QC report - Gestation and birth weight by gender

Hung Pham

04/21/2021

1 Synopsis

The "birth weight" data is a toy dataset that contains simulated information of 135 babies. The variables recorded are birth weight (grams), length of gestation (weeks), gender (Male/Female), and unique ids.

This report aims to accomplish two routine tasks: 1) Quality Control a dataset, and 2) produce preliminary statistical analyses. In this case, the statistical analyses of interest is the association between birth weight, length of gestation, and gender.

Data is not uploaded to github.

2 Libraries

The following code chunk loads the relevant R libraries - collections of pre- programmed R functions wrapped in discrete packages.

```r
library(tidyverse) # Data cleaning and wrangling
library(janitor) # Data cleaning
library(here) # For reproducibility and ease of collaboration
library(knitr) # Table formating
library(kableExtra) # This library formats our table to look nice
library(rlang) # For writing functions
library(patchwork) # For visualization
library(GGally) # For pairwise plot
library(magrittr) # Piping
library(treemap) # Visualize categorical variables
library(xtable) # Latex tables
```

3 QC

3.1 Data input

- Import the data with `read_csv`. There are *135 obs* and *4 variables*.
- Add another column `unique_id` in case there are problems with the `new_id` column.
- Convert to lower case and remove problematic symbols (e.g. `\`,,,. . . , etc.) from column names with `clean_names`.
- `my_dat` is our raw data. We print the first 10 observations from the data.

```r
loc<-here() # This variable tracks the root path of this file on a local machine

figure_n<-0 # Counter for figure number
count_fig<-function(reset = NULL){
        reset<- enexpr(reset)
        if(is.null(reset)){
        figure_n<<-figure_n+1} else {figure_n<<- 0}
} # This function automatically update our figure number everytime its ran

xtable2kable <- function(x) {
        out <- capture.output(print(x, table.placement = NULL))[-(1:2)]
        out <- paste(out, collapse = "\n")
        structure(out, format = "latex", class = "knitr_kable")
} # This function solves a bug with Latex

my_dat<-read_csv(paste0(loc,"/data/Play_Data_V1.csv")) %>%
        clean_names() # Load the data into R

my_dat$id<- 1:nrow(my_dat) # Add a unique id column.

head(my_dat, 10) %>%
        xtable(caption = "Top 10 obs of raw data") %>%
        xtable2kable() %>%
        kable_classic_2(latex_options = "HOLD_position", position = "center")
```

	birthweight	length_of_gestation	gender	new_id	id
1	3595.00	39.00	Female	1849	1
2	3515.00	37.43	Female	1850	2
3	2820.00	37.57	Male	1852	3
4	2636.55	37.86	Male	1857	4
5	1786.05	35.43	Female	1861	5
6	2590.00	37.57	37.57	1865	6
7		37.29	Male	1867	7
8	3875.00	41.00		1868	8
9	3590.00	39.14	Male	1875	9
10	3980.00	40.00	40	1876	10

Table 1: Top 10 obs of raw data

- Immediately, we see some issues:
  - `gender` has numerical values.
  - There are missing data in `birthweight` and `gender`.
- We first identify the problematic values in each variable. Then we will process missing data.

3.2 Identify individual errors

3.2.1 `birthweight`

```r
summary(my_dat$birthweight) %>%
        broom::tidy() %>%
        xtable(digits = 4, caption = "Summary stats of birth weights") %>%
        xtable2kable() %>%
        kable_classic_2(latex_options = "HOLD_position",position = "center")
```

	minimum	q1	median	mean	q3	maximum	na
1	0.2983	3033.1125	3430.0000	5514.4234	3660.0375	300033.4500	1.0000

Table 2: Summary stats of birth weights

`birthweight` is measured in grams. Value as low as 0.2983 or as high as 300033.5 are improbable. We print the `birthweight` values of < 5% percentile and > 95% percentile.

```
bw_quantile<-quantile(my_dat$birthweight, c(0.05,0.95), na.rm = T)

my_dat %>%
        filter(birthweight < bw_quantile[1] | birthweight > bw_quantile[2]) %>%
        arrange(birthweight) %>%
        xtable(digits = 3,
               caption = "Obs where birth weight are outside of the 5th-95th percent range") %>%
        xtable2kable() %>%
        kable_classic_2(latex_options = "HOLD_position",position = "center")
```

	birthweight	length_of_gestation	gender	new_id	id
1	0.298	40.571	Male	1909	23
2	0.386	39.857	Male	1930	34
3	1786.050	35.429	Female	1861	5
4	2012.850	34.571	Male	1969	62
5	2210.000	36.429	Female	2048	107
6	2235.000	38.429	Female	2018	84
7	2375.000	35.000	Female	2073	122
8	4167.000	40.571		Male	26
9	4224.000	40.000	Male	2023	88
10	4336.000	41.000	Female	Female	45
11	4337.000	38.571	Female	2029	93
12	4422.600	39.714	Male	1935	39
13	4906.000	41.857	Male	1897	16
14	300033.450	39.857	Female	1985	69

