Programming in R (OC+DL): Solution for exam part 2 (15/01/2024)

JUAN VANEGAS (2023/2024)

Part 1: the real_data_GDI data

In this part of the exam, the questions are focused on the real_data_GDI dataset which is a part of the genderstat R package. To access the data you need to install the package. More information can be found on https://cran.r-project.org/web/packages/genderstat/index.html. Use the code below to access the data.

```
library(genderstat)
data("real_data_GDI")
names(real_data_GDI)
```

```
## [1] "country"                 "female_life_expectancy" "male_life_expectancy"
## [4] "female_mean_schooling"   "male_mean_schooling"    "female_gni_per_capita"
## [7] "male_gni_per_capita"
```

Question 1

1. Create a new dataset, gdi, by removing all missing values from real_data_GDI. How many observations are included in the new data? Below you can see the first few lines of the data. As you can see, each country appears in one line in which information is available on both female and male.

Solution 1.1

```
gdi <- real_data_GDI %>%
  filter(complete.cases(.))
paste(nrow(gdi), "observations")
```

```
## [1] "174 observations"
```

2. Create a new data, gdi2, by transforming the gdi dataset from Q1.1 into a dataset for which the information for each country appears in two lines: one for female and one for male, see the first few lines below. For example, as shown below, for Afghanistan, the first line contains information about female and the second line information about male.

Solution 1.2

```
# TODO pivot instead
genders <- c("male", "female")
female_data <- gdi %>%
  mutate(gender="female") %>%
  mutate(gender=factor(gender, levels=genders)) %>%
  select(gender, country, female_life_expectancy, female_mean_schooling, female_gni_per_capita) %>%
  rename(life_expectancy = female_life_expectancy) %>%
  rename(mean_schooling = female_mean_schooling) %>%
  rename(gni_per_capita = female_gni_per_capita)

male_data <- gdi %>%
  mutate(gender="male") %>%
  mutate(gender=factor(gender, levels=genders)) %>%
  select(gender, country, male_life_expectancy, male_mean_schooling, male_gni_per_capita) %>%
  rename(life_expectancy = male_life_expectancy) %>%
  rename(mean_schooling = male_mean_schooling) %>%
  rename(gni_per_capita = male_gni_per_capita)

gdi2 <- bind_rows(female_data, male_data)  %>%
  arrange(country) %>%
```

```
  select(country, gender, life_expectancy, mean_schooling, gni_per_capita)
head(gdi2)
```

```
##       country gender life_expectancy mean_schooling gni_per_capita
## 1 Afghanistan female            65.3            2.3            533
## 2 Afghanistan   male            58.9            3.4           3089
## 3     Albania female            79.2           11.7          11637
## 4     Albania   male            74.1           10.9          16630
## 5     Algeria female            78.0            7.7           3550
## 6     Algeria   male            74.9            8.4          17787
```

3. Use the gdi2 dataset created in Q1.1 and define new variables: (1) Measurement, which includes the information about the type of the measurement (life_expectancy, mean_schooling and gni_per_capita) and (2) Values which contains the corresponding values of the measurement. Define a new dataset, gdi3, shown below. Note that only the first 18 lines of the data are printed and, as can be seen, the data for each country appears in 6 lines.

Solution 1.3

```
gdi3 <- gdi2 %>%
  pivot_longer(cols = c(life_expectancy, mean_schooling, gni_per_capita), names_to = "Measurement", valu
head(gdi3)
```

```
## # A tibble: 6 x 4
##   country     gender Measurement     Values
##   <chr>       <fct>  <chr>            <dbl>
## 1 Afghanistan female life_expectancy   65.3
## 2 Afghanistan female mean_schooling     2.3
## 3 Afghanistan female gni_per_capita   533
## 4 Afghanistan male   life_expectancy   58.9
## 5 Afghanistan male   mean_schooling     3.4
## 6 Afghanistan male   gni_per_capita  3089
```

4. For the new dataset gdi3, replace the "_" in column Measurement by " " (space), as shown below.

Solution 1.4

```
gdi3 <- gdi3 %>%
  mutate(Measurement = str_replace(Measurement, "_", " "))
head(gdi3)
```

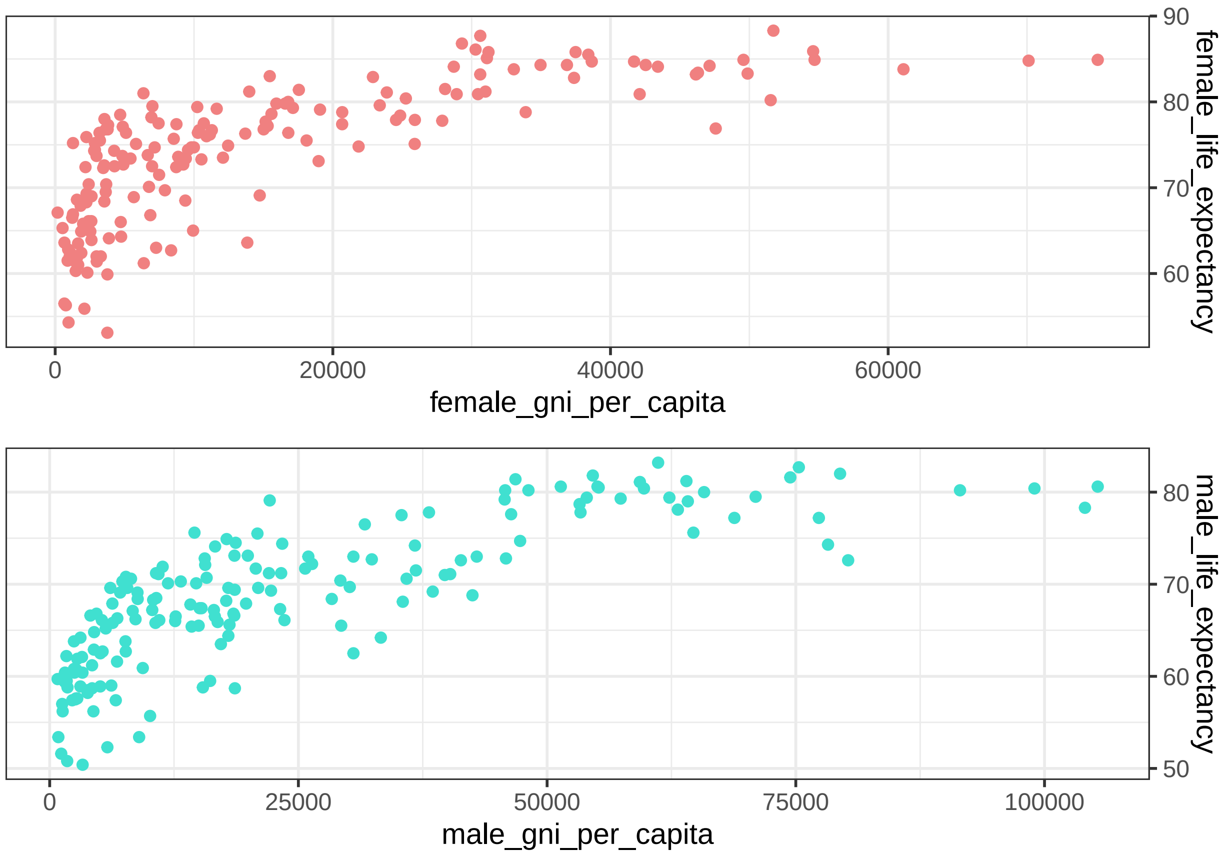
```
## # A tibble: 6 x 4
##   country     gender Measurement     Values
##   <chr>       <fct>  <chr>            <dbl>
## 1 Afghanistan female life expectancy   65.3
## 2 Afghanistan female mean schooling     2.3
## 3 Afghanistan female gni per_capita   533
## 4 Afghanistan male   life expectancy   58.9
## 5 Afghanistan male   mean schooling     3.4
## 6 Afghanistan male   gni per_capita  3089
```

Question 2

1. In this question, we use the gdi dataset created in Q1.1. Produce Figure 2.1 which presents a figure with **two** separate plots in one column and two rows.

