Introduction to R Session 2

Department of Sociology | University of Oxford

Casey Breen

2024-10-15

Intro to R – Session 2

- Reminder: course materials available from:
  - www.github.com/caseybreen/intro_r
- Questions on problem set 1?

Review of session 1

- What's the difference between `R` and `RStudio`?
- What's a `vector`? What's a `data.frame`?
- What does the `$` operator do? What does `data$column_b` do?
- What are two different data structures? What are three different data types?
- What does `%in%` operator do?
- What does the `!` operator do?

Session 2

- **Module 4**: Importing and exporting in data
- **Module 5**: Data manipulation (`dplyr`) and data visualization (`ggplot2`)
- **Module 6**: Best practices for R coding and resources for self-study

Module 4

Importing and exporting data

Learning objectives

- Common data formats
- Functions for importing / exporting data
- Types of file paths in R

Importing data

- Common formats for data
  - .csv, .xlsx, .txt, .dat (stata), etc.
- Key functions
  - `read_csv()` function from `tidyverse`: Read CSV files
    - Also built-in ("base") function: `read.csv()`
  - `read.table()`: Read text files
  - `readxl::read_excel()`: Read Excel files

```
## read in CSV file
df <- read_csv("/path/to/your/data.csv") ## faster

## read in stata file
library(haven)
data <- read_dta("path/to/file.dta")
```

File paths

- **Absolute Path**: Specifies the full path locate a file or directory, starting with the root directory.
  - Windows: `"C:\Users\username\folder\file.csv"`
  - macOS/Linux: `"/home/username/folder/file.csv"`
- **Relative Path**: Specifies how to find the file or directory based on the current working directory.
  - `folder/file.csv`

Working directories

- The working directory is the folder where your R session or script looks for files to read, or where it saves files you write
- Commands like `read_csv("file.csv")` or `write_csv(data, "file.csv")` will read from or write to this directory by default
- Key syntax:
  - `getwd()` — returns working directory
  - `setwd("/path/to/folder")` — sets working directory

```
1 getwd()
```

```
[1] "/Users/caseybreen/workspace/teaching/intro_r/slides"
```

Reading in .CSV files

- Recap: to read in .csv files use `read_csv()` function from `tidyverse`
  - This will read in the .csv file into memory as a `data frame`

```
1 library(tidyverse)
2 df <- read_csv("dataset.csv")
```

- Write out a `data frame` to a .csv file using `write_csv()`:

```
1 write_csv(df, "dataset_v2.csv")
```

Downloading data for exercises

- We will be using the CenSoc Numident Demo dataset
- Please download the .csv file from the course website (`intro_r/data`)
  - https://github.com/caseybreen/intro_r
- Short url: https://tinyurl.com/intro-r-data

Live coding demo

- Downloading demo file from Github
- Reading in a .csv file in `R` using `read_csv()`
  - Absolute and relative paths
- Using `tab` to auto-complete file paths
- Exploring a `data frame`: number of columns, rows, column names, etc.

In-class exercise 1

1. Load and install the `tidyverse` packages using the commands `install.packages()` and `library()`
2. Use the `read_csv()` function to read in the downloaded dataset and assign it to the object `censoc`
3. Use the `head` command to look at the first 5 rows
4. How many columns are in the dataset?
5. How many rows are in the dataset?
6. List the column names. What are a few research questions that could be addressed using this dataset?

Exercise 1 solutions

1. Load and install the `tidyverse` packages using the commands `install.packages()` and `library()`

```
install.packages(tidyverse) ## only have to do this once
library(tidyverse)
```

2. Use the `read_csv()` function to read in the dataset and assign it to the object `censoc`

```
censoc <- read_csv("censoc_numident_demo_v2.1.csv")
```

3. Use the `head()` command to look at the first 5 rows

```
head(censoc)
```

4. How many columns are in the dataset?

```
ncol(censoc)
```

```
[1] 39
```

Exercise 1 solutions (cont.)

5. How many rows are in the dataset?

```
1 nrow(censoc)
```

```
[1] 85865
```

6. List the column names.

```
1 colnames <- names(censoc)
2 head(colnames)
```

```
[1] "histid"    "byear"     "bmonth"    "dyear"     "dmonth"    "death_age"
```