Table 3: Obs where birth weight are outside of the 5th-95th percent range

Clearly, 0.2983, 0.3855, and 3e6 grams are not possible. We label and collect these errors for fixing.

```
problem_row<-tibble(id = c(23,34,69),
                    issue = c("bw too small", "bw too small", "bw too large"))
```

3.2.2 length_of_gestation

```
summary(my_dat$length_of_gestation) %>%
        broom::tidy() %>%
        xtable(digits = 3, caption = "Summary stats of gestation length") %>%
        xtable2kable() %>%
        kable_classic_2(latex_options = "HOLD_position",position = "center")
```

	minimum	q1	median	mean	q3	maximum
1	34.571	38.714	39.429	41.794	40.286	380.143

Table 4: Summary stats of gestation length

For `length_of_gestation`, value as high as 380.1429 is improbable. We gather values > 95% percentile.

```
lg_quantile<-quantile(my_dat$length_of_gestation, 0.95, na.rm = T)
my_dat %>%
        filter(length_of_gestation > lg_quantile[1]) %>%
        arrange(length_of_gestation) %>%
        xtable(digits = 3,
               caption = "Obs where gestation length are > 95th percentile") %>%
        xtable2kable() %>%
        kable_classic_2(latex_options = "HOLD_position",position = "center")
```

	birthweight	length_of_gestation	gender	new_id	id
1	3487.050	41.286	Female	1960	56
2	3798.800	41.286	Male	2038	101
3	3430.350	41.571	Male	1936	40
4	4906.000	41.857	Male	1897	16
5	3798.900	41.857		1929	33
6	3218.000	380.143	Male	2100	135

Table 5: Obs where gestation length are > 95th percentile

We collect the id where length of gestation is 380.1429 and move on.

```
problem_row<- problem_row %>%
        bind_rows(tibble(id = 135,
                         issue = c("gestation too long")))
```

3.2.3 new_id

```
head(my_dat$new_id) # Top new_id values
```

```
## [1] "1849" "1850" "1852" "1857" "1861" "1865"
```

```
class(my_dat$new_id) # Type of variable for new_id
```

```
## [1] "character"
```

The fact that the top values of `new_id` are all integers yet R imported `new_id` as a character variable raised concerns. We use REGEX to detect which values are not integers.

```
my_dat %>%
        filter(!str_detect(new_id, "\\d")) %>%
        xtable(digits = 3, caption = "Obs where id is not an integer") %>%
        xtable2kable() %>%
        kable_classic_2(latex_options = "HOLD_position",position = "center")
```

	birthweight	length_of_gestation	gender	new_id	id
1	4167.000	40.571		Male	26
2	4336.000	41.000	Female	Female	45

Table 6: Obs where id is not an integer

IDs shouldn't be `Male` or `Female`. Also, individual IDs should be unique. We check if there are any repeats in `new_id`.

```
my_dat %>%
        filter(str_detect(new_id, "\\d")) %>%
        pull(new_id) %>%
        unique(.) %>%
        length(.)
```

```
## [1] 133
```

The math checks out. There are 133 unique numerical IDs after removing two wrong IDs. We confirm that each observation is unique and collect the problematic IDs rows.

```
problem_row<- problem_row %>%
        bind_rows(tibble(id = c(26,45),
                         issue = c("wrong ID")))
```

3.2.4 gender

```
tabyl(my_dat,gender) %>%
        xtable(digits = 3, caption = "Frequency table of gender variable") %>%
        xtable2kable() %>%
        kable_classic_2(latex_options = "HOLD_position",position = "center")
```

	gender	n	percent	valid_percent
1	37.57	1.000	0.007	0.008
2	38.29	1.000	0.007	0.008
3	40	1.000	0.007	0.008
4	Female	59.000	0.437	0.454
5	Male	68.000	0.504	0.523
6		5.000	0.037	

Table 7: Frequency table of gender variable

Gender shouldn't be floats. We collect these rows.

```r
wrong_gender<- my_dat %>%
        filter(str_detect(gender,"[[\\d]]")) %>% pull(id)

problem_row<- problem_row %>%
        bind_rows(tibble(id = wrong_gender,
                         issue = c("wrong gender")))
```

