A Journey into the World of Tidyverse

Coffee, Cookie and Coding (C3) Workshop by the Public Health Data Science and Data Equity team

Shelby Golden, M.S.

February 16th, 2026

Shelby Golden, M.S.

- Worked 7 years as a Molecular Biologist and Biochemist.
- Received a Masters in Applied Computational Mathematics from Johns Hopkins University in 2024.

Today's Learning Objectives

01 Explore the tidyverse ecosystem and its integrated approach to data analysis using domain specific language (~ 10 minutes)

02 Develop proficiency with `tidyr`, `dplyr`, and `stringr` with a real-world worked-through example (~ 30 minutes)

03 Practice data manipulation skills by answering prepared questions using COVID-19 data (~ 20 minutes)

Our Choice Resources

- Yale's Center for Research Computing workshop "Tidying Data" by Benjamin Evans
- "Learn the tidyverse" webpage of resource by tidyverse
- dplyr, tidyr, and stringr package documentation and cheat sheets by tidyverse

Accessing the Codespaces

In this workshop, you'll access questions outlined in R. If you haven't already, please download the latest version of R to your device. We also recommend using the latest version of RStudio as your Integrated Development Environment (IDE).

Attribution and Ownership

Please note that all materials provided in this workshop, including any code added to your personal repository, belongs to DSDE. When using or referencing this material, please ensure to cite it correctly to give proper credit to the original authors.

Settings Used in Development

Initializing the Environment

1. Download the prepared codebase, which is configured as an RStudio project and includes code for in-class questions with solutions.

   Codespace

2. Unzipped the downloaded directory and move it to the file location you wish to house the project.

   Command-Line Application

```
cd "file_path/Downloads/"                          # Open the directory the file was downlo
unzip A-Journey-into-the-World-of-tidyverse.zip    # Unzip the file.
mv A-Journey-into-the-World-of-tidyverse "/new_path/"   # Move the unzipped directory to the new
```

programming_r_index.ht

`tidyverse` Worked-Through Example

Introduction

We will explore the `tidyverse` packages `tidyr`, `dplyr`, and `stingr` further with a worked-through example cleaning and visualizing COVID-19 daily death counts in the United States. The visualization will be prepared using another `tidyverse` package that was not discussed in this module. If you are interested in learning more about the `ggplot2` package, please check out our workshop on the topic: Data Visualization with ggplot2.

Set Up the Environment

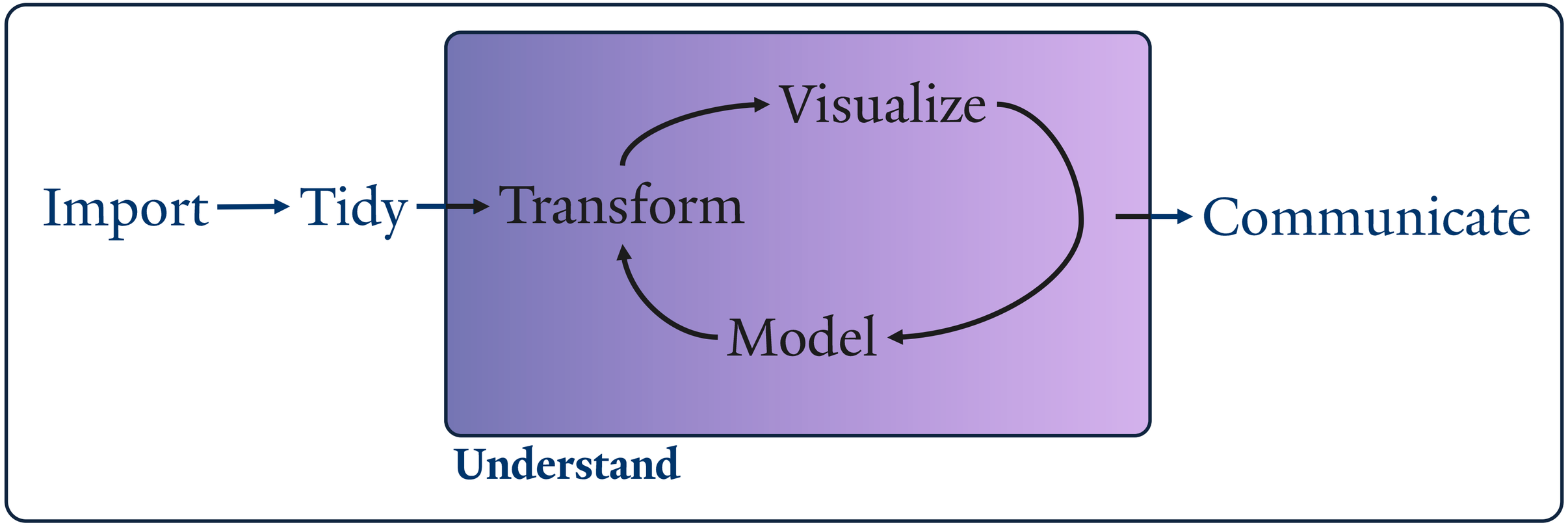
First, we will load the necessary libraries and any special functions used in the script.

```
# NOTE: renv initialization might need to be run twice after the repo is
#       first copied.
#renv::init()
renv::restore()
```

```
- The library is already synchronized with the lockfile.
```

```
suppressPackageStartupMessages({
  library("readr")      # For reading in the data
  library("tidyr")      # For tidying data
  library("dplyr")      # For data manipulation
  library("stringr")    # For string manipulation
  library("lubridate")  # For date manipulation
  library("ggplot2")    # For creating static visualizations
  library("scales")     # For formatting plots axis
```

