Eldor_Rakhmonaliev_48756997_Assignment2

```
library(tidyverse)
library(ggrepel)
library(here)
library(readr)
library(DBI)
library(duckdb)
library(dbplyr)
library(broom)
library(knitr)
```

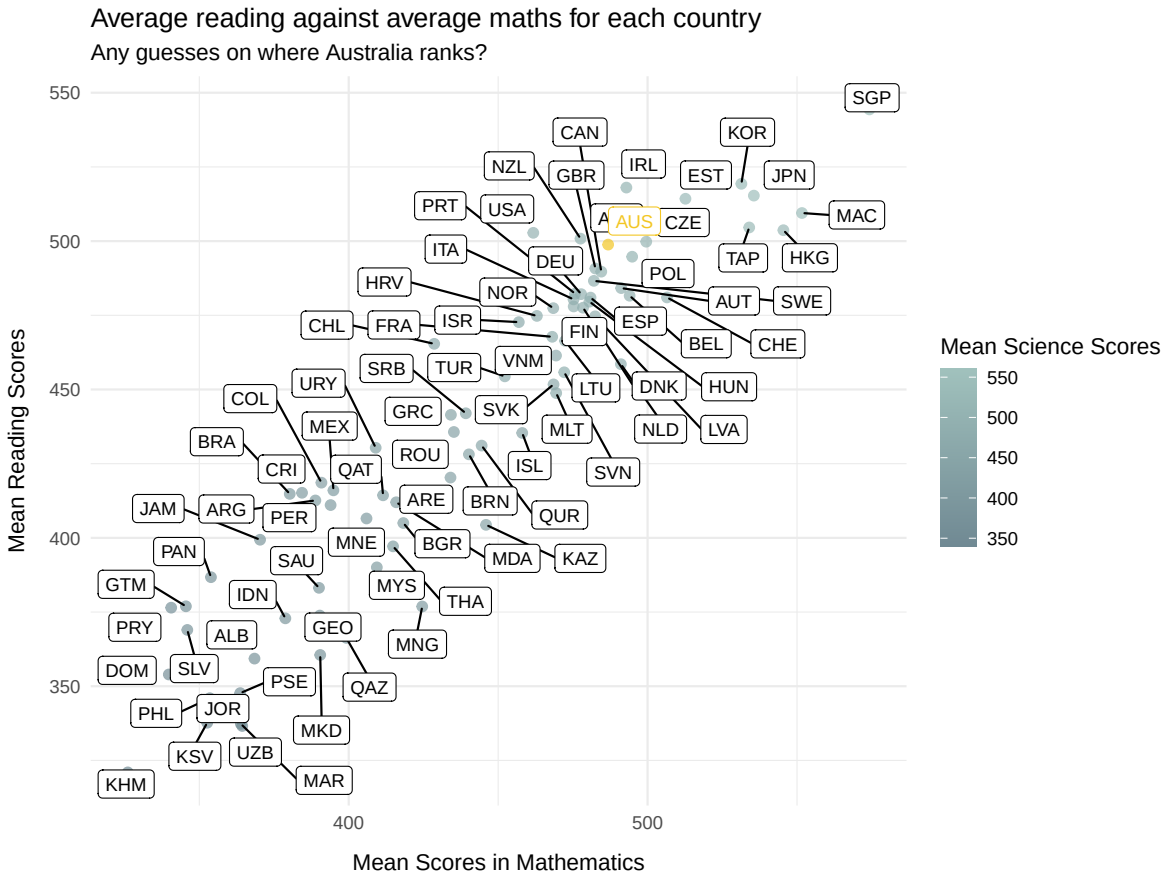
Exercise 1

```
# Loading the dataset
pisa <- read_csv(here("data", "pisa_2022.csv"), show_col_types = FALSE)

# Plot 1 part
pisa_avg <- pisa %>% # Calculating average rading and average math and science
  group_by(country) %>% # Group by country column
  summarise( # Summarise to be more efficient to find averages
    mean_math = mean(math, na.rm = TRUE),
    mean_read = mean(read, na.rm = TRUE),
    mean_science = mean(science, na.rm = TRUE),
    .groups = "drop" # Dropping the groubby after calculating
  ) %>%
  mutate(country = toupper(country))

# Creating the graph for first plot
ggplot() +
  geom_point(data = pisa_avg %>% filter(country != "AUS"),
             aes(x = mean_math, y = mean_read, color = mean_science),
```

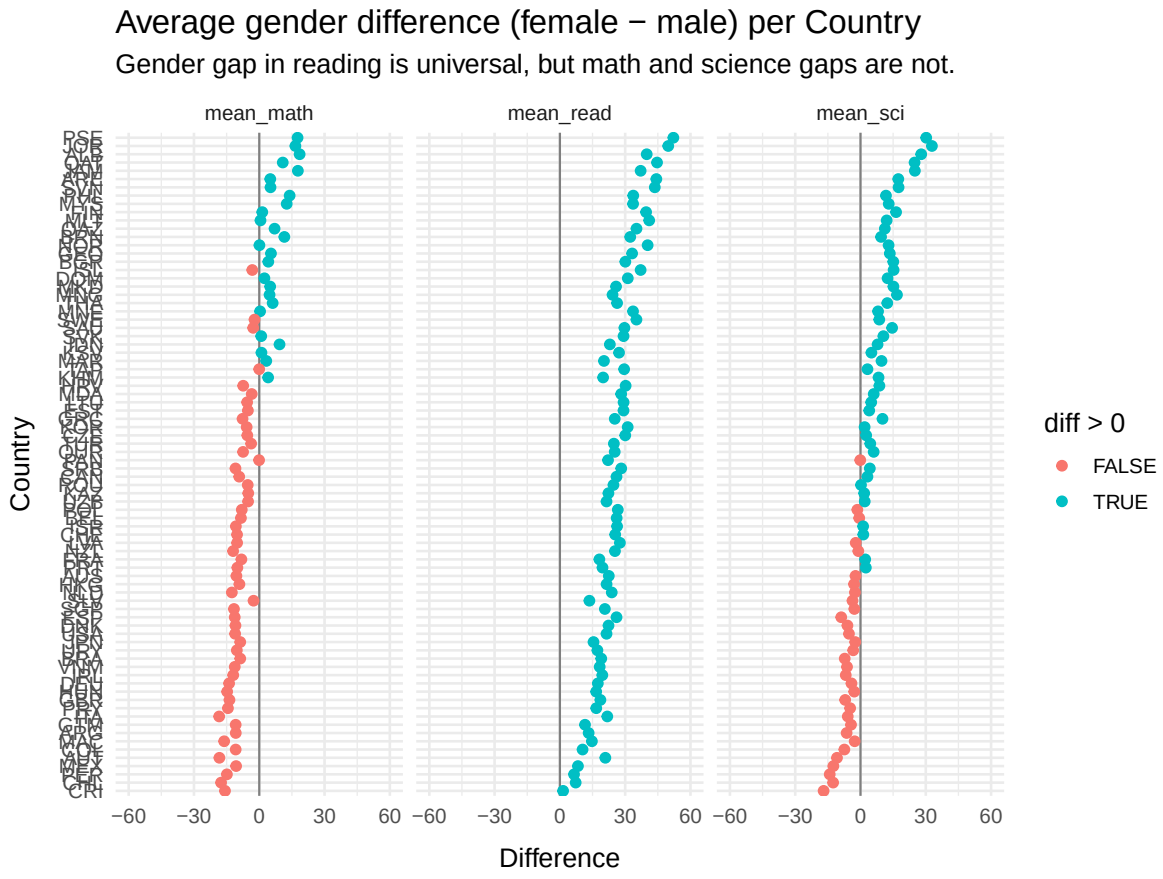
```
            size = 2, alpha = 0.7) +
geom_point(data = pisa_avg %>% filter(country == "AUS"),  # If country AUS then change the
            aes(x = mean_math, y = mean_read),
            size = 2, alpha = 0.7, color = "#F3C623") +
geom_label_repel(data = pisa_avg,
                 aes(x = mean_math, y = mean_read, label = country),
                 max.overlaps = 100, size = 3) +
geom_label_repel(data = pisa_avg %>% filter(country == "AUS"), # If country AUS then change
                 aes(x = mean_math, y = mean_read, label = country),
                 color = "#F3C623", max.overlaps = 100, size = 3) +
scale_color_gradient(low = "#708993", high = "#A1C2BD", # This row means coloring all other
                     name = "Mean Science Scores") +
labs(
  title = "Average reading against average maths for each country",
  subtitle = "Any guesses on where Australia ranks?",
  x = "Mean Scores in Mathematics", # X label
  y = "Mean Reading Scores" # Y label