Solution 2.1

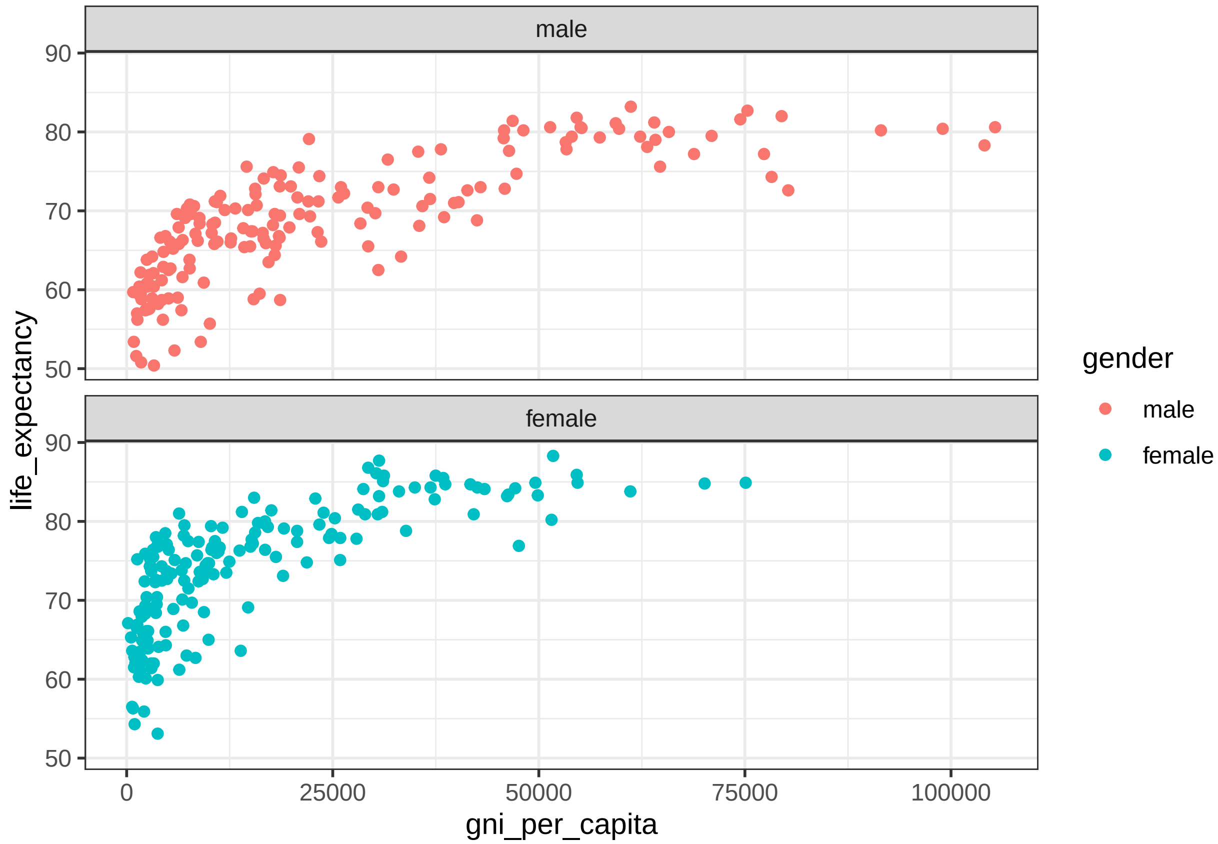
```
# ggplot dots in red and the y line should be on the right
p1 <- ggplot(gdi, aes(x=female_gni_per_capita, y = female_life_expectancy)) +
  geom_point(color="lightcoral") +
  theme_bw() +
  scale_y_continuous(position="right")
p2 <- ggplot(gdi, aes(male_gni_per_capita, y = male_life_expectancy)) +
  geom_point(color="turquoise") +
  scale_y_continuous(position="right") +
  theme_bw()
gridExtra::grid.arrange(p1, p2, ncol=1)
```



2. In this question, use the gdi2 dataset and produce Figure 2.2.

Solution 2.2

```
ggplot(gdi2, aes(x=gni_per_capita, y = life_expectancy, color = gender)) +
  geom_point() +
  theme_bw() +
  facet_wrap(~gender, ncol=1)
```



Question 3

1. Use the gdi dataset that was created in Q1.1. Create a new variable diff which is the difference between female and male life expectancy and add this variable to the dataset.

Solution 3.1

```
gdi <- gdi %>%
  mutate(diff = female_life_expectancy - male_life_expectancy)
head(gdi)
```

```
##       country female_life_expectancy male_life_expectancy female_mean_schooling
## 1 Afghanistan                   65.3                 58.9                   2.3
## 2     Albania                   79.2                 74.1                  11.7
## 3     Algeria                   78.0                 74.9                   7.7
## 4      Angola                   64.3                 59.0                   4.2
## 5   Argentina                   78.6                 72.2                  11.4
## 6     Armenia                   77.4                 66.6                  11.3
##   male_mean_schooling female_gni_per_capita male_gni_per_capita diff
## 1                 3.4                   533                3089  6.4
## 2                10.9                 11637               16630  5.1
## 3                 8.4                  3550               17787  3.1
## 4                 6.9                  4751                6197  5.3
## 5                10.9                 15581               26376  6.4
## 6                11.3                  8736               18558 10.8
```

2. Create a summary table of the minimum, maximum, mean and the 25% quantile of the variable diff. Create a new R object, q25, which is equal to the 25% quantile of the variable diff and print it.

Solution 3.2

```
data.frame(min = min(gdi$diff), max = max(gdi$diff), mean = mean(gdi$diff), q25 = quantile(gdi$diff, 0.2
```

```
##     min  max     mean   q25
## 25% 0.8 10.8 5.366667 4.025
```

```
q25 <- quantile(gdi$diff, 0.25)
q25
```

```
##   25%
## 4.025
```

3. Create a new dataset, gdi_new, that consists of countries with the life expectancy gap between gender (the variable diff) less than its q25. How many countries are included in the new data?

Solution 3.3

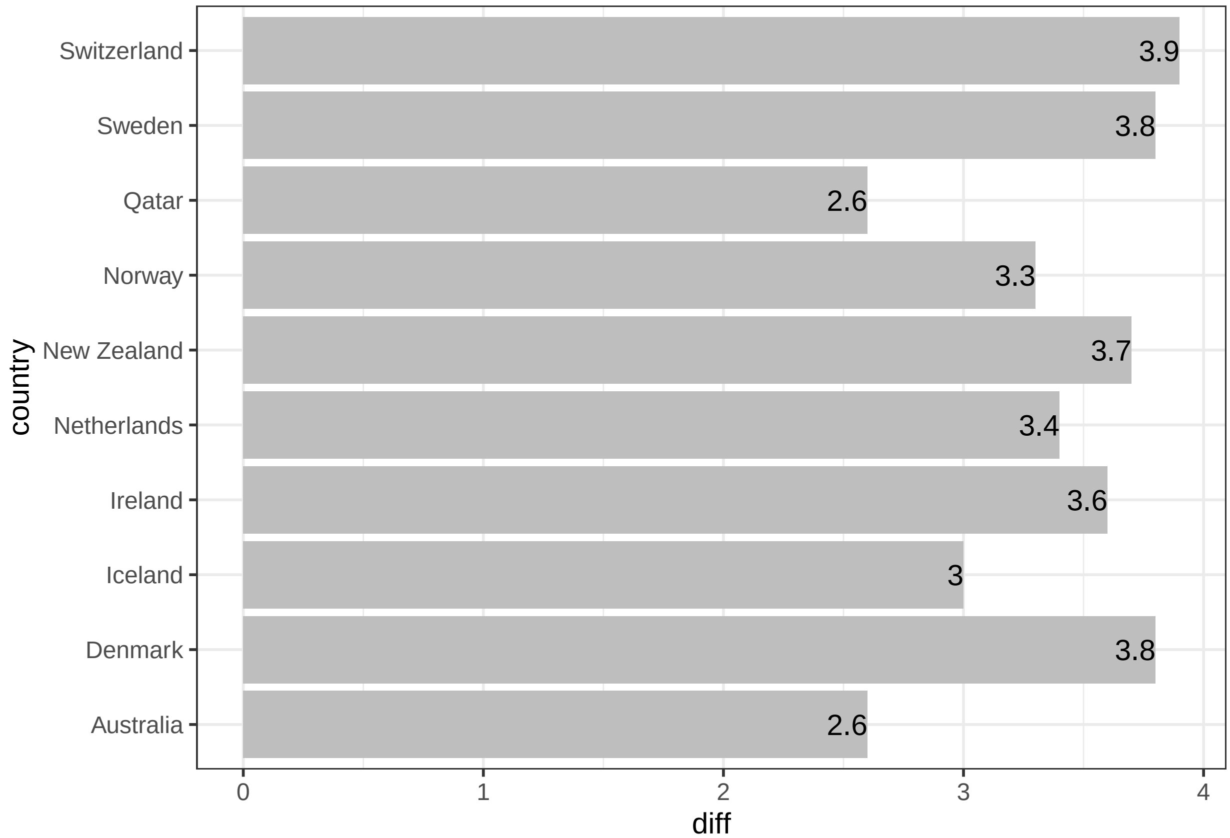
```
gdi_new <- gdi %>%
  filter(diff < q25)
head(gdi_new)
```

```
##      country female_life_expectancy male_life_expectancy female_mean_schooling
## 1    Algeria                   78.0                 74.9                   7.7
## 2  Australia                   85.8                 83.2                  12.8
## 3    Bahrain                   80.0                 77.8                  10.8
## 4 Bangladesh                   74.3                 70.6                   6.8
## 5   Barbados                   79.4                 75.6                  10.3
## 6      Benin                   61.4                 58.2                   3.3
##   male_mean_schooling female_gni_per_capita male_gni_per_capita diff
## 1                 8.4                  3550               17787  3.1
## 2                12.6                 37486               61161  2.6
## 3                11.2                 16786               53359  2.2
## 4                 8.0                  2811                8176  3.7
## 5                 9.1                 10235               14555  3.8
## 6                 5.4                  2998                3819  3.2
```

