke_color_discrete() +
  theme_duke()
```

Species Count by Island

4.18 Step Plot

```
stepplot <- ggplot(
  penguins,
  aes(as.factor(year), body_mass_g, color = species)
) +
  geom_step() +
  labs(
    itle = "Body Mass (g) By Year",
    x = "Year",
    y = "Body Mass (g)"
  )

stepplot +
  scale_duke_color_discrete() +
  theme_duke()
```
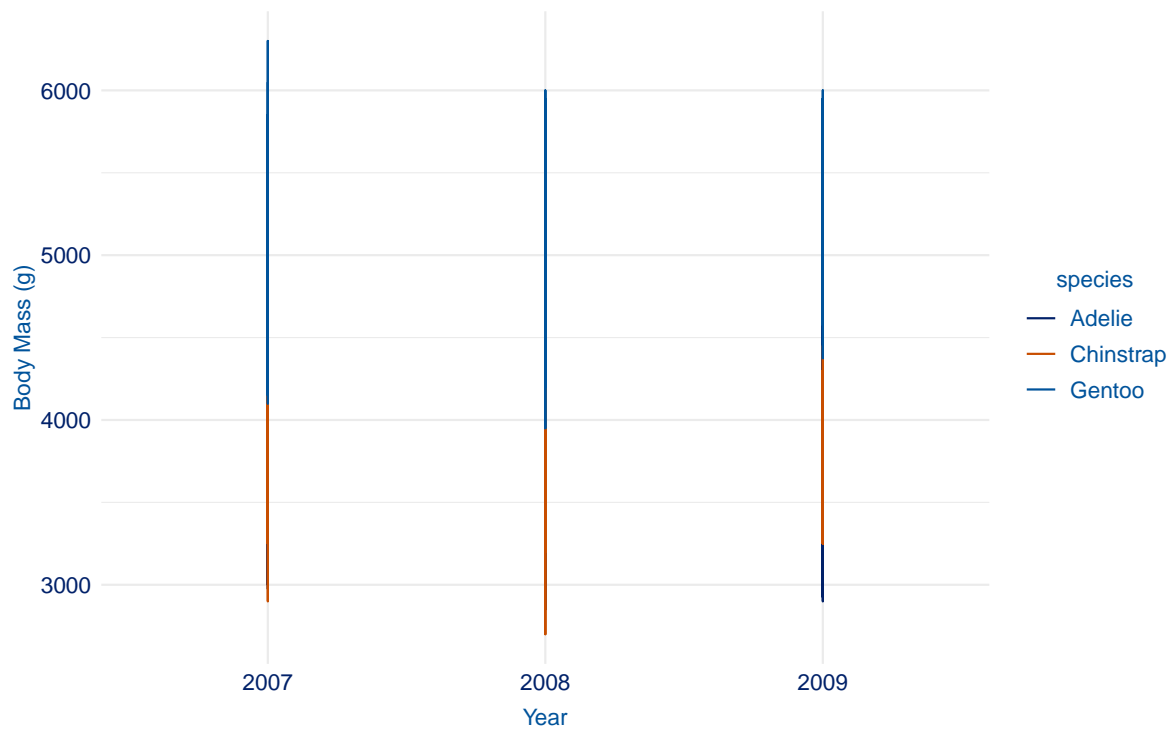
5 Case Study

5.1 Introduction

The **duke** package is intended to promote color blind accessibility in Duke official visualizations. This applied analysis aims to demonstrate how, when, and why the package may be used within the Duke community.

The data for this analysis is sourced from the Office of the University Registrar and details information about all Duke courses between 2013-2022. The data set has 10 variables and 63,433 observations, each of which is a unique class. Included in the data is information on the course department, subject, enrollment numbers, and room description. See Table 5.1 for a full data dictionary.

Table 5.1: **Registrar Data Dictionary**

Variable	Description
acad_year	Academic year of the course.
acad_org	Academic organization abbreviation of the course; proxy for department.
class_nbr	Unique class number for the course.
subject	Subject abbreviation for course.
catalog_number	Course number within academic organizations.
class_descr	Course name.
total_enrollment	Number of students enrolled in course.
enrollment_capacity	Enrollment capacity for course.
room_capacity	Capacity for students permitted in course room.
room_descr	Room where course is held.

5.2 Analysis

The Duke Office of the University Registrar facilitates the academic processes of the University by supporting students, faculty and staff in enrollment, academic records, and course scheduling - among many other tasks. It is crucial that the Office continuously explores the academic data they collect to identify the most successful policies for Duke's educational environment.

Their work is critical to the function of the University, making it all the more important that they communicate with the Duke community in an accessible way. Thus, communications, reports, and announcements from the University Registrar offer a tangible use case for the **duke** package.

The following tabs allows users without color blindness to see **duke** from their perspective and from the perspective of those who are deuteronamolous. This allows for a better understanding of how the package improves accessibility in Duke data visualizations.

