Introduction
Data Source and Nature
Data Cleaning
Initial Data Analysis
Packages Used
Part A - COVID Tests Distribution
Part B - Women in Data Science
Women in Data Science
Expected Data Scienctist Salary
Salary Perceptions with Gender
References

nice structure

# Report

Code

Student 1

University of Sydney | DATA2902 | Septempber 2021

Hide

```
# Load in R pacakges
library(tidyverse)
library(plotly)
```

Hide

```r
# Data cleaning

# Read in data
df <- read.csv(file = "DATA2x02 survey (Responses) - Form responses 1.csv"
)

# Rename the columns so they are more manageable
short_names <- c("timestamp", "covid_tests", "living_arrangments", "heigh
t", "wednesday_question", "in_au", "math_ability", "r_ability", "finding_d
ata2x02", "uni_year", "zoom_webcam", "vaccination", "fav_social_media", "g
ender", "steak_preference", "dominant_hand", "stress_level", "loneliness_l
evel", "emails", "email_sign_off", "data_scientist_salary", "in_advanced",
"majors", "exercising_hours")
colnames(df) <- short_names

# Split the "Timestamp" column into "date" and "time" columns.
# NB: "date" indicates the date that the the person completed the survey.
# NB: "time" indicates the times that the person completed the survey on t
he specified day.
df <- df %>%
  separate(timestamp, c("date", "time"), sep = "\\s+", remove = FALSE) #
 "remove = FALSE" to keep the original "timestamp" column

# Clean up height column so that all height measurements are in cm.
#   On visual inspection, it appears the majority of people wrote cm as th
eir height, but there were many variants:
#   i.e. (centimeters) need to clean for xcm, x cm, xcm<other characters>;
where x is the height in cm.
#   i.e. (metres) need to clean for x m, xm, x M, xM, x metres; where x is
the height in m.
#   i.e. (inches and feet) need to clean for x'n", x'n'', x' n", x' n'',
 x'n, x’n, xfoot ninches; where x is feet and n is inches
#   i.e. (random sentences or unreadable sequences) e.g. "Short enough to
 take offence at this question". These will be converted to NA.
#   i.e. (no unit specified) where a unit was not specified, the most natu
ral unit will be assumed. For example, it is assumed a height of 5.5 is 5
 feet and 5 inches,
#        as it is impossible to be 5.5 m or cm tall.
#   i.e. (multiple units specified) e.g. 168 cm (5 feet 5 inches)
#   i.e. (left blank) treated as NA.

# Function to convert feet and inches to cm.
# Values passed in are characters, and value passed out is numeric.
feet_and_inches_to_cm <- function(feet_string, inches_string) {
  feet = as.numeric(feet_string)
  inches = as.numeric(inches_string)
  inches = feet * 12 + inches
  cm = inches * 2.54
  return(cm)
}
```

```
# Function to convert a string representing a value in m to cm.
# cut_off is the number of characters to remove from the end of the strin
g, so that the function converts only the numeric part of the string.
m_to_cm <- function(m_string, cut_off) {
  m = as.numeric(substr(m_string, 1, nchar(m_string)-cut_off))
  m = m * 100
  return(m)
}

# Loop through all heights, and convert them to numbers in cm.
for (i in 1:length(df$height)){

  current_height = df$height[i]

  # NB: It is possible to pool some of these if statements together, but t
here are no current performance issues to warrant such a change.

  # Matches form: xcm
  if (grepl("^[0-9]+(\\.[0-9]+)?cm$", current_height)) {
    df$height[i] = substr(current_height, 1, nchar(current_height)-2)

    # Matches form: x cm
  } else if (grepl("^[0-9]+(\\.[0-9]+)? cm$", current_height)) {
    df$height[i] = substr(current_height, 1, nchar(current_height)-3)

    # Matches form: xcm<other characters>
  } else if (grepl("^[0-9]+(\\.[0-9]+)?cm", current_height)) {
    df$height[i] = str_split(current_height, "cm")[[1]][1]

    # Matches form: xm or xM
  } else if (grepl("^[0-9]+(\\.[0-9]+)?(m|M)$", current_height)) {
    df$height[i] = m_to_cm(current_height, 1)

    # Matches form: x m or x M
  } else if (grepl("^[0-9]+(\\.[0-9]+)? (m|M)$", current_height)) {
    df$height[i] = m_to_cm(current_height, 2)

    # Matches form: x metres or x Metres or x meters or x Meters
  } else if (grepl("^[0-9]+(\\.[0-9]+)? (m|M)(etres|eters)$", current_heig
ht)) {
    df$height[i] = m_to_cm(current_height, 7)

    # Matches form: x'n"
  } else if (grepl("^[0-9]*'[0-9]*\"$", current_height)) {
    feet = str_split(current_height, "'")[[1]][1]
    inches = str_split(current_height, "'")[[1]][2]
    inches = substr(inches, 1, nchar(inches)-1)
    df$height[i] = feet_and_inches_to_cm(feet, inches)

    # Matches form: x'n''
  } else if (grepl("^[0-9]*'[0-9]*''$", current_height)) {
    current_height = substr(current_height, 1, nchar(current_height)-2)
```

```
    feet = str_split(current_height, "'")[[1]][1]
    inches = str_split(current_height, "'")[[1]][2]
    df$height[i] = feet_and_inches_to_cm(feet, inches)

    # Matches form: x' n"
  } else if (grepl("^[0-9]*' [0-9]*\"$", current_height)) {
    current_height = substr(current_height, 1, nchar(current_height)-1)
    feet = str_split(current_height, "' ")[[1]][1]
    inches = str_split(current_height, "' ")[[1]][2]
    df$height[i] = feet_and_inches_to_cm(feet, inches)

    # Matches form: x' n''
  } else if (grepl("^[0-9]*' [0-9]*''$", current_height)) {
    current_height = substr(current_height, 1, nchar(current_height)-2)
    feet = str_split(current_height, "' ")[[1]][1]
    inches = str_split(current_height, "' ")[[1]][2]
    df$height[i] = feet_and_inches_to_cm(feet, inches)

    # Matches form: x'n
  } else if (grepl("^[0-9]*'[0-9]*$", current_height)) {
    feet = str_split(current_height, "'")[[1]][1]
    inches = str_split(current_height, "'")[[1]][2]
    df$height[i] = feet_and_inches_to_cm(feet, inches)

    # Matches form: x’n
  } else if (grepl("^[0-9]*’[0-9]*$", current_height)) {
    feet = str_split(current_height, "’")[[1]][1]
    inches = str_split(current_height, "’")[[1]][2]
    df$height[i] = feet_and_inches_to_cm(feet, inches)

    # Matches form: xfoot ninches
  } else if (grepl("^[0-9]*foot [0-9]*inches$", current_height)) {
    feet = str_split(current_height, " ")[[1]][1]
    inches = str_split(current_height, " ")[[1]][2]
    feet = substr(feet, 1, nchar(feet)-4)
    inches = substr(inches, 1, nchar(inches)-6)
    df$height[i] = feet_and_inches_to_cm(feet, inches)

    # Matches form: n cm <other text>
  } else if (grepl("^[0-9]* cm", current_height)) {
    cm = str_split(current_height, " cm")[[1]][1]
    df$height[i] = cm

    # Matches form: no unit specified
    # Anything between (inclusive) 54.6 and 243 should be converted to cm.
    # Anything between 0.546 (inclusive) and 1.8 (exclusive) will be assum
ed to be m.
    # Anything between (inclusive) 1.8 and 2.43 could be feet or m, so wil
l be converted to NA.
    # Anything between 2.43 (exclusive) and 7.97 (inclusive) will be assum
ed to be feet.
    # Anything left over will be converted to NA
```