Module 5

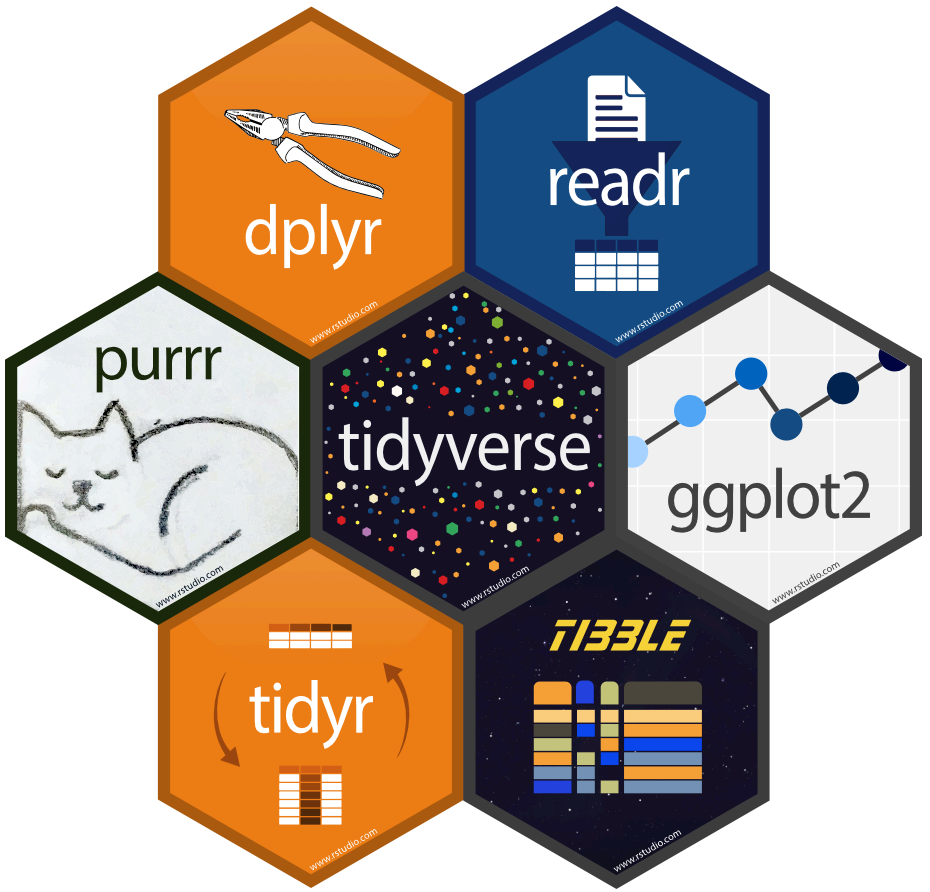
Data manipulation and visualization

Learning objectives

- Overview of `tidyverse` suite of packages
- Fundamentals of data manipulation with `dplyr`
- Data visualization with `ggplot`

Tidyverse

- Packages: Collection of R packages designed for data science.
- Data manipulation: Simplifies data cleaning and transformation with `dplyr`.
- Data Visualization: Enables advanced plotting with `ggplot2`.



Data Manipulation using `dplyr`

`filter`: Select rows based on conditions.

```
filtered_df <- filter(df, age > 21)
```

`select`: Choose specific columns

```
filtered_df <- select(df)
```

`mutate`: Add or modify columns

```
df <- mutate(df, age_next_year = age + 1)
```

`summarize` or `summarise`: Aggregate or summarize data based on some criteria

```
filtered_df <- summarize(df, mean(age))
```

`group_by`: Group data by variables. Often used with `summarise()`.

```
filtered_df <- df %>%
  group_by(gender) %>%
  summarize(mean(age))
```

The Pipe Operator %>% (or |>) in R

- Takes the output of one function and passes it as the first argument to another function
  - "And then do…"
- What's the below code doing?

```
filtered_df <- df %>%
  group_by(gender) %>%
  summarize(mean(age))
```

Recoding values in R

- Sometime you want to recode a variable to take different values (e.g., recoding exact income to binary high/low income variable)
- The `case_when()` function in `R` is part of the `dplyr` package and is used for creating new variables based on multiple conditions:

```
df_new <- df %>%
  mutate(new_var = case_when(
  condition1 ~ value1,
  condition2 ~ value2,
  TRUE ~ value_otherwise
))
```

Live coding demo

- Filter data
- Selecting data
- Calculating summary statistics by group
- Creating and recoding variables

In-class exercise 2

1. Filter the `censoc` data.frame to include only women (`sex == 2`). Use the `filter` command.
2. Filter the `censoc` data.frame to include only people born between 1905 and 1920 using the `byear` variable.
3. Select the columns `histid`, `death_age`, `sex`, and `ownershp`
4. Calculate the average age of death for women (hint: refer to question 1)

Exercise 2 solutions

1. Filter the `censoc` data.frame to include only women (sex == 2). Use the `filter` command.

```
## filter to only include women
censoc %>%
  filter(sex == 2)
```

2. Filter the `censoc` data.frame to include only people born between 1905 and 1920 using the `byear` variable.

```
## method 1
censoc %>%
  filter(byear %in% 1905:1920)

## method 2
censoc %>%
  filter(byear >= 1905 & byear <= 1920)
```

Exercise 2 solutions (cont.)

3. Select the columns `histid`, `death_age`, `sex`, and `ownershp`

```r
censoc_select <- censoc %>%
  select(histid, death_age, sex, ownershp)

head(censoc_select)
```

```
# A tibble: 6 × 4
  histid                               death_age   sex ownershp
  <chr>                                    <dbl> <dbl>    <dbl>
1 235C4FA2-B407-4E61-A31D-DBF299C1C120        85     1        1
2 0DE161A7-34A7-47EA-B053-EA8549172CCC        77     1        1
3 EFF79CEC-DA83-482A-AB9A-FFCAC3C9A6A5        77     1        1
4 B51D01FA-54A4-4E5E-8BCF-B6D9521A2983        73     2        2
5 D545AEB1-C5C3-4E32-BB22-4BF58CF50311        73     1        2
6 A71A537B-C440-4E85-A276-334B05B723A7        82     2        1
```

4. Calculate the average age of death for women (hint: refer to question 1)

```r
censoc %>%
  filter(sex == 2) %>%
  summarize(mean_death_age_women = mean(death_age))
```

```
# A tibble: 1 × 1
  mean_death_age_women
                 <dbl>
1                 78.2
```

Data visualization using ggplot

- `ggplot2` provides a powerful and flexible system for creating a variety of data visualizations
- `data`: specifies the dataset to be used for the plot
- `aes`: Defines what data to show
- `geoms`: Chooses the type of plot (e.g., histogram)