5.2.1 Number of courses by department

The Registrar is interested in understanding the breadth of academic offerings available for students. As we can see in Figure 5.1 and Figure 5.2, the academic organizations (proxy for department) that have offered the largest number of undergraduate courses from 2013-2022 are the Asian and Middle Eastern Studies (AMES), Chemistry, Math, Public Policy (PPS), and Romance Studies departments. With this information, the Registrar can better make decisions about the distribution of University resources and/or understand potential department popularity.
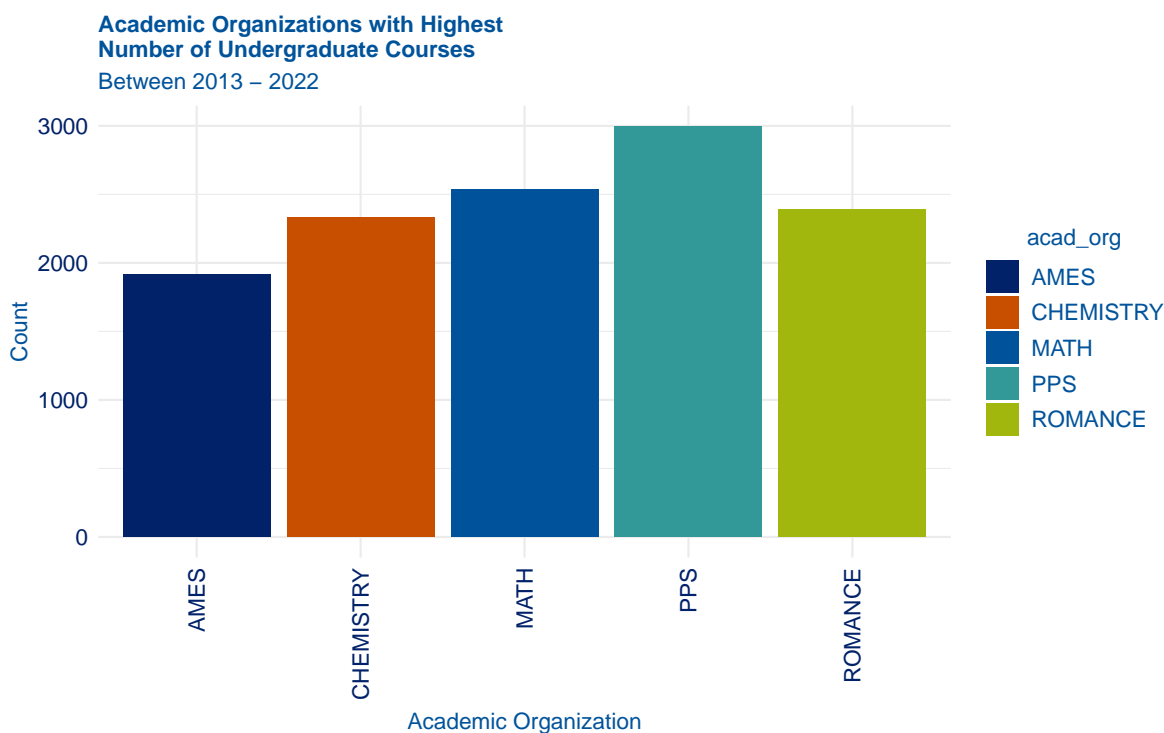

Figure 5.1: Academic Organizations with Highest Number of Undergraduate Courses - Non-Deuteronamolous Perspective"

**Academic Organizations with Highest
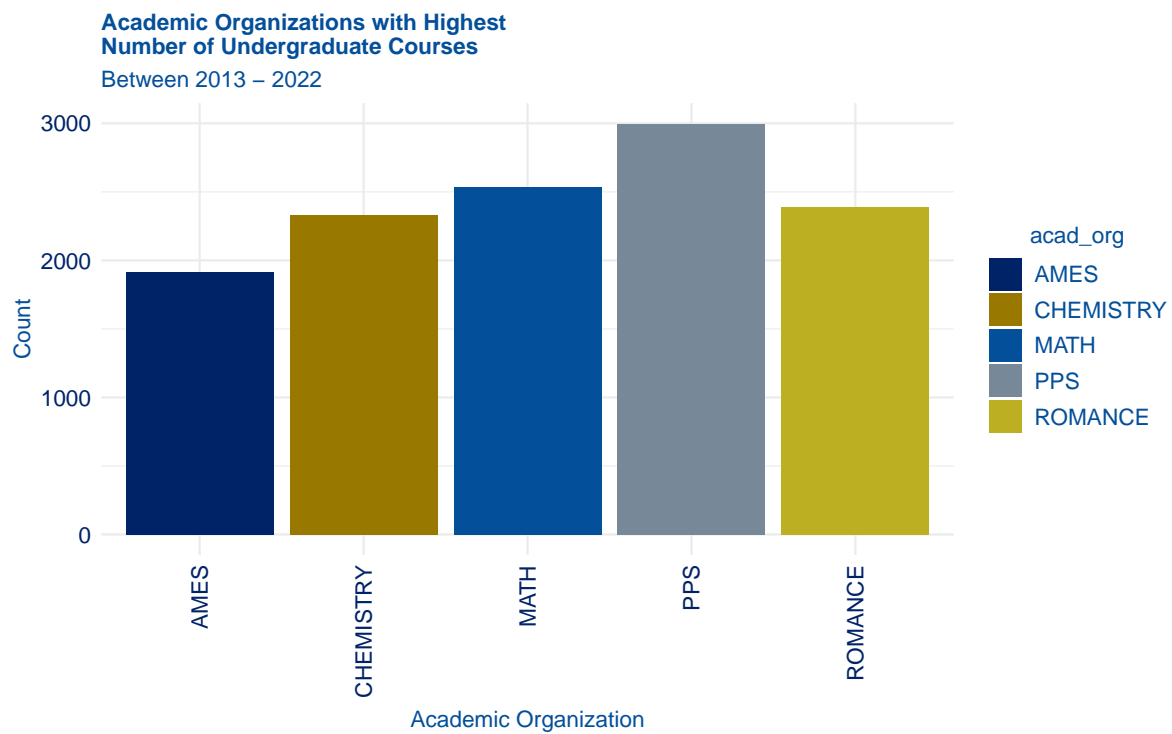Number of Undergraduate Courses**

Between 2013 – 2022

Figure 5.2: Academic Organizations with Highest Number of Undergraduate Courses - Deuteronamolous Perspective"

5.2.2 Average class size

Average class size is a staple concern of major universities, as prospective students and their families are attracted to small faculty to student ratios. In Figure 5.3 and Figure 5.4, we find that the Computer Science, Economics, Engineering, Environmental Sciences, and Earth and Ocean Sciences departments have the highest average undergraduate sciences. This is understandable given that they are majorly STEM departments, which tend to have more lecture-based courses. This visualization is a helpful resource for the Registrar to pinpoint departments that would benefit from reducing class size.
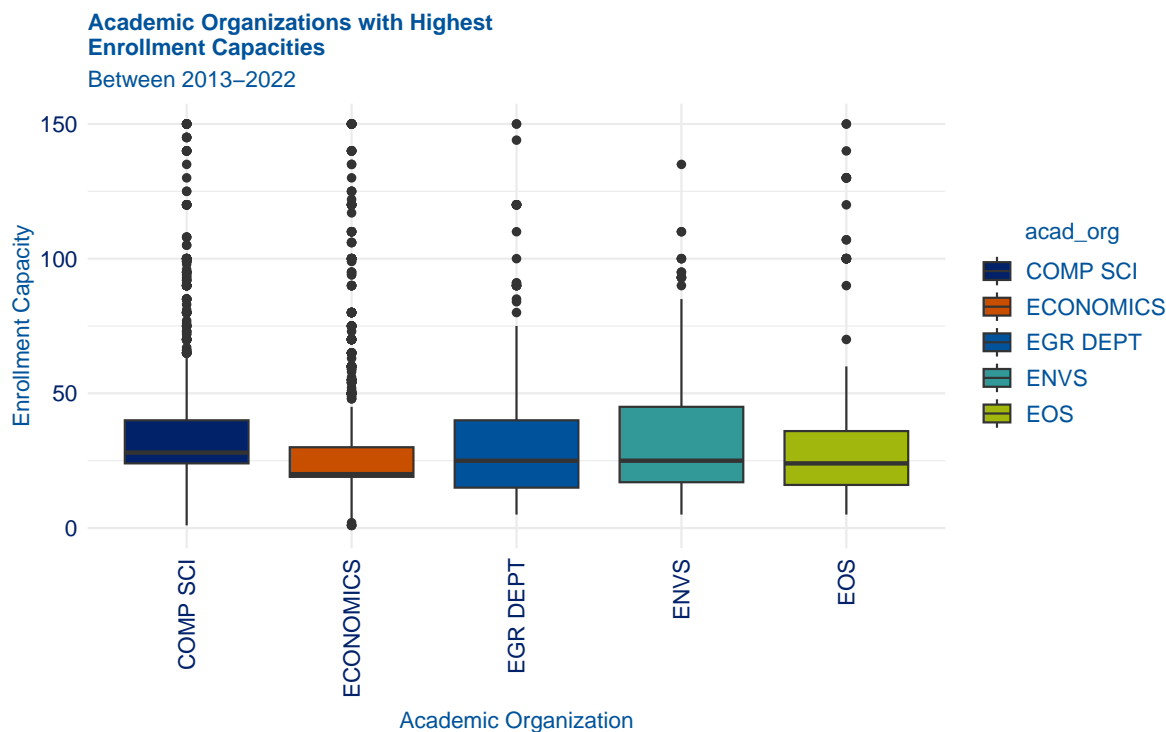

Figure 5.3: Academic Organizations with Highest Enrollment Capacities - Non-Deuteronamolous Perspective

5.2.3 Enrollment in Computer Science courses

Now that we have identified Computer Science as a department with larger class capacities, let's see if there is enough student demand to fill them. Figure 5.5 and Figure 5.6 exhibits both the distribution of student enrollment in Computer Science course and of class capacities in the department. While they share a right skew, the center of student enrollment appears less than that of course capacity. The Registrar would certainly want to investigate the data
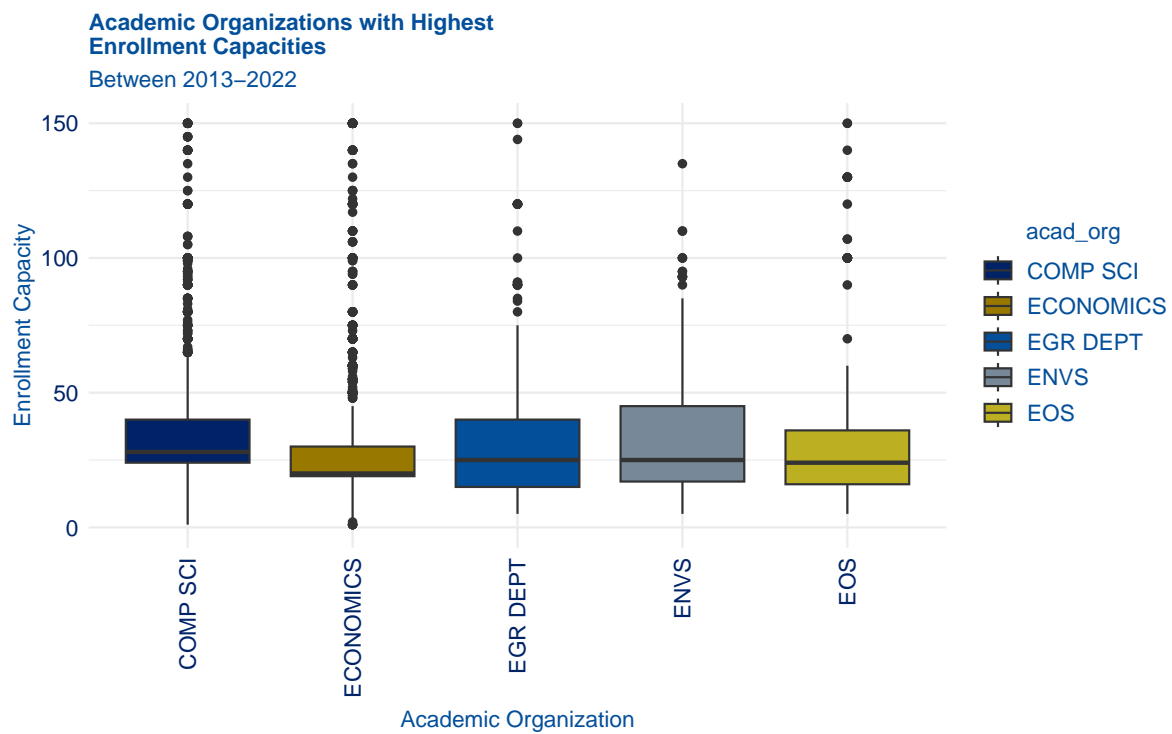
Figure 5.4: Academic Organizations with Highest Enrollment Capacities - Deuteronamolous Perspective

further, but the visualization suggests that students do not enroll at a high enough rate to fill the courses.