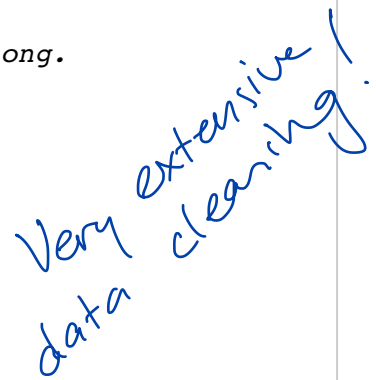
```
    # These justifications are based upon how the tallest person ever is 2
43cm and the shortest person ever is 54.6cm.
    # Tallest person -> https://www.guinnessworldrecords.com/records/hall-
of-fame/robert-wadlow-tallest-man-ever#:~:text=The%20twins%20then%20named%
20Robert,8%20ft%2011.1%20in)%20tall
    # Shortest person -> https://www.guinnessworldrecords.com/news/2021/8/
a-history-of-the-worlds-shortest-people-and-the-countries-theyre-from-6683
23
  } else if ( grepl("^[0-9]+\\.[0-9]*$", current_height) | grepl("^[0-9]+
$", current_height)){
    value = as.numeric(current_height)
    if (value >= 54.6 & value <= 243) {
      df$height[i] = value
    } else if (value >= 0.546 & value < 1.8) {
      df$height[i] = value * 100
    } else if (value >= 1.8 & value <= 2.43) {
      df$height[i] = NA
    } else if (value > 2.43 & value <= 7.97) {
      df$height[i] = value * 30.48
    } else {
      df$height[i] = NA
    }

    # Anything else that hasn't been converted yet is not in an appropriat
e form.
  } else {
    df$height[i] = NA
  }

}

# Convert "height" column to numeric type.
df$height = as.numeric(as.character(df$height))

# There were some values which must have been input incorrectly, so now cl
eaning for these.
# e.g. someone said their height is 1.90cm.
# Remove values larger than 243 and smaller than 54.6
df = df %>%
  filter(height >= 54.6 & height <= 243 )

# Clean "wednesday_question" column so that all days of the week are capat
ilised, and so that days of the week are in a consistent format.
# For now, ignore responses which are irregular such as "If I was notified
before Monday then Monday. Otherwise Friday."
df = df %>%
  mutate(
    wednesday_question = case_when(
      grepl("^((monday)|(Monday))(.)*", wednesday_question) ~ "Monday",
      grepl("^((teusday)|(Teusday))(.)*", wednesday_question) ~ "Teusday",
      grepl("^((wednesday)|(Wednesday))(.)*", wednesday_question) ~ "Wedne
sday",
```

```
      grepl("^((thursday)|(Thursday))(.)*", wednesday_question) ~ "Thursda
y",
      grepl("^((friday)|(Friday))(.)*", wednesday_question) ~ "Friday",
      grepl("^((saturday)|(Saturday))(.)*", wednesday_question) ~ "Saturda
y",
      grepl("^((sunday)|(Sunday))(.)*", wednesday_question) ~ "Sunday",
      TRUE ~ wednesday_question
    )
  )

# Create a new column "wednesday_question_cleaned" which when a response o
ccurs less than 5 time, it is classified as "Other"
df = df %>%
  mutate(wednesday_question_cleaned = fct_lump_min(wednesday_question, 5))

# Convert "in_au" into logical column
df = df %>%
  mutate(
    in_au = case_when(
      grepl("^No$", in_au) ~ "FALSE",
      grepl("^Yes$", in_au) ~ "TRUE",
      TRUE ~ in_au
    )
  )
df$in_au = as.logical(df$in_au)

# Clean "fav_social_media" column
# NB: "Weixin" has been classified as "WeChat"
df = df %>%
  mutate(fav_social_media = toupper(fav_social_media)) %>%
  mutate(
    fav_social_media = case_when(
      grepl("^((INSTAGRAM)|(IG)|(INSTA)|(INSGRAM))\\s*$", fav_social_media
) ~ "Instagram",
      grepl("^((FACEBOOK)|(FB))\\s*$", fav_social_media ) ~ "Facebook",
      grepl("^TWITTER\\s*$", fav_social_media ) ~ "Twitter",
      grepl("^NONE\\s*$", fav_social_media ) ~ "None",
      grepl("^((WECHAT)|(WEIXIN)|(WETCHAT)|(WECHAT IN CHINA))\\s*$", fav_s
ocial_media ) ~ "WeChat",
      grepl("^BILIBILI\\s*$", fav_social_media ) ~ "Bilibili",
      grepl("^YOUTUBE\\s*$", fav_social_media ) ~ "Youtube",
      grepl("^REDDIT\\s*$", fav_social_media ) ~ "Reddit",
      grepl("^((TIK TOK)|(TIKTOK))\\s*$", fav_social_media ) ~ "TikTok",
      grepl("^DISCORD\\s*$", fav_social_media ) ~ "Discord",
      grepl("^MESSENGER\\s*$", fav_social_media ) ~ "Messenger",
      grepl("^IDK\\s*$", fav_social_media ) ~ "",
      grepl("^QQ\\s*$", fav_social_media ) ~ "Tencent QQ",
      grepl("^ED\\s*$", fav_social_media ) ~ "Ed Stem",
      grepl("^WEIBO\\s*$", fav_social_media ) ~ "Sina Weibo",
      grepl("^SNAPCHAT\\s*$", fav_social_media ) ~ "Snapchat",
      grepl("^BILIBILI AND YOUTUBE$", fav_social_media) ~ "Bilibili and Yo
utube",
```

```
      grepl("^WECHAT FACEBOOK$", fav_social_media) ~ "WeChat and Facebook"
,
      grepl("^INSTAGRAM/SNAP$", fav_social_media) ~ "Instagram and Snapcha
t",
      TRUE ~ fav_social_media
    )
  )

# Add a new column named "fav_social_media_reduced" where social media pla
tforms which occur less than 5 times are classified as "Other"
df = df %>%
  mutate(fav_social_media_reduced = fct_lump_min(fav_social_media, 5))

# Clean "gender" column
df = df %>%
  mutate(gender = toupper(gender)) %>%
  mutate(
    gender = case_when(
      grepl("^((MALE)|(MAN)|(M))", gender) ~ "Male",
      grepl("^((FEMALE)|(WOMAN)|(F))", gender) ~ "Female",
      grepl("^NON-BINARY", gender) ~ "Non-binary",
      TRUE ~ gender
    )
  )

# Clean "email_sign_off" column
# NB: Entries that were incredibly similar were categorised together.
#     e.g. Kindest Regards, Any Adverb + Regards, Kind Regards, Regards we
re all categorised as Regards.
df = df %>%
  mutate(email_sign_off = toupper(email_sign_off)) %>%
  mutate(
    email_sign_off = case_when(
      grepl("(REGARD)|(REAGRD)", email_sign_off) ~ "Regards",
      grepl("SINCERELY", email_sign_off) ~ "Sincerely",
      grepl("THANK", email_sign_off) ~ "Thanks",
      grepl("CHEERS", email_sign_off) ~ "Cheers",
      grepl("FROM", email_sign_off) ~ "From",
      grepl("HAVE A NICE DAY!", email_sign_off) ~ "Have a nice day!",
      grepl("WISH", email_sign_off) ~ "Best wishes",
      grepl("BEST", email_sign_off) ~ "Best",
      grepl("TAKE CARE", email_sign_off) ~ "Take care",
      grepl("(NOTHING)|(NONE)", email_sign_off) ~ "Nothing",
      grepl("(DEPENDS)|(VARY)", email_sign_off) ~ "Depends",
      grepl("RESPECTFULLY YOURS", email_sign_off) ~ "Respectfully yours",
      grepl("LOOKING FORWARDS TO YOUR REPLY", email_sign_off) ~ "Looking f
orwards to your reply",
      grepl("GOOD", email_sign_off) ~ "Good",
      grepl("YOURS", email_sign_off) ~ "Yours",
      grepl("(IS THAT RIGHT AREA)|(DEAR)", email_sign_off) ~ "",
      TRUE ~ email_sign_off
    )
```

```
  )

# Add a new column named "email_sign_off_reduced" where email sign-offs wh
ich occur less than 5 times are classified as "Other"
df = df %>%
  mutate(email_sign_off_reduced = fct_lump_min(email_sign_off, 5))

# Clean "data_scientist_salary" column.
for (i in 1:length(df$data_scientist_salary)) {

  current_entry = df$data_scientist_salary[i]
  current_entry = str_replace_all(current_entry, ",|\\$| |\\?", "")
  current_entry = tolower(current_entry)

  # Case when user inputs a dollar value for the yearly salary.
  if (grepl("^[0-9]+$", current_entry)) {
    df$data_scientist_salary[i] = current_entry

    # Case when user inputs yearly salary in terms of thousands of dollar
s. e.g. 50k
  } else if (grepl("(^([0-9]+k)$)|(^([0-9]+k/year)$)", current_entry)) {
    current_entry = str_replace_all(current_entry, "[a-z]|[:punct:]", "")
    current_entry = as.numeric(current_entry)
    df$data_scientist_salary[i] = current_entry * 1000

    # Case where user inputs some other way of indicating that the quoted
 value is a yearly rate
  } else if (grepl("(peryear)|(/year)|(p\\.a\\.)|(p\\.a)|(^[0-9]+auds$)|(^
[0-9]+aud$)", current_entry)) {
    current_entry = str_replace_all(current_entry, "[a-z]|[:punct:]", "")
    df$data_scientist_salary[i] = current_entry

    # Manually adjust some more unique values
  } else if (current_entry == "mid60k") {
    df$data_scientist_salary[i] = "65000"
  } else if (current_entry == "50-60kannually") {
    df$data_scientist_salary[i] = "55000"
  } else if (current_entry == "2000au-4000au/month") {
    df$data_scientist_salary[i] = as.character(3000 * 12)
  } else if (current_entry == "1kpmprobs...") {
    df$data_scientist_salary[i] = 12000
  } else if (current_entry == "roughly60000aud") {
    df$data_scientist_salary[i] = 60000

    # Case where user inputs some way of indicated that the quoted value i
s a monthly rate
  } else if (grepl("month", current_entry)) {
    current_entry = str_replace_all(current_entry, "[a-z]|[:punct:]", "")
    df$data_scientist_salary[i] = as.numeric(current_entry) * 12

    # Case where user inputs some way of indicated that the quoted value i
s a weekly rate
```

```
  } else if (grepl("(pw)|(week)", current_entry)) {
    current_entry = str_replace_all(current_entry, "[a-z]|[:punct:]", "")
    df$data_scientist_salary[i] = as.numeric(current_entry) * 52

    # Case where user inputs some way of indicated that the quoted value i
s a weekly rate
    # The yearly rate was calculated based off the fact that the normal wo
rking week is 8 hours a day, 5 days a week.
  } else if (grepl("(perhour)|(/h)|(/hour)|(/hr)", current_entry)) {
    current_entry = str_replace_all(current_entry, "[a-z]|[:punct:]", "")
    df$data_scientist_salary[i] = as.numeric(current_entry) * 8 * 5 * 52

  } else {
    df$data_scientist_salary[i] = ""
  }

}

# Convert "data_scientist_salary" into numeric column
df$data_scientist_salary = as.numeric(df$data_scientist_salary)

# Convert "in_advanced" into logical column
df = df %>%
  mutate(
    in_advanced = case_when(
      in_advanced == "DATA2002"~ "FALSE",
      in_advanced == "DATA2902 (Advanced)" ~ "TRUE",
      TRUE ~ in_advanced
    )
  )
df$in_advanced = as.logical(df$in_advanced)

# Clean "uni_year" column so that responses are not as long.
df = df %>%
  mutate(
    uni_year = case_when(
      grepl("First", uni_year) ~ "First Year",
      grepl("Second", uni_year) ~ "Second Year",
      grepl("Third", uni_year) ~ "Third Year",
      TRUE ~ uni_year
    )
  )

# Re-arranged column positioning so that like columns are near each other.
df = df[,c(1:6,7,27,8:15,28,16:22,29,23:26)]
```