4. Select 10 countries with the largest female_gni_per_capita from the new data created in Q3.2. Produce Figure 3.1.

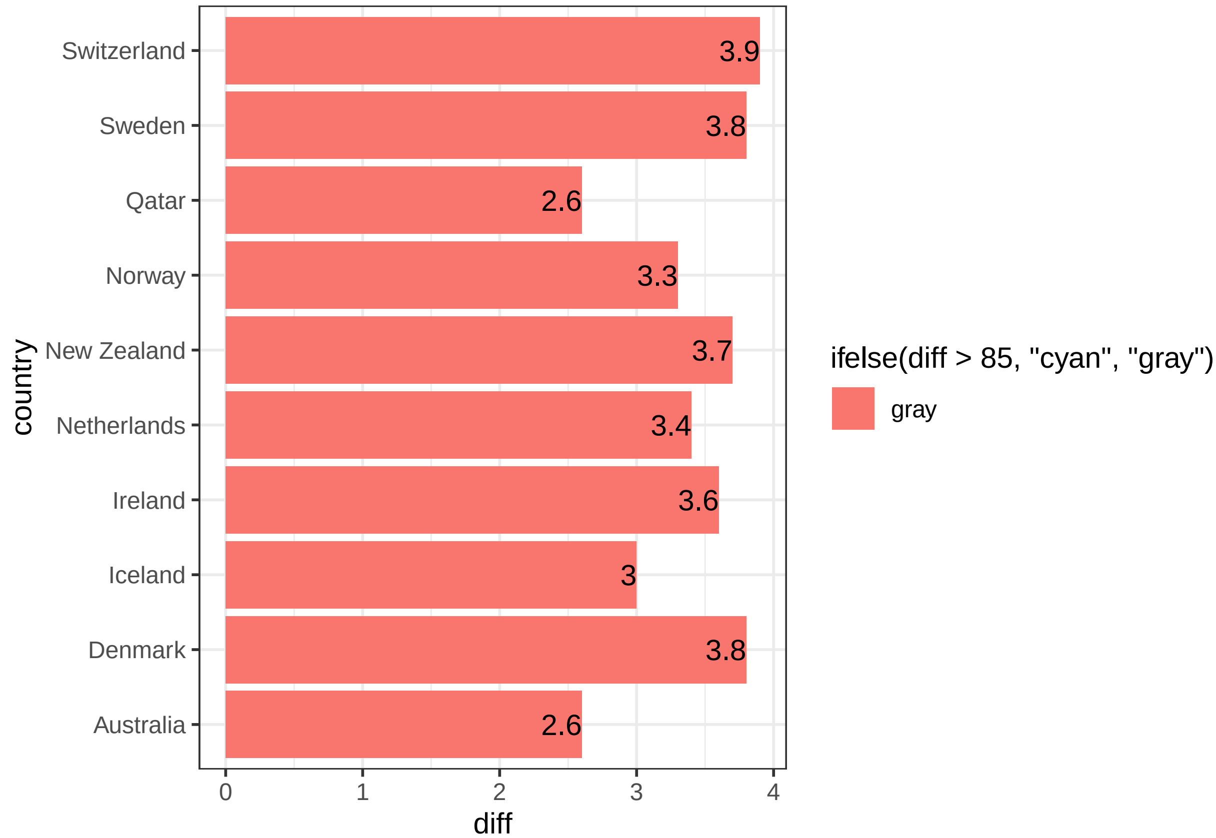
Solution 3.4

```
q3_4.df <- gdi_new %>%
  arrange(-female_gni_per_capita) %>%
  head(10)
ggplot(q3_4.df, aes(x=country, y=diff)) +
  geom_col(fill="grey") +
  theme_bw() +
  geom_text(aes(label = round(diff, 2)), hjust = 1, color = "black") +
  coord_flip()
```



5. Show the countries with female_life_expectancy higher than 85 in different colours as seen in Figure 3.2.

```r
# todo fix
ggplot(q3_4.df, aes(x=country, y=diff)) +
  geom_col(aes(fill = ifelse(diff > 85, "cyan", "gray"))) +
  theme_bw() +
  geom_text(aes(label = round(diff, 2)), hjust = 1, color = "black") +
  coord_flip()
```



Solution 3.5

Part 2 : the flying data

For the analysis of this part we use the flying data which is a part of the R package dropout. This is a modified version of the Flying Etiquette Survey data. More information can be found in https://CRAN.R-project.org/package=dropout. The code below can be used to access the data

```r
library(dropout)
data("flying")
names(flying)
```

```
##  [1] "respondent_id"                "travel_frequency"
##  [3] "seat_recline"                 "height"
##  [5] "children_under_18"            "two_armrests"
##  [7] "middle_armrest"               "window_shade"
##  [9] "moving_to_unsold_seat"        "talking_to_seatmate"
## [11] "getting_up_on_6_hour_flight" "obligation_to_reclined_seat"
## [13] "recline_seat_rudeness"        "eliminate_reclining_seats"
## [15] "switch_for_friends"           "switch_for_family"
## [17] "wake_passenger_bathroom"      "wake_passenger_walk"
## [19] "baby_on_plane"                "unruly_children"
## [21] "electronics_violation"        "smoking_violation"
## [23] "gender"                       "age"
## [25] "household_income"             "education"
## [27] "location_census_region"       "survey_type"
```

Question 4

For this question, use flying data without the missing values.

1. Calculate the frequency and the percentage of each response to the question "in a row of two seats, who should get to use the middle arm rest?" (variable middle_armrest) and define the dataframe shown below (note that the dataframe shows the different categories of the question, the number of answers and the percentage).

Solution 4.1

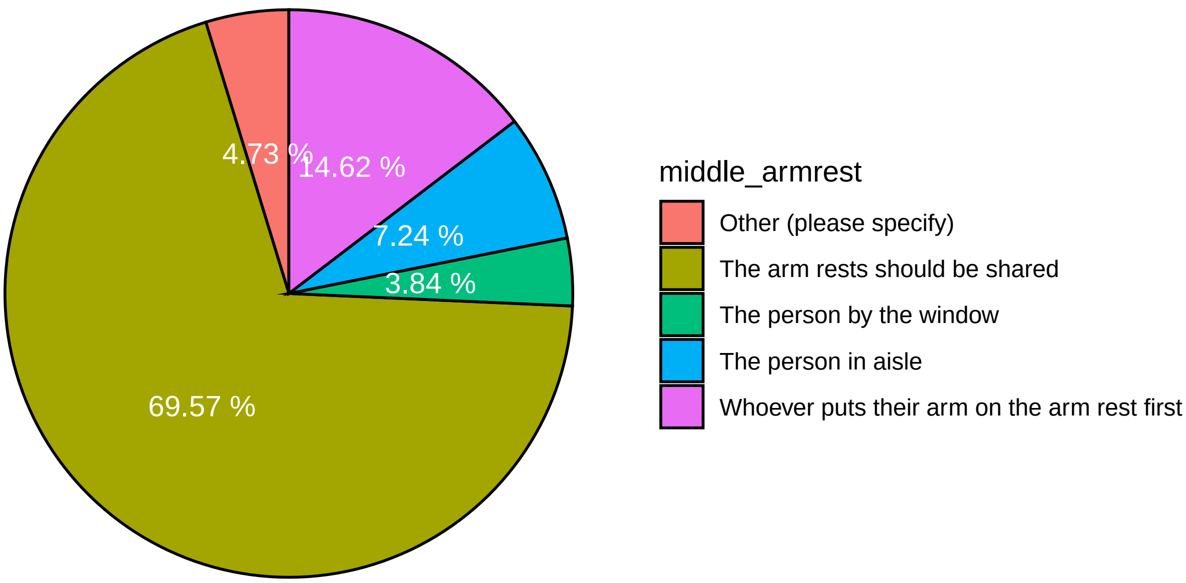
```
flying <- flying %>%
  filter(complete.cases(.))
q4_1.df <- flying %>%
  filter(!is.na(middle_armrest)) %>%
  group_by(middle_armrest) %>%
  summarise(n = n()) %>%
  mutate(percentage = round(n/sum(n)*100, 2))
q4_1.df
```

```
## # A tibble: 5 x 3
##   middle_armrest                                n percentage
##   <chr>                                     <int>      <dbl>
## 1 Other (please specify)                       32       4.73
## 2 The arm rests should be shared              471      69.6
## 3 The person by the window                     26       3.84
## 4 The person in aisle                          49       7.24
## 5 Whoever puts their arm on the arm rest first    99      14.6
```

2. Produce Figure 4.1.

Solution 4.2

```
ggplot(q4_1.df, aes(x = "", y = n, fill = middle_armrest, label = paste(percentage, "%"))) +
  geom_col(width = 1, color = "black") +
  geom_text(position = position_stack(vjust = 0.5), color="white") +
  coord_polar(theta = "y", direction = 1) +
  theme_void()
```

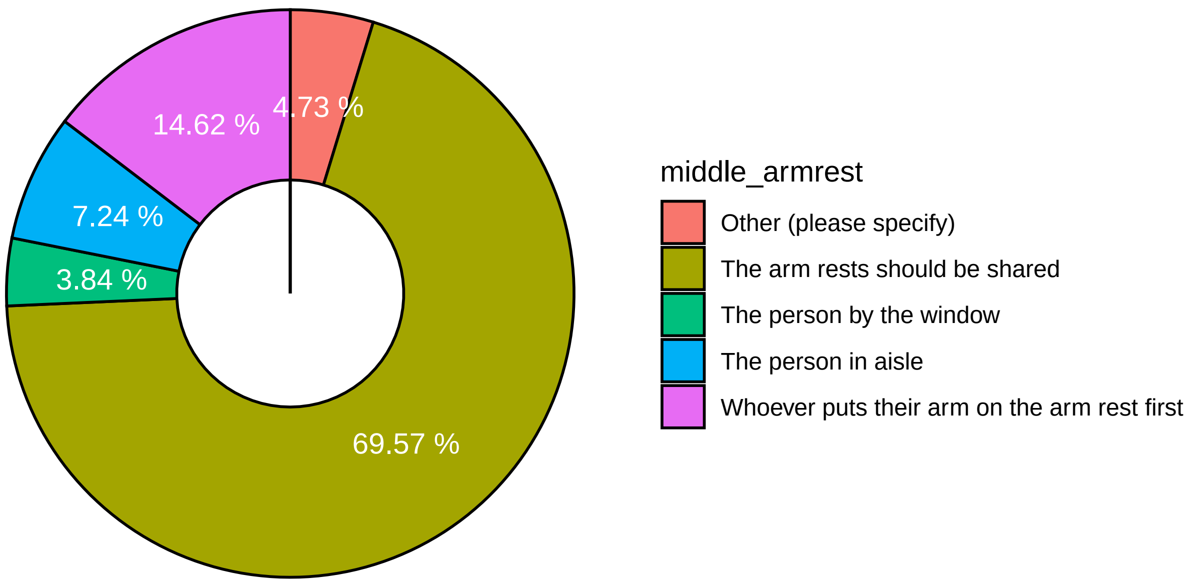


3. Produce Figure 4.2.

Solution 4.3

```
# generate the same as before but with the center not filled

ggplot(q4_1.df, aes(x = "", y = n, fill = middle_armrest, label = paste(percentage, "%"))) +
  geom_col(width = 1, color = "black") +
  geom_text(position = position_stack(vjust = 0.5), color="white") +
  coord_polar(theta = "y", direction = -1) +
  theme_void() +
  annotate("rect", xmin = 0, xmax = 0.6, ymin = 0, ymax = Inf, fill = "white", alpha = 1, color="black"
```



Question 5

1. For this question, use flying data without the missing values. We focus on the variables middle_armrest and age. Produce the table shown below, which shows the frequency of each response to the question “in a row of two seats, who should get to use the middle arm rest?” across the age of the respondents.