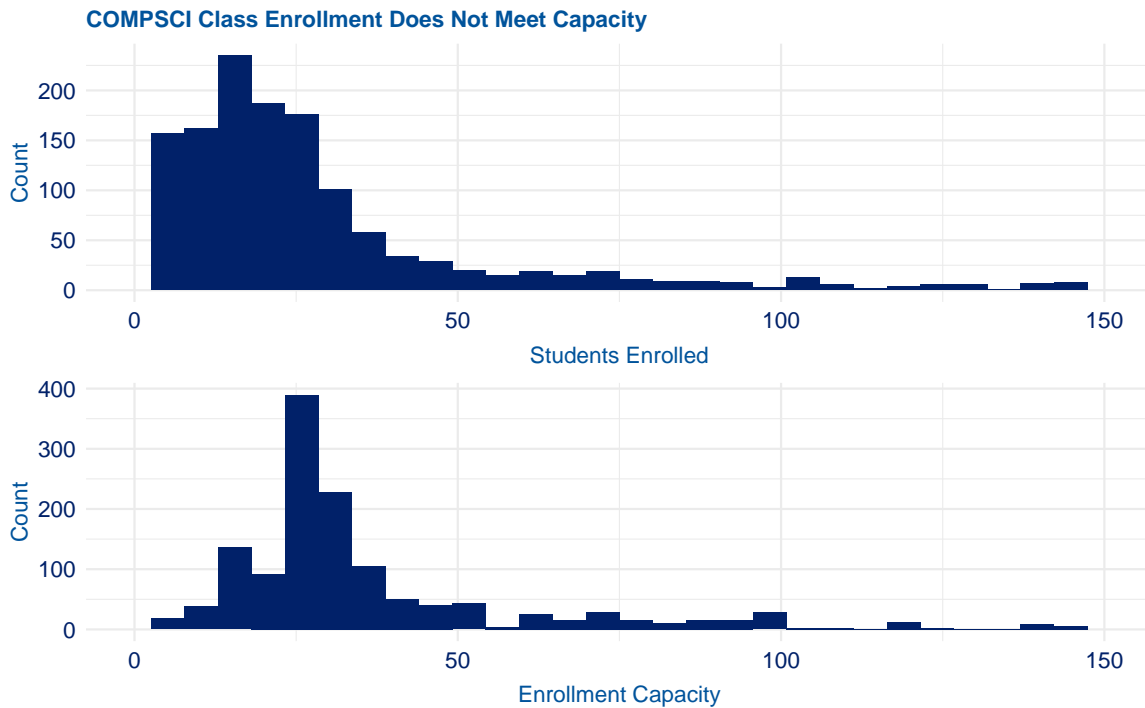

Figure 5.5: COMPSCI Class Enrollment and Capacity Distributions - Non-Deuteronamolous Perspective

5.2.4 Enrollment in introductory courses

The University often highlights their wide selection of introductory courses that allow new students to explore their interests. Figure 5.7 and Figure 5.8 explore the intro courses with the most successful enrollment rates. This is beneficial because it informs the Registrar's process of 1) adding sections to these courses and/or 2) identifying reasons for these courses' success and integrating them into others.

Conversely, the Registrar may want to identify intro courses with lower enrollment rates so that they can 1) adjust their structure/implementation and/or 2) remove them from the course catalog. Figure 5.9 and Figure 5.10 visualize the eight intro courses with the least successful enrollment rates.
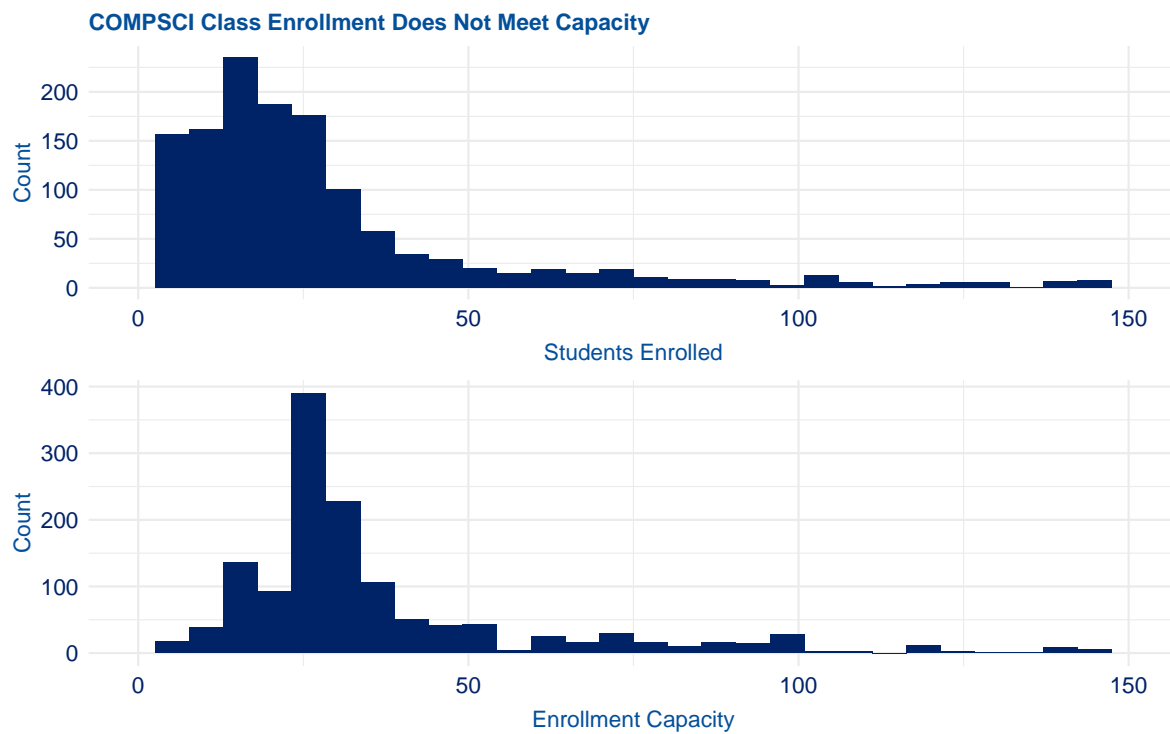
