ST661: Exam December 2023 Part B

Saad Siddiqui 2325668

- Download fin23.Rmd from Moodle and fill in your answers.
- Place your name and student number in the space for `author:`.
- Upload the html file produced by knitr to Moodle before 3.20pm. There will be a penalty of 10 marks for not uploading the html file.
- Do not include in your html file long data listings. There will be a 5 mark penalty for this.
- If your code for any answer does not run, use `{r, eval=FALSE}` for your code chunk. If the answer to a question relates to the answer to a previous question that you did not complete, you may still give code in an `eval=FALSE` code chunk.

```
library(tidyverse)
library(dplyr)
library(ggplot2)
```

Question 1 (48 marks, 8 marks per part)

The file provided on Moodle `children.Rdata` gives information on the age, height, weight and BMI of children measured multiple times. Download the data and place in the same folder as your Rmarkdown file. Access the data with

```
load("children.Rdata") # gives dataset called kids
```

Write code for each of the following. You can use base R, tidyverse or a mix.

1a. Change the class of the date variable to be class `POSIXct`. Show the result of `glimpse(d)` to verify your code.

```
#1a


str(kids)
```

```
spc_tbl_ [5,683 x 9] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ ID     : num [1:5683] 15 15 15 15 15 15 15 15 15 15 ...
 $ date   : chr [1:5683] "2008-04" "2008-10" "2009-04" "2009-10" ...
 $ age    : num [1:5683] 13.2 13.6 14.2 14.6 15.2 ...
 $ age.bin: num [1:5683] 13 13 14 14 15 15 16 16 17 17 ...
 $ gender : chr [1:5683] "boy" "boy" "boy" "boy" ...
 $ height : num [1:5683] 162 165 168 170 172 173 172 173 173 173 ...
 $ weight : num [1:5683] 49 51 55 55 57 62 62 64 64 66 ...
 $ BMI    : num [1:5683] 18.7 18.7 19.5 19 19.3 ...
```

```
 $ BMIcat : chr [1:5683] "normal" "normal" "normal" "normal" ...
 - attr(*, "spec")=
  .. cols(
  ..   ID = col_double(),
  ..   date = col_character(),
  ..   age = col_double(),
  ..   age.bin = col_double(),
  ..   gender = col_character(),
  ..   height = col_double(),
  ..   weight = col_double(),
  ..   BMI = col_double(),
  ..   BMIcat = col_character()
  .. )
 - attr(*, "problems")=<externalptr>
```

```
# The date wasnt in the correct format as POSIxct so we had to mutate it and add a day, So I added the

kids <- mutate(kids, date = paste(date, "-01", sep = ""))  # Adding day component as 01

str(kids)
```

```
tibble [5,683 x 9] (S3: tbl_df/tbl/data.frame)
 $ ID     : num [1:5683] 15 15 15 15 15 15 15 15 15 15 ...
 $ date   : chr [1:5683] "2008-04-01" "2008-10-01" "2009-04-01" "2009-10-01" ...
 $ age    : num [1:5683] 13.2 13.6 14.2 14.6 15.2 ...
 $ age.bin: num [1:5683] 13 13 14 14 15 15 16 16 17 17 ...
 $ gender : chr [1:5683] "boy" "boy" "boy" "boy" ...
 $ height : num [1:5683] 162 165 168 170 172 173 172 173 173 173 ...
 $ weight : num [1:5683] 49 51 55 55 57 62 62 64 64 66 ...
 $ BMI    : num [1:5683] 18.7 18.7 19.5 19 19.3 ...
 $ BMIcat : chr [1:5683] "normal" "normal" "normal" "normal" ...
```

```
# POSIXct converted
kids$date <- as.POSIXct(kids$date, format = "%Y-%m-%d")




glimpse(kids)
```

```
Rows: 5,683
Columns: 9
$ ID      <dbl> 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 21, 21, 21, 21~
$ date    <dttm> 2008-04-01, 2008-10-01, 2009-04-01, 2009-10-01, 2010-04-01, 2~
$ age     <dbl> 13.15, 13.65, 14.15, 14.65, 15.15, 15.65, 16.15, 16.65, 17.15,~
$ age.bin <dbl> 13, 13, 14, 14, 15, 15, 16, 16, 17, 17, 18, 18, 10, 11, 11, 12~
$ gender  <chr> "boy", "boy", "boy", "boy", "boy", "boy", "boy", "boy", "boy",~
$ height  <dbl> 162, 165, 168, 170, 172, 173, 172, 173, 173, 173, 173, 174, 15~
$ weight  <dbl> 49, 51, 55, 55, 57, 62, 62, 64, 64, 66, 66, 67, 59, 62, 70, 77~
$ BMI     <dbl> 18.67, 18.73, 19.49, 19.03, 19.27, 20.72, 20.96, 21.38, 21.38,~
$ BMIcat  <chr> "normal", "normal", "normal", "normal", "normal", "normal", "n~
```

