HW 3

AUTHOR
Brandon Keck

```
library(tidyverse)
```

```
Warning: package 'ggplot2' was built under R version 4.2.3

Warning: package 'readr' was built under R version 4.2.3

Warning: package 'dplyr' was built under R version 4.2.3
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.0
✔ ggplot2   3.5.1     ✔ tibble    3.2.1
✔ lubridate 1.9.3     ✔ tidyr     1.3.0
✔ purrr     1.0.1
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
```

```
library(dplyr)
library(ggplot2)
library(forcats)
```

Exercise 1

(a) Use the read_csv() function to read the tech_stock.csv

```
library(readr)
tech_stock <- read_csv("tech_stock.csv")
```

```
Rows: 756 Columns: 4
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr  (1): company
dbl  (2): high, low
date (1): date

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#View(tech_stock)
```

What are the dimensions of the data frame (i.e., number of rows and columns)?

```
dim(tech_stock)
```

```
[1] 756   4
```

There are 756 rows and 4 columns in the data frame.

What are the data types for the columns?

```
glimpse(tech_stock)
```

```
Rows: 756
Columns: 4
$ company <chr> "AAPL", "AAPL", "AAPL", "AAPL", "AAPL", "AAPL", "AAPL", "AAPL",…
$ date    <date> 2021-01-04, 2021-01-05, 2021-01-06, 2021-01-07, 2021-01-08, 2…
$ high    <dbl> 133.61, 131.74, 131.05, 131.63, 132.63, 130.17, 129.69, 131.45,…
$ low     <dbl> 126.76, 128.43, 126.38, 127.86, 130.23, 128.50, 126.86, 128.49,…
```

The types of data for the columns are: *character vector*, date: *date vector*, high: *double vector*, and low: *double vector*.

(b)

Make a side-by-side box plots of the high price for the three tech companies.

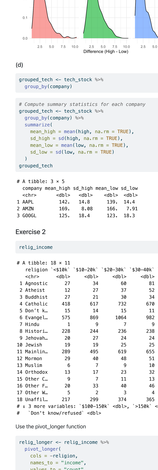
```
# Side-by-side boxplot for different companies by their high stock price
boxplot(high ~ company, data = tech_stock,
        main = "High Stock Price of three Tech Companies",
        xlab = "Company Names",
        ylab = "High Stock Price",
        col = c("lightblue", "royalblue", "lightgreen"))
```


```
# install.packages("dplyr")
library(dplyr)
```

```

Attaching package: 'dplyr'

The following objects are masked from 'package:dplyr':

    ident, sql
```

(c)

```
tech_stock <- tech_stock %>%
  mutate(diff = high - low)

# Create faceted histograms of 'diff' for each company
ggplot(tech_stock, aes(x = diff, fill = company)) +
  geom_density(color = "black", alpha = 0.7) +
  facet_wrap(~ company) +
  labs(title = "Histogram of Diff for Each Tech Company",
       x = "Difference (High - Low)",
       y = "Frequency") +
  theme_minimal()
```



(d)

```
grouped_tech <- tech_stock %>%
  group_by(company)

# Compute summary statistics for each company
grouped_tech <- tech_stock %>%
  group_by(company) %>%
  summarize(
    mean_high = mean(high, na.rm = TRUE),
    sd_high = sd(high, na.rm = TRUE),
    mean_low = mean(low, na.rm = TRUE),
    sd_low = sd(low, na.rm = TRUE)
  )
grouped_tech
```

```
# A tibble: 3 × 5
  company mean_high sd_high mean_low sd_low
  <chr>       <dbl>   <dbl>    <dbl>  <dbl>
1 AAPL        142.    14.0     139.   14.4
2 AMZN        169.     8.80    166.    7.91
3 GOOGL       125.    18.4     123.   18.3
```

