Day 7 Cheatsheet

Data Visualization

Major concepts:

- The `ggplot2` package is a tidyverse package for producing graphics.
- Tidy data: Each variable forms a column and each observation forms a row.
- Having data in tidy format and long format makes creating plots easier!
- Remember to use a `+` at the end of each line when adding new layers to plots.
  Pipes like `%>%` do not work to add layers to plots, but can be used to pipe data into plots.
  This works: `ggplot(data = iris, aes(x = Species, y = Petal.Length)) + geom_boxplot()`
  This works: `iris %>% ggplot(aes(x = Species, y = Petal.Length)) + geom_boxplot()`
  This doesn't: `iris %>% ggplot(aes(x = Species, y = Petal.Length)) %>% geom_plot()`

- Factor: A factor is a special character vector where the elements have pre-defined groups or "levels". You can think of these as qualitative or categorical variables. An example is 1st grade, 2nd grade, etc. It is important to specify factors as the class `factor` so that R recognizes it as such.

Functions

Library/Package	Piece of code	Example of usage	What it does
ggplot2	ggplot()	ggplot(data = iris)	Begins a plot that is finished by adding layers.
ggplot2	aes()	ggplot(data = iris, aes(x = Species, y = Petal.Length))	Aesthetic Mappings - Designates how variables in the data object will be mapped to the visual properties of the ggplot.
ggplot2	geom_boxplot()	ggplot(data = iris, aes(x = Species, y = Petal.Length)) + geom_boxplot()	Creates a boxplot when added as a layer to a `ggplot()` object.
ggplot2	geom_density()	ggplot(data = iris, aes(x = Petal.Length)) + geom_density()	Creates a smoothed density plot when added as a layer to a `ggplot()` object based on the computed density estimate.
ggplot2	geom_point()	ggplot(data = iris, aes(x = Species, y = Petal.Length)) + geom_point()	Creates a scatterplot when added as a layer to a `ggplot()` object.

Library/Package	Piece of code	Example of usage	What it does
ggplot2	`geom_line()`	`ggplot(data = iris, aes(x = Species, y = Petal.Length)) + geom_line()`	Creates a line plot when added as a layer to a `ggplot()` object by connecting the points in order of the x axis variable.
ggplot2	`geom_hline()`	`ggplot(data = iris, aes(x = Species, y = Petal.Length)) + geom_point() + geom_hline(yintercept = 4)`	Annotates a plot with horizontal lines when added as a layer to a `ggplot()` object with one of the `geom` functions used to draw the plot, for example, `geom_point()`.
ggplot2	`theme_classic()`	`ggplot(data = iris, aes(x = Petal.Length)) + geom_density() + theme_classic()`	Displays `ggplot` without grid lines. The `ggtheme` documentation has descriptions on additional themes that can be used.
ggplot2	`xlab(); ylab(); ggtitle()`	`ggplot(data = iris, aes(x = Petal.Length)) + geom_density() + xlab("Petal Length")`	Modifies the labels on the x axis and on the y axis, and sets the title of a `ggplot`, respectively.
ggplot2	`facet_wrap()`	`ggplot(data = iris, aes(x = Petal.Length)) + geom_density() + facet_wrap(~Species)`	Plots individual graphs using specified variables to subset the data.
ggplot2	`ggsave()`	`ggsave(filename = "plotname.pdf")`	Saves the last plot in working directory.
ggplot2	`last_plot()`	`Last plot()`	Returns the last plot produced.

Click here for a summary of the `ggplot2` theme system~

Factors

Library/Package	Piece of code	Example of usage	What it does
`base`	`factor()`	`fact_qual <-factor(c("poor", "fine", "good"))`	Creates a factor out of a vector.
`base`	`levels()`	`levels(fact_qual)`	Shows the levels (and order) of a factor vector.
`base`	`levels()`	`levels(fact_qual) <- c("poor", "fine", "good")`	Allows the order of the levels of a factor to be changed.
`forcats`	`fct_reorder()`	`fct_reorder(.f = grade, .x = absences)`	Allows the order of the levels of a factor variable (specified by the `.f` argument) to be changed according to another variable (specified by the `.x` argument). In this case grade is ordered by absences. This is especially helpful for plots or statistical output!

* This format was adapted from the cheatsheet format from AlexsLemonade.