# Introduction

This report analyses a subset of data collected from DATA2002 and DATA2902 (DATA2x02 hereafter) as part of an investigation into two main areas:

**Part A - COVID Tests Distribution**

Guiding research question: Does the number of COVID tests a student has taken in the past two months follow a Poisson distribution?

**Part B - Women in Data Science**

Guiding research questions: What is the gender distribution of Data Scientists in DATA2x02? Is there a distinction between the perceived graduate salaries in data science depending on gender?

---

# Data Source and Nature

### Data Collection

Google Forms

The data was collected via a short survey (run through Survey Monkey) through weeks 2 and 3 of Semester 2 (2021). A link to the survey to view its contents is provided here (https://docs.google.com/forms/d/1I3d2JY1IR32X-GbOWZxSfE9c5t-X9pEP8CoweZsyndo/viewform?edit_requested=true). Prior to the survey being released, students were invited to add questions of interest to ask their peers. After the survey was closed, the unit coordinator provided a link to the raw survey data, which was a table of rows with a schema, where each row represented one students entire set of responses to the questions in the survey.

### Data Nature

Each response can be treated as independent, as the survey was intended to be single response only. Further, on looking through the data, there does not seem to be any duplicate entries, further reinforcing the independence assumption.

(Assuming no siblings/close relatives in the sample)

Hide

```
# Get the number of standard students that responded to the survey
standard_respondents = df %>%
  filter(in_advanced == FALSE)
standard_respondents = nrow(standard_respondents)

# Get the number of advanced students that responded to the survey
advanced_respondents = df %>%
  filter(in_advanced == TRUE)
advanced_respondents = nrow(advanced_respondents)
```

### Respondents

There are currently 696 students enrolled in DATA2002 and 58 students enrolled in DATA2902. Of these students, only 169 DATA2002 students completed the survey, and only 32 DATA2902 students completed the survey.

could include response rate as a % to make the comparison easier.

### Data Limitations

not sure I'd characterise ~200 responses as a "very small number".

One of the key limitations of the data set is that it is not from a random sample. This is because from the entire population of DATA2x02 students (all of which were asked for a response), only a very small number responded. This means that selection bias can be introduced, as those that didn't respond may have represented a specific demographic of student. For example, we may expect the students which are really engaged with the course material to have noticed the survey from the unit coordinator, potentially making this demographic of student more likely to respond. This limitation is something that will be a key limitation when it comes to the hypothesis tests later in this report. ✓

Hide

```
# Get number of students not in Australia
international_students = df %>%
  filter(in_au == FALSE)
international_students = nrow(international_students)
```

**Variables Which Are Most Likely to be Subjected to Bias**

*What do you believe is the average entry salary in Australian Dollars of a data scientist who has just completed their undergraduate degree in data science?*

It is important to keep in mind before making generalisations, that only students enrolled in a data science course completed this survey. Many students will pick units based upon enjoyment of course material, as well as future job prospects. Because of this, data science students may perceive the starting salary as higher than those not in the data science field.

A further bias with this question is that 65 students are not currently in Australia. It would be safe to presume that for many of these students, they have lived in another country for most of their life. Because of differences in economies around the world, the salary international students provide may be heavily informed by the setting they were brought up in. ✓

*How many hours each week do you spend exercising?*

The variable which stems from this question is subject to bias as different people will define exercising differently. ✓

*How are you finding DATA2002 so far?*

Students in DATA2902 may feel excluded from answering this question, as the question specifically asks about DATA2002. OK

People might also respond more favourably than they really feel knowing it will be viewed by the teaching staff.

*How do you self assess your R coding ability? How do you self assess your mathematical ability?*

The variables which stem from these questions is potentially subject to question order bias. This is because, prior to these questions, the student is asked "How are you finding DATA2002 so far?", which gives the responded the options of "easy", "standard" or "difficult". If a student responds with "easy", they may feel more inclined to overstate their self assessment in R and mathematics.

The ordering of questions were randomised for each participant to try to mitigate any order bias.

# Data Cleaning

this was much more extensive data cleaning than we would ordinarily expect.

Before analysing, the entire data set was cleaned with the folded code chunk available at the top of this report. The entire data set was cleaned so the cleaned data could be used in the Shiny App part of the DATA2902 project (https://garthtarr.shinyapps.io/data2902studentB/).

There were many variables which required data cleaning, but below, only the variables used in this report will be mentioned:

**data_scientist_salary**

From Question: *What do you believe is the average entry salary in Australian Dollars of a data scientist who has just completed their undergraduate degree in data science?*

This variable required lots of cleaning due to the large array of formats which students used to answer this question. For example, responses included $100000, 100000, 100000AUD, 100000 AUD, 100k, 100000 p.a., mid100000, $10000 per month, $20 per h, etc.

In the survey, this question could have been improved by specifying that the user should enter a yearly salary, and only input the numerical value (i.e. no symbols). The survey could have then used validation to ensure import conformed with the suggested constraints.

This variable was cleaned by using regular expressions to match the different ways a user could write their response. From there, the numeric parts were extracted, and monthly, weekly and hourly rates were converted to a by year amount.

**gender**

From Question: *Gender*

This variable also required lost of cleaning due to the large array of responses which all meant the same value. For example, “M”, “Male”, “man”, all meant “Male”.

In the survey, this question could have been improved by actually writing a question rather than the one word “Gender”. Further, the issue of different values meaning the same thing could be reduced by having a few pre-defined options such as “Male”, “Female”, “Rather not say”, as well as the option to add in your own gender if the existing options do not suffice.

This variable was cleaned by using regular expressions inside a mutate combined with a case_when. If a value from a row matched the condition in the case_when block, that value would be over written into a consistent value, so that all values that mean the same thing have the same value. For example, “M” and “man” were converted to “Male”.

---

# Initial Data Analysis

Initial data analysis was conducted in the Shiny app here (https://garthtarr.shinyapps.io/data2902studentB/).

The data table can be found on the “Data Table” page (which can be selected from the left plane).

Visualisations of selected variables can be generated on the “Visualisations” page.

# Packages Used

**Tidyverse**

The tidyverse package was heavily utilised as part of the data cleaning phase. In particular, the pipe '%>%' operator, the mutate function and the case_when functions were heavily utilised in data cleaning. Further, the ggplot2 package which is a part of the tidyverse was utilised to create the plots and visualisations evident throughout this report (Wickham et al., 2019).

