TECHNISCHE UNIVERSITÄT BERLIN

Graphics for Communication

Data Science WiSe 24/25 | Lecture 07 | Nov 25, 2024

Today's Goal: learn to communicate your findings and make your graphics more accessible.

To follow along in today's class, please download **Lecture7.Rmd** from ISIS & open file in RStudio

Agenda

1. Mid-Semester Assignment
2. ggplot review
3. EDA
4. Communication
5. Accessibility

Mid-Semester Assignment

Objective:

- To use your R data wrangling and visualization skills to analyze MATSim output data.
- Present your workflow and outputs in a short presentation (16.12)
- Data to use:
  - Option 1) MATSim Berlin - Base Case: [link]
  - **Option 2) MATSim Berlin - Carfree Scenario [link]**
  - Option 3) Outputs from your own MATSim Scenario (from work, or a previous course)
  - Option 4) If you're interested in the output data from a particular MATSim city/scenario, feel free to ask us.

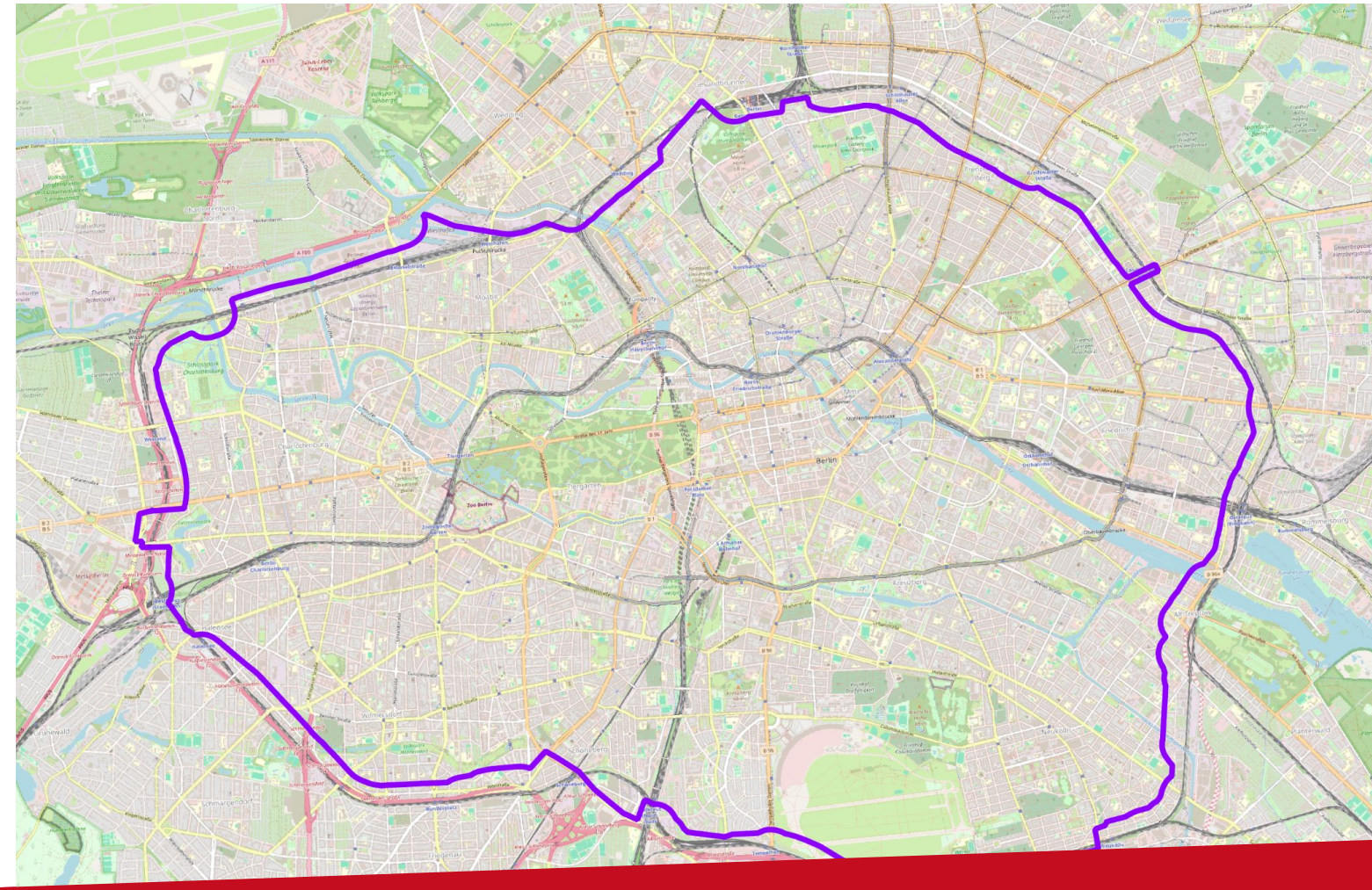
Deadlines:

- **SUNDAY - November 23rd, 2024**: Deadline to find a partner → There'll be a form on ISIS
- **FRIDAY - December 13, 2024**: Submission of slides and code
- **MONDAY - December 16, 2024**: Presentations (let us know by 23.11 if you can't make it).