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

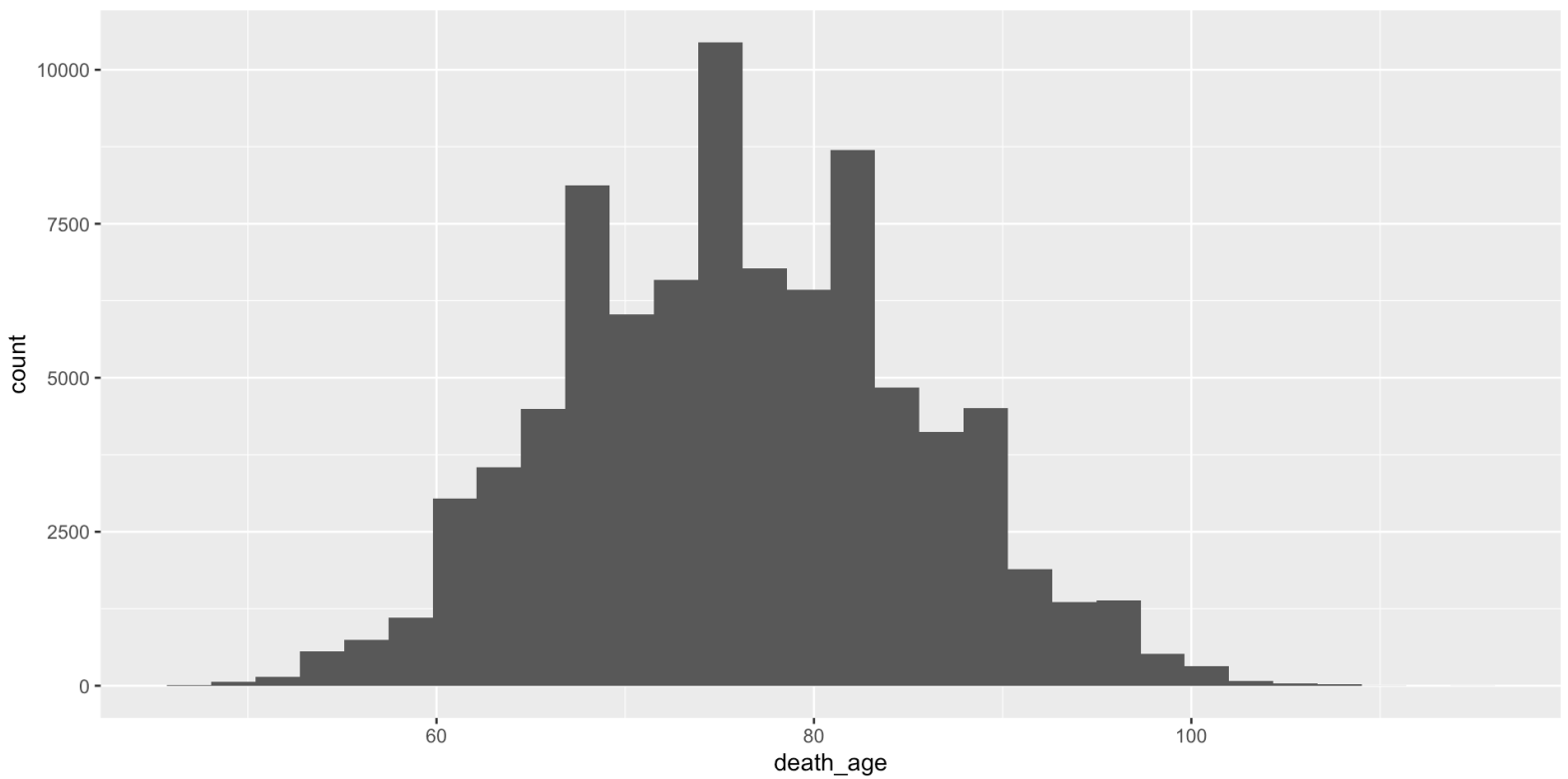
Types of plots

- `geom_point()`: Scatter plot
- `geom_bar()`: Bar chart
- `geom_histogram()`: Histogram

Basic histogram example

- Histogram of age of death in censoc dataset

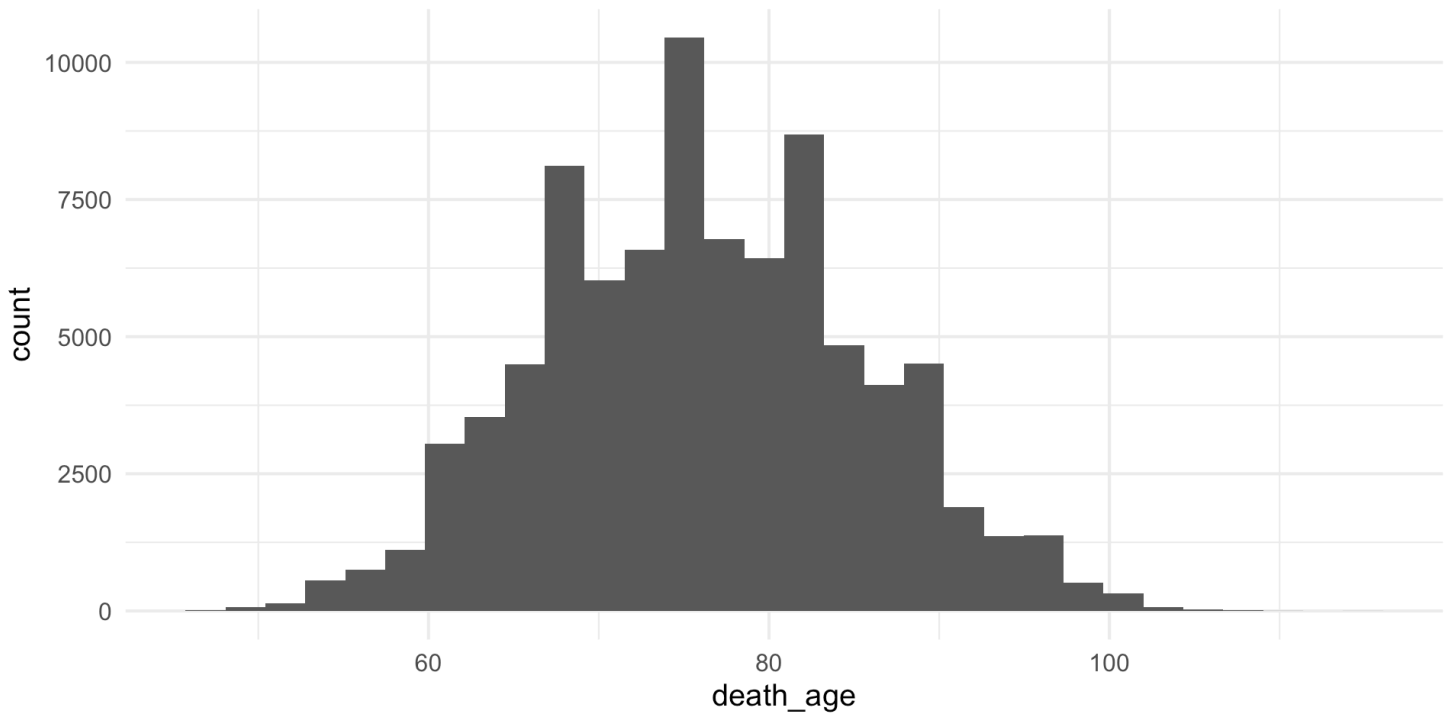
```
1  ggplot(data = censoc) +
2    geom_histogram(aes(x = death_age))
```