) +
theme_minimal() +
theme(
  axis.title.x = element_text(margin = margin(t = 10)), # Creating space between label x ar
  axis.title.y = element_text(margin = margin(r = 10))) # Creating space between label y ar
```




```
# Plot 2 part
pisa_gender <- pisa %>%
  group_by(country, gender) %>% # Groupby by Country and Gender
  summarise(across(c(math, read, science), \(x) mean(x, na.rm = TRUE)),
            .groups = "drop") %>%
  pivot_wider(names_from = gender, values_from = c(math, read, science)) %>%
  mutate( # Mutate and adding new columns
    diff_math = math_female - math_male,
    diff_read = read_female - read_male,
    diff_sci  = science_female - science_male
  ) %>%
  pivot_longer(starts_with("diff_"), names_to = "subject", values_to = "diff") %>%
  mutate(
    subject = recode(subject,
                     diff_math = "mean_math",
                     diff_read = "mean_read",
                     diff_sci  = "mean_sci"),
```

```
    diff_pos = diff > 0
  )

# Creating the graph for second plot
ggplot(pisa_gender, aes(x = diff, y = reorder(country, diff), color = diff_pos)) +
  geom_vline(xintercept = 0, color = "gray50") +
  geom_point(size = 2) +
  facet_grid(. ~ subject, scales = "free_x", space = "free_x") +
  scale_x_continuous(
    limits = c(-60, 60),
    breaks = seq(-60, 60, by = 30)
  ) +
  scale_color_manual(values = c("FALSE" = "#F8766D", "TRUE" = "#00BFC4")) +
  labs(
    title = "Average gender difference (female - male) per Country",
    subtitle = "Gender gap in reading is universal, but math and science gaps are not.",
    x = "Difference", y = "Country", color = "diff > 0" # X and Y labels
  ) +
  theme_minimal(base_size = 13) +
  theme(
    axis.title.x = element_text(margin = margin(t = 10)),  # Creating space between label x a
    axis.title.y = element_text(margin = margin(r = 10))) # Creating space between label y ar
```



Exercise 2

```