Cecilia Sanchez (DM) - YSP



R for Data Science (2e) - Introduction Figure 1 by Hadley Wickham, Mine Çetinkaya-Rundel, and Garrett Grolemund. Accessed November 15th, 2024.

Commonly used # core Packages

tidyr Tools for tidying data: i.e. `pivot_wider()`, `pivot_longer()`, and `drop_na()`.

dplyr Tools for transforming data: i.e. `filter()`, `arrange()`, and `mutate()`.

stringr Tools to manage character strings: i.e. `str_c()`, `str_detect()`, and `str_replace()`.

Tidyverse Package Graphic. Accessed November 15th, 2024.

2019 COPSS Presidents' Award Winner Hadley Wickham
Accessed November 18th, 2024.

Dr. Hadley Wickham

- Founder of tidyverse and leader of the current team of collaborators.
- Chief Scientist at Posit (formerly RStudio).
- Adjunct Professor at the University of Auckland, Stanford University, and Rice University.

In 2019, he was awarded the International COPSS Presidents' Award for

> "... developing and implementing an impressively comprehensive computational infrastructure for data analysis through R software..."
>
> - Citation on the COPSS plaque

Break complex nested operations into separate lines using a pipe "|>" or "%>%"

```r
# Nested
round(
  exp(
    diff(
      log(x)
    )
  )
, digits = 1)
```

```r
# Base R pipe
X |>
  log() |>
  diff() |>
  exp() |>
  round(1)
```

Example from *Pipes in R Tutorial For Beginners* by Karlijn Willems. Accessed June 11th, 2025.

Clean Messy Data with tidyr

Yale SCHOOL OF PUBLIC HEALTH
Data Science and Data Equity

“Tidy datasets are all alike, but every messy dataset is messy in its own way.” – Hadley Wickham

Variables

Country	Year	Cases	Pop
AFG	1999	745	20 M
AFG	2000	2667	20.5 M
Brazil	1999	37737	172 M
Brazil	2000	80488	174.5 M
China	1999	212258	1,272 M
China	2000	216766	1,280 M

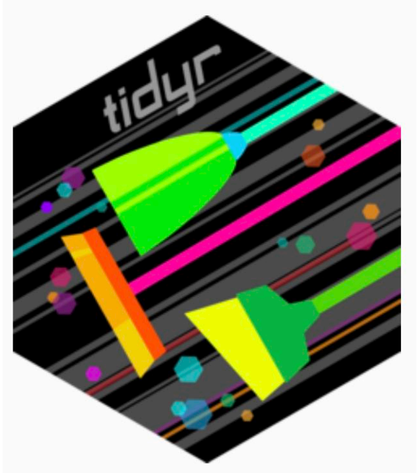
Observations

Country	Year	Cases	Pop
AFG	1999	745	20 M
AFG	2000	2667	20.5 M
Brazil	1999	37737	172 M
Brazil	2000	80488	174.5 M
China	1999	212258	1,272 M
China	2000	216766	1,280 M

Values

Country	Year	Cases	Pop
AFG	1999	745	20 M
AFG	2000	2667	20.5 M
Brazil	1999	37737	172 M
Brazil	2000	80488	174.5 M
China	1999	212258	1,272 M
China	2000	216766	1,280 M

R for Data Science (2e) - Data tidying Figure 5.1 by Hadley Wickham, Mine Çetinkaya-Rundel, and Garrett Grolemund. Accessed November 15th, 2024.



`tidyr` is a set of functions designed to reshape messy data into tidy data.

`pivot_longer()`	Lengthen by combining outcomes over multiple columns.
`pivot_wider()`	Widen by expanding outcomes into columns.
`expand()`	Make a new tibble with every combination.
`complete()`	Add missing combinations as NA.
`unite()`	Join the entries in multiple columns into one column.
`separate_wider_delim()`	Separate entries by a delimitator into new columns.
`separate_longer_delim()`	Separate entries by a delimitator into new rows.
`drop_na()`	Remove rows with NA's in that column.

tidyr Cheat Sheets. Select Examples are copied here. Accessed June 11th, 2025.

Open `tidyverse` Worked-Through Example page on the Book of Workshop website.