Customisable – specify theme

- `+ theme(<theme_choice>)` will add on a theme

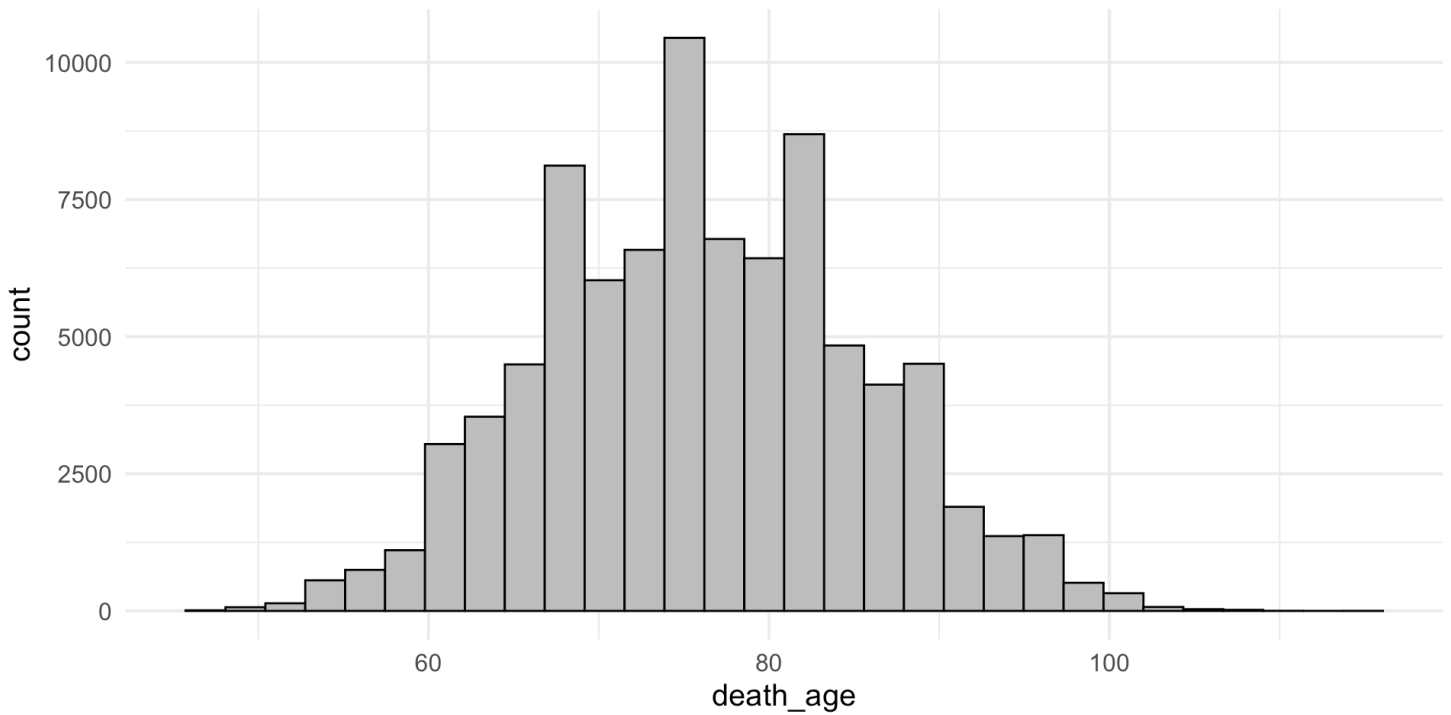
```
1  ggplot(data = censoc) +
2    geom_histogram(aes(x = death_age)) +
3    theme_minimal(base_size = 15)
```



Customisable – specify colors

- `color` and `fill` will can change color / fill of plot

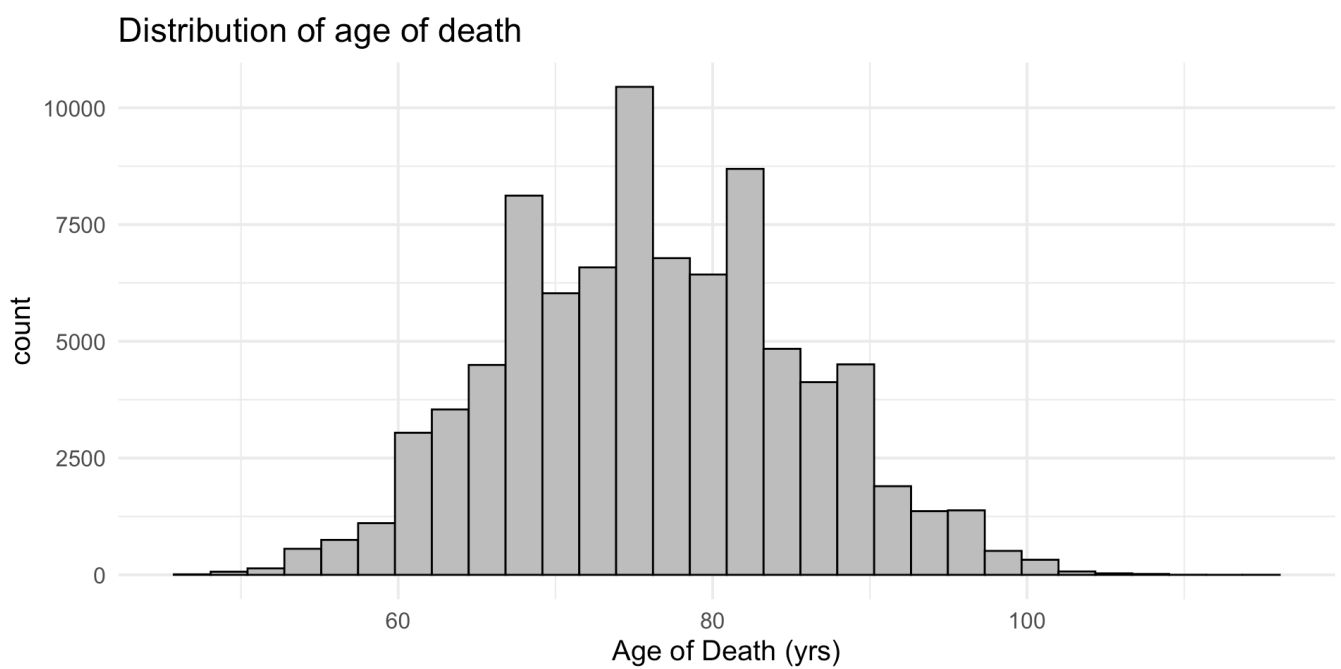
```
1  ggplot(data = censoc) +
2    geom_histogram(aes(x = death_age), color = "black", fill = "grey") +
3    theme_minimal(base_size = 15)
```