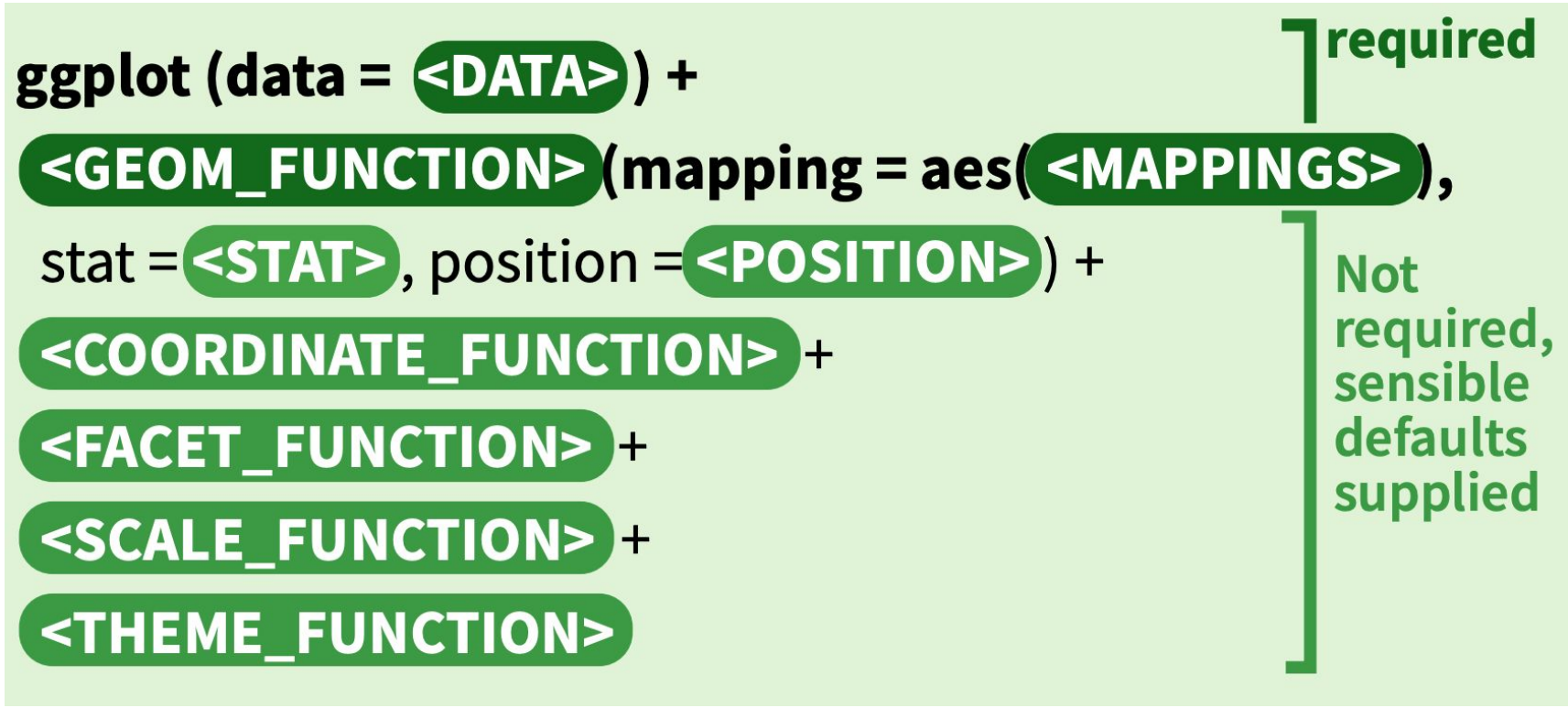
Mid-Semester Assignment | Berlin Autofrei

Extremely simple car-free scenario:

- “car” removed as allowed mode for all links within “Hundekopf” (Purple Outline)
- Code to create this scenario lives in matsim-berlin scenario, on “autofrei-datascience-v6.3” branch: [link]
- Outputs live here: [link]