Combined_Key	County	Province_State	Country_Region	2020-01-01	2020-02-01	2020-03-01
<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>
Tillamook, Oregon, US	Tillamook	Oregon	US	0	0	0
Green, Wisconsin, US	Green	Wisconsin	US	0	0	0
Jerauld, South Dakota, US	Jerauld	South Dakota	US	0	0	0
Johnston, North Carolina, US	Johnston	North Carolina	US	0	0	1
Holmes, Florida, US	Holmes	Florida	US	0	0	0
Union, Kentucky, US	Union	Kentucky	US	0	0	0

Variables

Country	Year	Cases	Pop
AFG	1999	745	20 M
AFG	2000	2667	20.5 M
Brazil	1999	37737	172 M
Brazil	2000	80488	174.5 M
China	1999	212258	1,272 M
China	2000	216766	1,280 M

Observations

Country	Year	Cases	Pop
AFG	1999	745	20 M
AFG	2000	2667	20.5 M
Brazil	1999	37737	172 M
Brazil	2000	80488	174.5 M
China	1999	212258	1,272 M
China	2000	216766	1,280 M

Values

Country	Year	Cases	Pop
AFG	1999	745	20 M
AFG	2000	2667	20.5 M
Brazil	1999	37737	172 M
Brazil	2000	80488	174.5 M
China	1999	212258	1,272 M
China	2000	216766	1,280 M

R for Data Science (2e) - Data tidying Figure 5.1 by Hadley Wickham, Mine Çetinkaya-Rundel, and Garrett Grolemund. Accessed November 15th, 2024.

Discussion:

Is the data in a “tidy” format? Why or why not?

`pivot_longer()`

country	1999	2000
A	0.7K	2K
B	37K	80K
C	212K	213K

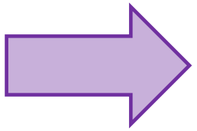
→

country	year	cases
A	1999	0.7K
B	1999	37K
C	1999	212K
A	2000	2K
B	2000	80K
C	2000	213K

Lengthen by combining outcomes over multiple columns.

```
df_raw |> pivot_longer(
    cols = "2020-03-01":"2020-04-01",
    names_to = "Date", values_to = "Deaths_Count_Cumulative")
```

Combined_Key	2020-03-01	2020-04-01
Arizona, US	25	320
Delaware, US	3	212
Utah, US	5	46



Combined_Key	Date	Deaths
Arizona, US	2020-03-01	25
Delaware, US	2020-03-01	3
Utah, US	2020-03-01	5
Arizona, US	2020-04-01	320
Delaware, US	2020-04-01	212
Utah, US	2020-04-01	46

tidyr Cheat Sheets. Select Examples are copied here. Accessed June 11th, 2025.

Manipulate Data with dplyr

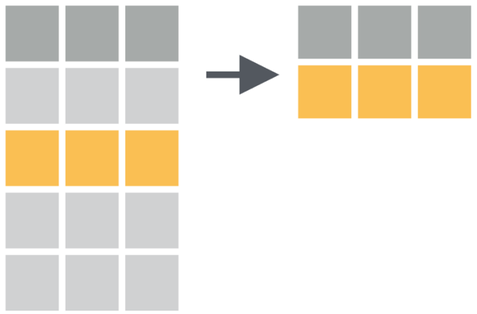
Yale SCHOOL OF PUBLIC HEALTH
Data Science and Data Equity

`dplyr` is a set of functions used to manipulate data.

`mutate()`	Add new variables that are functions of existing variables.
`select()`	Picks variables based on their names.
`filter()`	Picks entries based on conditional statements.
`glimpse()`	Basic data frame details.
`summarize()`	Reduces multiple values down to a summary metric.
`arrange()`	Changes the ordering of the rows.
`left/right_join()`	Combines tables by matching variables.
`group_by()`	Groups rows to prepare for calculations by groups.

dplyr Cheat Sheets. Select Examples are copied here. Accessed June 13th, 2025.

`filter()`



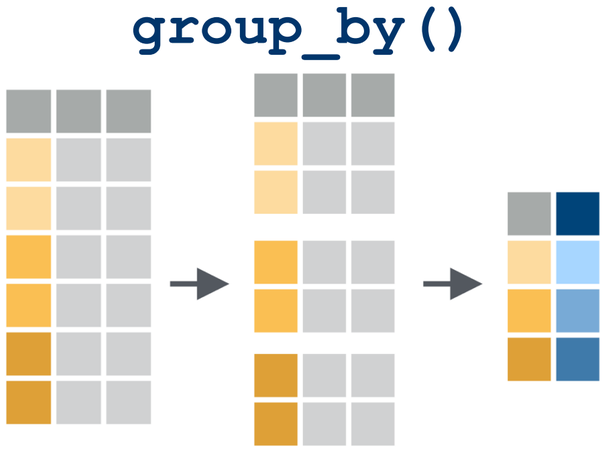
Select and extract rows that match a logical test.

```
df_filtered <- df_long |>
  filter(Combined_Key %in% str_c(c(datasets::state.name,
         "District of Columbia"), ", US"))
```

Combined_Key	Deaths_Count_Cumulative
Kentucky, US	4637
Monroe, Kentucky, US	38
Illinois, US	22735
Crawford, Illinois, US	23
Tennessee, US	11411

Combined_Key	Deaths_Count_Cumulative
Kentucky, US	4637
Illinois, US	22735
Tennessee, US	11411

dplyr Cheat Sheets. Select Examples are copied here. Accessed June 13th, 2025.