Customisable – add on labels/title

- `+ labs()` add on title/axis labels

```
1  ggplot(data = censoc) +
2    geom_histogram(aes(x = death_age), color = "black", fill = "grey") +
3    theme_minimal(base_size = 15) +
4    labs(title = "Distribution of age of death", x = "Age of Death (yrs)")
```



Live coding demo

- Create histogram using `ggplot`
- Demonstrate flexibility of `ggplot`
  - Themes
  - Axis labels, titles
  - Colors

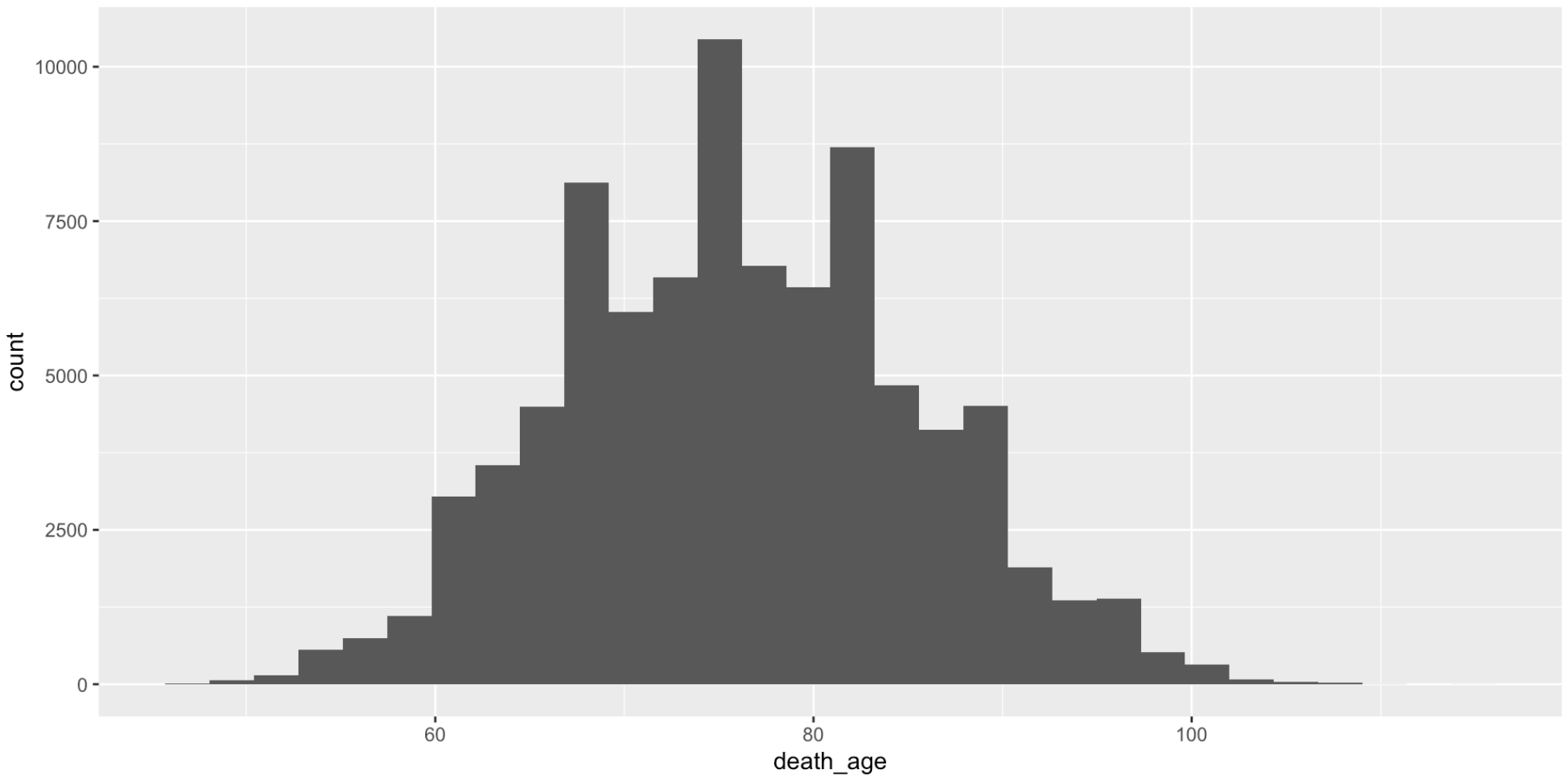
In-class exercise 3

1. Make a histogram of the variable `death_age`. When are most people dying?
2. Make a histogram of the variable `byear`. When are most people born?
3. Recode the variable `sex` from numeric values (1, 2) to take character values ("men" and "women"). Note that 1 = men, 2 = women.
4. Calculate the mean of of death for both men and women using `group_by()` and `summarize()`. Use the `death_age` variable. Do men or women live longer in this sample?
5. Make a histogram of the variable `death_age` for both men and women. Use the `filter()` command.
6. Now try adding the following line to the histogram you made in question 1: `+ facet_wrap(~sex)`

