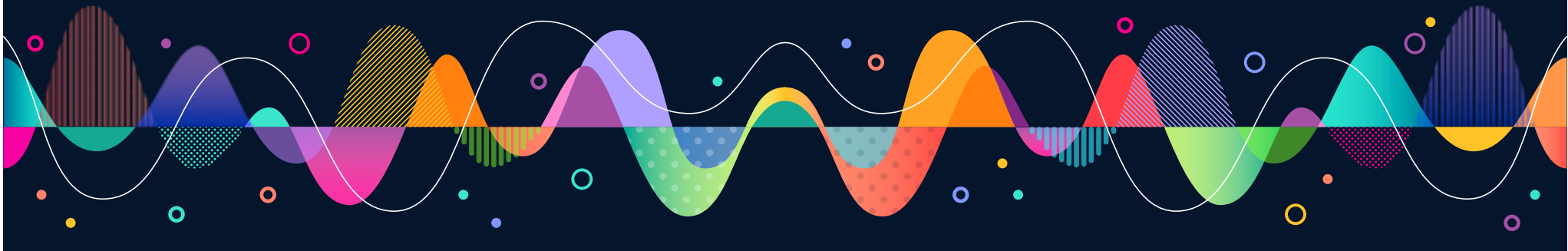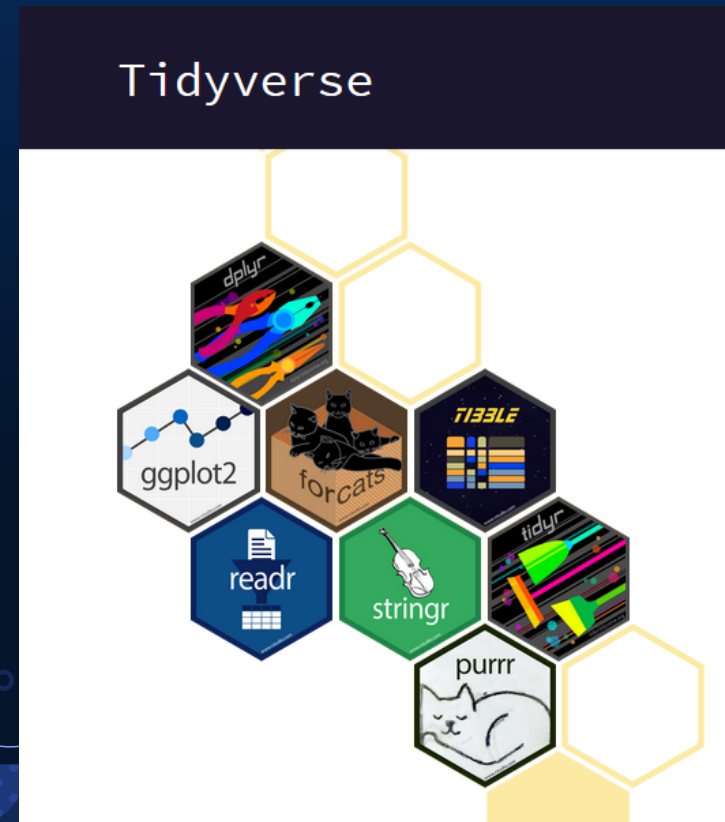# An Introduction to Tidyverse