Create a "grouped" copy of a table, allowing manipulation of and calculation by each "group" separately.

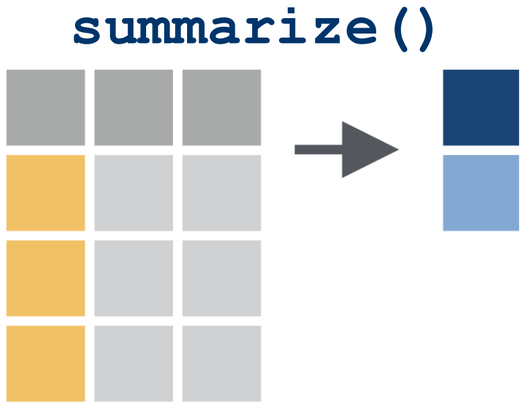
```
df_grouped <- df_filtered |> group_by(Date)
df_grouped |> tally() |>
    (\(x) list("Tally" = head(x), "Unique n" = unique(x$n)))()
```

$Tally

Date	n
<chr>	<int>
2020-01-01	51
2020-02-01	51
2020-03-01	51
2020-04-01	51

```
$Unique n
[1] 51
```


dplyr Cheat Sheets. Select Examples are copied here. Accessed June 13th, 2025.



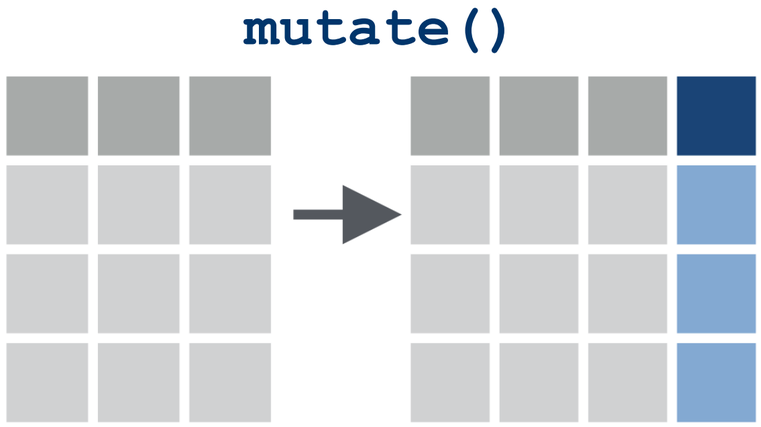
Reduce multiple values down to a single summary.

```
df_US <- df_grouped |>
  summarize(Deaths_Count_Cumulative = sum(Deaths_Count_Cumulative),
            .groups = "keep")
```

Combined_Key	Date	Deaths_Count_Cumulative
California, US	2021-10-01	71913
Maine, US	2021-10-01	1167
Maryland, US	2021-10-01	10895
New Mexico, US	2021-10-01	5049
South Dakota, US	2021-10-01	2235

Date	Deaths_Count_Cumulative
2021-09-01	693467
2021-10-01	741327
2021-11-01	775801
2021-12-01	821788
2022-01-01	884634

dplyr Cheat Sheets. Select Examples are copied here. Accessed June 13th, 2025.



Create new columns based on functions of existing columns or static values.

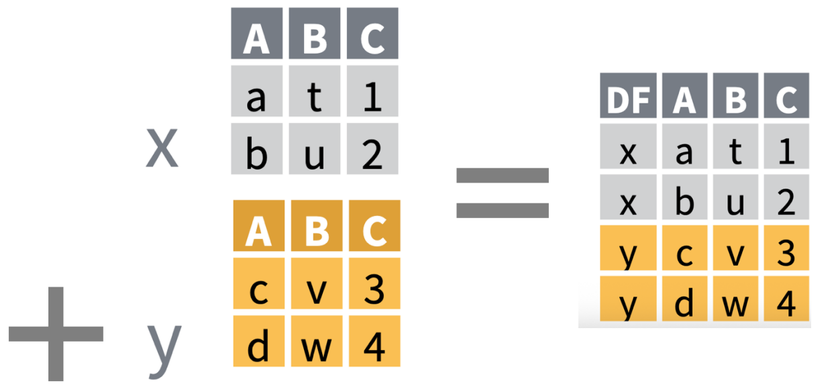
```
df_US <- df_US |>
  mutate(Combined_Key = "US", Country_Region = "US")
```

Date	Deaths_Count_Cumulative
2021-09-01	693467
2021-10-01	741327
2021-11-01	775801
2021-12-01	821788
2022-01-01	884634

Combined_Key	Date	Deaths_Count_Cumulative
US	2021-09-01	693467
US	2021-10-01	741327
US	2021-11-01	775801
US	2021-12-01	821788
US	2022-01-01	884634

dplyr Cheat Sheets. Select Examples are copied here. Accessed June 13th, 2025.