**Plotly**

The plotly packages was used to create the interactive visualisations (Sievert, 2020).

# Part A - COVID Tests Distribution

> **Guiding Research Question:** Does the number of COVID tests a student has taken in the past two months follow a Poisson distribution?

Receiving COVID tests has become such an essential part of combating the deadly COVID-19 virus (Hamzelou, 2020), that it is to no surprise that many global governments have been pushing for residents to test regularly, especially when there is fear of the disease spreading to new hosts via contact with an infected patient (Australian Government Department of Health, n.d.).

Because of this, it is to no surprise that of the 200 DATA2x02 students which completed the question regarding COVID tests in the survey, 80 students have had one or more COVID tests in the past two months.

The number of COVID tests which each student in DATA2x02 have undergone is summarised in Figure A.1.

Hide

```
# Get x ticks for graph
x_ticks = min(df$covid_tests, na.rm=T):max(df$covid_tests, na.rm=T)

# Generate bar plot of the number of COVID tests student in DATA2x02 have
 undergone
p = df %>% ggplot() +
            aes(x = covid_tests, fill = covid_tests) +
            geom_bar() +
            labs(x = "COVID tests in the past 2 months" , y = "Count", fil
l = "", title = "COVID Tests in the Last 2 Months") +
            scale_x_continuous(breaks = x_ticks) +
            theme_bw()

# Convert bar plot to interactive Plotly plot
ggplotly(p)
```

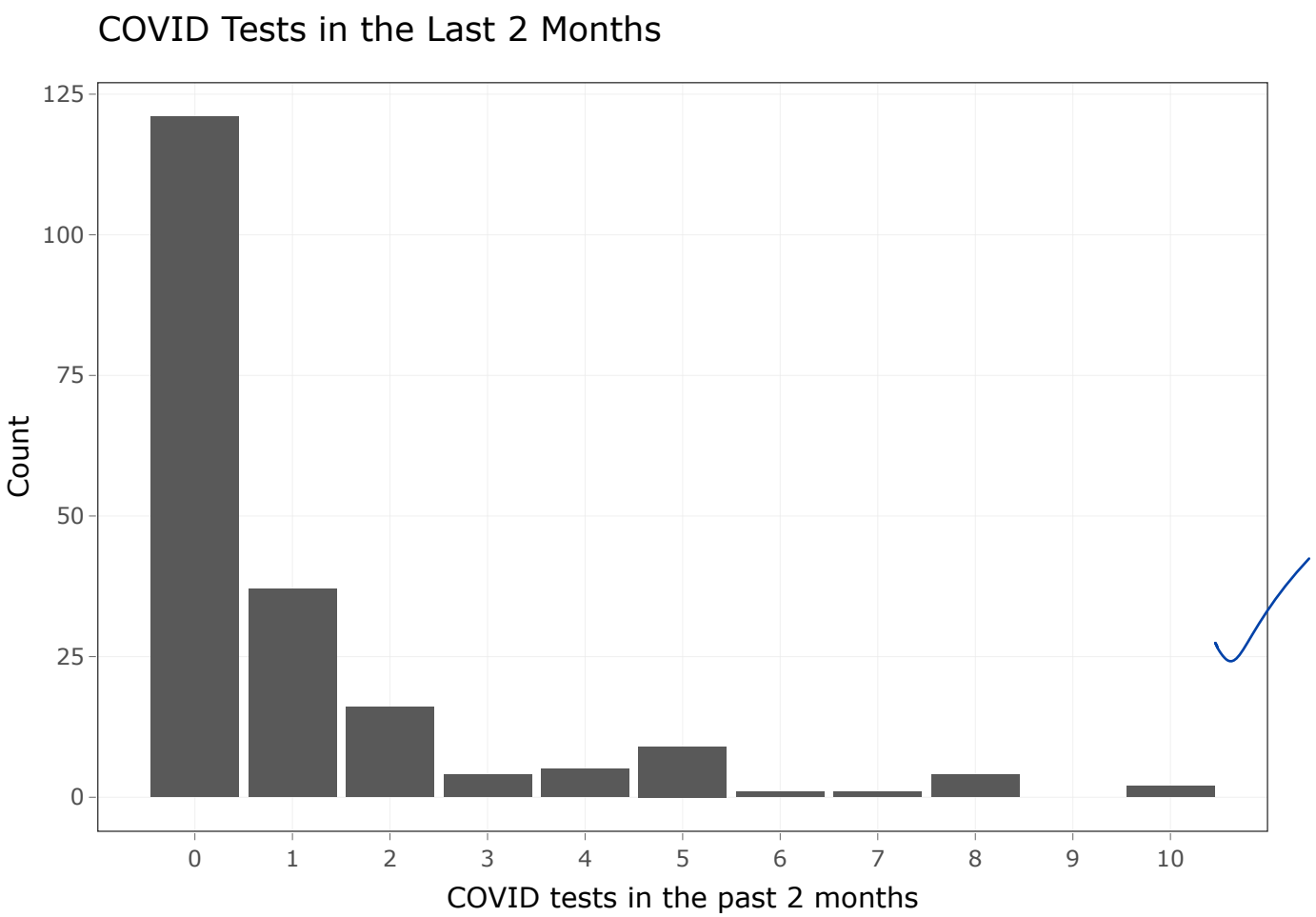



Figure A.1: The counts of COVID tests which students in DATA2x02 have undergone.

The distribution of data from Figure A.1 visually appears to be following a Poisson distribution, and so it was decided to test this observation using a Chi-Squared Goodness of Fit test.

A Poisson distribution is a discrete frequency distribution which provides the probability of a set number of independent events occurring in a fixed interval, where some known average rate per unit time ($\lambda$) is provided.

This is an appropriate distribution to test against as:

- the number of COVID tests is discrete.
- a fixed time interval (that being 2 months) was specified.
- $\lambda$ can be estimated by the mean number of COVID tests.

Hide

```
# Get "covid_tests" data into the correct format.
covid_data = df %>%
    select(covid_tests) %>%
    na.omit()
covid_data = covid_data[covid_data != ""]
covid_data = as.numeric(covid_data)

# Number of respondents for "covid_tests" column.
n = length(covid_data)

# Calculate lambda (aka the rate paramater).
# Because we have estimated this parameter using the sample data, we lose
 a further degree of freedom.
lam = mean(covid_data)

# Find Poisson probabilities.
p = dpois(min(covid_data):max(covid_data), lambda = lam)
r = p

# Redefine the 11th element P(>=10), NOT P(10).
p[max(covid_data) - min(covid_data) + 1] = 1 - sum(p[0:10])

# Calculate the expected frequencies.
ey = n * p
```

Before applying the hypothesis test, the expected counts for the Poisson distribution was calculated programatically, where $\lambda$ was estimated from the sample mean as:

$$\hat{\lambda} = \bar{x} = 1.028846$$

Figure A.2 shows the expected counts according to the Poisson distribution.

Hide

```
# Create a new data frame for the sole purpose of containing the expected
 values for graphing.
temporary_df = data.frame(ey, x_ticks)
colnames(temporary_df) <- c("expected", "index")

# Generate bar plot of the number of COVID tests student in DATA2x02 have
 undergone
p = temporary_df %>% ggplot() +
            aes(x = index, y = expected) +
            geom_bar(stat = 'identity') +
            labs(x = "COVID tests according to Poisson distribution" , y =
"Count", fill = "", title = "Expected Counts According to Poisson Distribu
tion") +
            scale_x_continuous(breaks = x_ticks) +
            theme_bw()

# Convert bar plot to interactive Plotly plot
ggplotly(p)
```