Roy Roberts
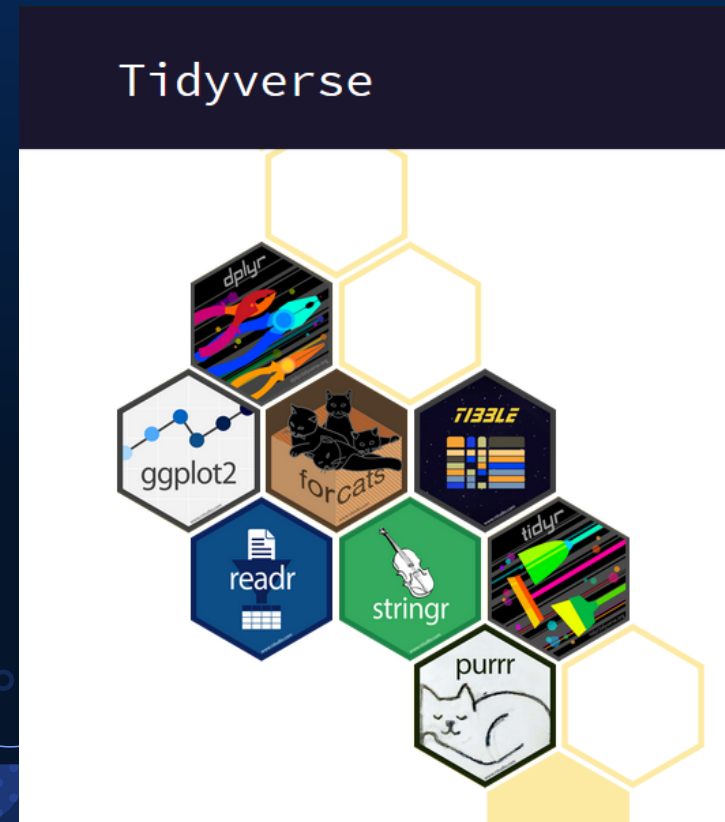
# The Tidyverse

▷ "The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures."

# The Tidyverse

▷ What is tidy data?

  ▸ Every column is a variable

  ▸ Every row is an observation

  ▸ Every cell is a single value

# Packages in Tidyverse

▷ Core Tidyverse contains:

- tibble
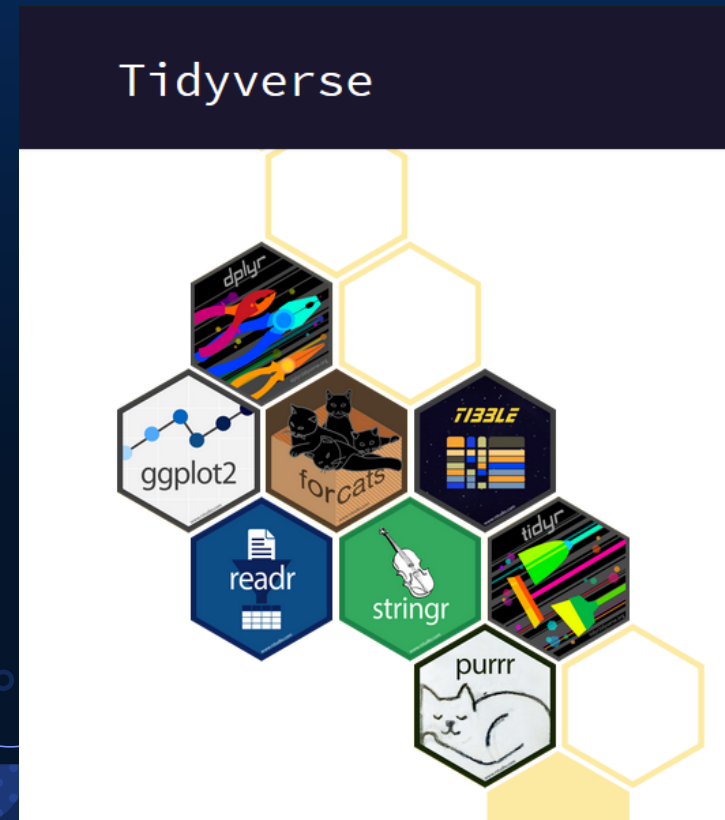- ggplot2
- dplyr
- tidyr
- readr
- purrr
- stringr
- forcats

# How to Install and Load Tidyverse

▷ Installing and loading Tidyverse

install.packages("tidyverse")

library(tidyverse)

# Miscellaneous Information Before Starting Tidyverse

```
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
```

%>%  view()  function(.)

source("path/to/file")

tibble$colunm_name

# Reading in Data With readr

- ▷ Package 'readr'

  - ▸ read_tsv()

  - ▸ read_csv()

  - ▸ read_delim()

  - ▸ read_table()

- ▷ Package 'readxl'