Solution 5.1

```
q5_1.df <- flying %>%
  group_by(middle_armrest, age) %>%
  summarise(n = n(), .groups = "drop") %>%
  pivot_wider(names_from = age, values_from = n)
q5_1.df
```

```
## # A tibble: 5 x 5
##   middle_armrest                              `18-29` `30-44` `45-60` `> 60`
##   <chr>                                         <int>   <int>   <int>  <int>
## 1 Other (please specify)                            7      12       7      6
## 2 The arm rests should be shared                   81     115     144    131
## 3 The person by the window                          4       6      12      4
## 4 The person in aisle                              20      14       8      7
## 5 Whoever puts their arm on the arm rest first     25      26      32     16
```

2. Use a chi-square test to test the hypothesis middle_armrest and age are independent.

Solution 5.2

```r
chisq.test(q5_1.df[,2:5])
```

```
##
##  Pearson's Chi-squared test
##
## data:  q5_1.df[, 2:5]
## X-squared = 30.909, df = 12, p-value = 0.002034
```

Part 3: the external opt datasets (Q6-Q8)

In this section, we focus on 3 **external** Excel files that contain data of Obstetrics and Periodontal Therapy.The external files are avilable online in BB. Description of each variable can be seen in https://rdrr.io/cran/medicaldata/man/opt.html. To access the datasets opt1, opt2 and opt3, these Excel files first need to be imported to R. This means that you need to read these external files to R. The excel files are available in BB. If you do not know how to do it you can look at https://datatofish.com/import-excel-r/.

The datasets opt1 and opt2 contain the same information about different trial participants (i.e., different subjects), while the dataset opt3 contains additional information about the medical condition (disease) of the participants. Variables names for all datasets are given below.

```r
opt1 <- read_excel("opt1.xlsx")
opt2 <- read_excel("opt2.xlsx")
opt3 <- read_excel("opt3.xlsx")
```

```r
names(opt1)
```

```
## [1] "PID"          "Clinic"       "Group"        "Age"          "Education"
## [6] "BMI"          "Birthweight"
```

```r
names(opt2)
```

```
## [1] "Participant_ID" "Clinic"         "Group"          "Age"
## [5] "Education"      "BMI"            "Birthweight"
```

```r
names(opt3)
```

```
## [1] "participantID" "clinic"        "Hypertension"  "Diabetes"
```

Question 6

In this question we focus on the dataset opt3.

1. In the opt3 dataset, the variable Diabetes is recorded as "Yes" and "No" while the variable Hypertension as "Y" and "N" (see the panel below). Replace values of Hypertension from "Y" into "Yes" and "N" into "No".

Solution 6.1

```r
opt3 <- opt3 %>%
  mutate(Hypertension = ifelse(Hypertension == "Y", "Yes", "No"))
head(opt3)
```

```
## # A tibble: 6 x 4
##   participantID clinic Hypertension Diabetes
```

```
##            <dbl> <chr> <chr>         <chr>
## 1        100034 NY    No            No
## 2        100042 NY    No            No
## 3        100067 NY    No            No
## 4        100083 NY    No            No
## 5        100091 NY    No            No
## 6        100109 NY    No            No
```

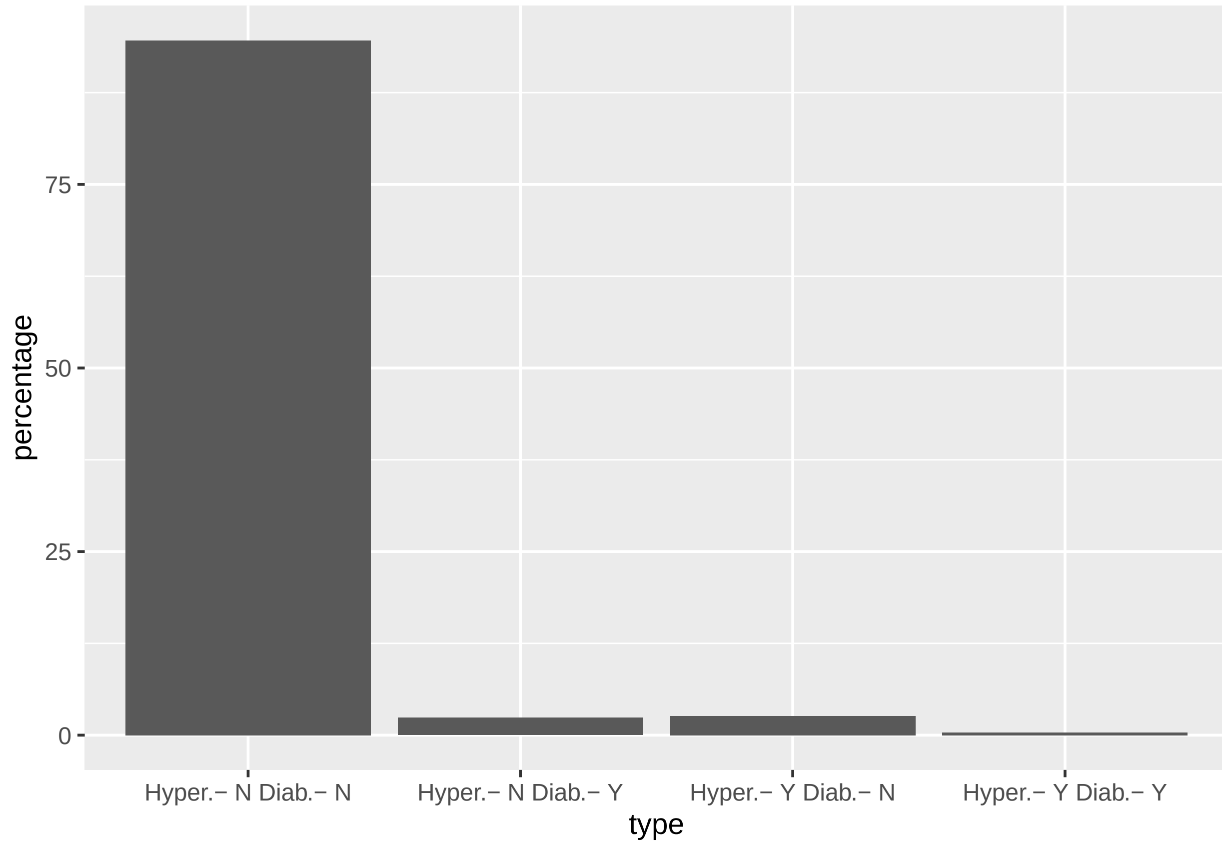
2. Produce a 2×2 table for **Hypertension X Diabetes** and based on this table, produce Figure 6.1 which shows the proportion of observations in each combination of Hypertension and Diabetes.

Solution 6.2

```
q6_2.df <- opt3 %>%
  group_by(Hypertension, Diabetes) %>%
  summarise(n = n(), .groups = "drop") %>%
  mutate(percentage = round(n/sum(n)*100, 2)) %>%
  mutate(type = paste("Hyper.-", substr(Hypertension, 1, 1), "Diab.-", substr(Diabetes, 1, 1)))
head(q6_2.df)
```

```
## # A tibble: 4 x 5
##   Hypertension Diabetes     n percentage type
##   <chr>        <chr>    <int>      <dbl> <chr>
## 1 No           No         757      94.6  Hyper.- N Diab.- N
## 2 No           Yes         19       2.38 Hyper.- N Diab.- Y
## 3 Yes          No          21       2.62 Hyper.- Y Diab.- N
## 4 Yes          Yes          3       0.38 Hyper.- Y Diab.- Y
```

```
ggplot(q6_2.df, aes(x = type, y = percentage)) +
  geom_col() +
  theme_gray()
```



3. Use the R function prop.test() to test the hypothesis that the proportion of Hypertension among the subjects with Diabetes is equal to the proportion of Hypertension among the subjects without Diabetes.

Solution 6.3

```
# TODO - add the solution
```

4. In the dataset created in Question 6.1, the information of each patients appears in one line. Define a new variable, Disease which includes the information about the two variables Hypertension and Diabetes in one "Yes/No" variable. Note that after the transformation the new data consists of **two lines per subject** as can be seen in the panel below.

Solution 6.4

```
q6_4.df <- opt3 %>%
  pivot_longer(cols = c(Hypertension, Diabetes), names_to = "Disease", values_to = "Yes/No")
head(q6_4.df)
```

```
## # A tibble: 6 x 4
##   participantID clinic Disease      `Yes/No`
##           <dbl> <chr>  <chr>        <chr>
## 1        100034 NY     Hypertension No
## 2        100034 NY     Diabetes     No
## 3        100042 NY     Hypertension No
## 4        100042 NY     Diabetes     No
## 5        100067 NY     Hypertension No
## 6        100067 NY     Diabetes     No
```

Question 7

1. For each of the three datasets that you imported to R, how many observations and variables are included in the dataset?

Solution 7.1

```
nrow(opt1)
```

```
## [1] 380
```

```
nrow(opt2)
```

```
## [1] 423
```

```
nrow(opt3)
```

```
## [1] 800
```

2. Merge the datasets opt1, opt2 and opt3 and include all participants from the datasets opt1 and opt2. How many observations (lines) and variables (columns) there are in the merged dataset?