Exercise 2

```
relig_income
```

```
# A tibble: 18 × 11
   religion `<$10k` `$10-20k` `$20-30k` `$30-40k` `$40-50k` `$50-75k` `$75-100k`
   <chr>      <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>      <dbl>
 1 Agnostic      27        34        60        81        76       137        122
 2 Atheist       12        27        37        52        35        70         73
 3 Buddhist      27        21        30        34        33        58         62
 4 Catholic     418       617       732       670       638      1116        949
 5 Don't k…      15        14        15        11        10        35         21
 6 Evangel…     575       869      1064       982       881      1486        949
 7 Hindu          1         9         7         9        11        34         47
 8 Histori…     228       244       236       238       197       223        131
 9 Jehovah…      20        27        24        24        21        30         15
10 Jewish        19        19        25        25        30        95         69
11 Mainli…      289       495       619       655       651      1107        939
12 Mormon        29        40        48        51        56       112         85
13 Muslim         6         7         9        10         9        23         16
14 Orthodox      13        17        23        32        32        47         38
15 Other f…       9         7        11        13        13        14         18
16 Other fa…     20        33        40        46        49        63         46
17 Other Re…      5         2         3         4         2         7          3
18 Unaffil…     217       299       374       365       341       528        407
# ℹ 3 more variables: `$100-150k` <dbl>, `>150k` <dbl>,
#   `Don't know/refused` <dbl>
```

Use the pivot_longer function

```
relig_longer <- relig_income %>%
  pivot_longer(
    cols = -religion,
    names_to = "income",
    values_to = "count"
  )
```

```
relig_longer
```

```
# A tibble: 180 × 3
   religion income           count
   <chr>    <chr>            <dbl>
 1 Agnostic <$10k              27
 2 Agnostic $10-20k            34
 3 Agnostic $20-30k            60
 4 Agnostic $30-40k            81
 5 Agnostic $40-50k            76
 6 Agnostic $50-75k           137
 7 Agnostic $75-100k          122
 8 Agnostic $100-150k         109
 9 Agnostic >150k              84
10 Agnostic Don't know/refused  96
# ℹ 170 more rows
```

Exercise 3

```
tb1 <- tibble(
id = c(1:4, 1:4),
group = c("t", "t", "t", "t", "c", "c", "c", "c"), vals = c(4, 6, 8, 11, 5, 6, 10, 9)
)
```

```
tb1
```

```
# A tibble: 8 × 3
     id group  vals
  <int> <chr> <dbl>
1     1 t         4
2     2 t         6
3     3 t         8
4     4 t        11
5     1 c         5
6     2 c         6
7     3 c        10
8     4 c         9
```

Use the pivot_wider() and mutate() functions to transform this data table into the following format, which has a column with the differences between the control and treatment group values.

```
tb1_wide <- tb1 %>%
  pivot_wider(
    names_from = group,
    values_from = vals
  ) %>%
  mutate(difference = t - c)
```

```
tb1_wide
```

```
# A tibble: 4 × 4
     id     t     c difference
  <int> <dbl> <dbl>      <dbl>
1     1     4     5         -1
2     2     6     6          0
3     3     8    10         -2
4     4    11     9         -5
```

Exercise 4

```
gss_cat2 <- gss_cat |>
mutate(partyid = fct_recode(partyid,
    "Republican, strong"    = "Strong republican",
    "Republican, weak"      = "Not str republican",
    "Independent, near rep" = "Ind,near rep",
    "Independent, near dem" = "Ind,near dem",
    "Democrat, weak"        = "Not str democrat",
    "Democrat, strong"      = "Strong democrat"
))
```

```
party_age <- gss_cat2 %>%
  group_by(partyid) %>%
  summarize(avg_age = mean(age, na.rm = TRUE))
```

```
party_age
```

```
# A tibble: 10 × 2
   partyid               avg_age
   <fct>                   <dbl>
 1 No answer                50.8
 2 Don't know               34
 3 Other party              45.2
 4 Republican, strong       51.9
 5 Republican, weak         47.2
 6 Independent, near rep    47.1
 7 Independent              43.3
 8 Independent, near dem    44.9
 9 Democrat, weak           46.5
10 Democrat, strong         51.2
```

```
ggplot(party_age, aes(x = avg_age, y = fct_reorder(partyid, avg_age))) +
  geom_point() +
  labs(title = "Average Age by Party ID",
       x = "Average Age",
       y = "Party Affiliation") +
  theme_minimal()
```

Exercise 5

```
ggplot(gss_cat2, aes(y = fct_rev(fct_infreq(partyid)))) +
  geom_bar() +
  labs(title = NULL,
       x = "Count",
       y = "Party Affiliation") +
  theme_minimal()
```