Exercise 3 solutions

1. Make a histogram of the variable `death_age`. When are most people dying?

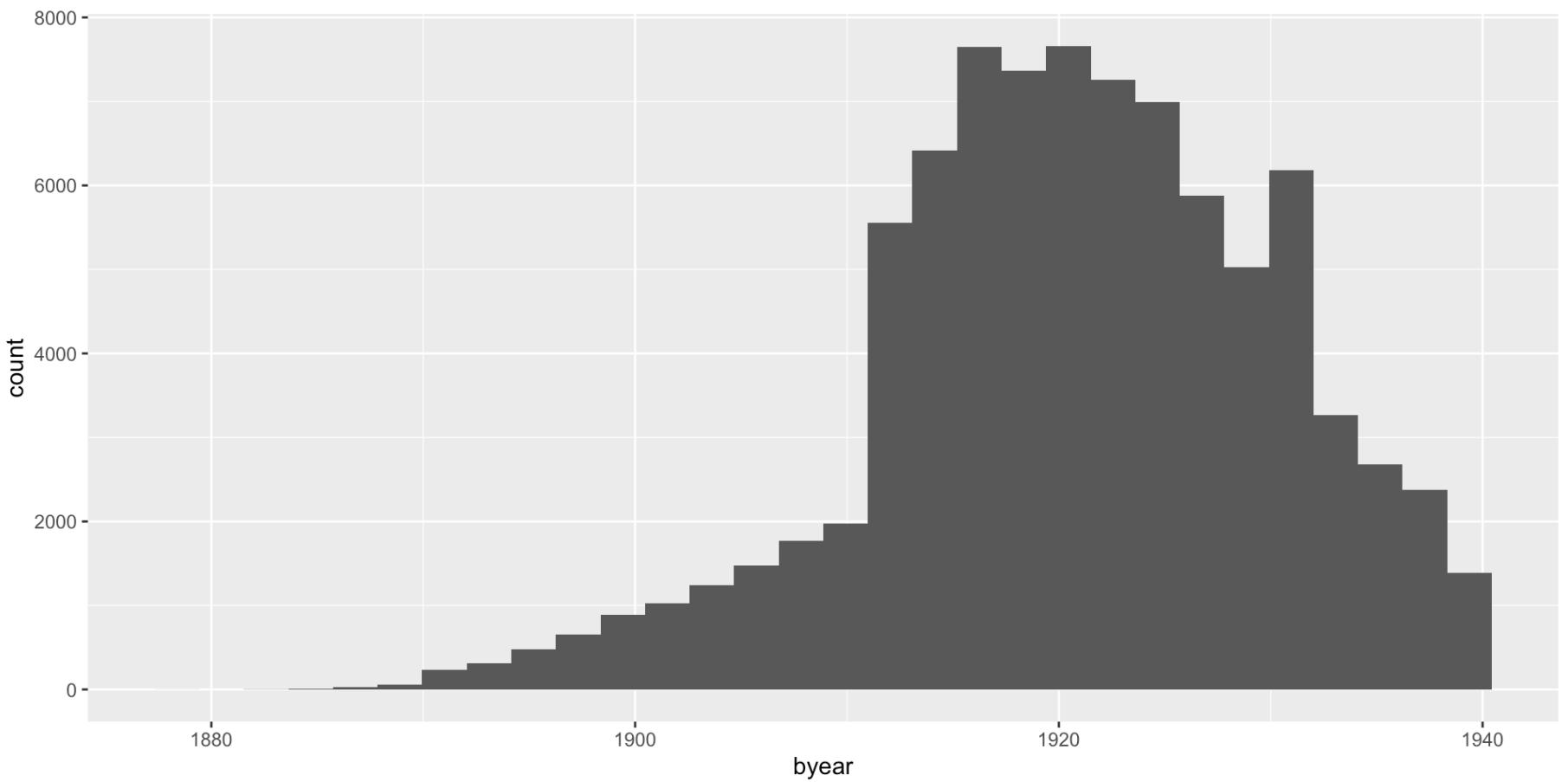
```
ggplot(data = censoc) +
  geom_histogram(aes(x = death_age))
```



Exercise 3 solutions (cont.)

2. Make a histogram of the variable `byear`. When are most people born?

```
ggplot(data = censoc) +
  geom_histogram(aes(x = byear))
```



Exercise 3 solutions (cont.)

3. Recode the variable `sex` from numeric values (1, 2) to take character values ("men" and "women"). Note that 1 = men, 2 = women.

```
## recode sex
censoc <- censoc %>%
  mutate(sex_recode = case_when(
    sex == 1 ~ "men",
    sex == 2 ~ "women"
  ))

## look at first few rows to check our recode worked
censoc %>%
  select(sex, sex_recode) %>%
  head()
```

```
# A tibble: 6 × 2
    sex sex_recode
  <dbl> <chr>
1     1 men
2     1 men
3     1 men
4     2 women
5     1 men
6     2 women
```

Exercise 3 solutions (cont.)

4. Calculate the mean of of death for both men and women using `group_by()` and `summarize()`. Do men or women live longer?

```
censoc %>%
  group_by(sex_recode) %>%
  summarize(mean(death_age))
```

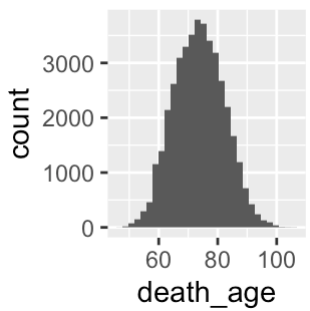
```
# A tibble: 2 × 2
  sex_recode `mean(death_age)`
  <chr>                  <dbl>
1 men                     73.9
2 women                   78.2
```

Exercise 3 solutions (cont.)