# Loading the dataset
aus_price <- read_csv(here("data", "aus_median_house_price.csv"), show_col_types = FALSE)
```

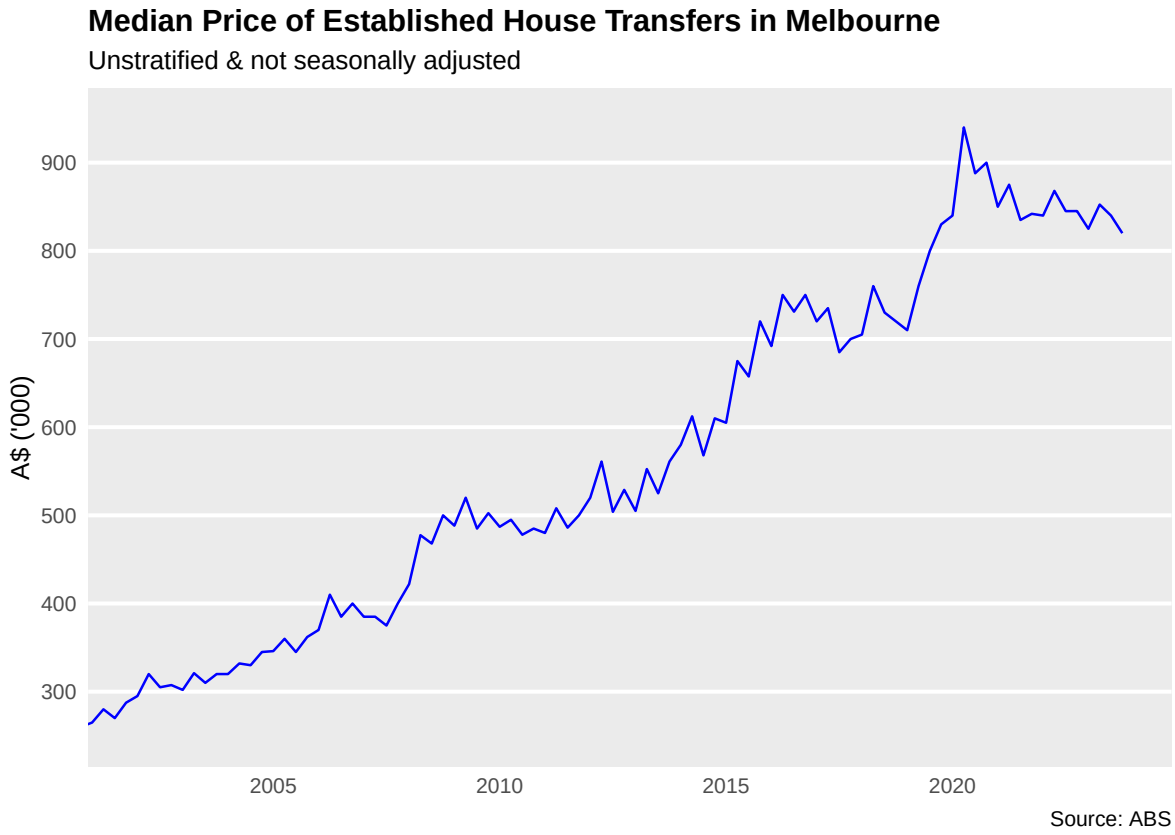
```
New names:
* `` -> `...1`
```

```
gdp_changes <- read_csv(here("data", "aus_GDP_changes.csv"), show_col_types = FALSE)

# Correction the house data
aus_price_clean <- aus_price %>% # Manipulating the dataset
  select(Melbourne_Median_Price = `Median Price of Established House Transfers (Unstratified)
  mutate(Melbourne_Median_Price = as.numeric(Melbourne_Median_Price)) %>% # Mutating the colu
```

```r
  filter(!is.na(Melbourne_Median_Price)) %>% # Filtering column
  mutate(
    Time_Period = row_number(),
    Year = 2000 + (Time_Period - 1) * 0.25  # Assuming quarterly data starting from 2000
  )

# Plot 1
ggplot(aus_price_clean, aes(x = Year, y = Melbourne_Median_Price)) + # Creating graph
  geom_line(color = "blue") +
  labs(
    title = "Median Price of Established House Transfers in Melbourne", # Title of the plot
    subtitle = "Unstratified & not seasonally adjusted",
    x = NULL, # X label name
    y = "A$ ('000)", # Y label name
    caption = "Source: ABS"
  ) +
  scale_x_continuous(breaks = seq(2005, 2020, 5)) + # Sequence of X label
  scale_y_continuous(breaks = seq(300, 900, 100)) + # Sequence of Y label
  coord_cartesian(xlim = c(2002, NA), ylim = c(250, 950)) +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold"), # Making bold text for title
    panel.grid.minor = element_blank(),
    panel.grid.major.x = element_blank(),
    panel.grid.major.y = element_line(color = "white", linewidth = 0.8), # Color of backgrou
    panel.background = element_rect(fill = "grey92", color = NA), # Panel background color
    plot.background = element_rect(fill = "white", color = NA) # Plot background color
  )
```

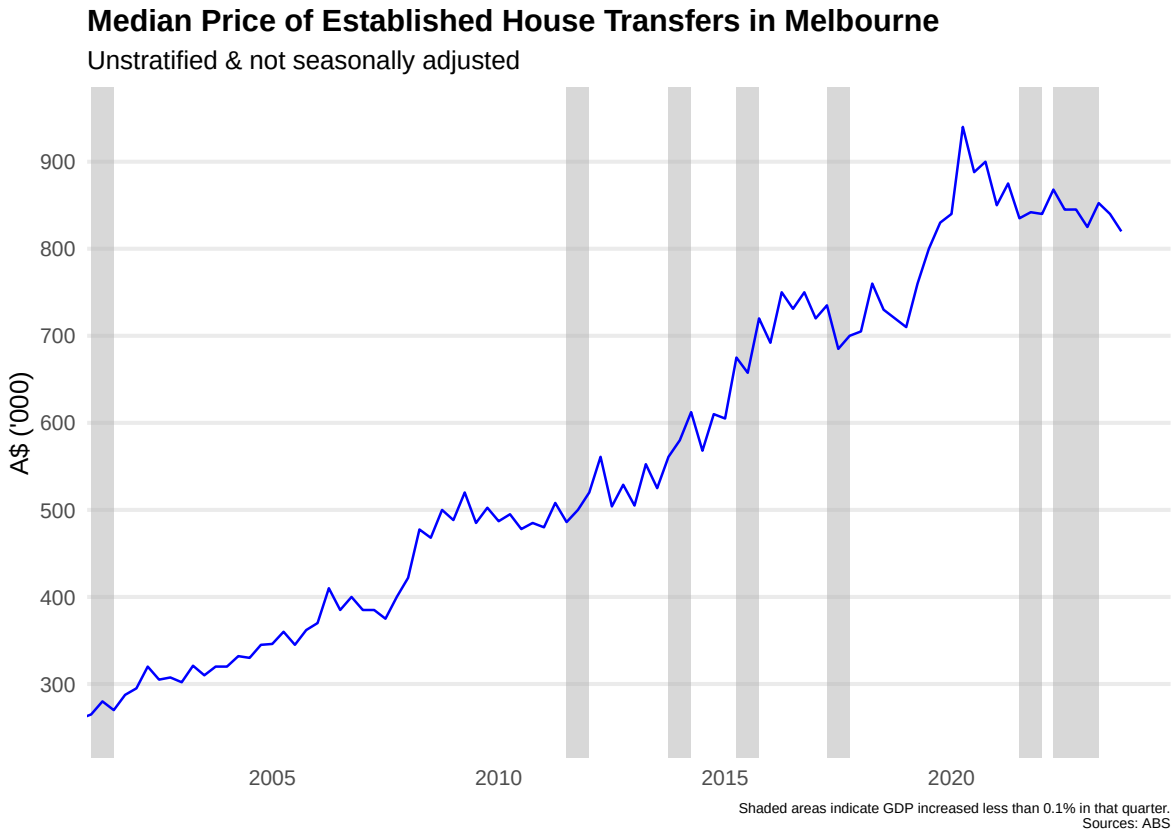



```
# Finding low GDP growth (GDP change < 0.1%)
low_gdp <- gdp_changes %>%
  mutate(
    GDP_changes = as.numeric(GDP_changes), # Mutating column into numeric