Solution 7.2

```
opt2.temp.df <- opt2 %>%
  rename(PID=Participant_ID)
q7_2.df <- bind_rows(opt1, opt2.temp.df)
q7_2.df <- q7_2.df %>%
  left_join(opt3, by = join_by(PID==participantID))
head(q7_2.df)
```

```
## # A tibble: 6 x 10
##       PID Clinic Group   Age Education   BMI Birthweight clinic Hypertension
##     <dbl> <chr>  <chr> <dbl> <chr>     <dbl>       <dbl> <chr>  <chr>
## 1 101529 NY     T        33 LT 8 yrs     NA        3160 NY     No
## 2 201867 MN     C        26 8-12 yrs     22        2160 MN     No
## 3 202253 MN     C        30 8-12 yrs     34        3470 MN     No
## 4 100851 NY     T        23 8-12 yrs     NA        4510 NY     No
## 5 200562 MN     C        30 LT 8 yrs     28        3360 MN     No
## 6 202014 MN     T        27 8-12 yrs     43        3636 MN     No
## # i 1 more variable: Diabetes <chr>
```

```
dim(q7_2.df)
```

```
## [1] 803  10
```

3. For the merged dataset, count the missing data in each variable and remove them.

Solution 7.3

```
q7_2.df %>%
  summarise_all(funs(sum(is.na(.))))
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## i Please use a list of either functions or lambdas:
##
## # Simple named list: list(mean = mean, median = median)
##
```

```
## # Auto named with 'tibble::lst()': tibble::lst(mean, median)
##
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

```
## # A tibble: 1 x 10
##     PID Clinic Group   Age Education   BMI Birthweight clinic Hypertension
##   <int>  <int> <int> <int>     <int> <int>       <int>  <int>        <int>
## 1     0      0     0     0         0    73          14     23           23
## # i 1 more variable: Diabetes <int>
```

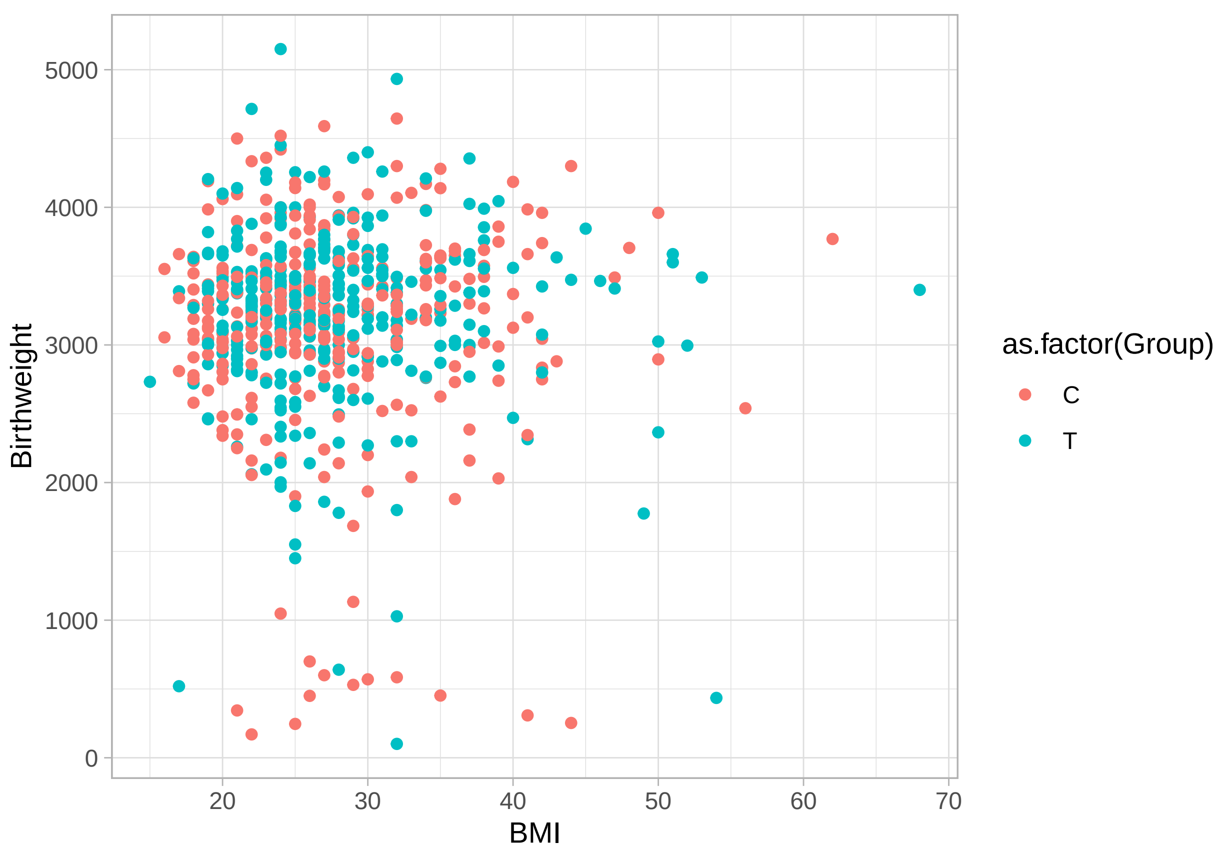
```
q7_3.df <- q7_2.df %>%
  filter(complete.cases(.))
dim(q7_3.df)
```

```
## [1] 694  10
```

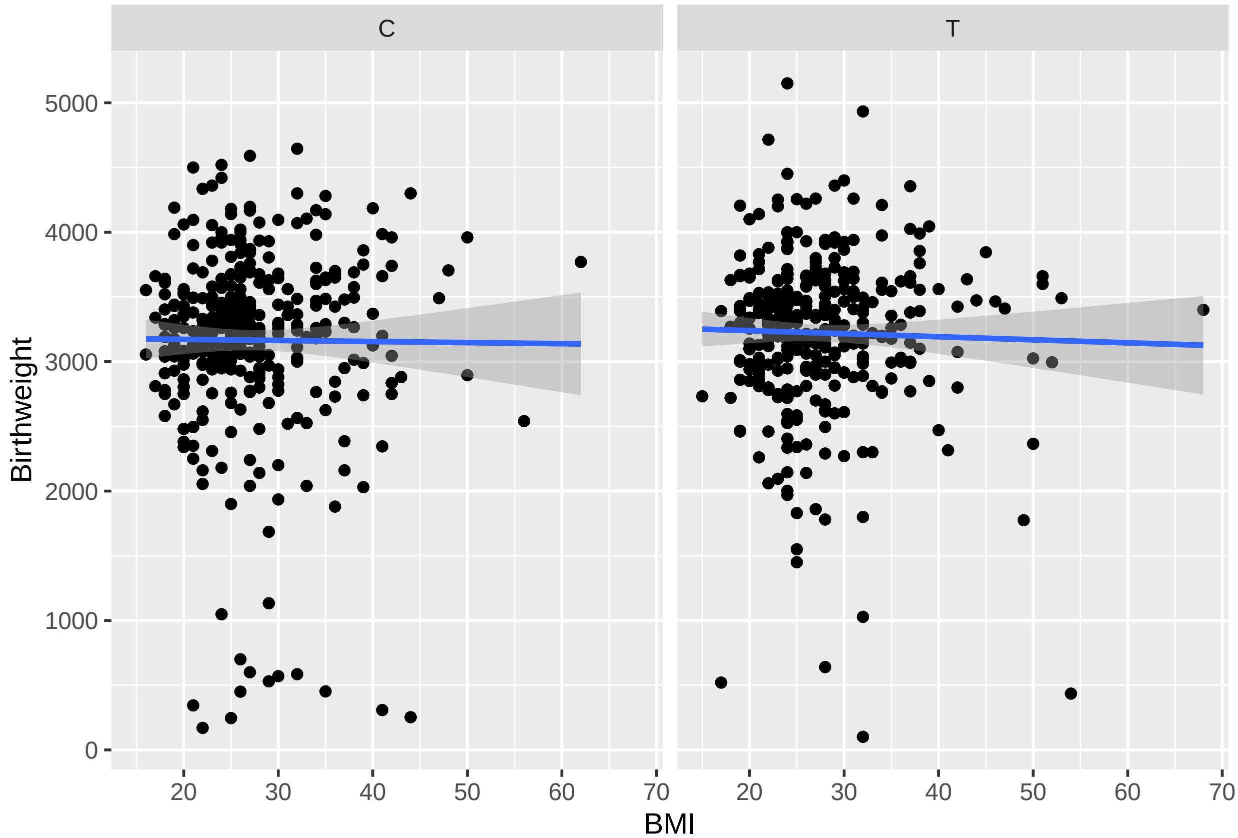
4. Produce Figure 7.1 and Figure 7.2 below and calculate the correlation between BMI and Birthweight for individuals from group C and individuals from group T (for each group separately, you can make the selection based on the variable Group).

Solution 7.4

```
ggplot(q7_3.df, aes(x = BMI, y = Birthweight, color=as.factor(Group))) +
  geom_point() +
  theme_light()
```



```r
ggplot(q7_3.df, aes(x = BMI, y = Birthweight)) +
  geom_point() +
  facet_wrap(~Group) +
  stat_smooth(method = "lm", se = T, formula = y ~ x)
```




```r
paste("group C")
```

```
## [1] "group C"
```

```r
cor(q7_3.df$BMI[q7_3.df$Group == "C"], q7_3.df$Birthweight[q7_3.df$Group == "C"])
```

```
## [1] -0.007758347
```

```r
paste("group T")
```

```
## [1] "group T"
```

```r
cor(q7_3.df$BMI[q7_3.df$Group == "T"], q7_3.df$Birthweight[q7_3.df$Group == "T"])
```