1b. Find the IDs of the children whose age.bin is 12 at the start of the study (the earliest date)

```
#1b



earliest_date <- min(kids$date)

# Filter IDs of children whose age.bin is 12 at the start of the study (earliest date)
ids12 <- kids$ID[kids$date == earliest_date & kids$age.bin == 12]

ids12
```

```
 [1]   121   179   436   454   468  1573  3818  3873  3877  3892  4142  4348
[13]  4493  4630  7367  9781 10105
```

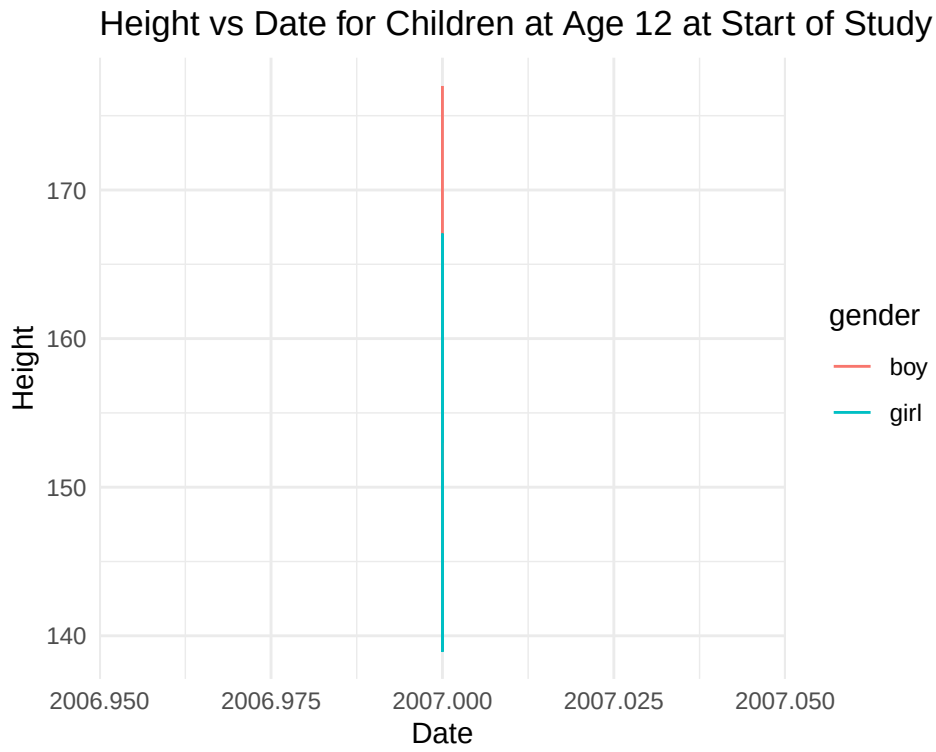
1c. For the children whose age.bin is 12 at the start of the study, plot `height` versus `date` as a line plot. Colour the lines by gender.

```
#1c

# for children whose age  is 12 at the start of the study
kids12_start <- subset(kids, age.bin == 12 & date == min(date))


Year <- year(kids12_start$date)

ggplot(kids12_start, aes(x =  Year , y = height, color = gender)) +
  geom_line() +
  labs(x = "Date", y = "Height") +
  ggtitle("Height vs Date for Children at Age 12 at Start of Study") +
  theme_minimal()
```



1d. Make a subset of the data consisting of kids whose `age.bin` is 12. For children who appear multiple times in this subset, remove all but the first occurrence. Call this subset `kids12`. Show the result with `glimpse(kids12)`.

```
#1d


# 12 year filter on kids
kids12 <- kids %>%
  filter(age.bin == 12) %>%
  distinct(ID, .keep_all = TRUE)

# Show the result with glimpse
glimpse(kids12)
```

```
Rows: 270
Columns: 9
$ ID      <dbl> 21, 49, 113, 121, 138, 142, 149, 167, 179, 218, 224, 225, 280,~
$ date    <dttm> 2009-04-01, 2009-10-01, 2008-04-01, 2007-10-01, 2012-10-01, 2~
$ age     <dbl> 12.24, 12.29, 12.80, 12.75, 12.28, 12.00, 12.13, 12.35, 12.04,~
$ age.bin <dbl> 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12~
$ gender  <chr> "girl", "boy", "boy", "boy", "boy", "boy", "girl", "girl", "bo~
$ height  <dbl> 161, 155, 145, 177, 152, 157, 153, 157, 144, 153, 162, 149, 14~
$ weight  <dbl> 77, 38, 48, 55, 41, 58, 40, 53, 59, 36, 49, 36, 39, 45, 38, 41~
$ BMI     <dbl> 29.71, 15.82, 22.83, 17.56, 17.75, 23.53, 17.09, 21.50, 28.45,~
$ BMIcat  <chr> "obese", "normal", "overweight", "normal", "normal", "overweig~
```