`bind_rows()`



Combine tables row-wise by matched column names while keeping all unique columns.

```
df <- bind_rows(df_US, df_long)
```

Combined_Key	Deaths_Count_Cumulative
US	945612

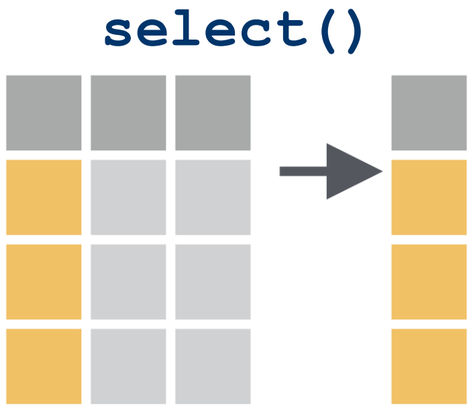
+

Combined_Key	Deaths_Count_Cumulative
Guam, US	322
Michigan, US	34505
Vermont, US	598

→

Combined_Key	Deaths_Count_Cumulative
US	945612
Guam, US	322
Michigan, US	34505
Vermont, US	598

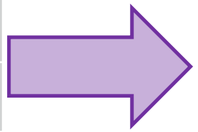
dplyr Cheat Sheets. Select Examples are copied here. Accessed June 13th, 2025.



Extract columns as a new table or reorganize columns.

```
df <- df |>
  select(Combined_Key, Province_State, Date)
```

Date	Combined_Key	Province_State
2021-10-01	US	NA
2021-11-01	Georgia, US	Georgia
2021-12-01	Puerto Rico, US	Puerto Rico
2021-01-01	Texas, US	Texas
2021-02-01	Utah, US	Utah



Combined_Key	Province_State	Date
US	NA	2021-10-01
Georgia, US	Georgia	2021-11-01
Puerto Rico, US	Puerto Rico	2021-12-01
Texas, US	Texas	2021-01-01
Utah, US	Utah	2021-02-01

dplyr Cheat Sheets. Select Examples are copied here. Accessed June 13th, 2025.

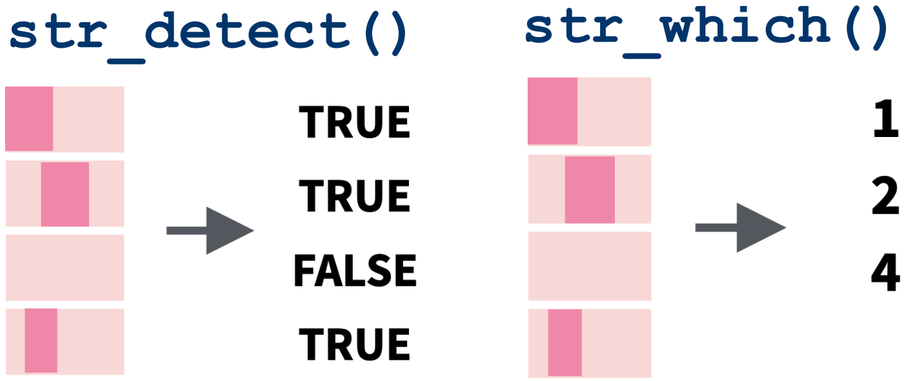
Introducing stringr

Yale SCHOOL OF PUBLIC HEALTH
Data Science and Data Equity

`stringr` contains a set of functions for manipulating and interpreting strings.

`str_c()`	Join discrete strings into one. Can specify spacers.
`str_detect()`	Find pattern match within strings.
`str_length()`	Counts the code points, or characters, in a string.
`str_replace()`	Replace the first match of a pattern in a string.
`str_lower()`	Convert strings to lower case.
`str_upper()`	Convert strings to upper case.
`str_title()`	Convert strings to tittle case.
`str_sort()`	Sorts the character vector.

stringr Cheat Sheets. Select Examples are copied here. Accessed November 15th, 2024.



Two functions that find rows where a string match is found. Reports as a Boolean or index.

```
df[!str_detect(df$Province_State, "Princess") %in% TRUE, ]
# OR
df[-str_which(df$Province_State, "Princess"), ]
```

Province_State	Deaths_Count_Cumulative
Connecticut	5995
Diamond Princess	0
Florida	21673
Georgia	10958
Grand Princess	3

Province_State	Deaths_Count_Cumulative
Connecticut	5995
Florida	21673
Georgia	10958

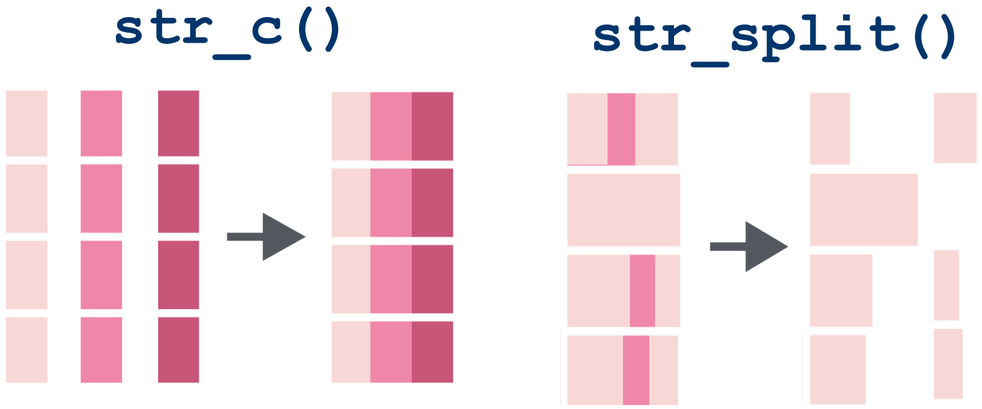
stringr Cheat Sheets. Select Examples are copied here. Accessed November 15th, 2024.