    Time_Period = row_number()
  ) %>%
  mutate(low_gdp_growth = GDP_changes < 0.1) %>%
  filter(Time_Period <= nrow(aus_price_clean)) %>%
  mutate(Year = 2000 + (Time_Period - 1) * 0.25)

# Finding periods
low_gdp_periods <- low_gdp %>%
  filter(low_gdp_growth) %>%
  arrange(Time_Period) %>%
  mutate(
    gap = c(1, diff(Time_Period)),
    new_group = gap > 1
  ) %>%
```

```r
  mutate(group = cumsum(new_group)) %>%
  group_by(group) %>%
  summarise(
    xmin = min(Year),
    xmax = max(Year) + 0.25,
    n_quarters = n(),
    .groups = "drop"
  ) %>%
  filter(n_quarters >= 2)  # Shows period of 2 consecutive quarters

# Plot 2
ggplot(aus_price_clean, aes(x = Year, y = Melbourne_Median_Price)) + # Creating plot
  geom_rect(
    data = low_gdp_periods,
    aes(xmin = xmin, xmax = xmax, ymin = -Inf, ymax = Inf),
    fill = "grey70",
    alpha = 0.5,
    inherit.aes = FALSE
  ) +
  geom_line(color = "blue") + # Color of the line in the graph
  labs(
    title = "Median Price of Established House Transfers in Melbourne", # Title of the graph
    subtitle = "Unstratified & not seasonally adjusted",
    x = NULL, # X label name
    y = "A$ ('000)", # Y label name
    caption = "Shaded areas indicate GDP increased less than 0.1% in that quarter.\nSources:
  ) +
  scale_x_continuous(breaks = seq(2005, 2020, 5)) + # Sequence of X label
  scale_y_continuous(breaks = seq(300, 900, 100)) + # Sequence of Y lalbel
  coord_cartesian(xlim = c(2002, NA), ylim = c(250, 950)) +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold"), # Making bold text for title
    panel.grid.minor = element_blank(),
    panel.grid.major.x = element_blank(),
    panel.grid.major.y = element_line(color = "grey92", linewidth = 0.8), # Color of backgrou
    panel.background = element_rect(fill = "white", color = NA), # Panel background color
    plot.background = element_rect(fill = "white", color = NA), # Plot background color
    plot.caption = element_text(hjust = 1, size = 6)