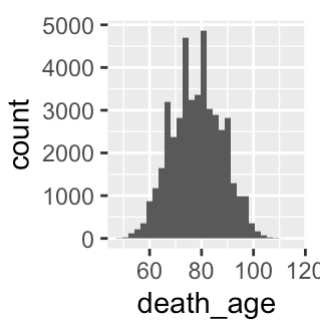
5. Make a histogram of the variable `death_age` for both men and women.

```
censoc_men <- censoc %>% filter(sex_recode == "men")
censoc_women <- censoc %>% filter(sex_recode == "women")

ggplot(data = censoc_men) + ## histogram for men
  geom_histogram(aes(x = death_age))
```



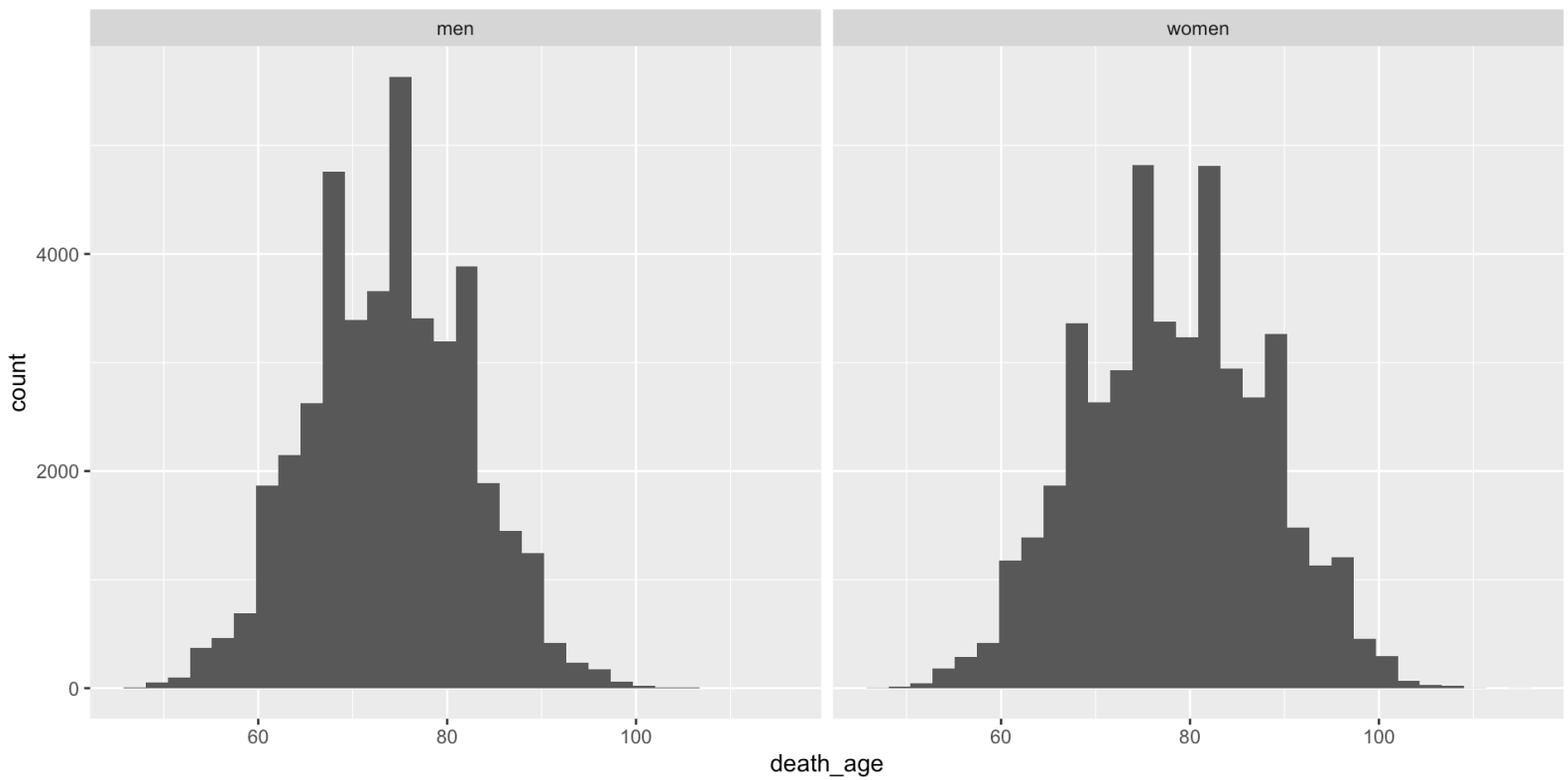

```
ggplot(data = censoc_women) + ## histogram for women
  geom_histogram(aes(x = death_age))
```



Exercise 3 solutions (cont.)

6. Now try adding the following line to the histogram you made in question 1: `+ facet_wrap(~sex)`

```
1  ggplot(data = censoc) +
2    geom_histogram(aes(x = death_age)) +
3    facet_wrap(~sex_recode)
```



Module 6

Best practices and resources for self-study

Learning objectives

- Best practices for writing and documenting code
- Where to go when you’re stuck
- Resources for learning more

Best practices (opinionated)

- **Style**: use descriptive names and “snake_case”
- **Documentation**: Start commenting your code early, it’s a good habit for the future
- **Learn `tidyverse`**: offers a more coherent syntax and is widely used in data science
- **Advanced topics**: R Projects, github integration, etc

When you're stuck

- Google
  - Lots of packages have documentation available online
  - Stack overflow – excellent resource
- Use help syntax (e.g., `?dplyr`)
- GPT4 (decent, but be careful!)

Resources for learning more

1. R for data science (https://r4ds.hadley.nz/)
2. Data visualization: a practical introduction (https://socviz.co/)

In-class exercise 4

Do homeowners in the United States live longer than renters in the United States?