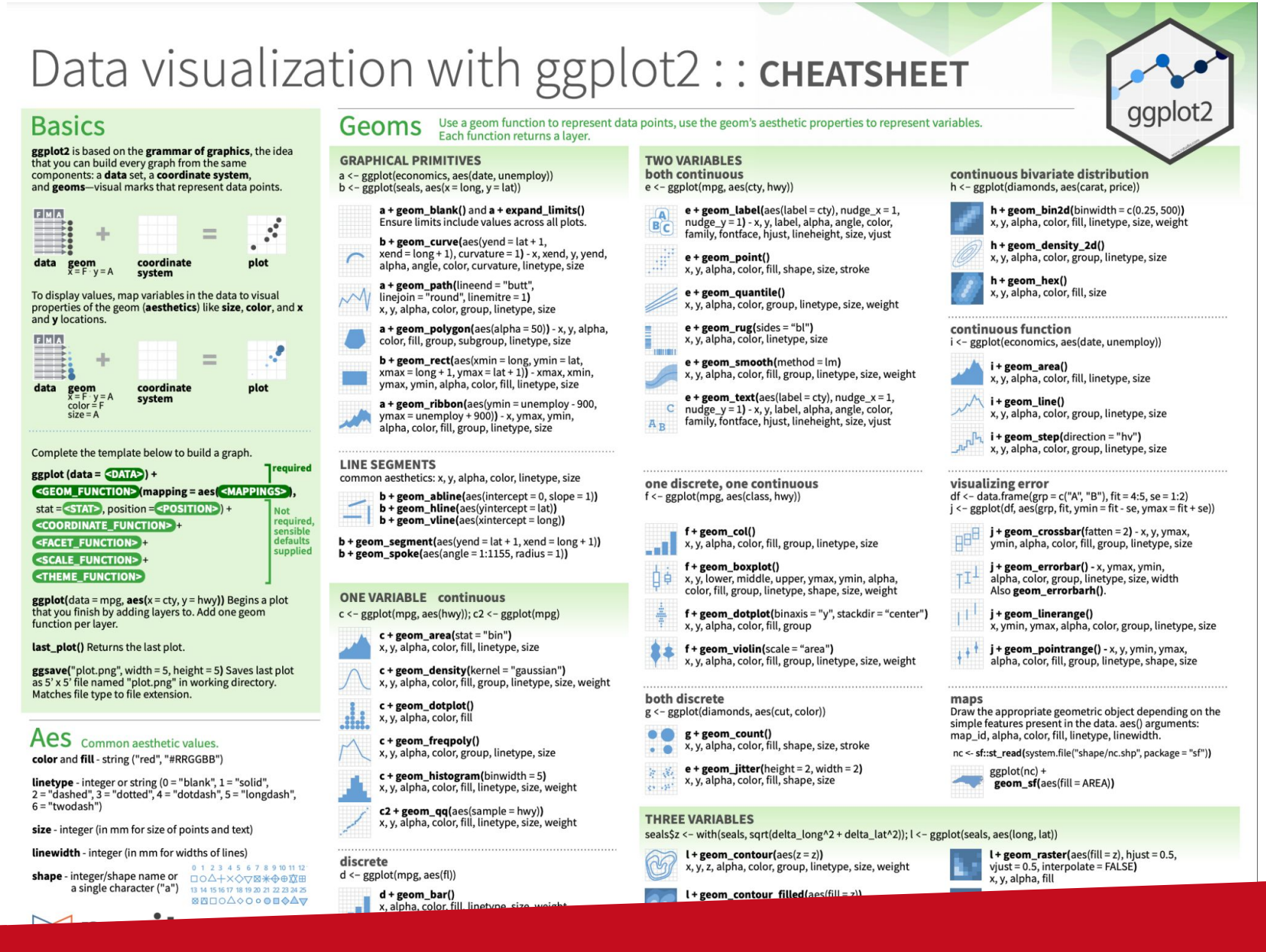


ggplot | Grammar of Graphics



ggplot Resources

- posit ggplot2 cheat sheet
- The R Graph Gallery
- from Data to Viz
- Friends Don’t Let Friends Make Bad Graphs



Graphics for Communication

- Thus far, in exploratory data analysis: We learned how to use plots as tools for *exploration:*
  - When making these plots, we knew (even before looking) which variables the plot would display.
  - We made these plots for a purpose, quickly looked at them, and then moved on to the next plot.
  - In the course of most analyses, you'll produce tens or hundreds of plots, most of which are immediately thrown away.
- Now that we understood our data, we need to *communicate* our understanding to others.
- **Always keep in mind:** We know the data better than the people looking at our charts! They (usually) don't have our background and knowledge. Therefore, we need to make our plots as self-explanatory as possible.

Exploratory analysis: Quick iterations, testing dozens of combinations
Communication: Proper labels, scales, headings, legends are needed

Graphics for Communication

- Labels
- Annotations
- Scales and Axes
- Colors
- Zooming
- Themes

Annotations

- It might be useful to label individual observations or groups of observations.
- The first tool to add annotations is geom_text().
  - geom_text() is similar to geom_point(), but it has an additional aesthetic: label. This makes it possible to add textual labels to your plots.
  - You might use a tibble to provide the labels or (as in our case) if there is just one label, we may set it manually.
- To overcome the overlapping of labels and points we can use the ggrepel package by Kamil Slowikowski.
- This useful package will automatically adjust labels so that they don't overlap:

Scales & Axes

- The third way you can make your plot better for communication is to adjust the scales.
- Scales control the mapping from data values to things that you can perceive.
- Normally, ggplot2 automatically adds scales for you.
- Default scales have been carefully chosen to do a good job for a wide range of inputs. Nevertheless, you might want to override the defaults for two reasons:
    - You might want to tweak some of the parameters of the default scale. This allows you to do things like change the breaks on the axes, or the key labels on the legend.
    - You might want to replace the scale altogether, and use a completely different algorithm. Sometimes you can do better than the default because you know more about the data.

