Visualization with R: tidyverse & ggplot2

Genomic Data Visualization & Integration

HUGEN 2073

(Slides borrowed/modified from Ryan Minster’s with permission)

Learning Objectives

By the end of the session, students will be able to:

- Use basic functions of R tidyverse *dplyr* to work with data frames
- Describe the basic syntax used by *ggplot2*
- Create a scatterplot with two categories of data points using R *ggplot2*

ACHTUNG!

- I have tried to catch Microsoft autocorrects but may have missed some.
- Beware the difference between ", “, and ”.
- Beware the difference between ', ‘, ’, and `.
- Beware the difference between -, –, and — (a hyphen, an en dash and an em dash, respectively). Also - vs --. Also = vs ==.
- Double check capitalization.
- Beware the difference between O, o and 0, and between 1, I and l.

tidyverse (and dplyr)

- tidyverse is an extension of R syntax that is meant to facilitate data science and encourages the use of a forward-pipe operator (%>%) over nesting although there is still a *lot* of nesting

- tidyverse is a suite of packages: `ggplot2`, `dplyr`, `tidyr`, `readr`, `purrr`, `tibble`, `stringr`, and `forcats`. A few other packages are useful, for example, `magrittr` (which adds %$%, and %<>%)

  ```
  library(tidyverse)
  library(magrittr)
  ```

Some Data Management Basics

- Reading in data

```
data <- read_csv("20220120-rlm-2073_Data.csv")
```

- Writing out data

```
write_csv(data, "20220120-YOURINITIALS-2073_Data.csv")
```

Some Data Management Basics

- Data read in via `read_csv()` (as opposed to `read.csv()`) technically creates a tibble rather than a data frame. The characteristics are slightly different, but don’t worry too much about them today.

- You can pipe the data to functions using `%>%`

  ```
  data %>% summarize(mean(height), sd(height))
  ```

  instead of

  ```
  mean(data$height); sd(data$height)
  ```

Some Data Management Basics

- You can subset using `filter()`

  ```
  data %>% filter(sex == "F")
  ```

  instead of

  ```
  data[data$sex == "F", ]
  ```

Optical Method cont’d

- “table()” categorical data

```
data %>% select(sex) %>% table()
```

- Look at basic histograms of continuous data using hist()

```
data %>% ggplot(aes(x = height)) + geom_histogram()
```

- Play around with binwidth to see different shapes of the histograms

```
data %>% ggplot(aes(x = height)) +
         geom_histogram(binwidth = 5)
data %>% ggplot(aes(x = height)) +
         geom_histogram(binwidth = 10)
```

ggplot2

- There *was* a `ggplot` package, with different syntax, but it was abandoned by its developer for `ggplot2`
- The syntax is built around constructing a plot in layers, for example:
  - supplying data set (`ggplot`) +
  - choosing type of plot (`geom`) +
  - applying mapping of data to x and y and colors (`aes`) +
  - etc. etc.
- Basic template

```
ggplot(data = <DATA>) +
  aes(x = <X_VARIABLE>, y = <Y_VARIABLE>) +
  <GEOM_FUNCTION>()
```

Create a Scatterplot using **`ggplot2`**

- If we are plotting height vs age
  - That is, by convention, **dependent** vs **independent** variable, *or*
  - ***y*** vs ***x***
  - So, ***y* = height** and ***x* = age**

```
data %>%
  ggplot() +
  aes(x = age, y = height) +
  geom_point()
```

Create a Scatterplot using **ggplot2**

- Alternatively

```
data %>%
  ggplot(aes(x = age, y = height)) +
  geom_point()
```

- Or

```
ggplot(data, aes(x = age, y = height)) +
  geom_point()
```

- Or

```
data %>% ggplot() + aes(age, height) + geom_point()
```

Create a Scatterplot using **`plot()`**

- Change points shape and color

```
data %>% ggplot() + aes(age, height) +
  geom_point(shape = 16, color = alpha("black", 0.25))
```

- `alpha()` is a function that lets you add transparency to a color, such that `0.25` means 25% opaque and 75% transparent.

Stratify Scatterplot by Category

- Simple with `ggplot2`, specify that the color and the shape are set by the `sex` field from `data`.

```
data %>%
  ggplot() +
  aes(x = age, y = height, color = sex, shape = sex) +
  geom_point()
```

Change the Default Colors

- Add a layer that specifies colors (`+ scale_color_manual()`) and set the point transparency in `geom_point()`

```
data %>%
  ggplot() +
  aes(x = age, y = height, color = sex, shape = sex) +
  geom_point(alpha = 0.25) +
  scale_color_manual(values = c("darkgreen", "purple"))
```

Add a Trendline

- Add a layer that specifies a trendline with + `geom_smooth()`

```
data %>%
  ggplot() +
  aes(x = age, y = height, color = sex, shape = sex) +
  geom_point(alpha = 0.25) +
  scale_color_manual(values = c("darkgreen", "purple")) +
  geom_smooth(method = “lm”, alpha = 0.5, size = 2,
              linetype = "42")
```

Plot Separately Using Facets

- Add a layer that specifies a split in the data (a facet) with
  `+ facet_wrap()`

```
data %>%
  ggplot() +
  aes(x = age, y = height, color = sex, shape = sex) +
  geom_point(alpha = 0.25) +
  scale_color_manual(values = c("darkgreen", "purple")) +
  stat_smooth(method = "lm", geom = "line", alpha = 0.5,
              size = 2) +
  facet_wrap(~sex)
```

Set Labels

- Add a layer that changes the labels with `+ labs()`

```
data %>%
  ggplot() +
  aes(x = age, y = height, color = sex, shape = sex) +
  geom_point(alpha = 0.25) +
  scale_color_manual(values = c("darkgreen", "purple")) +
  stat_smooth(method = "lm", geom = "line", alpha = 0.5,
              size = 2) +
  facet_wrap(~sex) +
  labs(title = "Height vs Age", x = "Height (cm)",
       y = "Age (years)")
```

Move Legend

- Move the legend with `+ theme()`

```
data %>%
  ggplot() + aes(age, height, color = sex, shape = sex) +
  geom_point(alpha = 0.25) +
  scale_color_manual(values = c("darkgreen", "purple")) +
  stat_smooth(method = "lm", geom = "line", alpha = 0.5,
              size = 2) +
  facet_wrap(~sex) +
  labs(title = "Height vs Age", x = "Height (cm)",
       y = "Age (years)") +
  theme(legend.position = "top")
```

Set Black and White Background

- Change the background appearance to black and white, for example, with `+ theme_bw()`

```
data %>%
  ggplot() + aes(age, height, color = sex, shape = sex) +
  geom_point(alpha = 0.25) +
  scale_color_manual(values = c("darkgreen", "purple")) +
  stat_smooth(method = "lm", geom = "line", alpha = 0.5,
              size = 2) +
  facet_wrap(~sex) +
  labs(title = "Height vs Age", x = "Height (cm)",
       y = "Age (years)") +
  theme(legend.position = "top") + theme_bw()
```

Other **geom**’s

```
geom_point()
geom_histogram()
geom_density()
geom_bar()
geom_dotpoint()
geom_boxplot()
geom_violin()
```

Resources

https://ggplot2.tidyverse.org/index.html

https://sthda.com/english/wiki/ggplot2-essentials