3.3 Missing data

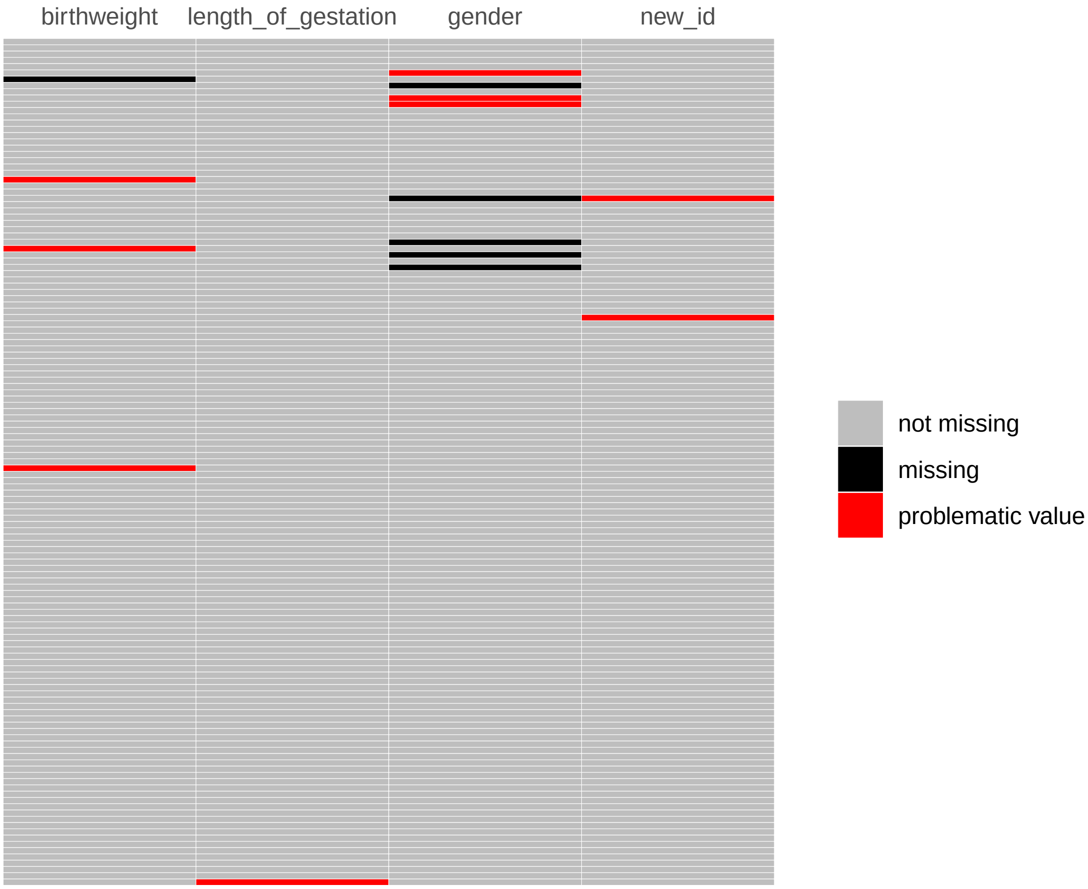
We investigate the pattern of missing values.

```r
missing_to_binary<-function(x){if_else(is.na(x),1,0)}

# `for_miss_viz` converts missing values to 1, non missing values to 0, and
# problematic values to 2.
for_miss_viz<-my_dat %>%
        mutate(across(.cols = -id,
                      .fns = missing_to_binary)) %>%
        pivot_longer(cols = -id,
                     names_to = "column_name") %>%
        left_join(problem_row, by = "id") %>% ### Join with the problem table
        mutate(value = case_when(
                ### Convert problematic values to 2
                grepl("bw too", .$issue) & column_name == "birthweight" ~ 2,
                grepl("gestation", .$issue) & column_name == "length_of_gestation" ~ 2,
                grepl("wrong ID", .$issue) & column_name == "new_id" ~ 2,
                grepl("gender", .$issue) & column_name == "gender" ~ 2,
                T ~ value),
               ### Convert data to factor for visualization
               x_coord = fct_inseq(as.character(rep(1:4, 135))),
               y_coord = fct_rev(fct_inseq(as.character(id))),
               my_color = factor(value))

count_fig() # Count figure
ggplot(for_miss_viz, aes(x_coord, y_coord)) +
        geom_tile(aes(fill = my_color),color = "white")+
        scale_fill_manual(values = c("grey","black","red"),
                          labels = c("not missing", "missing", "problematic value"))+
        scale_x_discrete(labels = setdiff(names(my_dat),"id"), position = "top")+
        theme_minimal()+
        labs(title = paste0("Figure ",figure_n,": Missing matrix of birth weight data"))+
        theme(axis.title = element_blank(),
              axis.text.y = element_blank(),
              panel.grid = element_blank(),
              legend.title = element_blank(),
              legend.position = "right")
```

Figure 1: Missing matrix of birth weight data



From Figure 1, we conclude that **available data analysis** is appropriate. Meaning bias is not an issue if we proceed with the analysis only using the available data. This is because the amount of missing data is not significant and the pattern can be safely concluded as missing at completely random (MCAR). Sensitivity analysis with imputed data using multiple imputation can be performed to confirm the robustness of the available data analysis but it wouldn't be needed in this scenario.