Scales & Axes

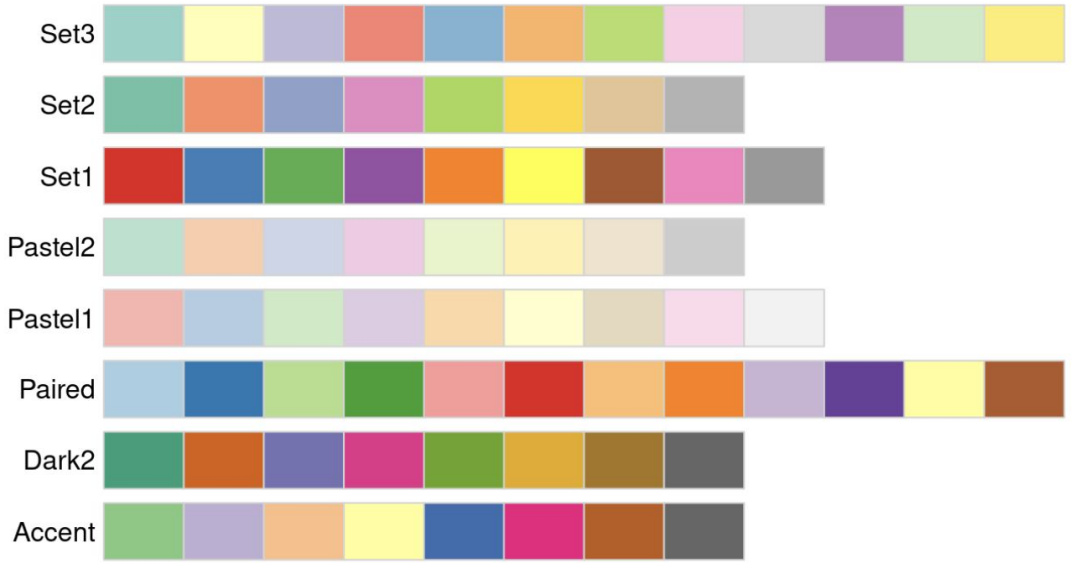
- Some examples:
  - scale_x_continuous()
  - scale_y_continuous(breaks = seq(15, 40, by=5))
  - scale_x_date()
  - scale_x_log10()
- Note the naming scheme for scales: scale_ followed by the name of the aesthetic, then _, then the name of the scale. The default scales are named according to the type of variable they align with: continuous, discrete, datetime, or date.

Colors

- Another scale that is frequently customised is color.
- The default categorical scale picks colors that are evenly spaced around the color wheel.
- Useful alternatives are the ColorBrewer scales which have been hand tuned to work better for people with common types of color blindness.
- Two plots below look similar, but there is enough difference in the shades of red and green that the dots on the right can be distinguished even by people with red-green color blindness.

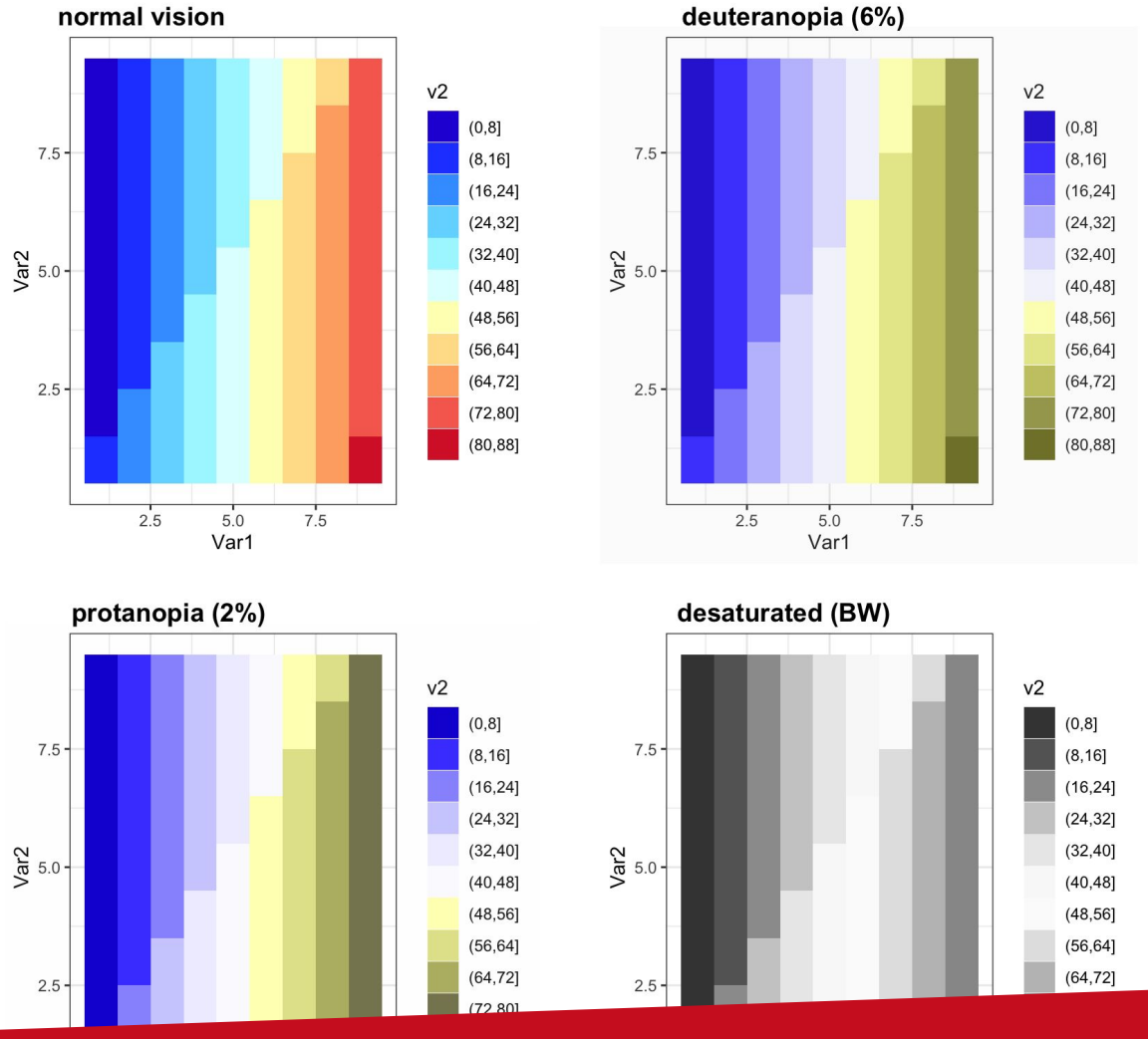
Colors

- Documentation of ColorBrewer scales can be found on http://colorbrewer2.org/
- The made available in R via the RColorBrewer package by Erich Neuwirth
- The sequential (left) and diverging (right) palettes are particularly useful if your categorical values are ordered, or have a “middle”. This often arises if you’ve used cut() to make a continuous variable into a categorical variable.