Figure 5.6: COMPSCI Class Enrollment and Capacity Distributions - Deuteronamolous Perspective

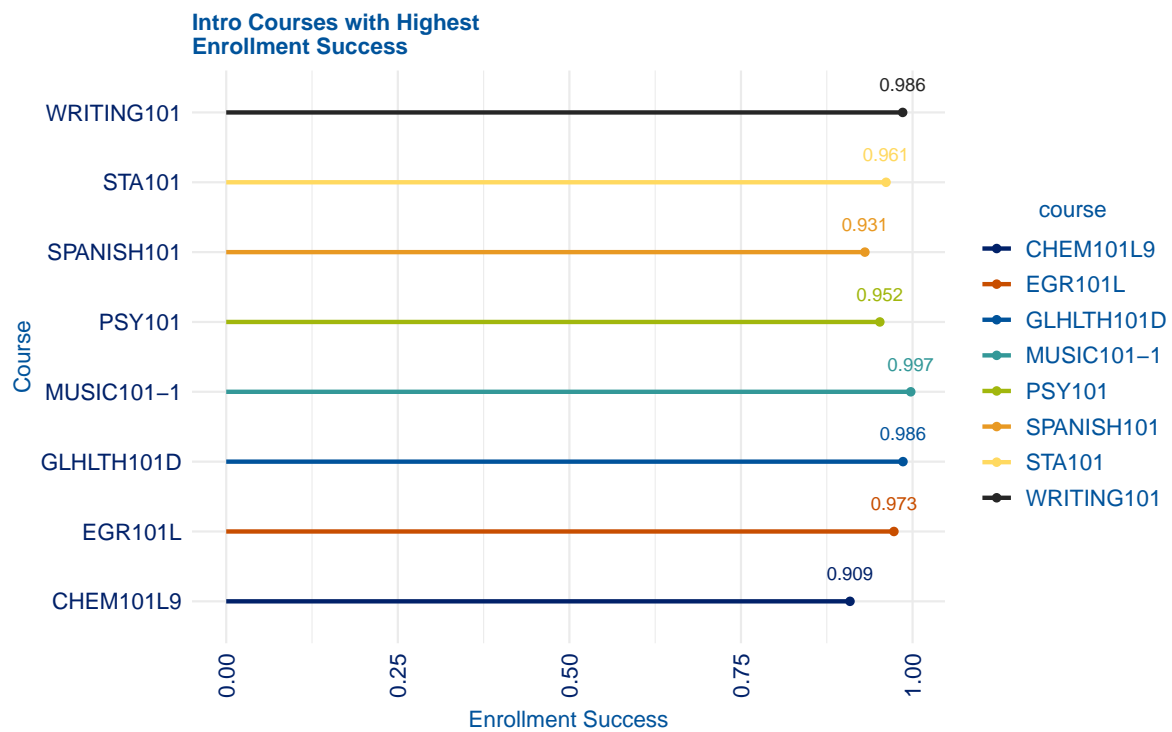**Intro Courses with Highest Enrollment Success**

Figure 5.7: Intro Courses with Highest Enrollment Success - Non-Deuteronamolous Perspective
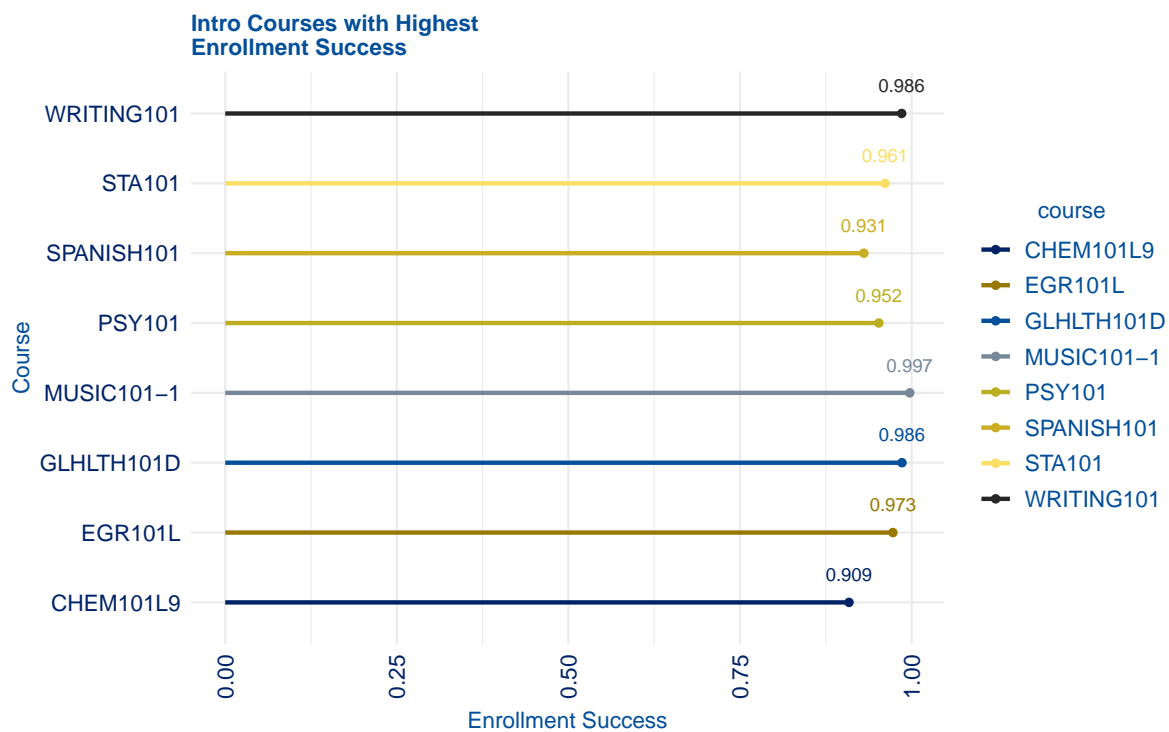
Figure 5.8: Intro Courses with Highest Enrollment Success - Deuteronamolous Perspective
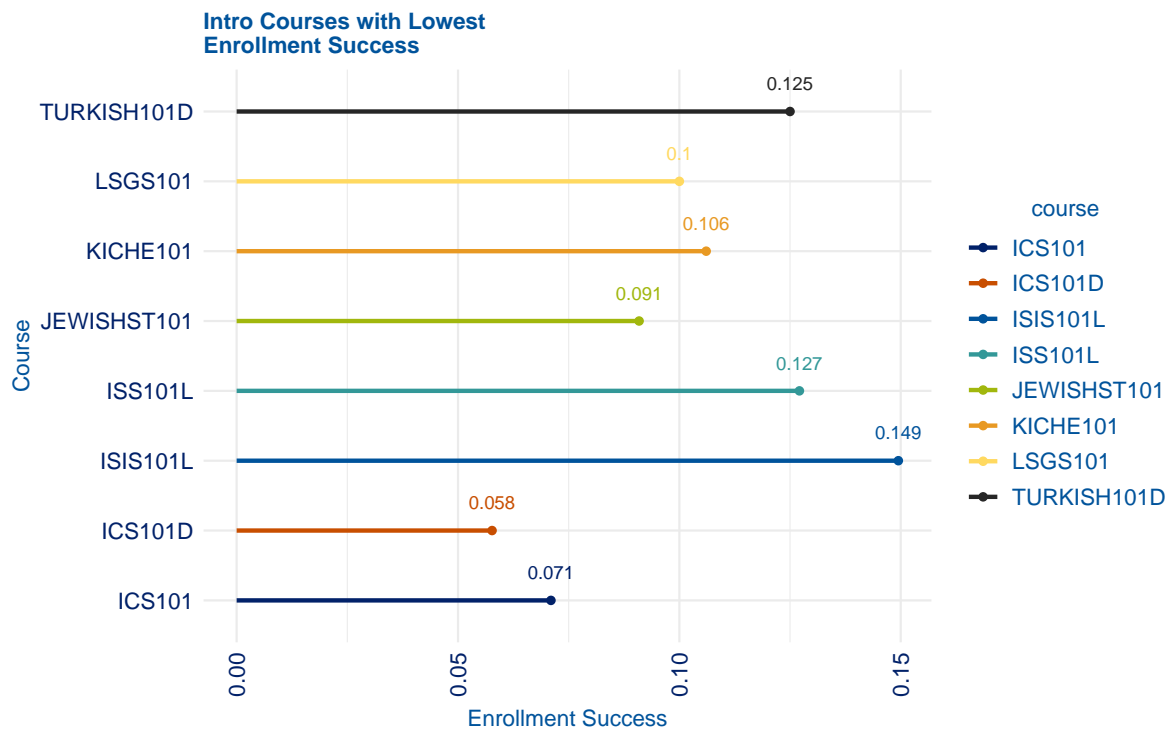
Figure 5.9: Intro Courses with Least Enrollment Success - Non-Deuteronamolous Perspective
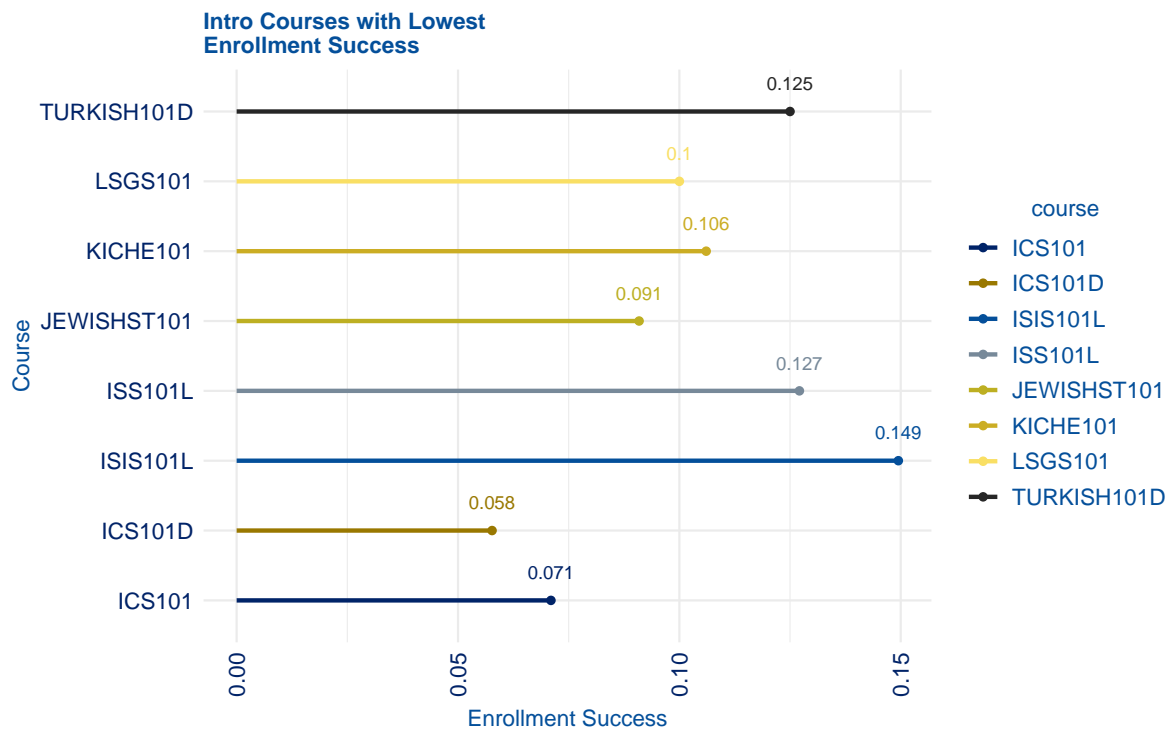
Figure 5.10: Intro Courses with Least Enrollment Success - Deuteronamolous Perspective

5.2.5 Seminar courses

The University is continuously interested in increasing the amount of face-to-face time students have with their professors. Seminar courses are a great course format for doing so. Figure 5.11 and Figure 5.12 demonstrate the five departments with the highest proportion