1e. Using `kids12` , find the proportion of boys and girls in each of the `BMIcat` groups. Omit those whose `BMIcat` is `NA` from the calculation.

```r
#1e

# the formula of proportion is count over sum of total count

BMIcat_filtered <- kids12 %>%
  filter(!is.na(BMIcat))


proportion_by_BMIcat <- BMIcat_filtered %>%
  group_by(BMIcat, gender) %>%
  summarise(count = n()) %>%
  group_by(BMIcat) %>%
  mutate(proportion = count / sum(count))

print(proportion_by_BMIcat)
```

```
# A tibble: 10 x 4
# Groups:   BMIcat [5]
   BMIcat        gender count proportion
   <chr>         <chr>  <int>      <dbl>
 1 normal        boy       69      0.457
 2 normal        girl      82      0.543
 3 obese         boy       27      0.643
 4 obese         girl      15      0.357
 5 overweight    boy       28      0.571
 6 overweight    girl      21      0.429
 7 severely thin boy        1      0.333
 8 severely thin girl       2      0.667
 9 thin          boy        4      0.8
10 thin          girl       1      0.2
```

1f. Make a barplot showing the proportion of boys and girls in `kids12` in each of the BMI categories. Omit those whose `BMIcat` is `NA`. Make sure that the `BMIcat` labels are in order of increasing `BMIcat`.

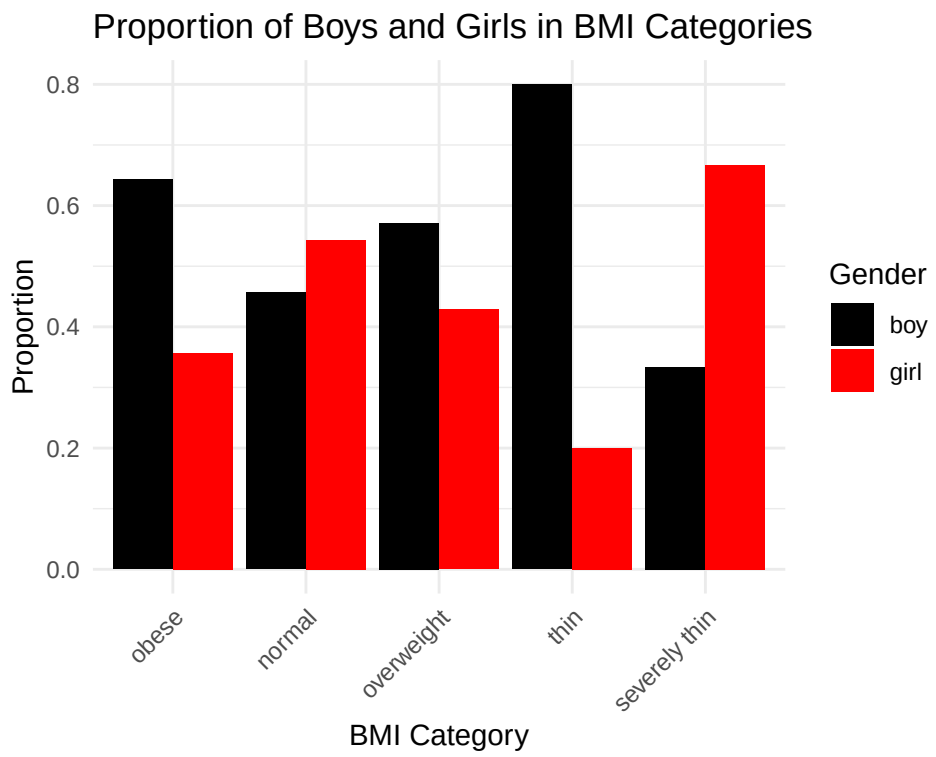
```r
#1f

library(ggplot2)


filtered_data <- kids12 %>%
  filter(!is.na(BMIcat)) %>%
  mutate(BMIcat = factor(BMIcat, levels = unique(BMIcat)))


proportion_by_BMIcat <- filtered_data %>%
  group_by(BMIcat, gender) %>%
  summarise(count = n()) %>%
  group_by(BMIcat) %>%
  mutate(proportion = count / sum(count))
```

```r
ggplot(proportion_by_BMIcat, aes(x = BMIcat, y = proportion, fill = gender)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "BMI Category", y = "Proportion", title = "Proportion of Boys and Girls in BMI Categories") +
  scale_fill_manual(values = c("black", "red"), name = "Gender") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Question 2. (18 marks)

Write a function called `runzero` that given a numeric vector return the length of the longest consecutive subsequence of zeros. Run the following to test your answers.

runzero(c(4,5,6,1))

runzero(c(4,0,0,0,6,0,0,0,0))

runzero(c(0,0,3,3,3,3))

```r
#calculate longest zero sequence = clzs
#max sequence length = msl
clzs <- function(vector) {
  msl <- 0
  current_sequence_length <- 0

  for (element in vector) {
    if (element == 0) {
      current_sequence_length <- current_sequence_length + 1
      msl <- max(msl, current_sequence_length)
    } else {
```

```
      current_sequence_length <- 0
    }
  }

  return(msl)
}

# Test cases
result1 <- clzs(c(4, 5, 6, 1))
result2 <- clzs(c(4, 0, 0, 0, 6, 0, 0, 0, 0))
result3 <- clzs(c(0, 0, 3, 3, 3, 3))

# Print the results
print(result1)
```

```
[1] 0
```

```
print(result2)
```

```
[1] 4
```

```
print(result3)
```

```
[1] 2
```

install.packages('tinytex') tinytex::install_tinytex()