Colors | Color Blindness

- Color blindness is a synonym for color visio
  color blindness may be unaware of differen
  the people without it.
- There are different kinds of color blindness,
  (defective green cone cells) and protanopia
- There are several packages designed to hel
  read by those with color blindness
- We only briefly touch upon
  - colorblindr
  - viridis

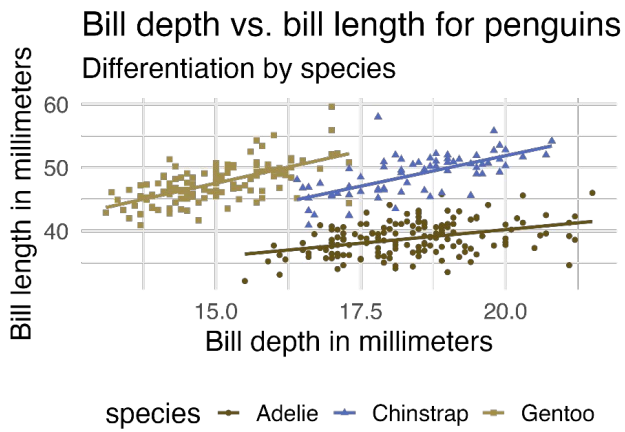


Colors | Color Blindness

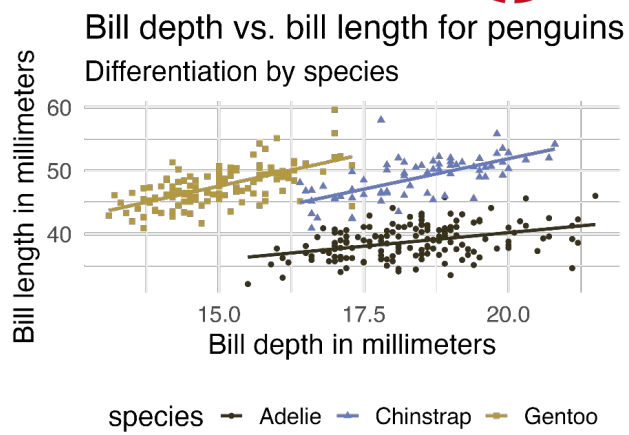
colorblindr

- According to the developers, “Simulate colorblindness in production-ready R figures.”
- Installation guide: https://github.com/clauswilke/colorblindr
- Use cvd_grid() to create the following:

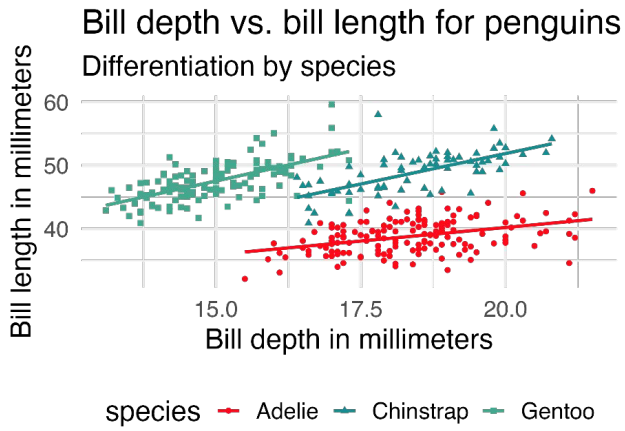
Deutanomaly



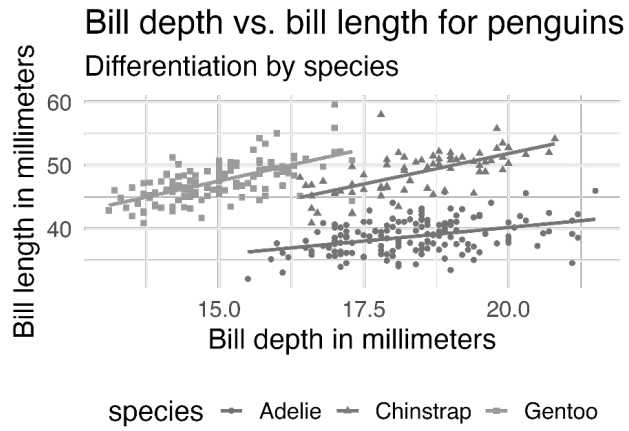
Protanomaly



Tritanomaly



Desaturated



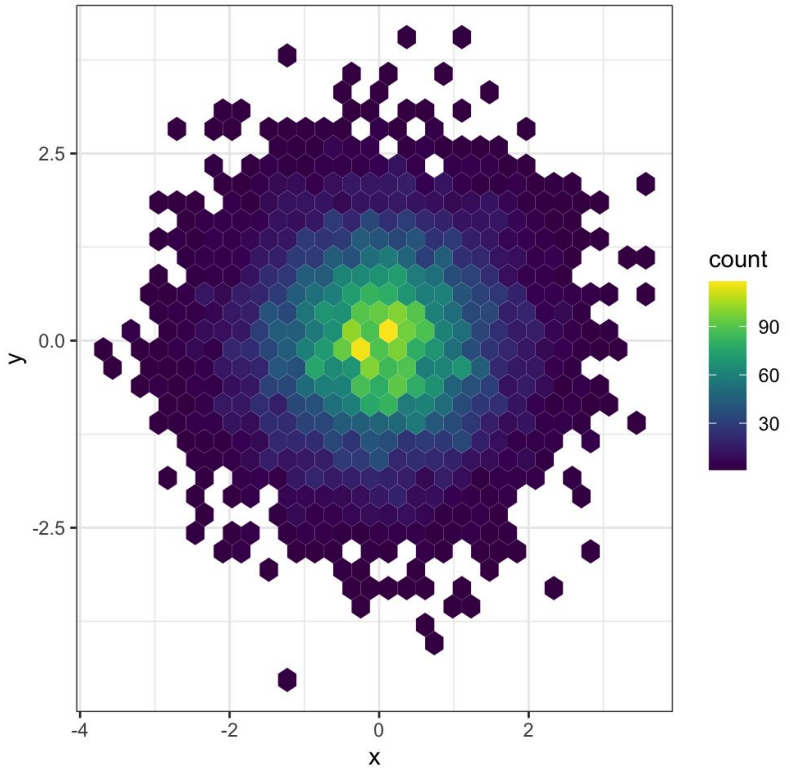
Colors | Viridis

viridis

- When you have a continuous variable and you need to visually discern the different levels, try out viridis

For more details see:

- youtube.com/watch?v=xAoljeRJ3lU
- https://cran.r-project.org/web/packages/viridis/vignettes/intro-to-viridis.html