1. Using the `censoc` data.frame, create a new data.frame `censoc_homeownership` that filters out any "missing" value for the `ownershp` variable (missing = 0). Use the `filter()` command.
2. In the `censoc_homeownership` data.frame, create a new variable `homeowner` using the `mutate()` command and the `case_when()` command. Assign this new variable `homeowner` a value of "own" if `ownershp == 1` and a value of "rent" if `ownershp == 2`.
3. Make a histogram on the age of death for "homeowner" and "renter" groups using `ggplot` using the `censoc_homeownership` data.frame. Use the `+ facet_wrap(~homeowner)` command.
4. Calculate the average age of death for "homeowner" and "renter" groups. Which group lives longer, on average? Use the `group_by()` and `summarize()` functions. What are some possible explanations for homeowners living longer than renters in the US?

Exercise 4 solution

Do homeowners in the United States live longer than renters in the United States?

1. Using the `censoc` data.frame, create a new data.frame `censoc_homeownership` that filters out any "missing" value for the `ownershp` variable (missing = 0). Use the `filter()` command.

```
censoc_homeownership <- censoc %>%
  filter(ownershp != 0)
```

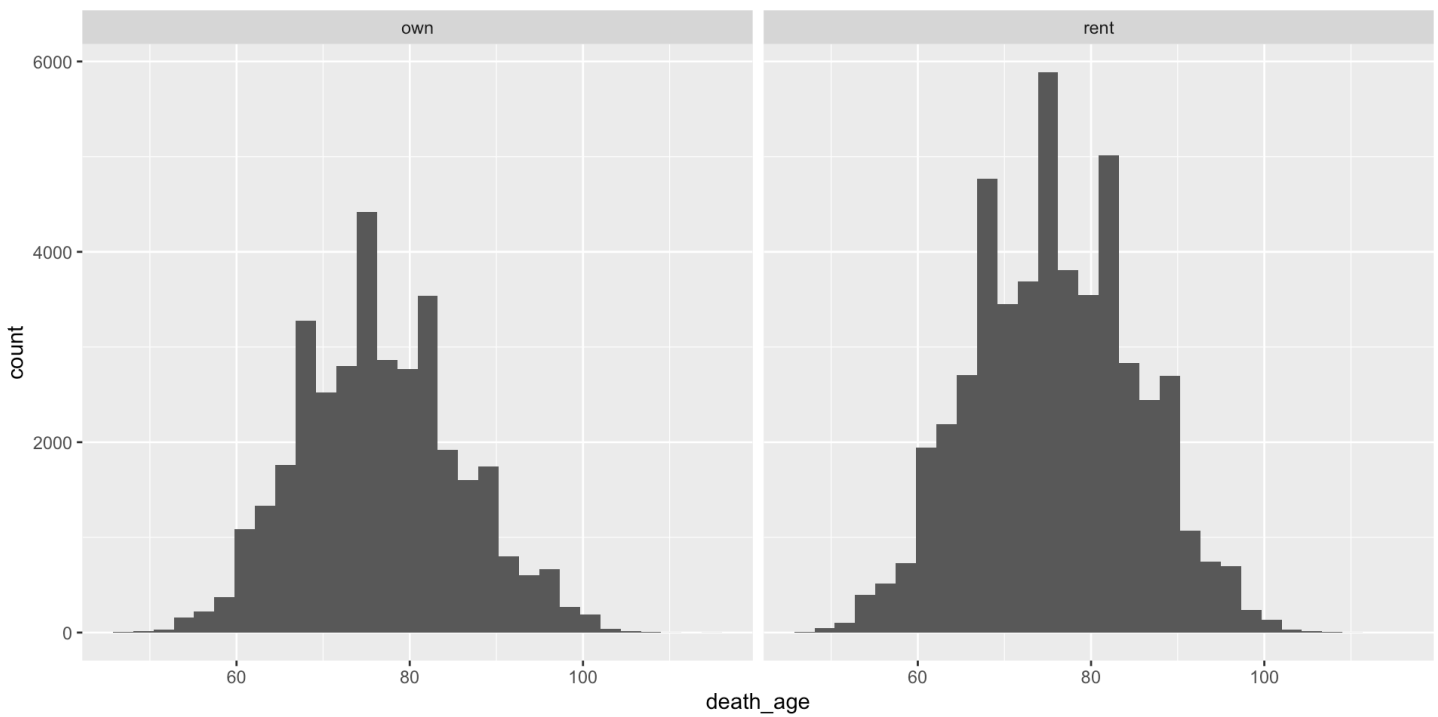
2. In the `censoc_homeownership` data.frame, create a new variable `homeowner` using the `mutate()` command and the `case_when()` command. Assign this new variable `homeowner` a value of "own" if `ownershp == 1` and a value of "rent" if `ownershp == 2`.

```
## create new homeowner variable
censoc_homeownership <- censoc_homeownership %>%
  mutate(homeowner = case_when(
    ownershp == 1 ~ "own",
    ownershp == 2 ~ "rent"
  ))
```

Exercise 4 solution (cont.)

3. Make a histogram on the age of death for “homeowner” and “renter” groups using `ggplot` using the `censoc_homeownership` data.frame. Use the `+ facet_wrap(~homeowner)` command.

```
1  ggplot(data = censoc_homeownership) +
2    geom_histogram(aes(x = death_age)) +
3    facet_wrap(~homeowner)
```



Exercise 4 solution (cont.)

4. Calculate the average age of death for “homeowner” and “renter” groups. Which group lives longer, on average? Use the `group_by()` and `summarize()` functions What are some possible explanations for homeowners living longer than renters in the US?

```
censoc_homeownership %>%
  group_by(homeowner) %>%
  summarize(mean(death_age))
```

```
# A tibble: 2 × 2
  homeowner `mean(death_age)`
  <chr>                 <dbl>
1 own                    76.5
2 rent                   75.8
```

Thank you

- Course materials available from:
  - www.github.com/caseybreen/intro_r
- Please independently complete all exercises in problem set 2 (and review solutions)
- Questions: casey.breen@demography.ox.ac.uk