3.4 Removing problemmatic values

We can setup a for-loop to remove the problematic values.

```
tempt_dat<-for_miss_viz %>% filter(value == 2) %>% select(id,column_name)

my_dat_clean<-my_dat

remove_walk<-function(x,y){
        my_dat_clean[which(my_dat_clean$id == x), y]<<-NA
}

walk2(tempt_dat$id,tempt_dat$column_name,remove_walk)
```

We check the imputation.

```
before_remove<-problem_row %>% left_join(my_dat,by = "id") %>%
        rename(gestation_l = length_of_gestation)
after_remove<-problem_row %>% left_join(my_dat_clean,by = "id") %>%
        rename(gestation_l = length_of_gestation)

xtable(digits = 3, x =  before_remove,
       caption = "Obs where there are improbable values, before replacing the values with NA") %>%
        xtable2kable() %>%
        kable_classic_2(latex_options = "HOLD_position",position = "center")
```

	id	issue	birthweight	gestation_l	gender	new_id
1	23.000	bw too small	0.298	40.571	Male	1909
2	34.000	bw too small	0.386	39.857	Male	1930
3	69.000	bw too large	300033.450	39.857	Female	1985
4	135.000	gestation too long	3218.000	380.143	Male	2100
5	26.000	wrong ID	4167.000	40.571		Male
6	45.000	wrong ID	4336.000	41.000	Female	Female
7	6.000	wrong gender	2590.000	37.571	37.57	1865
8	10.000	wrong gender	3980.000	40.000	40	1876
9	11.000	wrong gender	2830.000	38.286	38.29	1877

Table 8: Obs where there are improbable values, before replacing the values with NA

```
xtable(digits = 3,x =  after_remove,
    caption = "After replacing the values with NA") %>%
        xtable2kable() %>%
        kable_classic_2(latex_options = "HOLD_position",position = "center")
```

	id	issue	birthweight	gestation_l	gender	new_id
1	23.000	bw too small		40.571	Male	1909
2	34.000	bw too small		39.857	Male	1930
3	69.000	bw too large		39.857	Female	1985
4	135.000	gestation too long	3218.000		Male	2100
5	26.000	wrong ID	4167.000	40.571		
6	45.000	wrong ID	4336.000	41.000	Female	
7	6.000	wrong gender	2590.000	37.571		1865
8	10.000	wrong gender	3980.000	40.000		1876
9	11.000	wrong gender	2830.000	38.286		1877

Table 9: After replacing the values with NA

This table shows the list of all our problematic values and the successful removal of those values. `my_dat_clean` is our cleaned data.

4 Statistical Analysis

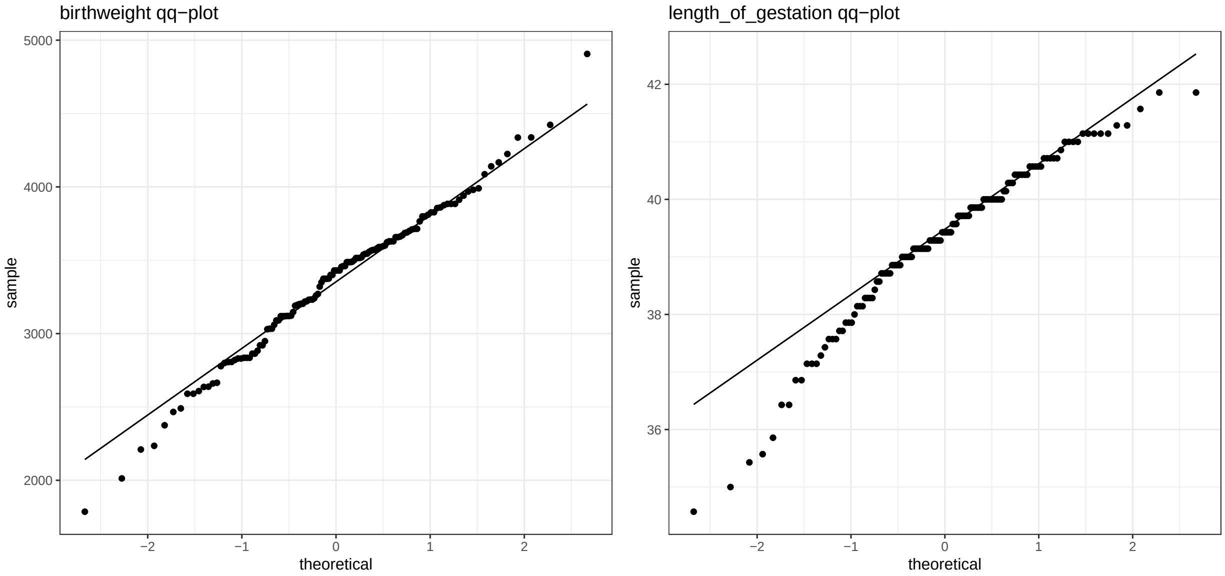
4.1 Birthweight and length of gestation

4.1.1 Normality assumption

```r
## This function draws qq plots
draw_qq<-function(my_var,dat,strata = NULL){
        if(!is.null(strata)){
                p<-dat %>% ggplot(aes(sample = .data[[my_var]])) +
                        geom_qq()+
                        geom_qq_line()+
                        theme_bw()+
                        facet_wrap(~ .data[[strata]])+
                        labs(subtitle = paste0(my_var," qq-plot"))+
                        theme(plot.subtitle = element_text(hjust = 0, size = 13))
                return(p)
        }
        p<-dat %>% ggplot(aes(sample = .data[[my_var]])) +
                        geom_qq()+
                        geom_qq_line()+
                        theme_bw()+
                        labs(subtitle = paste0(my_var," qq-plot"))+
                        theme(plot.subtitle = element_text(hjust = 0, size = 13))
        p
}
p2<-draw_qq("birthweight",my_dat_clean)
p3<-draw_qq("length_of_gestation",my_dat_clean)
```

```r
count_fig() # Update figure number
p2+p3+plot_annotation(title = paste0("Figure ",figure_n,
                                     ": qqplots of birthweight and gestation length"))
```