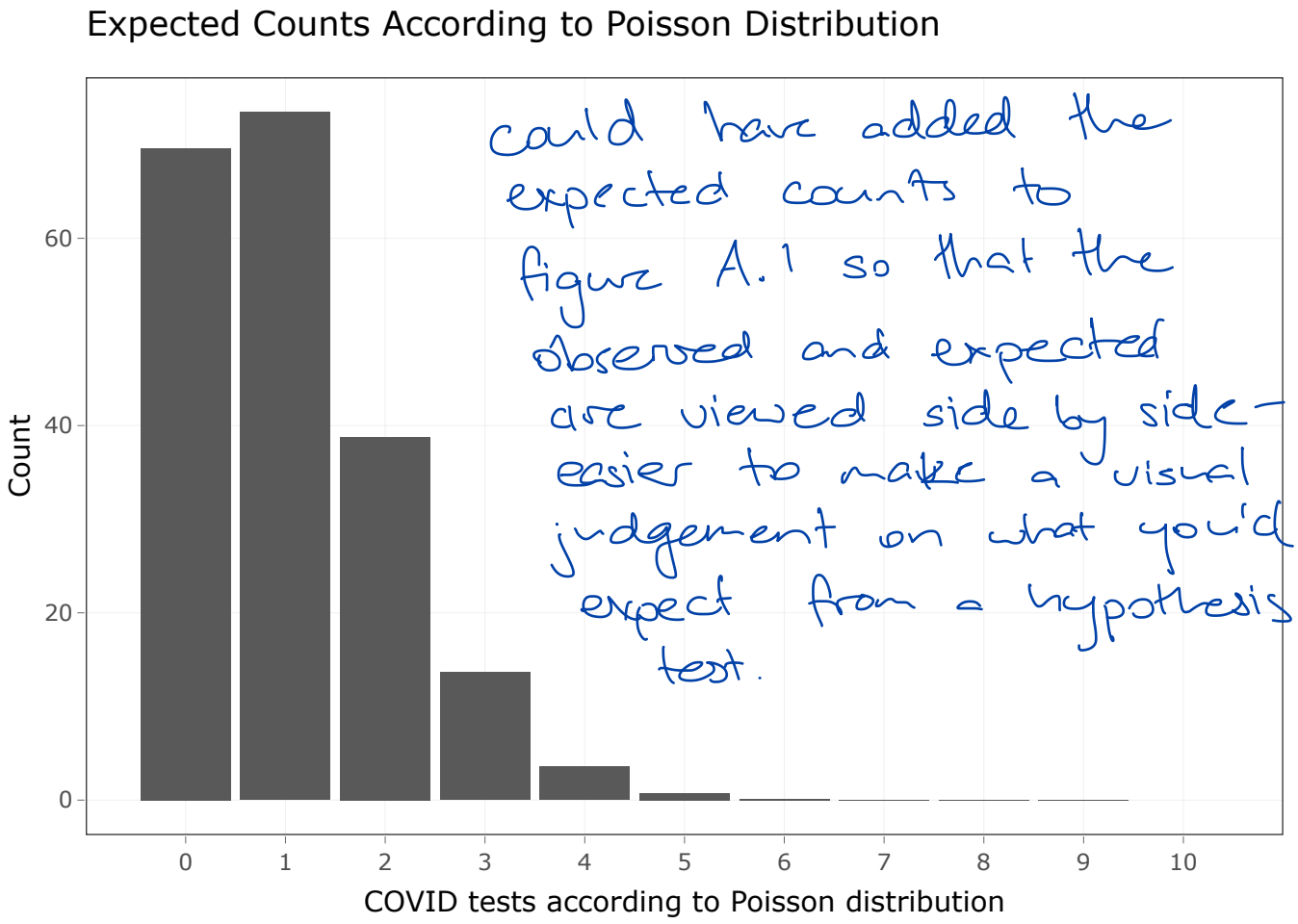



Figure A.2: The expected counts of COVID tests from a Poisson distribution with a rate parameter of 1.054

## Goodness of Fit Test Framework

**Hypothesis:**

$H_0$ (null hypothesis): The data comes from a Poisson distribution,

vs

$H_1$ (alternate hypothesis): The data do not come from a Poisson distribution.

**Assumptions:**

The assumption that observations are independent is satisfied under the data collection methodology.

The assumption that expected frequencies are $\geq 5$ is incorrect. This is visually evident in the small counts for 4 or more tests in Figure A.2. To satisfy this assumption, expected frequencies for occurrences of 4 or more tests were merged into the 3 tests column.

Hide

```
# Code to fix up assumption
adjusted_ey = c(ey[1:3], sum(ey[4:11]))

# Discrete counts
counts <- vector(mode = "numeric", length = 11 )
for (i in 1:n) {
  val = covid_data[i]
  counts[val+1] = counts[val+1] + 1
}

adjusted_covid_tests_sample = c(counts[1:3], sum(counts[4:11]))
```

**Test Statistic:**

$$T = \sum_{i=1}^{k} \frac{(Y_i - np_i)^2}{np_i}$$

Under $H_0, T \sim \chi^2_2$.

Note that there are two degrees of freedom. This is because after adjusting the frequencies to satisfy the expected frequencies assumption, we had 4 groups. We lose one degree of freedom by definition, and we lose another degree of freedom because $\lambda$ was estimated using the sample mean.

Hide

```
t0 = sum((adjusted_covid_tests_sample - adjusted_ey)^2/adjusted_ey)
```

**Observed Test Statistic:**

The observed test statistic was calculated programatically. It was found that $t_0 = 70.90525$.

Hide

```
pval = 1 - pchisq(70.90525, df = 2)
```

**P-value:**

$$P(T \geq t_0) = P(\chi_2^2 \geq 70.90525) = 4.440892e\text{-}16$$

This p-value was found by finding the area under the curve, which is seen in Figure A.3, which utilises a function by Garth (2021).

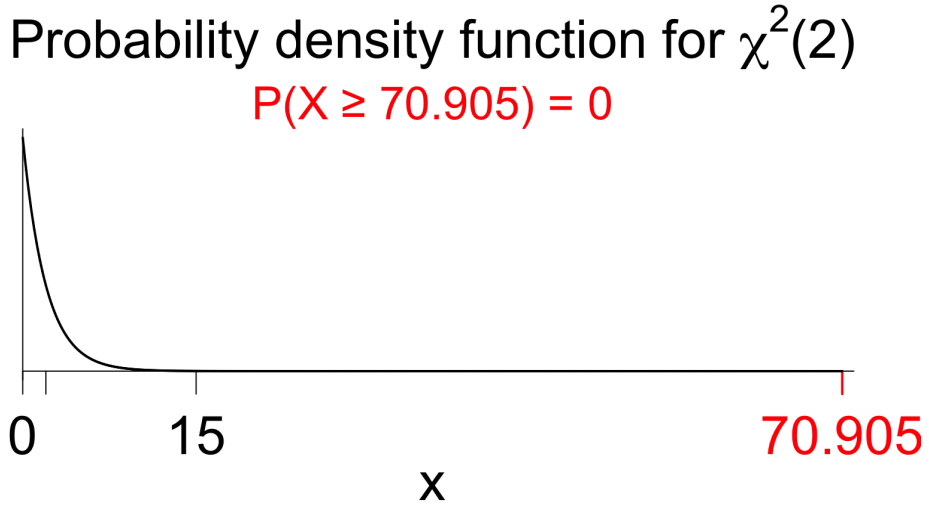
$< 0.001$

nicer to write it like this

Hide

```
chisquaredplot = function(t0,df,prob_statement=NULL,tail="lower",cex = 2,
 upper_quantile = 0.9998){
  x=LB=UB=NA
  LB = 0
  if(!is.na(t0)){
    UB = max(qchisq(upper_quantile,df),t0)
    x <- seq(LB, UB, by = .001)
    xlims = c(LB,UB)
  } else {
    UB = qchisq(upper_quantile,df)
    x <- seq(LB, UB, by = .001)
    xlims = c(LB,UB)
  }
  y <- dchisq(x, df)
  ylims = NULL
  if(df<=2){
    ylims = c(0,0.5)
  }

  if(is.null(prob_statement)){
    if(tail=="lower"){
      prob_statement = paste("P(X \u2264 ",round(t0,3),") = ",
                             round(pchisq(t0,df,lower.tail=TRUE),4),sep=""
)
    } else {
      prob_statement = paste("P(X \u2265 ",round(t0,3),") = ",
                             round(pchisq(t0,df,lower.tail=FALSE),4),sep=
"")
    }
  }

  par(mgp=c(3,1,0), mar = c(3, 1, 4, 1), cex = cex)
  plot(x, y, type = "n", axes = FALSE,
       main=bquote(paste("Probability density function for ",chi^2,"(",.(p
aste0(df)),")",sep="")),
       xlab = NA, ylab = NA, lwd = 2, xlim = xlims, ylim=ylims,
       cex=1.4,cex.lab=1.4,cex.axis=1.4,cex.main=1.4)
  axis(side = 1, at = c(0, df, round(qchisq(0.999,df)+1,0)),
       xlim = xlims, pos = 0 , cex.axis=1.4)
  title(xlab = "x",cex.lab=1.4,mgp=c(1.8,1,0))
  abline(v=0)

  if(!is.na(t0)){
    axis(side = 1, at = round(t0,3), pos = 0, col = "red",
         col.axis = "red", lwd = 2, cex.axis=1.4)
    if(tail=="lower"){
      #shade area to left of obtained test statistic:
      seqs = seq(0, t0, by = 0.001)
      cord.x <- c(0, seqs, t0)
      cord.y <- c(0, dchisq(seqs, df), 0)
      cord.y[cord.y>0.5] = 1
```

```
      polygon(cord.x, cord.y, col= rgb(1,0,0,alpha=0.5))
      segments(round(t0,3), 0, round(t0,3), dchisq(t0, df), col = "red", l
wd = 2)
    } else {
      #shade area to right of obtained test statistic:
      upper = max(UB,t0+1)
      seqs = seq(t0, upper, by = 0.001)
      cord.x <- c(t0, seqs, upper)
      cord.y <- c(0, dchisq(seqs, df), 0)
      cord.y[cord.y>0.5] = 1
      polygon(cord.x, cord.y, col= rgb(1,0,0,alpha=0.5))
      segments(round(t0,3), 0, round(t0,3), dchisq(t0, df), col = "red", l
wd = 2)
    }
    mtext(prob_statement,side=3,col = "red",cex=1.2*cex)
  }
  lines(x, y, xlab = NA, ylab = NA, lwd = 2)
}

chisquaredplot(70.90525, 2, tail = "upper")
```




not really necessary.