```
## [1] -0.02672523
```

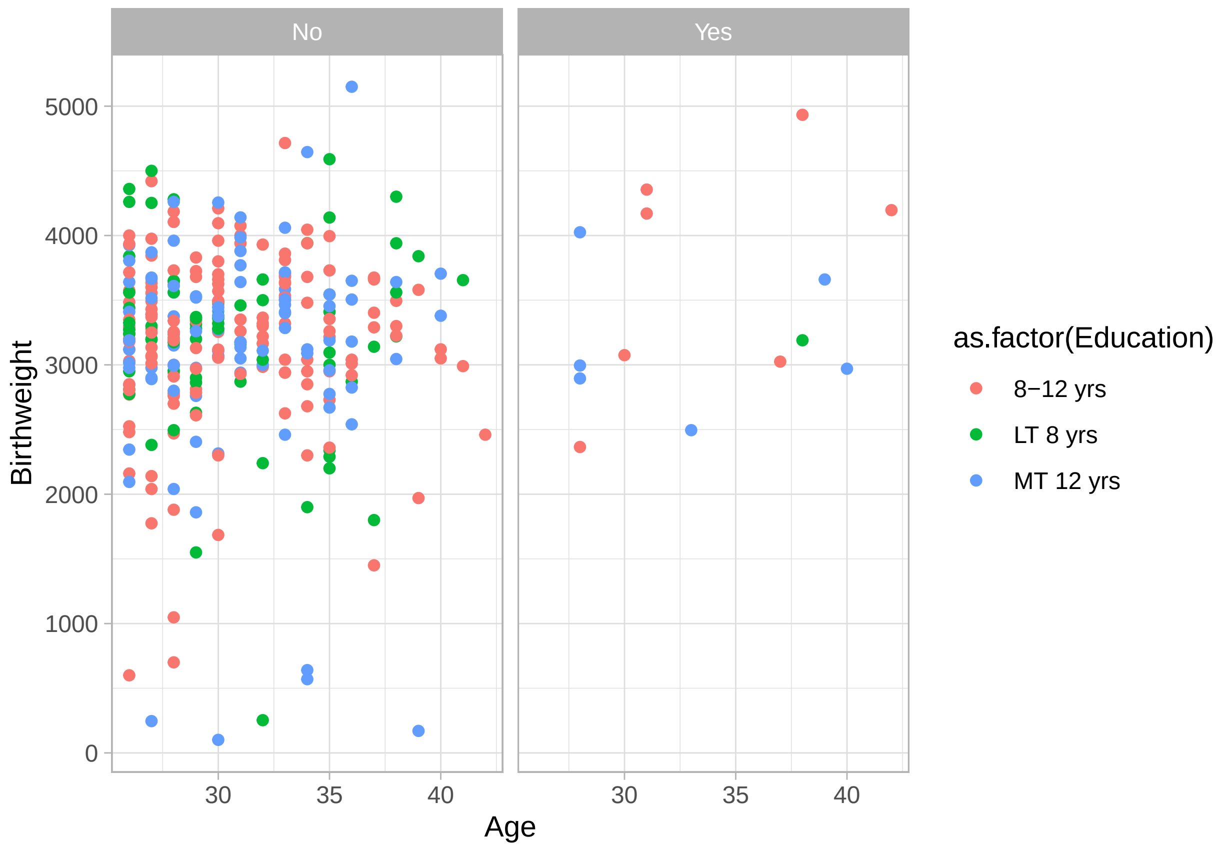
5. Use the merged data that you created in Question 7.3 to create a new data frame for all individuals older than 25 years old. How many observations are included in the new dataset? Produce Figure 7.3 for the new data which presents the Age Vs. Birthweight for each level of Diabetes colored by education level.

Solution 7.5

```
q7_5.df <- q7_3.df %>%
  filter(Age > 25)
nrow(q7_5.df)
```

```
## [1] 323
```

```
ggplot(q7_5.df, aes(x = Age, y = Birthweight, color=as.factor(Education))) +
  geom_point() +
  theme_light() +
  facet_wrap(~Diabetes)
```



6. Fit a linear regression model which includes the effect of the treatment group (the variable Group), hypertension and diabetes on the birth weight.

Solution 7.6

```
lm(Birthweight ~ Group + Hypertension + Diabetes, data = q7_3.df)
```

```
##
## Call:
## lm(formula = Birthweight ~ Group + Hypertension + Diabetes, data = q7_3.df)
##
## Coefficients:
##     (Intercept)           GroupT  HypertensionYes      DiabetesYes
##         3170.64            55.71          -367.32           267.92
```

7. Use the merged data that you created in Question 7.3 to create a new data frame for all individuals participated in the Intervention (T) group from Enrollment Center: Harlem Hospital (NY). For selection, you can use the variable Clinic.

Solution 7.7

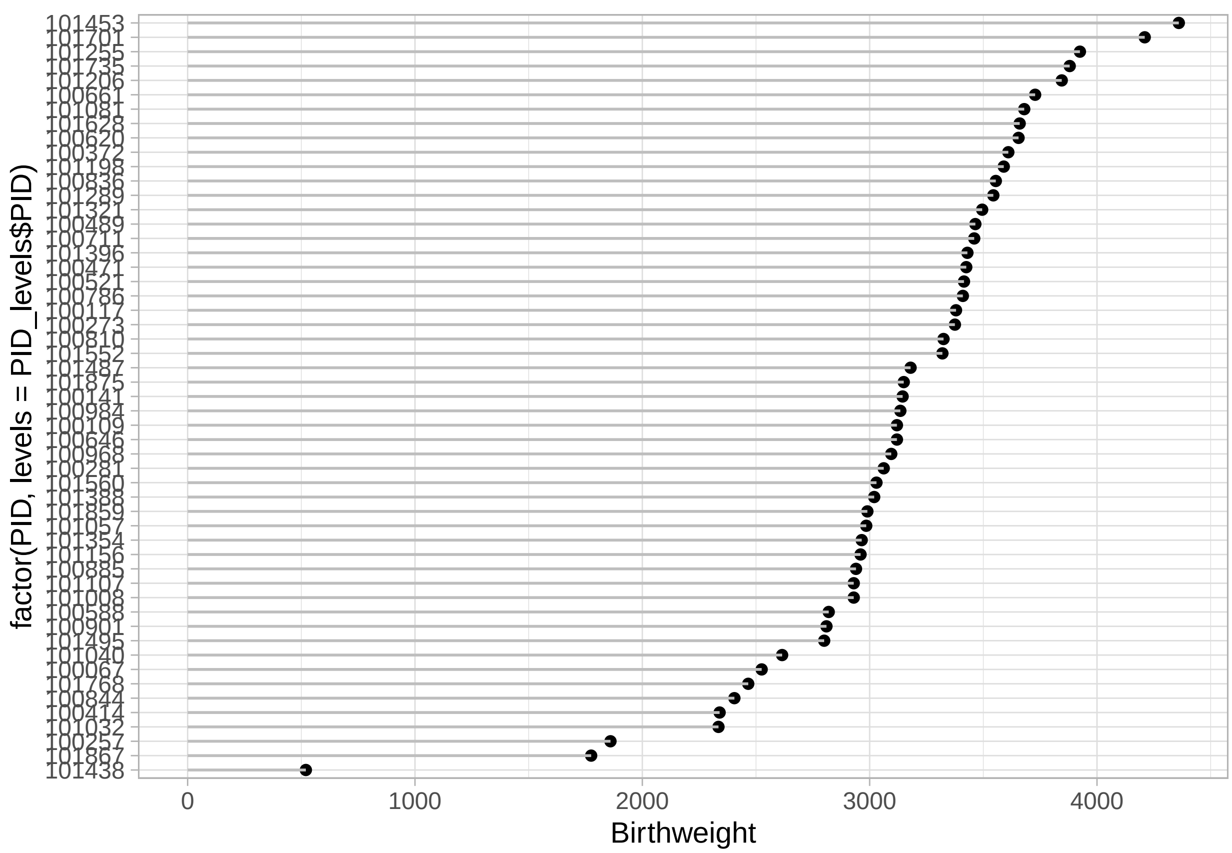
```
q7_7.df <- q7_3.df %>%
  filter(Group == "T" & Clinic == "NY")
nrow(q7_7.df)
```

```
## [1] 53
```

8. Produce Figure 7.4, to visualize the infant birth weight at time of birth of each participant.

Solution 7.8

```
PID_levels <- q7_7.df%>%
  arrange(Birthweight) %>%
  select(PID)
ggplot(q7_7.df, aes(x = factor(PID, levels=PID_levels$PID), y = Birthweight)) +
  geom_point() +
  theme_light() +
  geom_segment(aes(xend = factor(PID, levels=PID_levels$PID), y = 0, yend = Birthweight), color = "grey
  coord_flip()
```

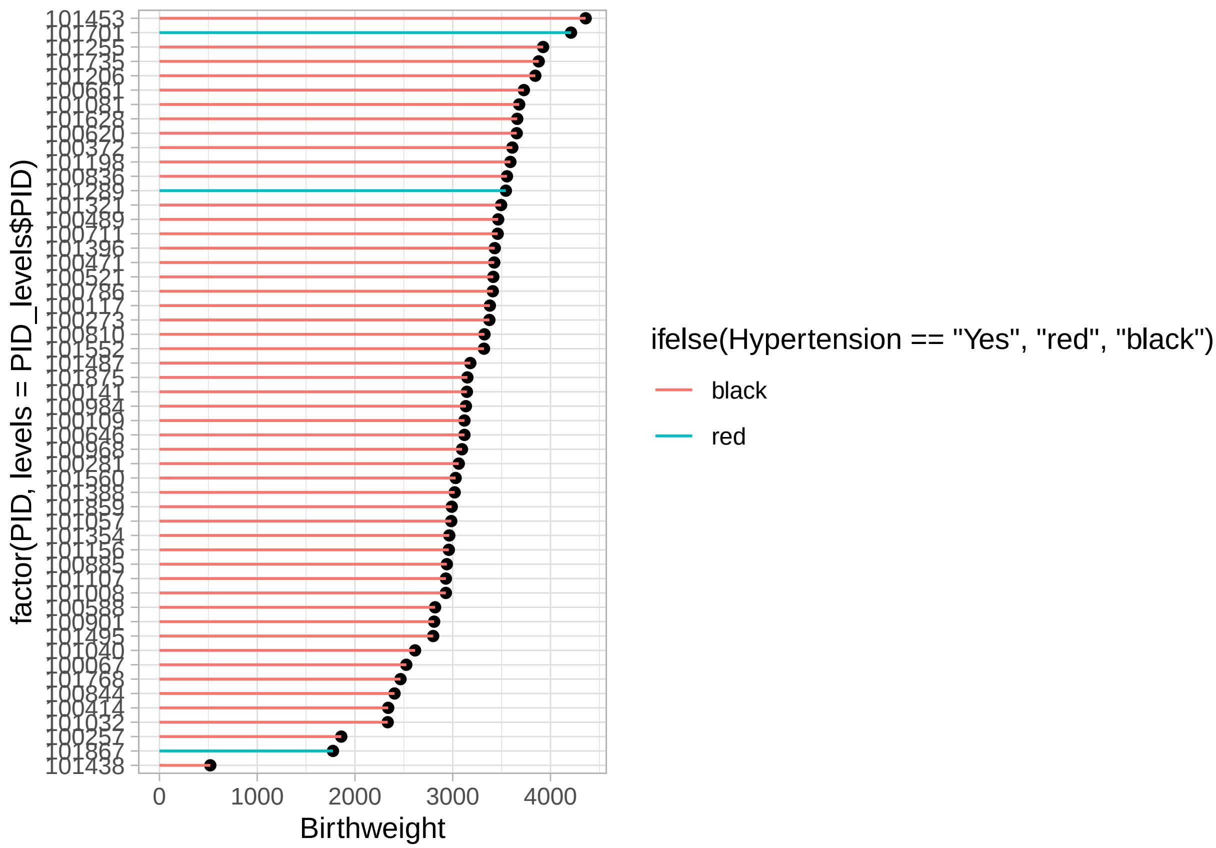


9. Highlight participants who had chronic hypertension at baseline with different color as shown in Figure 7.5.

Solution 7.9

TODO add color