Figure 2: qqplots of birthweight and gestation length



birthweight qq−plot

length_of_gestation qq−plot

From the qq-plots, we see that length of gestation might not be normally distributed. We use the Shapiro-Wilk test to formally test the normal assumptions of `birthweight` and `length_of_gestation`.

```
## This function performs the shapiro-wilk test and automatically generate
## decisions when pvalue is >= 0.05

my_shapiro<-function(my_var,dat){
        for_map<-function(x){
                res<-shapiro.test(dat[[x]]) %>%
                        broom::tidy()
                # If p.value is larger than 0.05 then we reject the null hypothesis
                if (res$p.value >= 0.05) {
                        res %<>% mutate(H0 = paste0(x, " is normally distributed"),
                                        decision = "fail to reject H0",
                                        n = length(na.omit(dat[[x]]))) %>%
                                select(statistic, p.value, n, everything())
                } else {
                        res %<>% mutate(H0 = paste0(x, " is normally distributed"),
                                        decision = "reject H0",
                                        n = length(na.omit(dat[[x]]))) %>%
                                select(statistic, p.value, n, everything())
                }
        }
        res<-map(my_var,for_map) %>% reduce(rbind)
        res
}
my_shapiro(c("birthweight","length_of_gestation"), my_dat_clean) %>%
        mutate(method = "Shapiro-Wilk") %>%
        xtable(digits = 3,
               caption = "Testing the normality assumption of birth weight and gestation length") %>%
        xtable2kable() %>%
        kable_classic_2(latex_options = "HOLD_position",position = "center")
```

	statistic	p.value	n	method	H0	decision
1	0.988	0.289	131	Shapiro-Wilk	birthweight is normally distributed	fail to reject H0
2	0.952	0.000	134	Shapiro-Wilk	length_of_gestation is normally distributed	reject H0

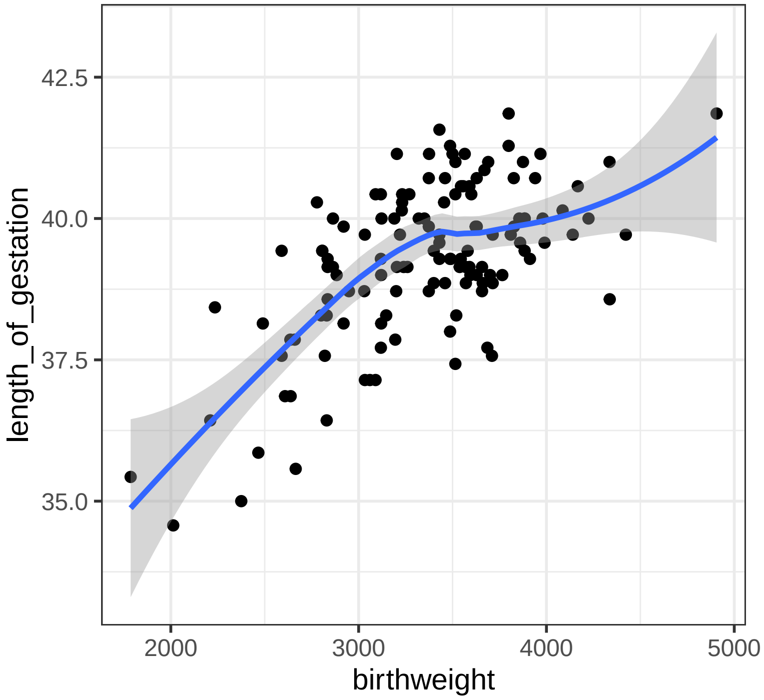
Table 10: Testing the normality assumption of birth weight and gestation length

We reject the null hypothesis that length of gestation is normally distributed and fail to reject the null hypothesis that birth weight is normally distributed.