  )
```



Exercise 3

```
# Creating DuckDB connection
con <- dbConnect(duckdb::duckdb())

# Reading csv files
actors <- read_csv(here("data", "Actors.csv"), show_col_types = FALSE)
castings <- read_csv(here("data", "Castings.csv"), show_col_types = FALSE)
directors <- read_csv(here("data", "Directors.csv"), show_col_types = FALSE)
movies <- read_csv(here("data", "Movies.csv"), show_col_types = FALSE)

# Copying data to DuckDB
dbWriteTable(con, "actors", actors)
dbWriteTable(con, "castings", castings)
dbWriteTable(con, "directors", directors)
dbWriteTable(con, "movies", movies)
```

```r
# Creating dplyr table references
actors_db <- tbl(con, "actors")
castings_db <- tbl(con, "castings")
directors_db <- tbl(con, "directors")
movies_db <- tbl(con, "movies")

# Part a
# Identifying the most common director and
# actor parterships through collection total movies
part_a <- actors_db %>%
  semi_join(castings_db, by = "ActorID") %>%  # Keep only actors with castings
  inner_join(castings_db, by = "ActorID") %>%  # Join to get movie associations
  inner_join(movies_db, by = "MovieID") %>%    # Join to get movie details
  inner_join(directors_db, by = "DirectorID") %>%  # Join to get directors
  group_by(Actor = Name.x, Director = Name.y) %>%  # Group by partnership
  summarise(Movies_total = n(), .groups = "drop") %>%  # Count movies per pair
  arrange(desc(Movies_total)) %>%  # Sort by frequency
  head(5) %>%  # Limit to top 10
  collect
part_a
```

```
# A tibble: 5 x 3
  Actor         Director             Movies_total
  <chr>         <chr>                       <dbl>
1 Robert Duvall Francis Ford Coppola            2
2 Steve Buscemi Joel Coen                       2
3 John Cazale   Francis Ford Coppola            2
4 Ronald Lacey  Steven Spielberg                1
5 Gene Hackman  Tony Scott                      1
```

```r
# Part b
# Finding actors who worked with multiple workers
# and counting unique directors with total movies per actor
part_b <- actors_db %>%