of seminar courses in their catalog. If the Registrar wants to tout their seminar offerings, they can highlight the Documentary Studies, English, Gender, Sexuality Feminist Studies, Policy and Journalism, and Theater Studies departments.
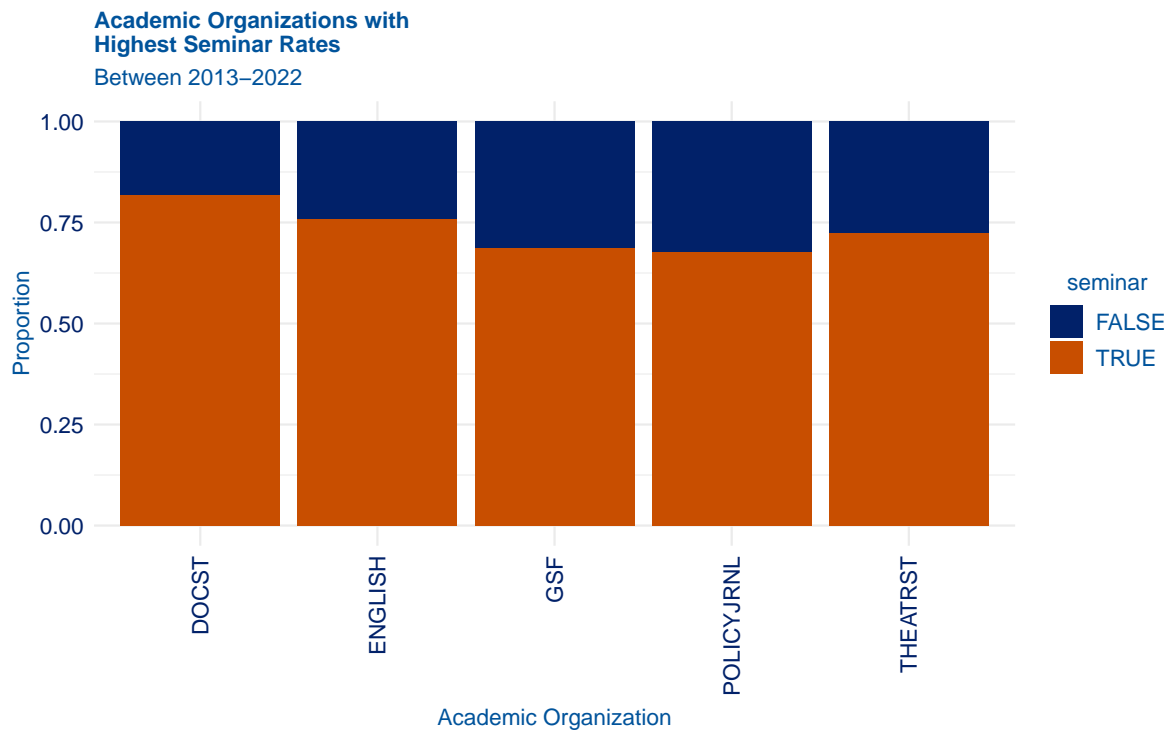

Figure 5.11: Academic Organizations with Highest Seminar Rates - Non-Deuteronamolous Perspective

On the other hand, if the University aims to improve their overall seminar offerings, it is important that they identify departments with lowest proportions. The Registrar may be interested in adding additional seminar courses to the departments in Figure 5.13 and Figure 5.14.

These data insights are critical for both the Registrar and members of the Duke community to make informed educational decisions. To ensure that those who are colorblind have the same level of access, the Registrar should use the **duke** package as part of their visualization best practices.