Figure A.3: Graphical visualisation depicting how the area under the chi-squared curve was utilised to find the p-value. The p-value in red is not actually 0, rather the caption has rounded the p-value down to 0 as the actual value is very close to 0.

**Decision:**

Since the p-value is below the 0.05 threshold, we reject $H_0$.

## Conclusion

As we reject the null hypothesis, this means that the number of COVID tests in a 2 month period for our survey sample is not consistent with a Poisson distribution. ✓

Independence was satisfied but can we make strong conclusions about the whole DATA2x02 population given the sample may not be truly representative?

# Part B - Women in Data Science

> **Guiding Research Questions:** What is the gender distribution of Data Scientists in DATA2x02? Is there a distinction between the perceived graduate salaries in data science depending on gender?

### Introduction

Technology has revolutionised the jobs of the future, as there has been a seismic shift towards graduates seeking to enter fields centred around technology. In particular, jobs in data science and data analytics have seen a large increase in popularity, especially because of their high relevance in a technological world. In fact, Seet and Jones from Australia's 'The Conversation' (2020) attribute the growth of data science not only to its relevance, but also that it is a "job with a future" with "high growth prospects and a low risk of automation".

This research aims to investigate questions surrounding the gender of students studying data science, as well as the perceived salaries that a job in data scientist provides, and how this perception relates to gender.

## Women in Data Science

Whilst specific data for the gender distribution of data science jobs has not been located, it is still well known that in the computer science-related sphere, women represent a very small proportion of the workforce.

This is evident in Daley's (2021) article which shares some statistics which shed some light towards this issue, such as:

- "Women currently remain highly underrepresented in … computer science-related jobs (25% of total workforce)."
- "66% of women report that there is no clear path forward for them in their career at their current companies."
- "39% of women view gender bias as a primary reason for not being offered a promotion."

Whilst not all students enrolled in DATA2x02 will be aiming for a computer or data science-related job, it is still sensible to compare the gender distribution of students studying this course with Daley's statistic that only 25% of computer-science related jobs are represented by women.

To test this hypothesis, a Chi-squared goodness of fit test was conducted. This test will compare two samples:

1. The students that they identified as female.
2. The students that identified as a gender OTHER than female.

We will also be checking whether the split between "Female" and Other genders is indeed a 25:75 split.

hypothesised due to Daley (2021)?

## Goodness of Fit Test Framework

Hide

```
# Extract gender data
gender_data = df %>%
    select(gender) %>%
    na.omit()
covid_data = covid_data[covid_data != ""]
covid_data = as.numeric(covid_data)

# Clean so that any response that is not "Female" is encoded as "Other"
gender_data = gender_data %>%
  mutate(gender = toupper(gender)) %>%
  mutate(
    gender = case_when(
      grepl("FEMALE", gender) ~ "Female",
      TRUE ~ "Other"
    )
  )
```

### Hypothesis

$H_0$ (null hypothesis): $p_{female} = 0.25, p_{other} = 0.75.$

vs

$H_1$ (alternate hypothesis): at least one equality does not hold.

### Assumptions

The assumption that observations are independent is satisfied under the data collection methadology.

Hide

```
# Checking assumption that expected frequencies are >= 5
p = c(0.25, 0.75)

# Number of people who filled out this question
n = nrow(gender_data)

# Expected counts
ey = n * p

# Check cell counts are >= 5 (un-comment for code to run showing expected
 cell counts)
# ey >= 5
```

The assumption that expected frequencies are $\geq 5$ is satisfied. This is because under $H_0$, we expect 52.75 women to study DATA2x02, and 158.25 women to study DATA2x02, and both values are clearly $\geq 5$.

**Test Statistic:**

$$T = \sum_{i=1}^{k} \frac{(Y_i - np_i)^2}{np_i}$$

Under $H_0, T \sim \chi^2_1$.

Note that there is only 1 degree of freedom, because there were only two caregories to start with (Female and Other), and no parameters were estimated (rather they were found from literature).

Hide

```
# Observed counts
counts <- vector(mode = "numeric", length = 2)
for (i in 1:n) {
  string = gender_data$gender[i]
  if (string == "Female") {
    counts[1] = counts[1] + 1
  } else {
    counts[2] = counts[2] + 1
  }
}

t0 = sum((counts - ey)^2/ey)
```

counts[1] = sum(gender_data$gender == "female")

counts[2] = sum(gender_data$gender != "female")

totally unnecessary loop.

**Observed Test Statistic:**

The observed test statistic was calculated programatically. It was found that $t_0 = 13.66351$.

Hide

```
pval = 1 - pchisq(13.66351, df = 1)
```

**P-value:**

$$P(T \geq t_0) = P(\chi^2_2 \geq 202.11) = 0.0002186624$$

This p-value was found by finding the area under the curve, which is seen in Figure B.1, which utilises a function by Garth (2021).

Hide

```
chisquaredplot(13.66351, 1 ,tail = "upper")
```

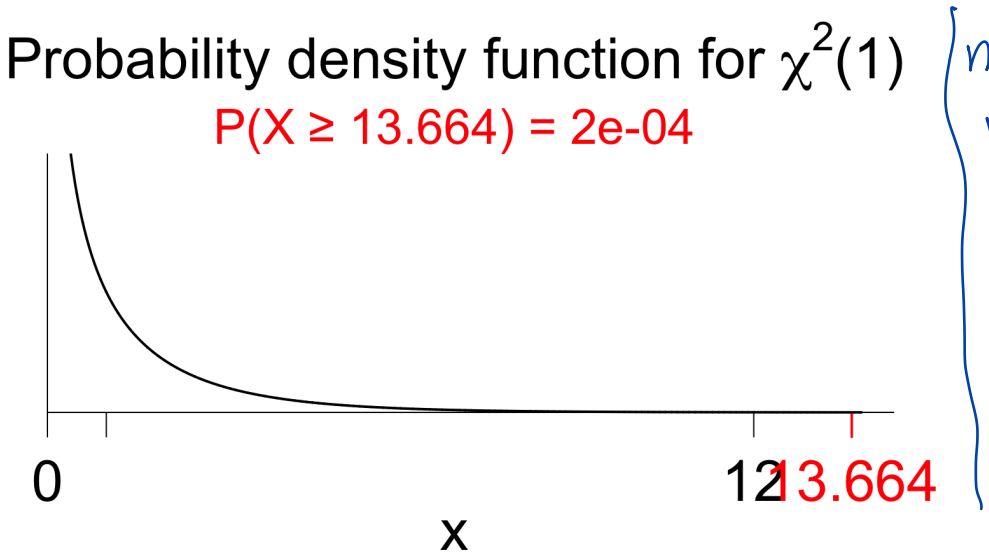



Figure B.1: Graphical visualisation depicting how the area under the chi-squared curve was utilised to find the p-value.

**Decision:**

Since the p-value is below the 0.05 threshold, there is strong evidence in the data against $H_0$.

### Conclusion

Despite concerns that in industry, women only account for 25% of computer-science related jobs, this does not hold among the people which responded to the DATA2x02 survey according to the Chi-squared goodness of fit test. In fact, 36% of DATA2x02 students identify as female, which is much larger than 25%.

# Expected Data Scienctist Salary

Before looking at the difference between the expected starting salaries for different genders in data science, we will first look at the overall perception of what the survey respondents think is a graduate data scientist salary.

We can investigate this by comparing the mean response from the survey data with the average salary of a graduate data scientist. Whilst the salary a graduate makes will vary, the average salary that will be used for the purpose of this test is $65,000 AUD. This is based off the average of five entry salaries found on Glassdoor (n.d.)