4.1.2 Correlation

```
count_fig()
my_dat_clean %>% ggplot(aes(x = birthweight, y = length_of_gestation)) +
        geom_point()+
        geom_smooth(method = "loess")+
        labs(title = paste0("Figure ",figure_n,
                            ": Scatterplot of birthweight by length of gestation"))+
        theme_bw()
```

Figure 3: Scatterplot of birthweight by lengt



From the scatter-plot, we can safely assume that there exists a linear correlation between `birthweight` and `length_of_gestation` for the range of values in our data. Hence, we can use a Pearson's correlation to measure this correlation.

```
pearson_res<- cor.test(my_dat_clean$birthweight, my_dat_clean$length_of_gestation) %>%
        broom::tidy() %>% rename(correlation = estimate) %>%
        t() %>% as.data.frame()

pearson_res %>% tibble(term = rownames(.)) %>%
        rename("Value" = V1) %>% select(term,Value) %>%
        kable(digits = 3, caption = "Pearson's correlation test") %>%
        kable_classic_2(latex_options = "HOLD_position",position = "center")
```

Table 11: Pearson's correlation test

term	Value
correlation	0.622155
statistic	8.990847
p.value	2.769694e-15
parameter	128
conf.low	0.503956
conf.high	0.71748
method	Pearson's product-moment correlation
alternative	two.sided

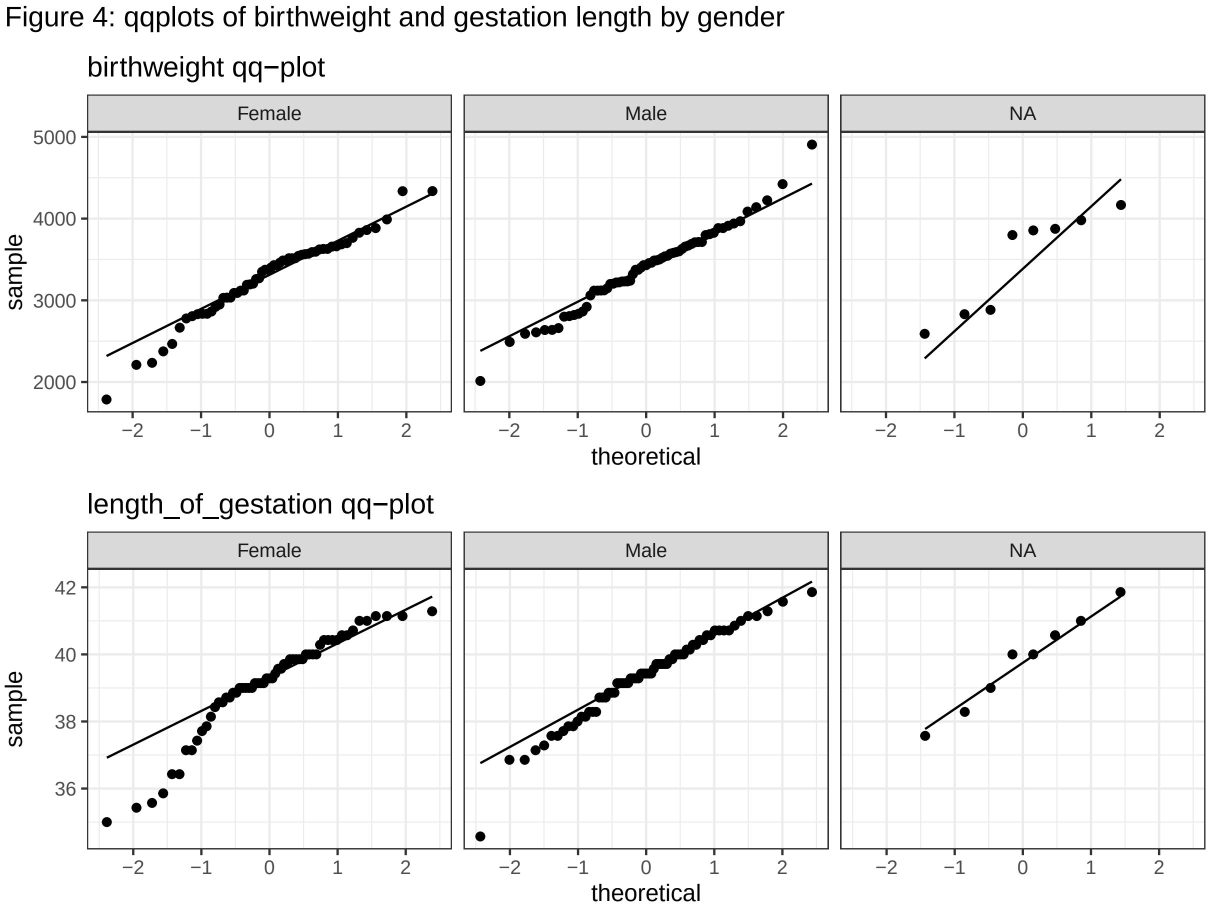
The Pearson's correlation between `birthweight` and `length_of_gestation` of 0.622 *is significant* at a two sided alpha of 0.05.