Discussion:

An alternative form of `str_which()` is to use its native negate setting:

```
str_which(df$Province_State, "Princess", negate = TRUE)
```

This works in many scenarios, but it's not appropriate here. Why?



Two functions that generate a new string by joining discrete strings or splitting a composite.

```r
new <- str_c(
  df_state_level$Province_State,
  df_state_level$Country_Region,
  sep = ", ")
```

```r
# OR
```

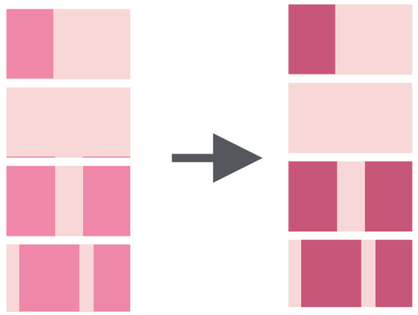
```r
new <- str_split(
  df_state_level$Combined_Key,
  ",", simplify = TRUE,
  n = 2)[, 2]
```

Province_State	Country_Region	new
Montana	US	Montana, US
Oregon	US	Oregon, US
Hawaii	US	Hawaii, US

Combined_Key	new
Cheyenne, Colorado, US	Colorado, US
McMinn, Tennessee, US	Tennessee, US
Accomack, Virginia, US	Virginia, US

stringr Cheat Sheets. Select Examples are copied here. Accessed November 15th, 2024.

`str_replace_all()`



Find string matches and replace those substrings with a new one.

```
df_filtered[index, c("Province_State", "Combined_Key")] |>
  (\(x) {
    sapply(x, function(y)
      str_replace(y, "^Virgin Islands", "Virgin Islands"))
  }) ()
```

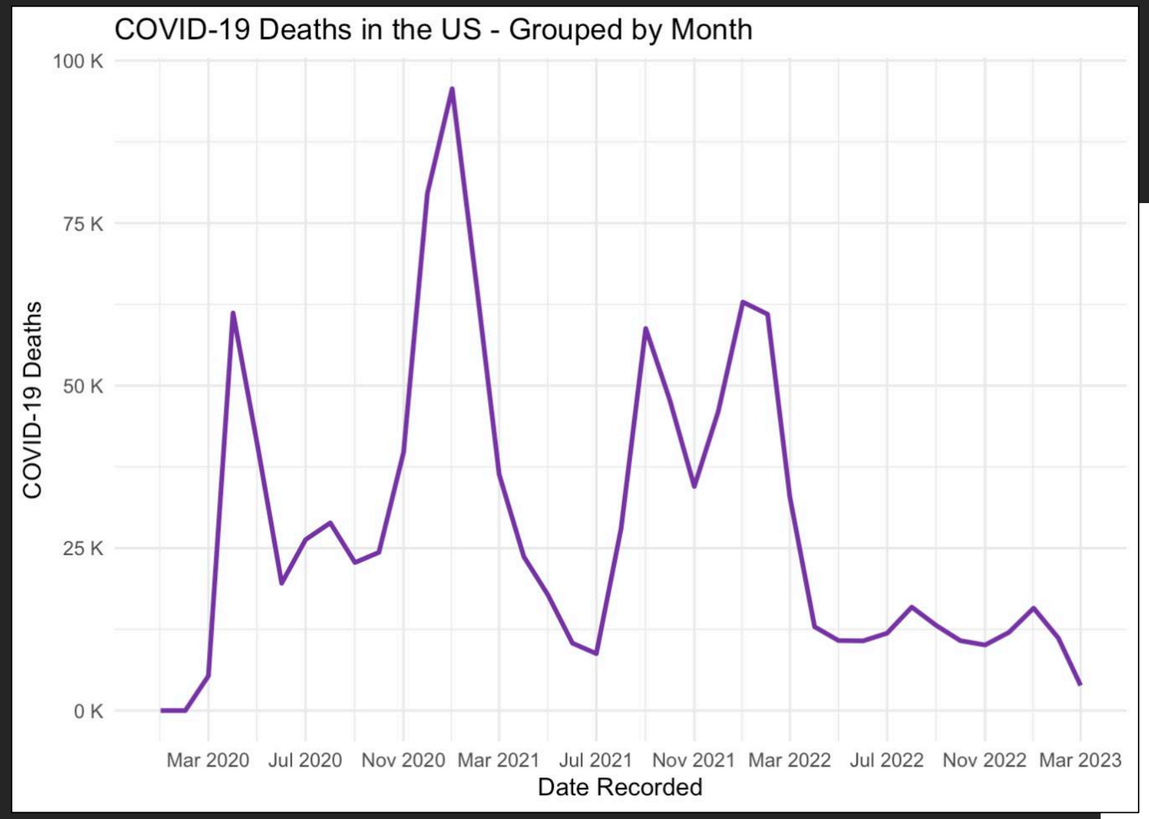
Province_State	Combined_Key
Virginia	Virginia, US
Virgin Islands	Virgin Islands, US
Texas	Texas, US