  - ▸ read_excel()

```r
#### Reading in data ####
read_tsv("fastqc_data.tsv")
read_delim("fastqc_data.tsv",
           delim = "\t")
fastqc_tbl <- read_tsv(file = "fastqc_data.tsv",
                       skip = 1,
                       col_names = c("id",
                                     "total_sequences",
                                     "percent_failed",
                                     "percent_gc_content",
                                     "percent_duplication"),
                       col_types = cols(id = col_character(),
                                        total_sequences = col_double(),
                                        percent_failed = col_double(),
                                        percent_gc_content = col_double(),
                                        percent_duplication = col_double()))
```

# Reading in Data With readr

Helpful options:

```r
fastqc_tbl <- read_tsv(file = "fastqc_data.tsv",
                       skip = 1,
                       col_names = c("id",
                                     "total_sequences",
                                     "percent_failed",
                                     "percent_gc_content",
                                     "percent_duplication"),
                       col_types = cols(id = col_character(),
                                        total_sequences = col_double(),
                                        percent_failed = col_double(),
                                        percent_gc_content = col_double(),
                                        percent_duplication = col_double()))
```

# Tidyverse Activity 1

▷ Read in altered "Davis" from car dataset located in the current working directory.

▷ Save as a variable.

▷ Try and use some of the other options when saving as a variable.

▷ Pipe in drop_na() and compare with and without the new command.

▷ Save the above tibble as a variable to use for later activities.

# Subsetting Data in R

- ▷ select()
  - ▸ Subsets columns of a tibble
- ▷ filter()
  - ▸ Subsets rows of a tibble

```
#### Subseting Your Data ####

fastqc_tbl %>%
  select(id,
         total_sequences)
fastqc_tbl %>%
  select(-starts_with("percent"))

fastqc_tbl %>%
  filter(total_sequences >= 50000000)

fastqc_tbl %>%
  filter(percent_gc_content == 44 | percent_gc_content == 45,
         total_sequences <= 50000000 & total_sequences >= 30000000)

fastqc_tbl %>%
  filter(total_sequences >= 50000000) %>%
  select(id,
         percent_duplication)
```

# Transforming Data in R

▷ mutate()

  ▸ Create, modify, and delete columns

  ▸ data %>% mutate(New_Column = Column_1 * Number)

  ▸ data %>% mutate(Column_1 = Column_1 * Number)

▷ case_when()

  ▸ A general vectorized conditional (if) statement that can be used with mutate

  ▸ data %>% mutate(New_Column = case_when(Column1 == Value ~ Value,

  Column1 == Value ~ Value))

```r
#### Changing and Creating Columns ####
fastqc_tbl %>%
  mutate(log_total_sequences = log10(total_sequences))

fastqc_tbl %>%
  mutate(percent_duplication = percent_duplication/100)

fastqc_tbl %>%
  mutate(percent_duplication_as_decimal = percent_duplication/100,
         number_of_nonduplicate_sequences = percent_duplication_as_decimal * total_sequences,
         number_of_duplicated_sequences = (1 - percent_duplication_as_decimal) * total_sequences) %>%
  select(id,
         total_sequences,
         number_of_nonduplicate_sequences,
         number_of_duplicated_sequences)

fastqc_tbl %>%
  mutate(gc_interpretation = case_when(percent_gc_content < 45 ~ "Less than 40% GC Content",
                                       percent_gc_content > 50 ~ "Greater then 50% GC Content",
                                       TRUE ~ "Inbetween 40 and 50% GC Content")) %>%
  select(id,
         gc_interpretation)
```

# Tidyverse Activity 2

▷ Using the previously read in davis tibble, remove both heights from the data set using select.

▷ Filter your data by Male or Female and also by 1 or 2 different hair colors.