To investigate this, a one-sample t-test will be conducted, as evident in the hypothesis test framework below:

### One Sample t-test Framework

**Hypothesis:**

$H_0$ (null hypothesis): $\mu = 65000$

vs

$H_1$ (alternate hypothesis): $\mu \neq 65000$

**Assumptions:**

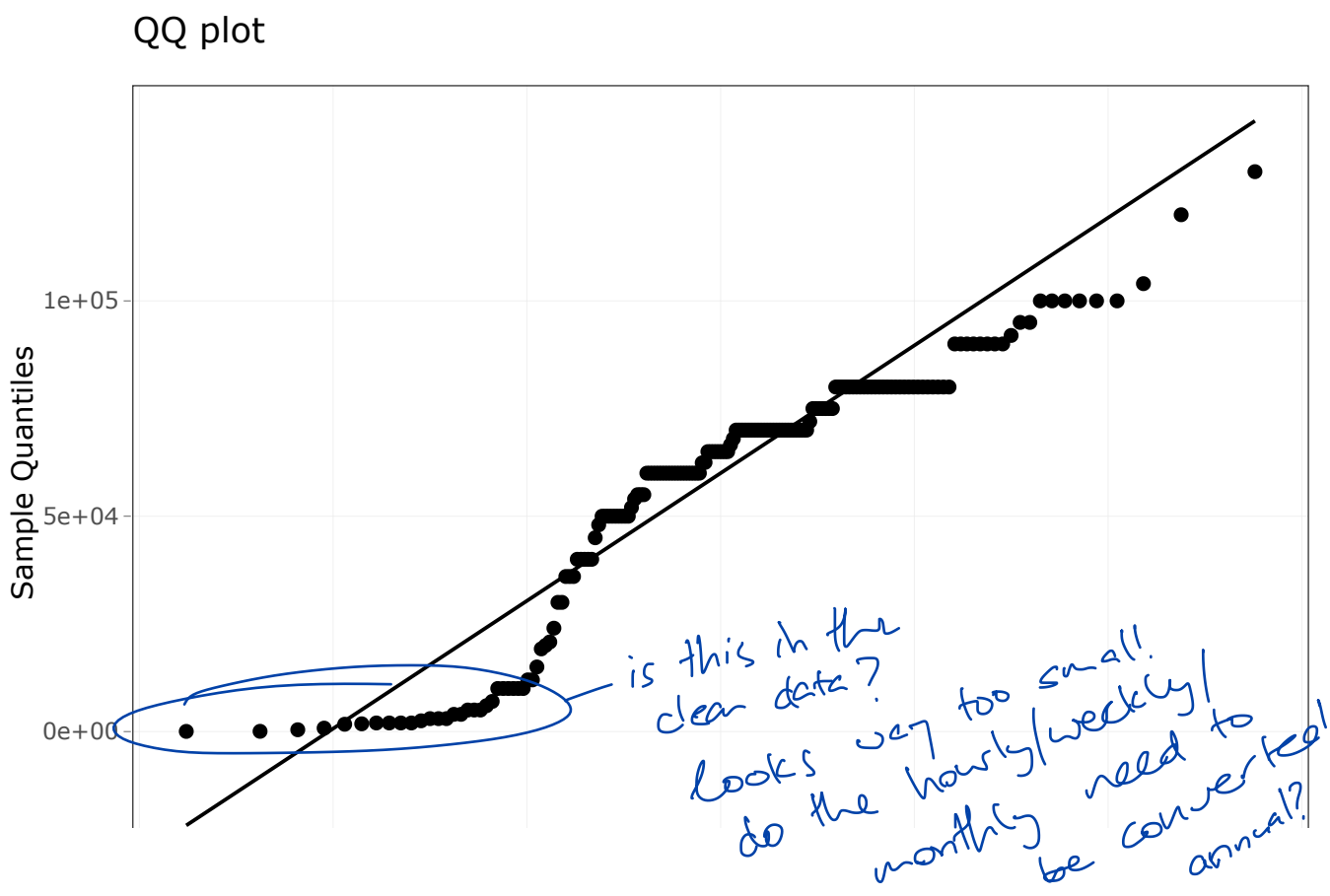
The assumption that the sample was chosen at random is not satisfied, as explained in "Data Limitations" part of the "Data Source and Nature" section. This represents a key limitation of this t-test, and is something to keep in mind.

The assumption that the data is independently and identically distributed is satisfied under the data collection methodology.

To check the normality assumption, Figure B.2 (a QQ plot) was constructed.

Hide

```
# Code to make QQ plot
p <- df %>%
    ggplot() +
    aes(sample = data_scientist_salary) +
    stat_qq() +
    stat_qq_line() +
    xlab("Theoretical Quantiles") +
    ylab("Sample Quantiles") +
    ggtitle("QQ plot") +
    theme_bw()
ggplotly(p)
```



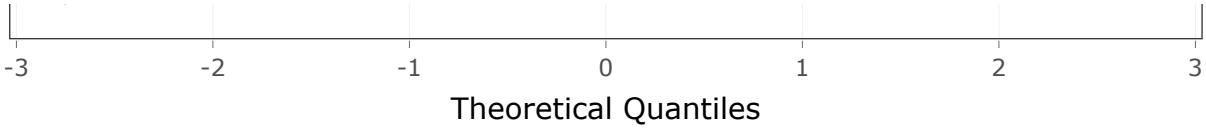



Figure B.2: QQ plot of the survey responses for the graduate data scientist salary question. As is evident by the deviation of the points from the line, this data does not appear to be normally distributed.

Hide

```
# Extract salary data
salary_data = df %>%
    select(data_scientist_salary) %>%
    na.omit()
salary_data = salary_data[salary_data != ""]
salary_data = as.numeric(salary_data)

mean_salary = mean(salary_data)

# Number of respondents for salary question
n = length(salary_data)
#n
```

Although in Figure B.2 it appears that the data does not follow a normal distribution because of the deviation from the line, the Central Limit Theorem applies here because of the sufficiently large sample size ($n = 177$). This means that the sample's mean will come from a normal distribution, regardless of the original data's distribution. Hence, the normality assumption is also satisfied.

**Test Statistic:**

$T = \frac{\bar{X} - \mu_0}{S/\sqrt{n}}$. Under $H_0, T \sim t_{176}$

Hide

```
t0 = t.test(salary_data, mu = 65000)$statistic
```

**Observed Test Statistic:**

The observed test static was found programatically. It was found to be $t_0 =$ -3.352237

Hide

```
p = t.test(salary_data, mu = 65000)$p.value
```

**P-Value**

$2P(t_{176} \geq |t_0|)$ = 0.000981274

**Decision**

As the p-value is less than 0.05, there is evidence against $H_0$.

### Conclusion

The very small p-value of the t-test indicates that there is evidence that the sample mean graduate data scientist salary is not equal to $65000 AUD.

What was the observed sample mean?

A boxplot of the data could have been included in the report (not just in the shiny app).

# Salary Perceptions with Gender

It has been well documented that there has historically been a large gender pay gap between men and women. According to a publication from the 27th of August, 2021, the Australian Governments Workplace Gender Equality Agency notes that the current national gender pay gap is 14.2%. Whilst Australian sources have been difficult to find, a United States article states that in 2019, the gender pay gap for women is 8.4% in data science (2019).

With such a large gap between salary, this section aims to understand whether the perceptions of how much a female thinks she will earn as a data science graduate is less than how much the other genders think.

To do this, the salary data was separated into two independent samples based on gender. The expected salaries of respondents who identified as “female” were put together, and the expected salaries of everyone who did not identify as “female” were put together. Figure B.3 shows the outcome of this graphically by using box plots.