4.1.3 Differential effect by gender

We remake the qq-plots by gender using `draw_qq` programmed on line 257.

```
p4<-draw_qq("birthweight",my_dat_clean,"gender")
p5<-draw_qq("length_of_gestation",my_dat_clean,"gender")
```

```
count_fig()
(p4/p5)+plot_annotation(title = paste0("Figure ",figure_n,
                                       ": qqplots of birthweight and gestation length by gender"))
```



4.1.3.1 Is the birthweight between males and females babies the same? We use the t-test to test the following H0: *the mean* `birthweight` *between males and females are the same.*

```
t_test_res<- t.test(my_dat_clean[which(my_dat_clean$gender == "Female"),
                                 "birthweight"],
       my_dat_clean[which(my_dat_clean$gender == "Male"),"birthweight"],
       var.equal = T) %>%
        broom::tidy() %>%
        rename("mean diff" = estimate,
               "mean female" = estimate1,
               "mean male" = estimate2) %>% t() %>% as.data.frame()
```

```r
t_test_res %>% tibble(term = rownames(.)) %>% rename("Value" = V1) %>%
        select(term,Value) %>%
        xtable(digits = 3,
               caption = "T-test mean difference between male and female birth weight, assuming
            equal variance") %>%
        xtable2kable() %>%
        kable_classic_2(latex_options = "HOLD_position",position = "center")
```

	term	Value
1	mean diff	-105.6554
2	mean female	3284.972
3	mean male	3390.627
4	statistic	-1.174279
5	p.value	0.2425894
6	parameter	121
7	conf.low	-283.784
8	conf.high	72.47327
9	method	Two Sample t-test
10	alternative	two.sided

Table 12: T-test mean difference between male and female birth weight, assuming equal variance

4.1.3.2 Is the length of gestation between males and females babies the same?

We use the non-parametric Wilcoxson-rank sum test to test the following H0: *the location shift (or median) of `length_of_gestation` between male and female is zero.*

```r
wil_cox_test<-wilcox.test(my_dat_clean[which(my_dat_clean$gender == "Female"),"length_of_gestation", dro
       my_dat_clean[which(my_dat_clean$gender == "Male"),"length_of_gestation",
                    drop = T]) %>%
        broom::tidy() %>% t() %>% as.data.frame()

wil_cox_test %>% tibble(term = rownames(.)) %>% rename("Value" = V1) %>%
        select(term,Value) %>%
        xtable(digits = 3,
               caption = "Wilcox test of median shift between male and female length of gestation") %>%
        xtable2kable() %>%
        kable_classic_2(latex_options = "HOLD_position",position = "center")
```

	term	Value
1	statistic	1857.5
2	p.value	0.5620061
3	method	Wilcoxon rank sum test with continuity correction
4	alternative	two.sided

Table 13: Wilcox test of median shift between male and female length of gestation

- We fail to reject the null hypothesis that the mean `birthweight` is the same between male and female with a two sided alpha = 0.05.
- We fail to reject the null hypothesis that the median of `length_of_gestation` is the same between males and females with a two sided alpha = 0.05.

5 Recode

We create another variable "term" where `length_of_gestation` < 37 weeks is categorized as "pre-term" and "to term" otherwise. Then we create a treemap to visualize the proportion of to-term and pre-term by gender.

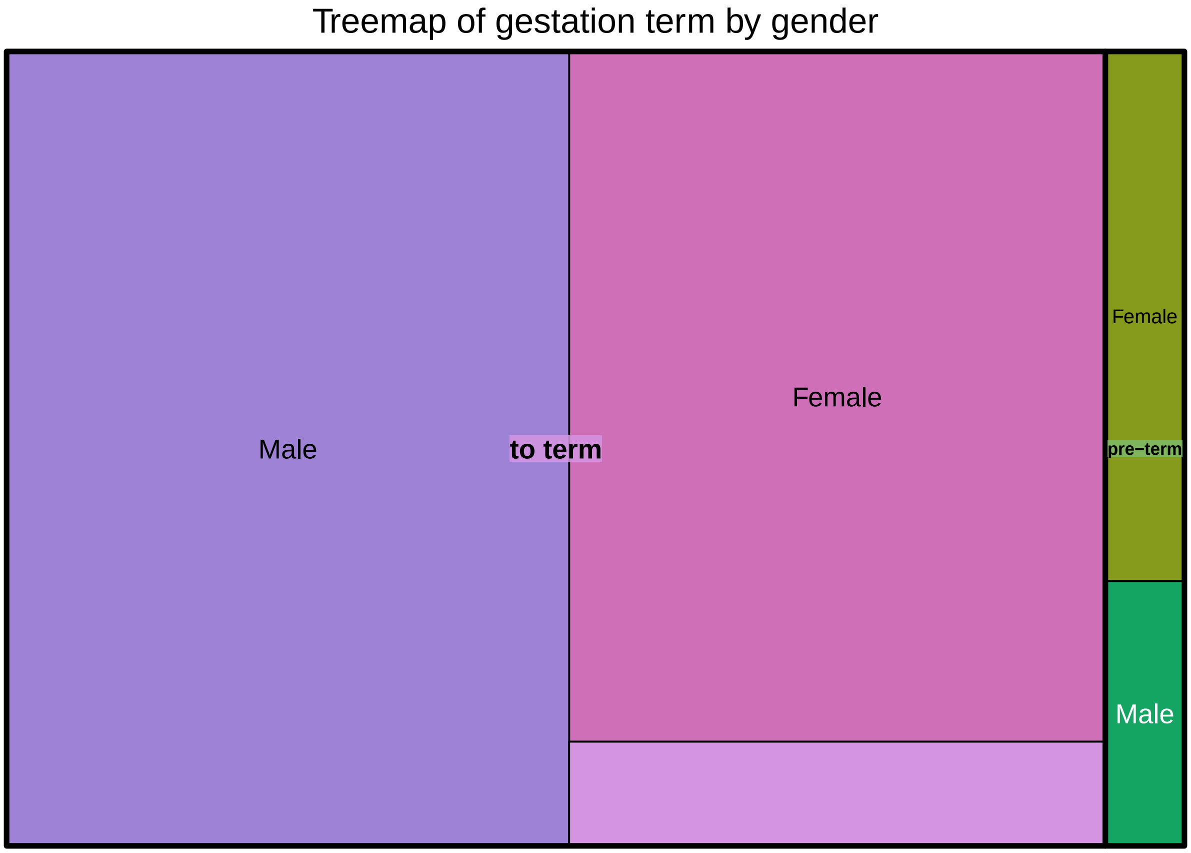
```
my_dat_clean$term<-ifelse(my_dat_clean$length_of_gestation < 37,
                           "pre-term",
                           "to term")

tree_dat<-my_dat_clean %>%
        group_by(term, gender) %>%
        summarize(n = n()) %>%
        rbind(my_dat_clean %>%
                      group_by(term) %>%
                      summarize(n = n()) %>%
                      mutate(gender = "both")
        )
tree_dat %>%
        kable(digits = 3, caption = "Frequency table of `term` alone and by gender") %>%
        kable_classic_2(latex_options = "HOLD_position",position = "center")
```