▷ Create 2 new columns

▸ The mean of weight 1 and 2

▸ The difference of weight 1 and 2

▷ Create a new column that's value is either negative if the difference of weight 1 and 2 is negative, positive if positive, and no change if 0

# Transforming Data in R

▷ **relocate()**

　　▶ Change column order

▷ **unite()**

　　▶ Unite multiple columns into one by pasting strings together

▷ **separate()**

　　▶ Separate a character column into multiple columns with a regular expression or numeric locations

# Transforming Data in R

- unite()

  - data %>% unite("New_Column",

    columns_to_select,

    sep = "delimiter",

    remove = FALSE)

- separate()

  - data %>% separate(Column_1,

    into = c("New_Col1", "New_Col2", "New_Col3"),

    sep = "delimiter",

    remove = FALSE)

Left window:

Source on Save | Run | Source

```
143
144
145   (fastqc_tbl_separate <- fastqc_tbl %>%
146     separate(id,
147             into = c("wga",
148                      "species",
149                      "repair",
150                      "ng_dna"),
151             sep = "_"))
152
153
154
155
```

144:1 | # Changing and Creating Columns | R Script

Console | Terminal ✕ | Jobs ✕

R 4.1.2 · C:/Users/Roy_R/Texas A&M University - Corpus Christi/Bird, Chris - Thesis_Roy_Roberts/repos/

```
> (fastqc_tbl_separate <- fastqc_tbl %>%
+   separate(id,
+           into = c("wga",
+                    "species",
+                    "repair",
+                    "ng_dna"),
+           sep = "_"))
# A tibble: 101 x 8
   wga    species   repair  ng_dna  total_sequences
   <chr>  <chr>     <chr>   <chr>              <dbl>
 1 NoWGA  Adu-AMat  NR      10              22626345
 2 NoWGA  Adu-AMat  NR      3.2             18538669
 3 NoWGA  Adu-AMat  R       10              32061574
 4 NoWGA  Adu-AMat  R       3.2             20795529
 5 NoWGA  Adu-CHam  NR      10              37212563
 6 NoWGA  Adu-CHam  NR      1               56793298
 7 NoWGA  Adu-CHam  NR      3.2             55032990
 8 NoWGA  Adu-CHam  R       10              56331948
 9 NoWGA  Adu-CHam  R       1               52844376
10 NoWGA  Adu-CHam  R       3.2             50799485
# ... with 91 more rows, and 3 more variables:
#   percent_failed <dbl>,
#   percent_gc_content <dbl>,
#   percent_duplication <dbl>
>
```

Right window:

Source on Save | Run | Source

```
154
155
156   (fastqc_tbl_unite <- fastqc_tbl_separate %>%
157     relocate(species,
158             ng_dna,
159             repair,
160             wga) %>%
161     unite("id",
162           species:wga,
163           sep = "-"))
164
165
```

165:1 | # Changing and Creating Columns | R Script

Console | Terminal ✕ | Jobs ✕

R 4.1.2 · C:/Users/Roy_R/Texas A&M University - Corpus Christi/Bird, Chris - Thesis_Roy_Roberts/repos/rroberts_t

```
> (fastqc_tbl_unite <- fastqc_tbl_separate %>%
+   relocate(species,
+           ng_dna,
+           repair,
+           wga) %>%
+   unite("id",
+         species:wga,
+         sep = "-"))
# A tibble: 101 x 5
   id                       total_sequences  percent_failed
   <chr>                              <dbl>           <dbl>
 1 Adu-AMat-10-NR-NoWGA            22626345            36.4
 2 Adu-AMat-3.2-NR-NoWGA          18538669            45.5
 3 Adu-AMat-10-R-NoWGA             32061574            36.4
 4 Adu-AMat-3.2-R-NoWGA           20795529            36.4
 5 Adu-CHam-10-NR-NoWGA            37212563            36.4
 6 Adu-CHam-1-NR-NoWGA             56793298            27.3
 7 Adu-CHam-3.2-NR-NoWGA          55032990            36.4
 8 Adu-CHam-10-R-NoWGA             56331948            18.2
 9 Adu-CHam-1-R-NoWGA              52844376            18.2
10 Adu-CHam-3.2-R-NoWGA           50799485            18.2
# ... with 91 more rows, and 2 more variables:
#   percent_gc_content <dbl>,
#   percent_duplication <dbl>
>
```

# Tidyverse Activity 3

▷ Separate name into surname and given_name

▷ Combine haircolor and sex into haircolor_sex

▷ Move the columns around for fun!