  left_join(castings_db, by = "ActorID") %>%
  left_join(movies_db, by = "MovieID") %>%
  left_join(directors_db, by = "DirectorID") %>%
  group_by(Name.x) %>%
  summarise(
    unique_directors = n_distinct(Name.y),
    total_movies = n(),
```

```
    .groups = "drop"
  ) %>%
  filter(unique_directors > 1) %>%
  arrange(desc(unique_directors)) %>%
  collect()

part_b
```

```
# A tibble: 6 x 3
  Name.x         unique_directors total_movies
  <chr>                     <dbl>        <dbl>
1 Ingrid Bergman                2            2
2 Gene Hackman                  2            2
3 Harrison Ford                 2            2
4 John Cazale                   2            3
5 Al Pacino                     2            2
6 Vito Scotti                   2            2
```

```
# In my opinion, the question for part b can be :
# Who are the most adaptable actors that can work with various directors
# This question is more fit based on the sql query

# Disconnecting from database
dbDisconnect(con)
```

Exercise 4

Overview

```
# Creating matrix from questions example
obs <- matrix(c(3,33,27,9), nrow = 2, byrow = TRUE,
dimnames = list(Gender = c("Men","Women"), Status = c("Studying","Not")))
obs
```

```
       Status
Gender  Studying Not
  Men          3  33
  Women       27   9
```

```r
chi <- chisq.test(obs, correct = FALSE) # Pearson's chi-squared test
chi
```

```
    Pearson's Chi-squared test

data:  obs
X-squared = 32.914, df = 1, p-value = 9.631e-09
```

```r
fisher <- fisher.test(obs, alternative = "two.sided") # Fisher's exact test (two-sided)
fisher
```

```
    Fisher's Exact Test for Count Data

data:  obs
p-value = 8.418e-09
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.005139362 0.138390511
sample estimates:
odds ratio
 0.0324741
```

Simulation

```
# A tibble: 24 x 3
   Description                                 Test_Level Values