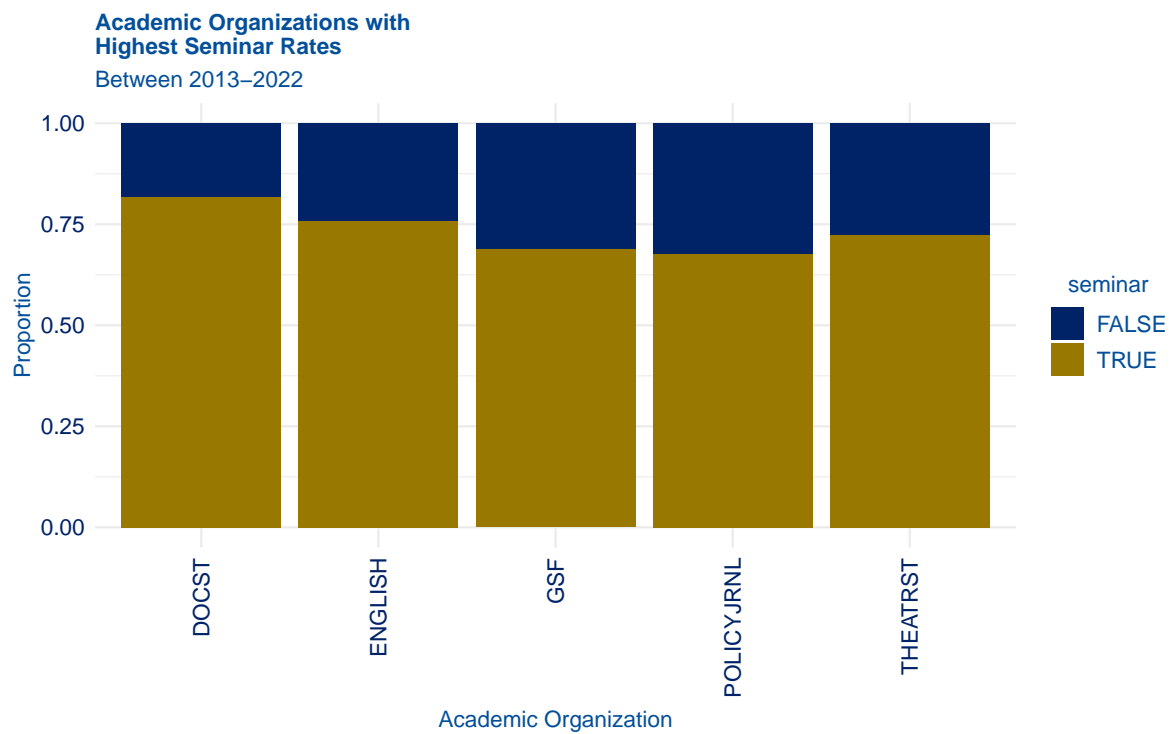Figure 5.12: Academic Organizations with Highest Seminar Rates - Deuteronamolous Perspective

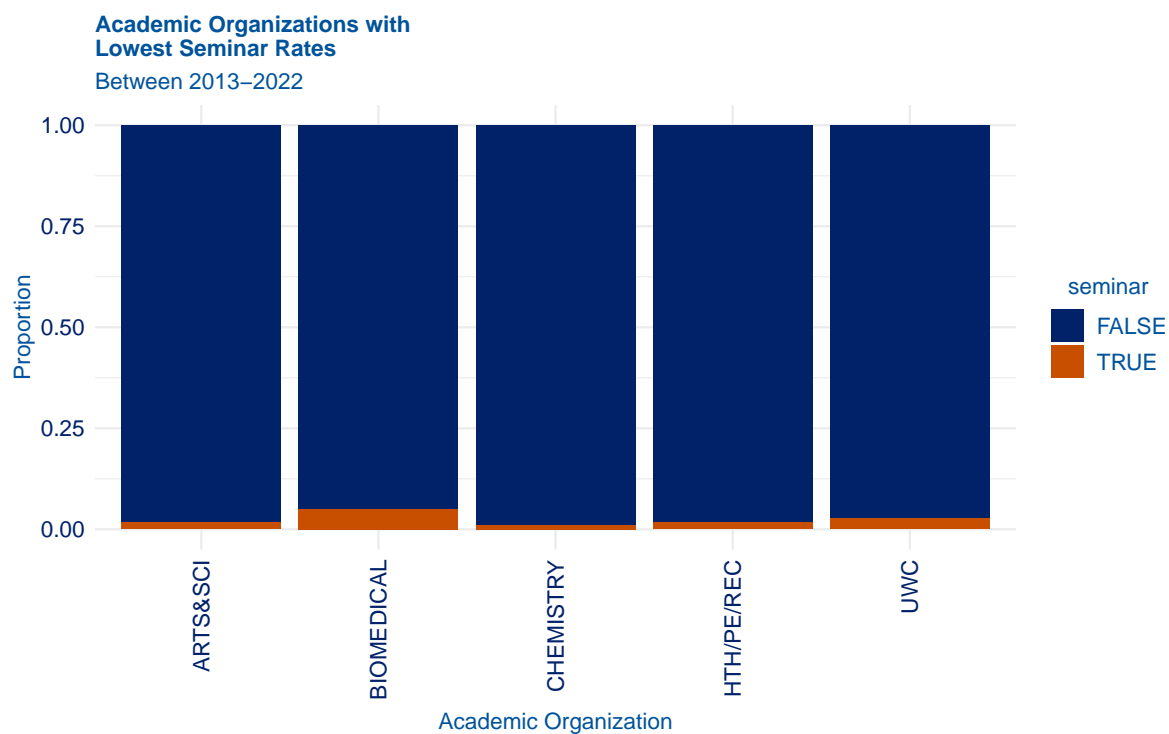**Academic Organizations with Lowest Seminar Rates**

Between 2013–2022

Figure 5.13: Academic Organizations with Highest Seminar Rates - Non-Deuteronamolous Per-spective
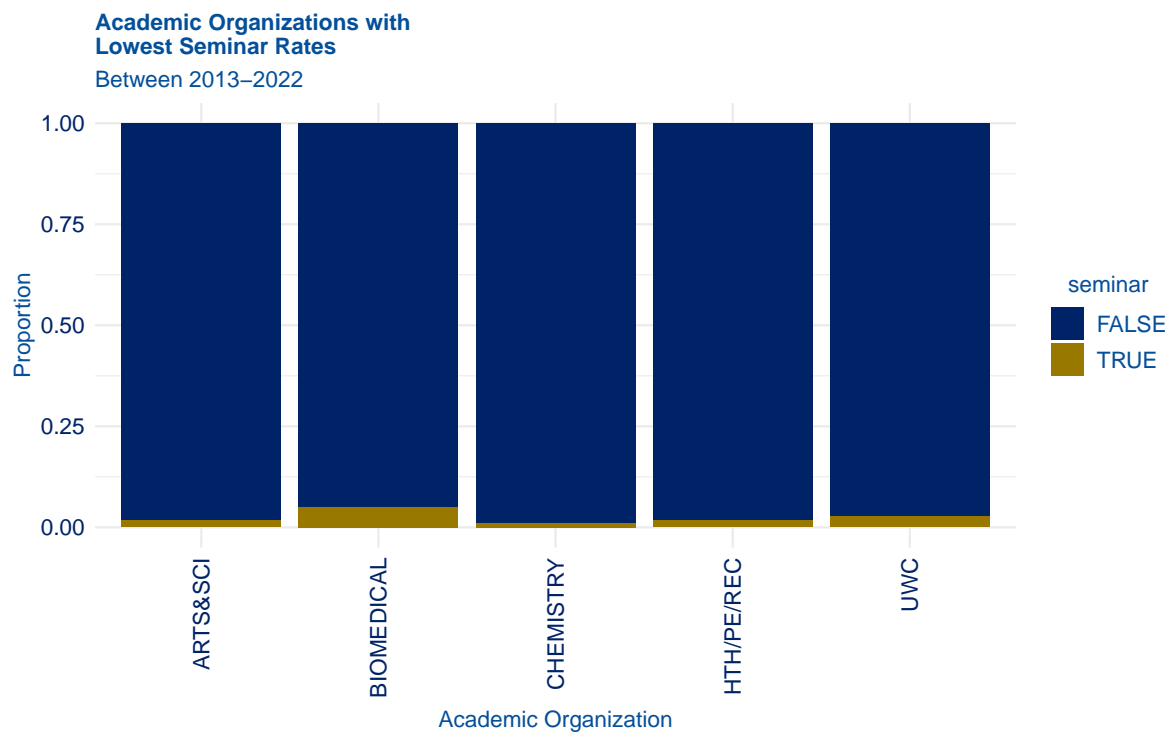
Academic Organizations with Lowest Seminar Rates

Between 2013–2022

Figure 5.14: Academic Organizations with Highest Seminar Rates - Deuteronamolous Perspective

5.3 Acknowledgement

6 Discussion

The **duke** package allows users to transform their visualizations so that they are simultaneously colorblind friendly and representative of the Duke brand. To achieve this goal, the package incorporates a minimalist thematic design along with a high contrast color scale (both discrete and continuous) for ggplot geometric objects. As explained in the methods section, these functionalities are necessary and efficacious, but only a part of creators' toolbox for advancing accessibility in data visualization. There are several other best practices in data visualization that can make all the difference for those with color vision deficiency and low vision and should be highlighted.