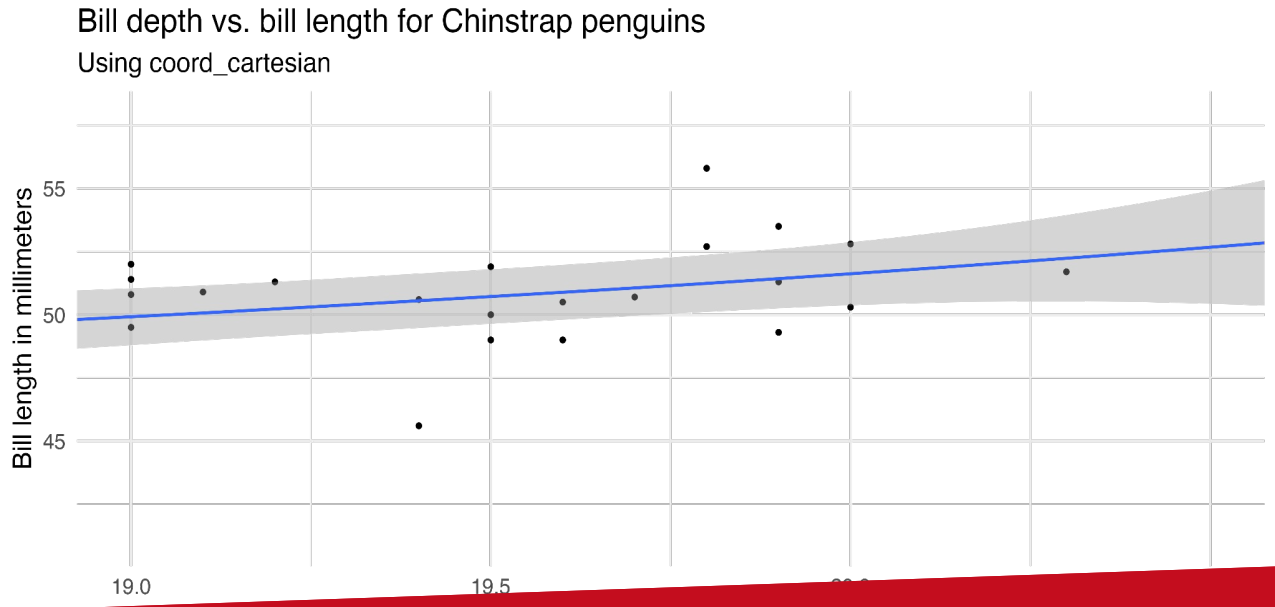


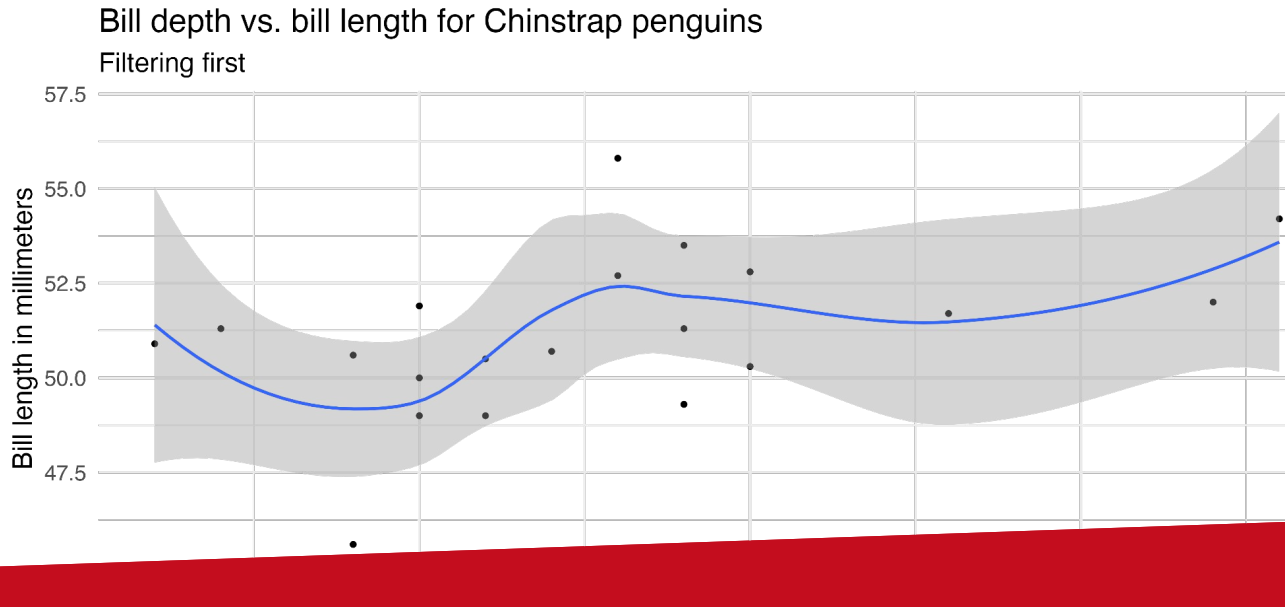
Zooming

3 ways to control the plot limits:

1. Adjusting what data are plotted
2. Setting the limits in each scale
3. Setting xlim and ylim in coord_cartesian()

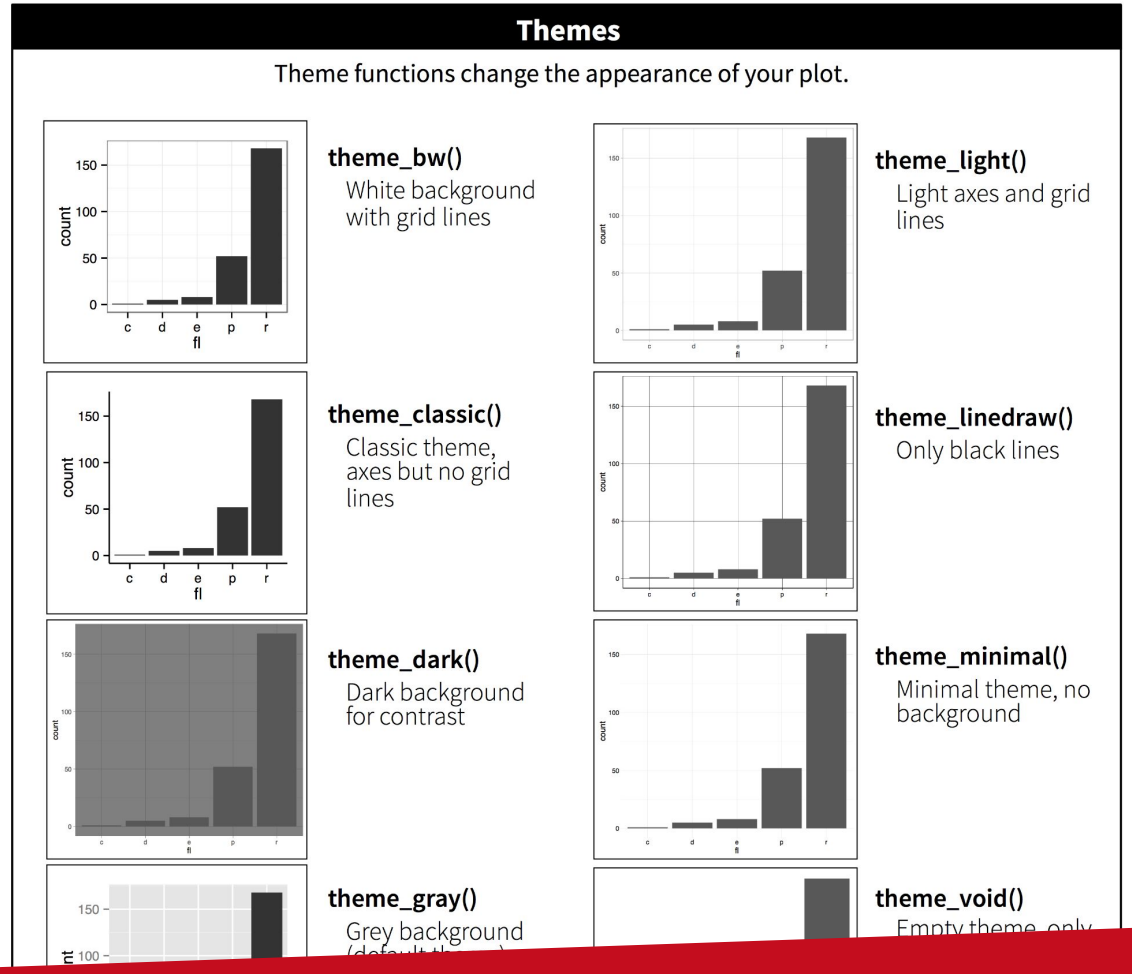
To zoom in on a region of the plot, it's generally best to use coord_cartesian(). Compare the following two plots:





Themes

- Finally, you can customise the non-data elements of your plot with a theme.
- ggplot2 includes eight themes by default. Many more are included in add-on packages like ggthemes by Jeffrey Arnold.
- theme_minimal() is my personal favorite (as you might have noticed)