```
# Todo fix color
PID_levels <- q7_7.df%>%
  arrange(Birthweight) %>%
  select(PID)
ggplot(q7_7.df, aes(x = factor(PID, levels=PID_levels$PID), y = Birthweight)) +
  geom_point() +
  theme_light() +
  geom_segment(aes(xend = factor(PID, levels=PID_levels$PID), y = 0, yend = Birthweight, color=ifelse(Hy
  coord_flip()
```



Question 8

For this question, use the merged dataset that was created in Question 7.

1. The table below presents the summary statistics (mean, standard deviation, median, minimum, maximum, sample size) of the variables Birthweight by group (Group). Produce the same table.

Solution 8.1

```
q8.1.df <- q7_3.df %>%
  group_by(Group) %>%
  summarise(
```

```r
    mean = mean(Birthweight, na.rm = T),
    sd = sd(Birthweight, na.rm = T),
    median = median(Birthweight, na.rm = T),
    min = min(Birthweight, na.rm = T),
    max = max(Birthweight, na.rm = T),
    n = n()
  )
q8.1.df
```

```
## # A tibble: 2 x 7
##   Group  mean    sd median   min   max     n
##   <chr> <dbl> <dbl>  <dbl> <dbl> <dbl> <int>
## 1 C     3166.  742.   3260   170  4645   348
## 2 T     3221.  623.   3285   101  5150   346
```

2. Write a function that received as an input a dataset data1, a numerical variable name x, a factor z and produces as an output the table of the summary statistics in presented in Question 8.1. Apply this function for the variables Age from the dataset opt1 and the variable Birthweight of the dataset that you used in Question 8.1. For both, the factor should be Group.

Solution 8.2

```r
summary_stats <- function(data1, x, z){
  data1 %>%
    group_by({{z}}) %>%
    summarise(
      mean = mean({{x}}, na.rm = T),
      sd = sd({{x}}, na.rm = T),
      median = median({{x}}, na.rm = T),
      min = min({{x}}, na.rm = T),
      max = max({{x}}, na.rm = T),
      n = n()
    )
}
summary_stats(q7_3.df, Age, Group)
```

```
## # A tibble: 2 x 7
##   Group  mean    sd median   min   max     n
##   <chr> <dbl> <dbl>  <dbl> <dbl> <dbl> <int>
## 1 C      25.8  5.54     25    16    42   348
## 2 T      26.0  5.59     25    16    42   346
```

Part 4: the unemp data

In this part of the exam, the questions are focused on the unemp dataset which is a part of the viridis R package. To access the data you need to install the package. More information can be found in https://cran.r-project.org/web/packages/viridis/viridis.pdf. Use the code below to access the data.

```r
library(viridis)
data(unemp)
names(unemp)
```

```
## [1] "id"          "state_fips"  "county_fips" "name"        "year"
## [6] "rate"        "county"      "state"
```

Question 9

1. How many counties have unemployment rates (the variable rate) higher than the 0.75 quantile? How many counties have unemployment rates lower than the 0.25 quantile?

Solution 9.1

```
q9.1.df <- unemp %>%
  filter(rate > quantile(rate, 0.75) | rate < quantile(rate, 0.25))
nrow(q9.1.df)
```

```
## [1] 1588
```

2. Create a subset of the dataset with states starting with the letter 'N' (NC,ND,NY, etc.,,,). How many observations and states are included in the new dataset? How many observations per state there are in the new dataset?

```
# letter N
q9.2.df <- unemp %>%
  filter(startsWith(state, "N"))
nrow(q9.2.df)
```

```
## [1] 389
```

```
q9.2_2.df <- q9.2.df %>%
  group_by(state) %>%
  summarise(n = n())
q9.2_2.df
```

```
## # A tibble: 8 x 2
##   state     n
##   <chr> <int>
## 1 NC      100
## 2 ND       53
## 3 NE       93
## 4 NH       10
## 5 NJ       21
## 6 NM       33
## 7 NV       17
## 8 NY       62
```

Solution 9.2

3. For the new dataset created in in Q9.2. Produce the pie plot and the barplot in a figure with one row of two panels, as presented in Figure 9.1. Note that both the pie and the bar plot show the number of observations per state.

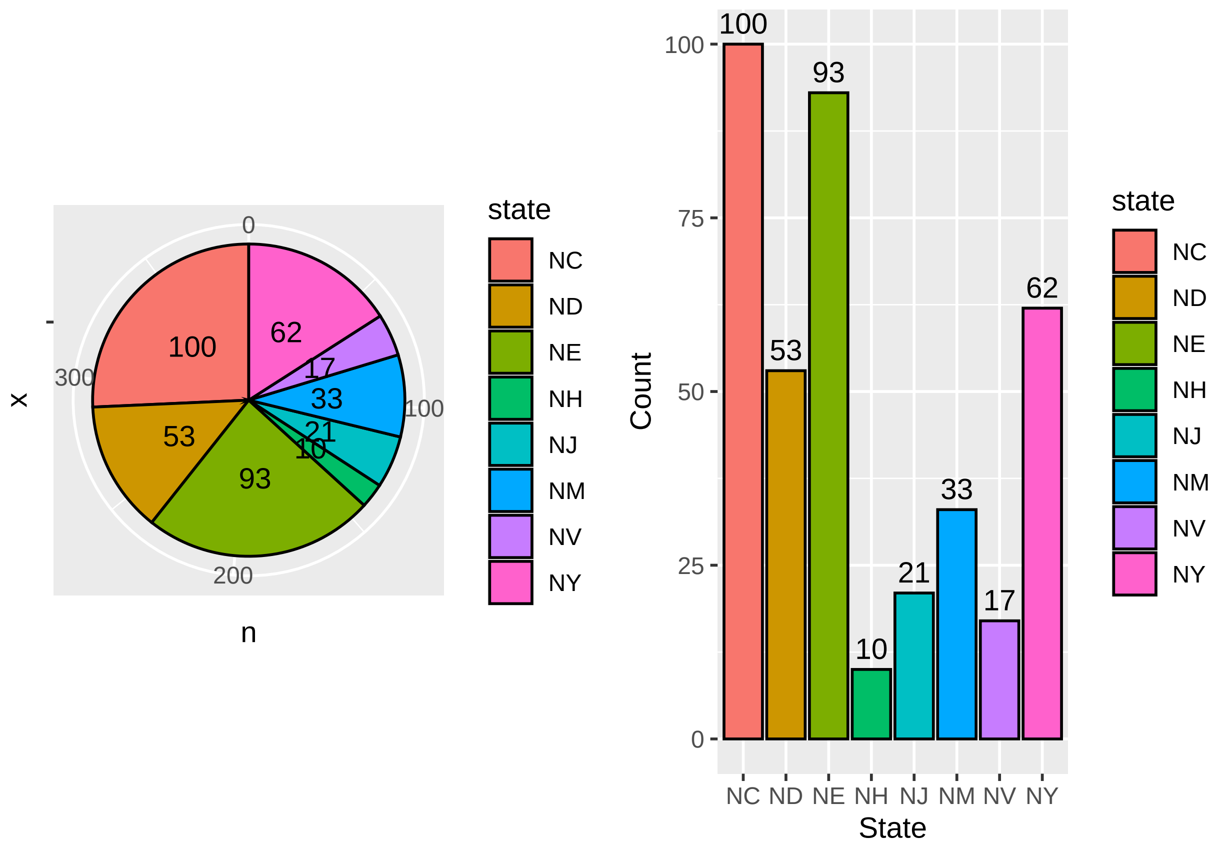
Solution 9.3

```
p2 <- ggplot(q9.2_2.df, aes(x="", y = n, fill=state)) +
  geom_col(width = 1, color="black") +
  coord_polar("y", start = 0, direction=1) +
  geom_text(aes(label = n), position = position_stack(vjust = 0.5))

p1 <- ggplot(q9.2_2.df, aes(x = state, y =n, fill=state)) +
  geom_col(color="black") +
  geom_text(aes(label = n), vjust = -0.5) +
```

```
  labs(x = "State", y = "Count")

gridExtra::grid.arrange(p2, p1, nrow = 1)
```



4. Create a subset of the dataset with states starting with the letter 'W'. Compute summary statistics (count, mean, sd) of the rate (the variable rate) by the variable state and produce that dataframe shown below.

Solution 9.4