   <chr>                                       <chr>       <dbl>
 1 Setting 1: n = 20 men, 20 women, p = 0.2    Chi_10     0.106
 2 Setting 1: n = 20 men, 20 women, p = 0.2    Chi_05     0.0554
 3 Setting 1: n = 20 men, 20 women, p = 0.2    Chi_01     0.0116
 4 Setting 1: n = 20 men, 20 women, p = 0.2    Fish_10    0.0499
 5 Setting 1: n = 20 men, 20 women, p = 0.2    Fish_05    0.026
 6 Setting 1: n = 20 men, 20 women, p = 0.2    Fish_01    0.0033
 7 Setting 2: n = 20 men, 20 women, p = 0.5    Chi_10     0.0848
 8 Setting 2: n = 20 men, 20 women, p = 0.5    Chi_05     0.0432
 9 Setting 2: n = 20 men, 20 women, p = 0.5    Chi_01     0.0094
10 Setting 2: n = 20 men, 20 women, p = 0.5    Fish_10    0.0432
11 Setting 2: n = 20 men, 20 women, p = 0.5    Fish_05    0.0201
```

```
12 Setting 2: n = 20 men, 20 women, p = 0.5   Fish_01    0.0033
13 Setting 3: n = 100 men, 100 women, p = 0.2 Chi_10     0.0974
14 Setting 3: n = 100 men, 100 women, p = 0.2 Chi_05     0.0497
15 Setting 3: n = 100 men, 100 women, p = 0.2 Chi_01     0.0097
16 Setting 3: n = 100 men, 100 women, p = 0.2 Fish_10    0.0666
17 Setting 3: n = 100 men, 100 women, p = 0.2 Fish_05    0.032
18 Setting 3: n = 100 men, 100 women, p = 0.2 Fish_01    0.0053
19 Setting 4: n = 100 men, 100 women, p = 0.5 Chi_10     0.104
20 Setting 4: n = 100 men, 100 women, p = 0.5 Chi_05     0.0566
21 Setting 4: n = 100 men, 100 women, p = 0.5 Chi_01     0.0094
22 Setting 4: n = 100 men, 100 women, p = 0.5 Fish_10    0.0769
23 Setting 4: n = 100 men, 100 women, p = 0.5 Fish_05    0.0409
24 Setting 4: n = 100 men, 100 women, p = 0.5 Fish_01    0.0062
```

Reference

Downey, Allen B. 2014. *Think Stats: Exploratory Data Analysis*. Sebastopol, CA: O'Reilly Media.

R Core Team. 2023. "R: A Language and Environment for Statistical Computing." Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Wickham, Hadley, and Garrett Grolemund. 2016. *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. Sebastopol, CA: O'Reilly Media.