6.1 Colorblind Accessibility Best Practices

Keep It Simple

The **duke** package selects a simple thematic design as to not distract from the key takeaways of the data visualization. It is in this vein that you should constantly consider how you can more succinctly convey the insights from your data. While the **duke** discrete palette allows for up to eight distinct colors, you should always strive to minimize the number of dimensions in your plots. Reducing the number of categories present in your visualizations is not only a best practice for colorblind accessibility. It also allows all users to more effectively differentiate between distinctions in the data and quickly comprehend the primary takeaways.

To reduce plot complexity, ask questions like: What do I need to highlight in the data? What are the primary insights readers should understand? This thought process will reveal what characteristics and/or categories must stand out and allow you to direct your attention on making them clear - especially for colorblind readers. Inversely, you can identify data groups that are insignificant to your goal and combine them into a single category, in turn, eliminating the need for excess colors. By keeping it simple in data visualization, you can be more confident that your audience will both comprehend and appreciate the story of your data visualization.

De-Emphasize the Need for Color

The **duke** package allows users to integrate a more colorblind friendly palette into their visualizations; however, this is not the end-all solution. There are several other plot features present in **ggplot** that alleviate the need for the color palette to be perfectly distinguishable - or in some cases - be used at all. Essentially, if you "get it right in black and white", you no longer

have to worry about access issues for colorblind readers (Muth 2020). This can be done by adding, "icons, directional arrows, labels, annotations, or other indicators" that differentiate between different categories (Shaffer 2020).

Here are a few examples of how these supplementary features can improve the accessibility of visualizations using the **duke** package.

Scatterplots can quickly become difficult to read when there a several distinct categories in the data. To improve your plot's readability, consider adding shapes to your data points.
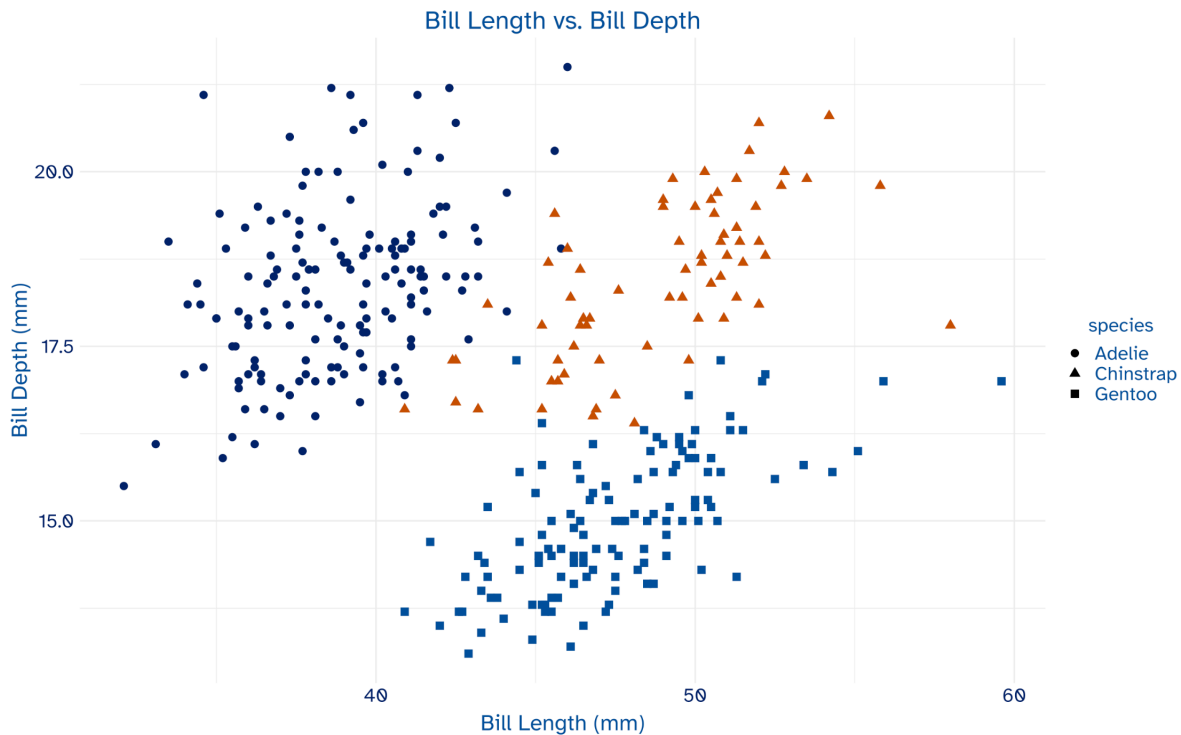

Figure 6.1: Scatter plot with shape distinctions

For geometric objects with large fill spaces (ie. bar plot, pie chart), the use of the colorblind friendly palettes can be effective on its own. Nevertheless, adding fill textures to your visualization is an effective way to increase plot clarity. To do so, I used the **ggpattern** package.

Finally, the efficacy of color in line plots deteriorates as the number of categories increases and/or the lines are intersecting. By varying the patterns and widths of the lines in your plot, your visualization can more effectively convey trend differences.

The use of these complimentary features - among many others - are easy and effective. By integrating them alongside the **duke** package, you can promote accessibility for not only
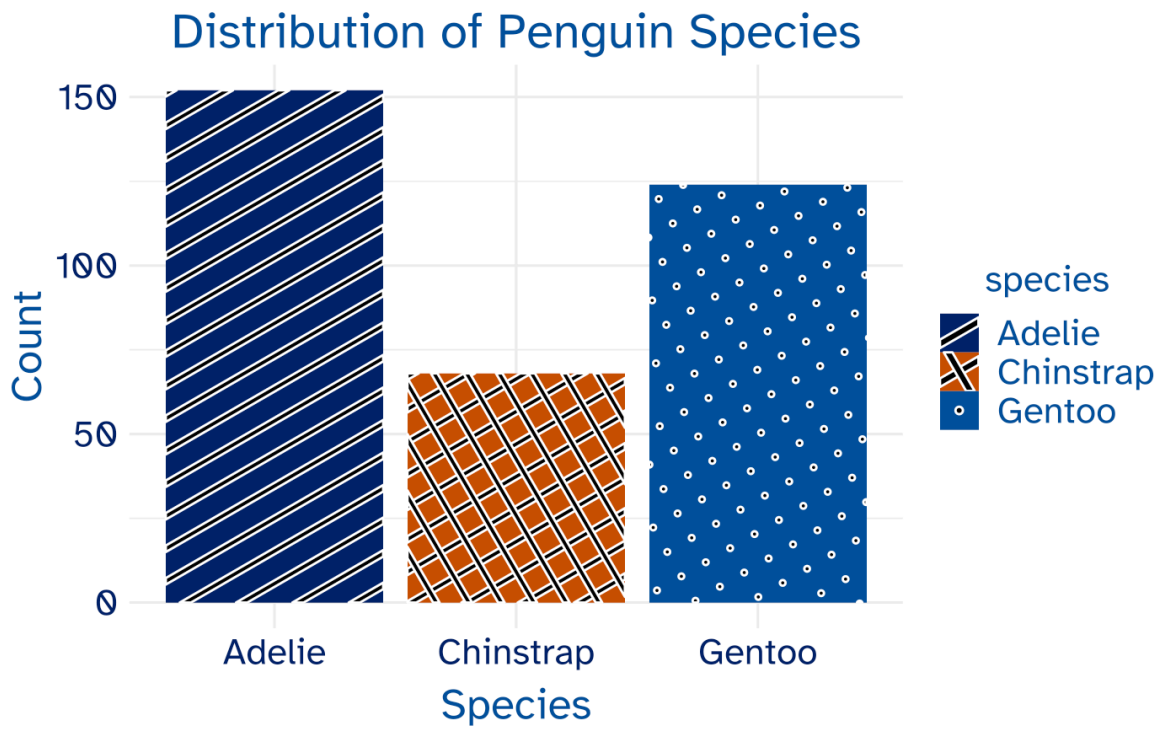
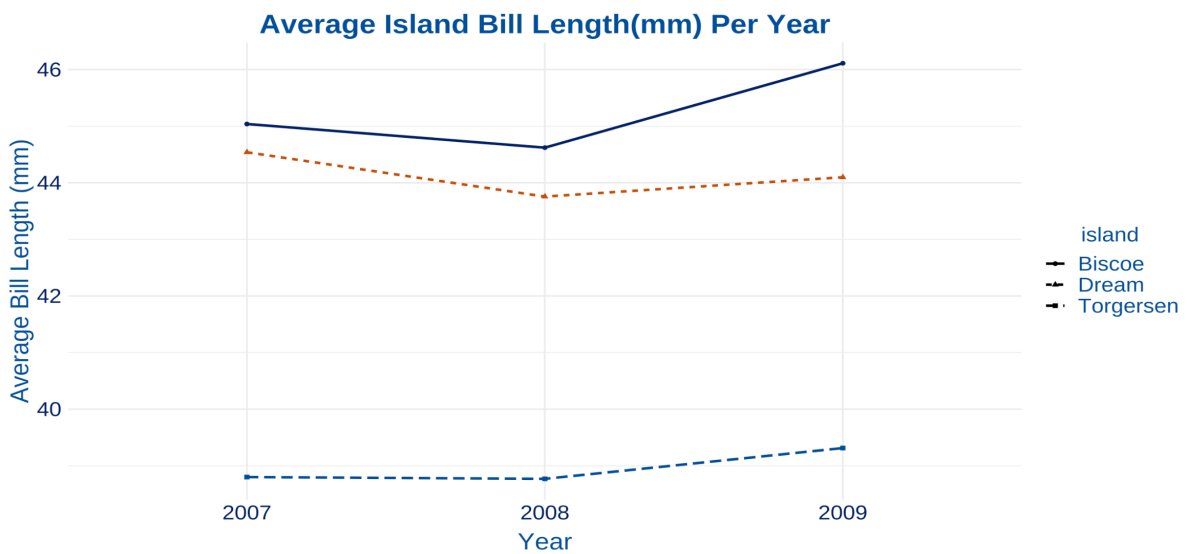Figure 6.2: Bar plot with fill texture

Figure 6.3: Line plot with line pattern and point shapes.