Province_State	Combined_Key
Virginia	Virginia, US
US Virgin Islands	US Virgin Islands, US
Texas	Texas, US

stringr Cheat Sheets. Select Examples are copied here. Accessed November 15th, 2024.

```
plot_deaths <- df |>
  filter(Combined_Key == "US") |>
  ggplot(aes(x = Date, y = Deaths_Count_Monthly) +
    geom_line(color = "#7634A6", linewidth = 1) +
    ...
    labs(x = "Date Recorded", y = "COVID-19 Deaths" ,
         title = "COVID-19 Deaths in the US - Grouped by Month") +
    theme_minimal()

plot_deaths
```



Discussion:

Work with your neighbors to answer the two questions in “Questions.R.”

Appendix

Glossary

Import Data Loading data from a stored file, database, or application programming interface (API) into the R environment.

Tidy Data Formatting data into a consistent structure without anomalies. Each column represents a variable, and each row represents an observation.

Transform Data The process of converting raw, inconsistent, or unstructured data into a clean, standardized, and actionable format suitable for storage, analysis, or reporting.

Glossary

Wrangle Data The process of tidying and transforming data.

Anonymous Function A temporary function that is not stored permanently. It is often used for short-term operations where defining a full function is unnecessary.

References

Slide 4

1. B. Evans, “Tidying Data,” Yale Center for Research Computing (YCRC). Accessed: Feb. 15, 2026. [Online]. Available: https://www.youtube.com/watch?v=jZPWFf_rrxc
2. H. Wickham, M. Çetinkaya-Rundel, and G. Grolemund, “Learn the tidyverse,” Tidyverse. Accessed: Nov. 15, 2024. [Online]. Available: https://www.tidyverse.org/learn/
3. H. Wickham, R. François, L. Henry, K. Müller, and D. Vaughan, “A Grammar of Data Manipulation • dplyr,” Tidyverse. Accessed: Nov. 15, 2024. [Online]. Available: https://dplyr.tidyverse.org/
4. H. Wickham, D. Vaughan, and M. Girlich, “Tidy Messy Data • tidyr,” Tidyverse. Accessed: Nov. 15, 2024. [Online]. Available: https://tidyr.tidyverse.org/
5. H. Wickham, “Simple, Consistent Wrappers for Common String Operations • stringr,” Tidyverse. Accessed: Nov. 15, 2024. [Online]. Available: https://stringr.tidyverse.org/

References

Slide 6

1. H. Wickham, M. Çetinkaya-Rundel, and G. Grolemund, R for Data Science (2e). O'Reilly Media. Accessed: Nov. 13, 2024. [Online]. Available: https://r4ds.hadley.nz/intro
2. "Tidyverse," Wikipedia. Accessed: Nov. 13, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Tidyverse

Slide 7

1. H. Wickham, M. Çetinkaya-Rundel, and G. Grolemund, R for Data Science (2e). O'Reilly Media. Accessed: Nov. 13, 2024. [Online]. Available: https://r4ds.hadley.nz/intro
2. H. Wickham, "Tidyverse." Accessed: Nov. 14, 2024. [Online]. Available: https://www.tidyverse.org/
3. H. Wickham *et al.*, "Welcome to the Tidyverse," *J Open Source Softw*, vol. 4, no. 43, p. 1686, Nov. 2019, doi: 10.21105/JOSS.01686.

References

Slide 8

1. Committee of Presidents of Statistical Societies, "2019 - Committee of Presidents of Statistical Societies (COPSS)." Accessed: Nov. 16, 2024. [Online]. Available: https://community.amstat.org/copss/awards/presidents/2018151
2. Hadley Wickham, "Personal Website." Accessed: Feb. 15, 2026. [Online]. Available: https://hadley.nz/
3. "Tidyverse," Wikipedia. Accessed: Nov. 13, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Tidyverse
4. "Hadley Wickham," Wikipedia. Accessed: Nov. 13, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Hadley_Wickham

Slide 9

1. H. Wickham, M. Çetinkaya-Rundel, and G. Grolemund, R for Data Science (2e). O'Reilly Media. Accessed: Nov. 13, 2024. [Online]. Available: https://r4ds.hadley.nz/intro

References

Slide 9 continued

2. Committee of Presidents of Statistical Societies, "2019 - Committee of Presidents of Statistical Societies (COPSS)." Accessed: Nov. 16, 2024. [Online]. Available: https://community.amstat.org/copss/awards/presidents/2018151