# Pivoting Data in R

▷ pivot_longer()

  ▸ Pivots wide data to long data

    ■ Wide data sets do not contain repeating values in the ID

    ■ Long data sets do contain repeating values in the ID

  ▸ pivot_longer(cols = ColumnX:ColumnY,

          names_to = "Column_Name1"

         values_to = "Column_Name2")

# Pivoting Data in R

▷ pivot_wider()

  ▸ Pivots long data to wide data

    ▪ Wide data sets do not contain repeating values in the ID

    ▪ Long data sets do contain repeating values in the ID

  ▸ pivot_longer(id_cols = ColumnX,

           names_from = Column_Name1

           values_from = Column_Name2)

# Tidyverse Activity 4

▷ Pivot longer all davis numeric davis values longer and save the new tibble as a variable.

▷ Pivot wider the previous tibble back to its original state.

# Summarising a Tibble in R

▷ summarise()

  ‣ To use summarise() a tibble must be grouped by one or more variables using group_by()

    ■ group_by(ColumnX, ColumnY)

  ‣ summarise(NewColumn1 = function(Column),

              NewColumn2 = function(Column),

              ...)

  ‣ Keeping a tibble grouped can cause problems with other functions so ungroup() should be used after grouping is no longer needed

```r
99
100  #### Summarising a Tibble ####
101
102  fastqc_tbl_separate %>%
103    group_by(species) %>%
104    summarise(number_of_observations = n()) %>%
105    ungroup()
106
107  fastqc_tbl_separate %>%
108    group_by(wga, ng_dna) %>%
109    summarise(number_of_observations = n(),
110              mean_failed_percent = mean(percent_failed),
111              median_failed_percent = median(percent_failed),
112              mean_duplicate_percent = mean(total_sequences),
113              median_duplicate_percent = median(total_sequences),
114              mean_number_of_nonduplicate_sequences = mean((1-(percent_duplication/100)) * total_sequences)) %>%
115    ungroup()
116
```

116:1  # Summarising a Tibble                                                                                          R Script

Console  Terminal  Jobs

R 4.1.2 · C:/Users/Roy_R/Texas A&M University - Corpus Christi/Bird, Chris - Thesis_Roy_Roberts/repos/2022_PIRE_omics_workshop/R_workshop/

```r
> fastqc_tbl_separate %>%
+   group_by(wga, ng_dna) %>%
+   summarise(number_of_observations = n(),
+             mean_failed_percent = mean(percent_failed),
+             median_failed_percent = median(percent_failed),
+             mean_duplicate_percent = mean(total_sequences),
+             median_duplicate_percent = median(total_sequences),
+             mean_number_of_nonduplicate_sequences = mean((1-(percent_duplication/100)) * total_sequences)) %>%
+   ungroup()
`summarise()` has grouped output by 'wga'. You can override using the `.groups` argument.
# A tibble: 6 x 8
```

| | wga | ng_dna | number_of_observations | mean_failed_percent | median_failed_percent | mean_duplicate_p~ | median_duplicate~ | mean_number_of_no~ |
|---|---|---|---|---|---|---|---|---|
| | <chr> | <chr> | <int> | <dbl> | <dbl> | <dbl> | <dbl> | <dbl> |
| 1 | NoWGA | 1 | 16 | 27.3 | 27.3 | 41158135. | 47215947 | 23223565. |
| 2 | NoWGA | 10 | 20 | 30.0 | 31.8 | 40486239. | 44599912. | 27615043. |
| 3 | NoWGA | 3.2 | 22 | 28.5 | 31.8 | 37670124. | 44573735 | 24570108. |
| 4 | WGA | 0.32 | 12 | 15.2 | 18.2 | 43117634 | 42104900. | 30987209. |
| 5 | WGA | 1 | 13 | 18.2 | 18.2 | 47664618. | 52688931 | 30218688. |
| 6 | WGA | 3.2 | 12 | 16.7 | 18.2 | 42977095. | 46882569 | 32863658. |

```r
> |
```

# Tidyverse Activity 5

▷ Create mean_weight and summarise mean_weight by grouping by haircolor and sex, calculating the following:

▸ Count, n()

▸ Mean, mean()

▸ Standard Deviation, sd()

▸ Median, median()

▸ Interquartile Range, IQR()