colorblind individuals, but older readers, audiences with lower technical exposure, and people reading in low light.

Adjust When Necessary

The colorblind community is not a monolith, therefore, it is understandable that you may need to adjust your visualizations given the context. The development of the **duke** package was intended to promote accessibility for the largest possible audience, which is why red-green colorblindness was the primary focus. Some features of **duke** may not be fitting for you, so adapt them when necessary. For example, text sizing is scaled relative to the plot dimensions in `theme_duke()`, but it is reasonable that even larger text may be required for audience readability. In this case, engage with the specifications of the `theme()` function to make sure the text is sufficiently large.

The **duke** palette offers the most accessible colors within Duke's official branding guide, but modifying these colors may be needed to perfectly meet WCAG guidelines. If your visualization does not fall within the purview of the Duke official brand, using the the rule light vs. dark can be an effective strategy. Even when colors are similar, readers can typically distinguish between lighter and darker shades. This also to applies the monochromatic palettes which only use different shades of the same color. The Duke brand does not permit color adjustments and therefore the **duke** package cannot manually incorporate the light vs. dark strategy. However, if colorblind accessibility is your primary emphasis over representing Duke, this can be a successful tool for creating intelligible visualizations.

6.2 Summary

duke empowers users to develop Duke branded and colorblind friendly visualizations and should be used in University communications. By implementing a minimalist thematic design and a high contrast color palette in both continuous and discrete geometric objects, the package is an efficacious tool for data visualization in the Duke community. Nevertheless, it is not the only tool. To further enhance the accessibility of visualizations, users can engage with several complimentary strategies:

Keep It Simple: Identify the most important features of your data and remove those that are unnecessary. Reducing the complexity of the plot will more succinctly convey your data insights.

De-Emphasize Color: Integrate additional indicators into your visualizations (labels, shapes, patterns) to clearly direct the attention of readers. When used effectively, these features can eliminate the need for color differentiation at all.

Adjust When Necessary: Be creative and craft a unique visualization fitting to your needs. By manually adjusting text sizing and color shades, you can ensure the visualization is appropriate for your audience.

The use of the **duke** package, in tandem with these best practices, will foster a more accessible and inclusive approach to data visualization in the Duke community.

References

"About Colour Blindness." n.d.a. https://www.colourblindawareness.org/colour-blindness/.

———. n.d.b. https://www.colourblindawareness.org/colour-blindness/.

"Atkinson Hyperlegible." n.d. https://fonts.google.com/specimen/Atkinson+Hyperlegible.

"Colors." n.d.a. https://brand.duke.edu/colors/.

———. n.d.b. https://brand.duke.edu/colors/.

Duke Research & Innovation. n.d. "Annual Report 21–22." https://report.research.duke.edu/.

"Duke University Chapel." n.d. https://dribbble.com/shots/3937382-Duke-University-Chapel.

"Hatteras Gray." n.d. https://www.ppgpaints.com/color/color-families/neutrals/hatteras-gray.

Horst, Allison, Alison Hill, and Kristen Gorman. n.d. "Palmerpenguins r Data Package • Palmerpenguins." https://allisonhorst.github.io/palmerpenguins/.

"How to Calculate Colour Contrast." 2019. https://www.accessibility-developer-guide.com/knowledge/colours-and-contrast/how-to-calculate.

Initiative, W3C Web Accessibility. n.d. "WCAG 2 Overview." https://www.w3.org/WAI/standards-guidelines/wcag/.

"Introduction to Understanding WCAG 2.0 | Understanding WCAG 2.0." n.d. https://www.w3.org/TR/UNDERSTANDING-WCAG20/intro.html#introduction-fourprincs-head.

Iyer, Luc. 2020. "Euclidian Distance in RGB Color / Luc Iyer | Observable." https://observablehq.com/@luciyer/euclidian-distance-in-rgb-color-space.

Madsen, Rune. n.d.a. "Perceptually Uniform Color Spaces - Programming Design Systems." https://programmingdesignsystems.com/color/perceptually-uniform-color-spaces/.

———. n.d.b. "Perceptually Uniform Color Spaces - Programming Design Systems." https://programmingdesignsystems.com/color/perceptually-uniform-color-spaces/.

Muth, Lisa Charlotte. 2020. "What to Consider When Visualizing Data for Colorblind Readers - Datawrapper Blog." https://blog.datawrapper.de/colorblindness-part2/.

Rouse, Margaret. 2021. "Color Hex Code." https://www.techopedia.com/definition/29788/color-hex-code.

Shaffer, Jeffrey. 2020. "5 Tips on Designing Colorblind-Friendly Visualizations." https://www.tableau.com/blog/examining-data-viz-rules-dont-use-red-green-together.

UNC School of Medicine. n.d. "Using Accessible Web Colors." https://uncch.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=57624079-8af2-481d-be42-ace100e9b5ef.

Vissichelli, Lisa. 2022. "Blooming Bar Charts: The Growth of the Data Visualization In-

dustry.” https://uxdesign.cc/blooming-bar-charts-the-growth-of-the-data-visualization-industry-5095c59a17fb.

“Web Content Accessibility Guidelines (WCAG) 2.0.” n.d.a. https://www.w3.org/TR/WCAG20/#relativeluminancedef.

———. n.d.b. https://www.w3.org/TR/WCAG20/#relativeluminancedef.

Wilke, Claus. n.d. “Fundamentals of Data Visualization.” https://clauswilke.com/dataviz/color-pitfalls.html.

Wilke, Claus O. n.d. *Fundamentals of Data Visualization.* https://clauswilke.com/dataviz/color-pitfalls.html.

Woods, Rob. 2021a. “Types of Color Blindness.” https://www.allaboutvision.com/conditions/color-blindness/types-of-color-blindness/.

———. 2021b. “Types of Color Blindness.” https://www.allaboutvision.com/conditions/color-blindness/types-of-color-blindness/.