```
q9.4.df <- unemp %>%
  filter(startsWith(state, "W"))

q9.4.df %>%
  group_by(state) %>%
  summarise(
    count = n(),
    mean = mean(rate, na.rm = T),
    sd = sd(rate, na.rm = T)
  )
```

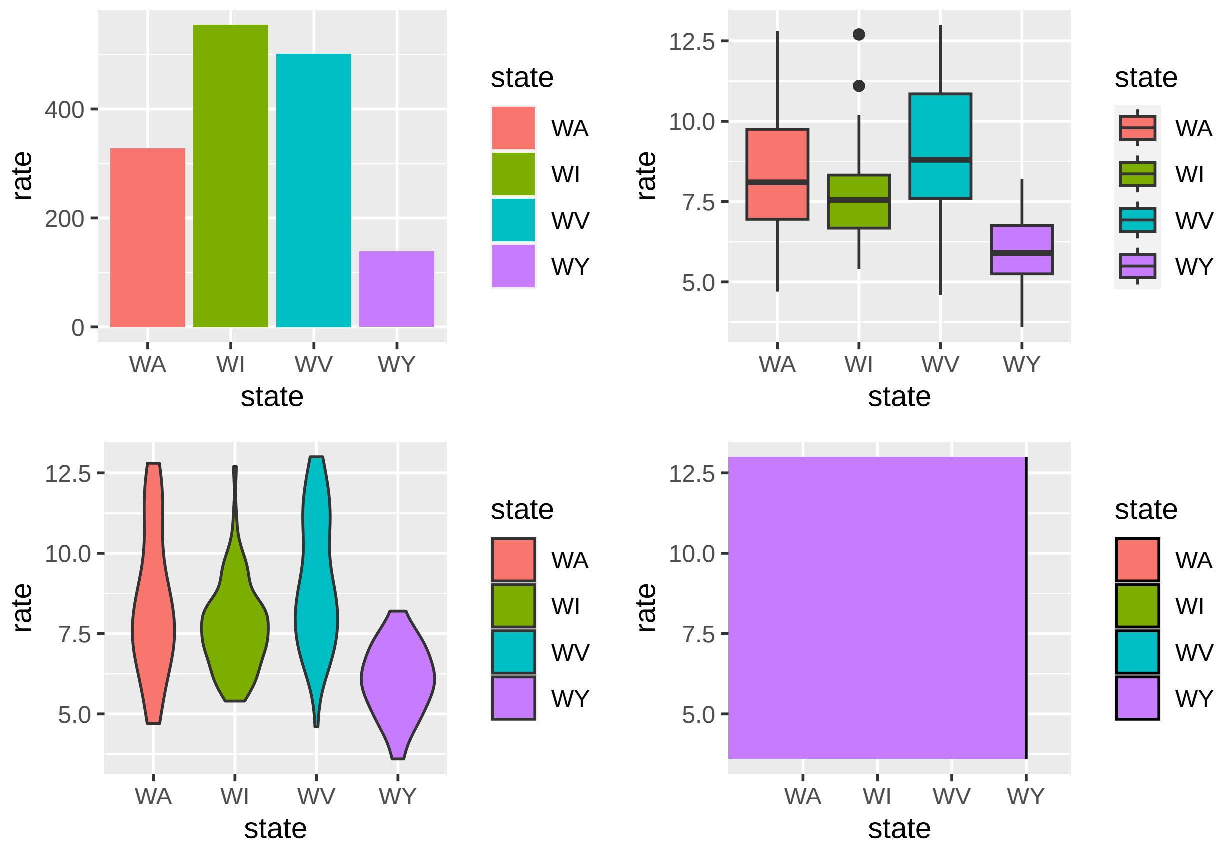
```
## # A tibble: 4 x 4
##   state count  mean    sd
##   <chr> <int> <dbl> <dbl>
## 1 WA       39  8.40  2.22
## 2 WI       72  7.70  1.38
```

```
## 3 WV        55  9.11  2.04
## 4 WY        23  6.04  1.11
```

5. For the dataset created in Q9.4, produce Figure 9.2 presented below.

Solution 9.5

```
p1 <- ggplot(q9.4.df, aes(x = state, y = rate, fill=state)) +
  geom_col()
p2 <- ggplot(q9.4.df, aes(x = state, y = rate, fill=state)) +
  geom_boxplot()
p3 <- ggplot(q9.4.df, aes(x = state, y = rate, fill=state)) +
  geom_violin()
p4 <- ggplot(q9.4.df, aes(x = state, y = rate, fill=state)) +
  geom_density()
gridExtra::grid.arrange(p1, p2, p3, p4, nrow = 2)
```



6. For that dataset created in Q9.4, fit a one-way ANOVA model in which the rate (the variable rate) is the independent variable and the state (state) is the factor. Print the F test statistics.

Solution 9.6

```
q9_6 <- anova(lm(rate ~ state, data = q9.4.df))
q9_6$`F value`
```

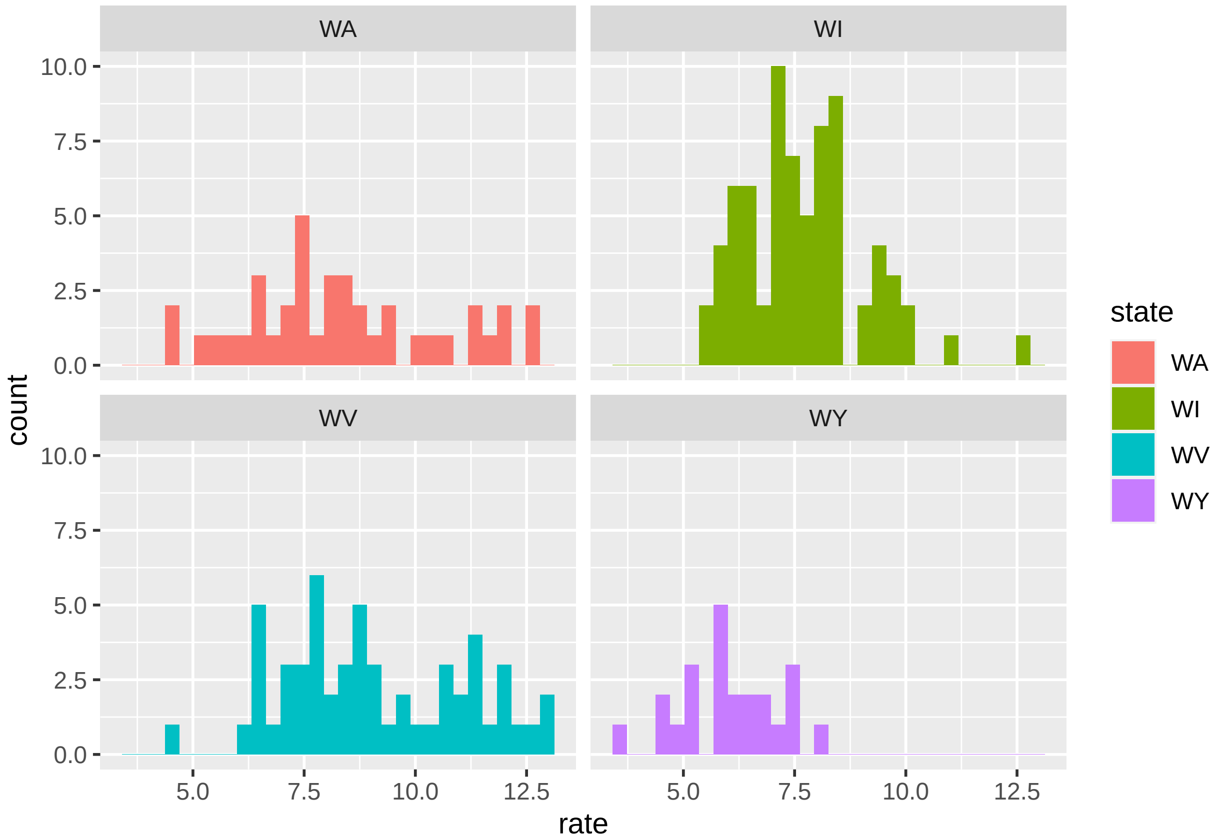
```
## [1] 18.0838      NA
```

7. produce Figure 9.3.

Solution 9.7

```
ggplot(q9.4.df, aes(x = rate, fill=state)) +
  geom_histogram() +
  facet_wrap(~state, ncol = 2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Part 5: the fish data

In this part we use the data fish which is a part of the rrcov R package. To access the data you need to install the package. More information can be found in https://search.r-project.org/CRAN/refmans/rrcov/html/fish.html. You can use the code below to access the data.

```
library(rrcov)
data(fish)
names(fish)
```

```
## [1] "Weight"  "Length1" "Length2" "Length3" "Height"  "Width"   "Species"
```

Question 10

Our aim in Q10.1-Q10.5 is to explore the correlation between the first 6 variables (i.e., the first 6 columns) in the fish dataset.

1. Produce Figure 10.1 using the pairs() function. Note that data are colored according to the variable Species.

Solution 10.1

2. Produce Figure 10.2 using R package GGally.

Solution 10.2

3. Produce Figure 10.3 using R package psych.

Solution 10.3

4. Produce Figure 10.4 using R package corrplot. Use the fish dataset without missing values.

Solution 10.4

5. Produce Figure 10.5 using R package corrplot. Use the fish dataset without missing values. Note that the variables are presented according to alphabet order.

Solution 10.5

6. In the fish dataset, observation 14 has a missing value in the variable Weight. Replace the missing value for this observation with the value 1253 and create a new dataset, fish2. Use the new dataset, create a scatter plot with connecting dots by a line as Figure 10.6 below:

Solution 10.6

7. For the dataset created in Question 10.4, test if the variances of the Length3 and Length1 variables are equal. Use significance level of 5%.

Solution 10.7

8. Based on the result of Question 10.7, conduct a t-test to test the hypothesis that the mean of Length3 is greater than the mean of Length1.

Solution 10.8