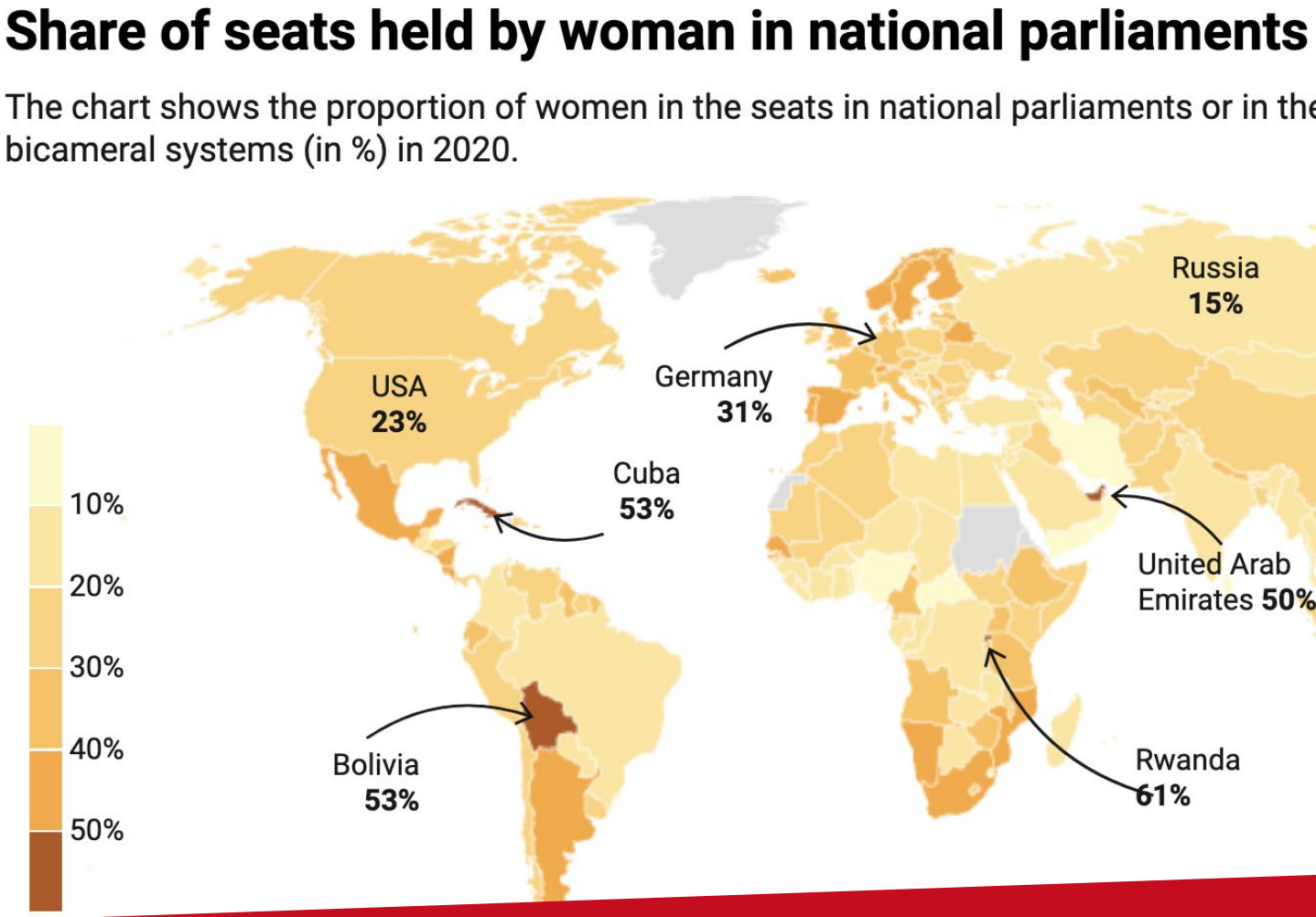
Accessibility | Alt Text

- Alt text (sometimes called Alt tags or alternative text): Written descriptions added to images that convey the meaning of the visual.
- Why is this important? Because assistive technology (like screen readers) reads the alt text out loud.
- Users of screen readers include the totally blind, but also people with low/impaired vision.
- Hence, using alt text helps users to hear (and hence more users to understand) visual content.
- But how to write alt text that conveys your data visualization in a meaningful way?
  - Suggestion: alt = "Chart type of type of data where reason for including chart/main inference"
- Where to include alt text in your graph?
  - Set the "alt" argument in `labs()`
- How to check a graphic for alt text?
  - Use `get_alt_text()`

Accessibility | Alt Text

A map: When writing alt text for a map, try not to focus on visual elements. Instead: Highlight the important message of the map.

Possible alt text:
“A choropleth map of the proportion of women in national parliaments in the world in 2020, where Rwanda stands out as the country with the highest percentage of women (61%) in a national parliament, followed by Bolivia and Cuba, with a 53%. Papua New Guinea is the country that scored the worst with 0% of women in their parliament.”



Accessibility | BrailleR

In addition, there's also the BrailleR package

- According to it's developers, "The BrailleR package is a collection of tools to make use of R a happier experience for blind people."
- You can install BrailleR using install.packages("BrailleR"), but you might have to install some of its dependencies manually

Two helpful functions

- VI() : "A set of methods that extract the most relevant information from a graphical object (or implied set of graphical objects) and display the interpreted results in text or HTML form. [...]"
- VI.ggplot() : "Prints a textual description of a graph produced by ggplot or qplot."