Slides 11-12

1. H. Wickham, M. Çetinkaya-Rundel, and G. Grolemund, R for Data Science (2e). O'Reilly Media. Accessed: Nov. 13, 2024. [Online]. Available: https://r4ds.hadley.nz/intro

Slide 12

1. H. Wickham, D. Vaughan, and M. Girlich, "Tidy Messy Data • tidyr," Tidyverse. Accessed: Nov. 15, 2024. [Online]. Available: https://tidyr.tidyverse.org/

2. M. Çetinkaya-Rundel, "tidyr Cheat Sheets," GitHub. Accessed: Feb. 15, 2026. [Online]. Available: https://github.com/rstudio/cheatsheets/blob/main/tidyr.pdf

References

Slides 14-15

1. H. Wickham et al., "Welcome to the Tidyverse," J Open Source Softw, vol. 4, no. 43, p. 1686, Nov. 2019, doi: 10.21105/JOSS.01686.

Slide 15

1. H. Wickham, D. Vaughan, and M. Girlich, "Tidy Messy Data • tidyr," Tidyverse. Accessed: Nov. 15, 2024. [Online]. Available: https://tidyr.tidyverse.org/
2. M. Çetinkaya-Rundel, "tidyr Cheat Sheets," GitHub. Accessed: Feb. 15, 2026. [Online]. Available: https://github.com/rstudio/cheatsheets/blob/main/tidyr.pdf

Slides 17-23

1. H. Wickham, R. François, L. Henry, K. Müller, and D. Vaughan, "A Grammar of Data Manipulation • dplyr," Tidyverse. Accessed: Nov. 15, 2024. [Online]. Available: https://dplyr.tidyverse.org/
2. H. Wickham, M. Çetinkaya-Rundel, and G. Grolemund, R for Data Science (2e). O'Reilly Media. Accessed: Nov. 13, 2024. [Online]. Available: https://r4ds.hadley.nz/intro

References

Slides 17-23 continued

3. H. Wickham, M. Çetinkaya-Rundel, and G. Grolemund, "3 Data transformation – R for Data Science (2e)," in R for Data Science (2e), O'Reilly Media, ch. 14. Accessed: Feb. 15, 2026. [Online]. Available: https://r4ds.hadley.nz/data-transform

4. M. Çetinkaya-Rundel, "dplyr Cheat Sheets," GitHub. Accessed: Feb. 15, 2026. [Online]. Available: https://github.com/rstudio/cheatsheets/blob/main/data-transformation.pdf

Slides 25-26

1. H. Wickham, "Simple, Consistent Wrappers for Common String Operations • stringr," Tidyverse. Accessed: Nov. 15, 2024. [Online]. Available: https://stringr.tidyverse.org/

2. H. Wickham and L. Vaudor, "stringr Cheat Sheet," GitHub. Accessed: Nov. 15, 2024. [Online]. Available: https://github.com/rstudio/cheatsheets/blob/main/strings.pdf

3. H. Wickham, M. Çetinkaya-Rundel, and G. Grolemund, R for Data Science (2e). O'Reilly Media. Accessed: Nov. 13, 2024. [Online]. Available: https://r4ds.hadley.nz/intro

References

Slides 25-26 continued

4. H. Wickham, “Function reference • stringr,” Tidyverse. Accessed: Feb. 15, 2026. [Online]. Available: https://stringr.tidyverse.org/reference/index.html

Slides 28-29

1. H. Wickham, “Simple, Consistent Wrappers for Common String Operations • stringr,” Tidyverse. Accessed: Nov. 15, 2024. [Online]. Available: https://stringr.tidyverse.org/

2. H. Wickham and L. Vaudor, “stringr Cheat Sheet,” GitHub. Accessed: Nov. 15, 2024. [Online]. Available: https://github.com/rstudio/cheatsheets/blob/main/strings.pdf

3. H. Wickham, M. Çetinkaya-Rundel, and G. Grolemund, R for Data Science (2e). O’Reilly Media. Accessed: Nov. 13, 2024. [Online]. Available: https://r4ds.hadley.nz/intro

4. H. Wickham, “Function reference • stringr,” Tidyverse. Accessed: Feb. 15, 2026. [Online]. Available: https://stringr.tidyverse.org/reference/index.html

References

Glossary

1. H. Wickham, R. François, L. Henry, K. Müller, and D. Vaughan, “A Grammar of Data Manipulation • dplyr,” Tidyverse. Accessed: Nov. 15, 2024. [Online]. Available: https://dplyr.tidyverse.org/
2. H. Wickham, D. Vaughan, and M. Girlich, “Tidy Messy Data • tidyr,” Tidyverse. Accessed: Nov. 15, 2024. [Online]. Available: https://tidyr.tidyverse.org/
3. H. Wickham, “Simple, Consistent Wrappers for Common String Operations • stringr,” Tidyverse. Accessed: Nov. 15, 2024. [Online]. Available: https://stringr.tidyverse.org/

ysph.yale.edu
sph.yale.edu/dsde

@YaleSPH

Data Science and Data Equity
Yale School of Public Health
60 College Street, New Haven, CT 06510

Yale SCHOOL OF PUBLIC HEALTH