Hide

Ideally would also report the sample mean income for the two populations in the text.

```
# Data for students that identified as "female"
female_salaries = df %>%
      select(c(gender, data_scientist_salary)) %>%
      na.omit()
female_salaries <- filter(female_salaries, female_salaries[1] == "Female")

# Data for students that did not identify as "female"
not_female_salaries = df %>%
      select(c(gender, data_scientist_salary)) %>%
      na.omit()
not_female_salaries <- filter(not_female_salaries, not_female_salaries[1]
 != "Female")

female_salaries['sample'] = "Female"
not_female_salaries['sample'] = "Other"

merge = bind_rows(female_salaries, not_female_salaries)

# Create box plot
p <- merge %>%
    ggplot() +
    aes(x = data_scientist_salary, y = sample) +
    geom_boxplot() +
    xlab("Graduate Salaries") +
    ylab("Samples") +
    ggtitle("Boxplot of Expected Graduate Salaries Categorised by Gender")
+
    theme_bw()

p + scale_x_continuous(labels = scales::dollar_format())
```

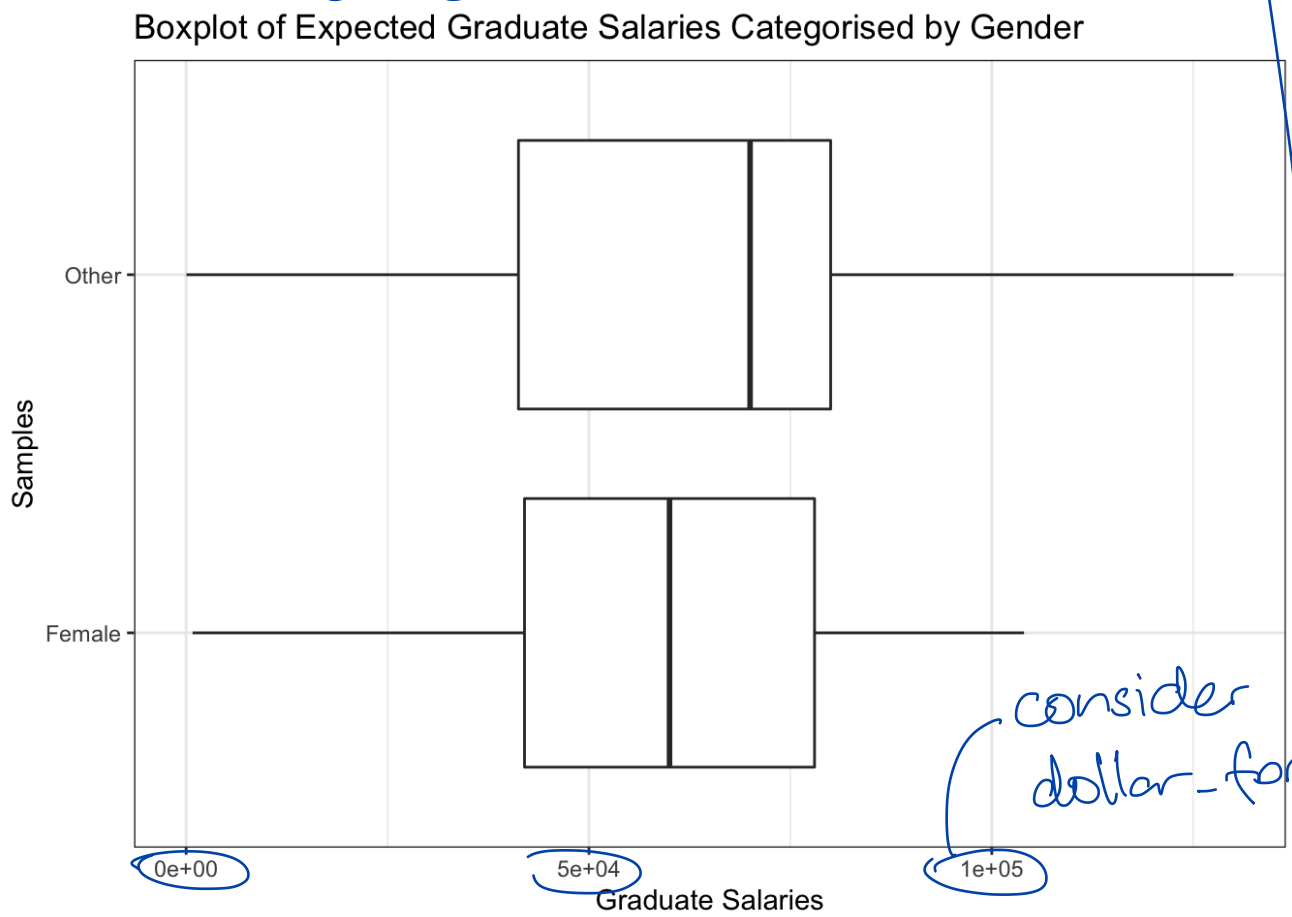



Figure B.3: Boxplots showing the range of reponses for graduate salaries for females and other groups combined.

To investigate this, a Welch two-sample t-test was conducted, as seen in the hypothesis framework below:

## Welch Two-Sample t-test Framework

**Hypothesis:**

$H_0$ (null hypothesis): $\mu_{female} = \mu_{other}$

vs

$H_1$ (alternate hypothesis): $\mu_{female} < \mu_{other}$

**Assumptions:**

The assumption that the sample was chosen at random is not satisfied, as explain in the initial parts of the report.

The assumption that the data is independently and identically distributed is satisfied under the data collection methodology.

The assumption that the samples are independent is also satisfied. This is because a respondent can only be assigned one distinct group (i.e. the groups are mutually exclusive).

Note that the normality assumption is not required in a Welch two-sample t-test. if the sample size is large enough such that we can rely on the CLT for the approximate distribution of the test statistic.

**Test Statistic:**

The Welch statistic is given by the expression: $\frac{\bar{X}-\bar{Y}}{\sqrt{\frac{S_X^2}{n_X}+\frac{S_Y^2}{n_Y}}}$.

(distribution?)

Hide

```
# Code to find test statistic
t0 = t.test(female_salaries$data_scientist_salary, not_female_salaries$dat
a_scientist_salary, alternative = "less")$statistic
```

**Observed Test Statistic:**

The observed test static was found programatically. It was found $t_0 = -0.819458$

Hide

```
# Code to find test statistic
p = t.test(female_salaries$data_scientist_salary, not_female_salaries$data
_scientist_salary, alternative = "less")$p.value
```

**P-Value**

The p-value was determined programatically. It was found $p = 0.2070323$

**Decision**

As the p-value is greater than 0.05, ~~there is evidence for $H_0$~~. the data are consistent with $H_0$.
OR
there is no evidence against $H_0$.
(it's a subtle difference in terminology)

## Conclusion

The p-value being above 0.05 suggests that the average expected salary indicated by respondents who identified as Female is not ~~significantly~~ any less than the other genders combined.

→ worth including an overall summary/conclusion at the end reiterating key findings and limitations.

# References

Australian Government Department of Health (n.d.) What you need to know about coronavirus (COVID-19) https://www.health.gov.au/news/health-alerts/novel-coronavirus-2019-ncov-health-alert/ongoing-support-during-coronavirus-covid-19/what-you-need-to-know-about-coronavirus-covid-19 (https://www.health.gov.au/news/health-alerts/novel-coronavirus-2019-ncov-health-alert/ongoing-support-during-coronavirus-covid-19/what-you-need-to-know-about-coronavirus-covid-19)

Bayern, M. (2019) The data science gender pay gap is shrinking—barely. https://www.techrepublic.com/article/the-data-science-gender-pay-gap-is-shrinking-barely/ (https://www.techrepublic.com/article/the-data-science-gender-pay-gap-is-shrinking-barely/)

Sievert, C. (2020) Interactive Web-Based Data Visualization with R, plotly, and shiny. Chapman and Hall/CRC Florida

Daley, S. (2021) Women in Tech Statistics Show the Industry Has a Long Way to Go. https://builtin.com/women-tech/women-in-tech-workplace-statistics (https://builtin.com/women-tech/women-in-tech-workplace-statistics)

Glassdoor. (n.d.) How much does a Graduate Data Scientist make? https://www.glassdoor.com.au/Salaries/graduate-data-scientist-salary-SRCH_KO0,23.htm (https://www.glassdoor.com.au/Salaries/graduate-data-scientist-salary-SRCH_KO0,23.htm)

Hamzelou, J. (2020) WHO expert: We need more testing to beat coronavirus. https://www.newscientist.com/article/2237544-who-expert-we-need-more-testing-to-beat-coronavirus/ (https://www.newscientist.com/article/2237544-who-expert-we-need-more-testing-to-beat-coronavirus/)

Seet, P., Jones, J. (2020) If you're preparing students for 21st century jobs, you're behind the times. https://theconversation.com/if-youre-preparing-students-for-21st-century-jobs-youre-behind-the-times-131567 (https://theconversation.com/if-youre-preparing-students-for-21st-century-jobs-youre-behind-the-times-131567)

Tarr, G (2021). DATA2002 Data Analytics: Learning from Data. University of Sydney, Sydney Australia.

Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686 (https://doi.org/10.21105/joss.01686)

good to cite packages, could also cite R itself.

Workplace Gender Equality Agency. (2021) Australia's Gender Pay Gap Statistics. https://www.wgea.gov.au/publications/australias-gender-pay-gap-statistics (https://www.wgea.gov.au/publications/australias-gender-pay-gap-statistics) ***

A well written report — very concise. A little more information could have been given (for example summary statistics along with the t-tests and a boxplot (or similar) for the one sample t-test - possibly including the hypothesised mean.) Well referenced and good use of external sources going above and beyond in that respect. Good job!