Table 14: Frequency table of ‘term‘ alone and by gender

term	gender	n
pre-term	Female	6
pre-term	Male	3
to term	Female	53
to term	Male	64
to term	NA	8
NA	Male	1
pre-term	both	9
to term	both	125
NA	both	1

```
treemap(tree_dat[1:6,],
        index = c("term", "gender"),
        vSize = "n",
        title = "Treemap of gestation term by gender")
```



6 Session Info

Session information for reproducibility.

```
sessionInfo()
```

```
## R version 4.0.5 (2021-03-31)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19042)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
```

```
##  [1] xtable_1.8-4     treemap_2.4-2    magrittr_2.0.1   GGally_2.1.1
##  [5] patchwork_1.1.1  rlang_0.4.10     kableExtra_1.3.4 knitr_1.31
##  [9] here_1.0.1       janitor_2.1.0    forcats_0.5.1    stringr_1.4.0
## [13] dplyr_1.0.5      purrr_0.3.4      readr_1.4.0      tidyr_1.1.3
## [17] tibble_3.1.0     ggplot2_3.3.3    tidyverse_1.3.0
##
## loaded via a namespace (and not attached):
##  [1] httr_1.4.2         splines_4.0.5      jsonlite_1.7.2     viridisLite_0.4.0
##  [5] modelr_0.1.8       shiny_1.6.0        assertthat_0.2.1   cellranger_1.1.0
##  [9] yaml_2.2.1         lattice_0.20-41    pillar_1.6.0       backports_1.2.1
## [13] glue_1.4.2         digest_0.6.27      RColorBrewer_1.1-2 promises_1.2.0.1
## [17] rvest_1.0.0        snakecase_0.11.0   colorspace_2.0-0   Matrix_1.3-2
## [21] htmltools_0.5.1.1  httpuv_1.5.5       plyr_1.8.6         pkgconfig_2.0.3
## [25] broom_0.7.6        haven_2.3.1        scales_1.1.1       webshot_0.5.2
## [29] svglite_2.0.0      later_1.1.0.1      mgcv_1.8-34        farver_2.1.0
## [33] generics_0.1.0     ellipsis_0.3.1     withr_2.4.1        cli_2.4.0
## [37] crayon_1.4.1       readxl_1.3.1       mime_0.10          evaluate_0.14
## [41] fs_1.5.0           fansi_0.4.2        nlme_3.1-152       xml2_1.3.2
## [45] tools_4.0.5        data.table_1.14.0  hms_1.0.0          lifecycle_1.0.0
## [49] gridBase_0.4-7     munsell_0.5.0      reprex_2.0.0       compiler_4.0.5
## [53] systemfonts_1.0.1  grid_4.0.5         rstudioapi_0.13    igraph_1.2.6
## [57] labeling_0.4.2     rmarkdown_2.7      codetools_0.2-18   gtable_0.3.0
## [61] DBI_1.1.1          reshape_0.8.8      R6_2.5.0           lubridate_1.7.10
## [65] fastmap_1.1.0      utf8_1.2.1         rprojroot_2.0.2    stringi_1.5.3
## [69] Rcpp_1.0.6         vctrs_0.3.7        dbplyr_2.1.1       tidyselect_1.1.0
## [73] xfun_0.22
```