Programming in R (OC+DL): Exam part 1 (15/12/2023)

Juan VANEGAS (12/01/2024)

Contents

1 Introduction

1.1 General information

- Solutions to 20 questions using R in the Programming with R class part of the Msc. of Science in Statistics and Data Science.
- First part out of three parts of the exam.

2 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"
```

2.1 Question 1

1. How many countries are included in the data? Count missing values in each variable in the data.
2. Create a new data frame without the missing data. How many countries are left in the data?
3. Calculate the minimum and maximum for the variables life expectancy, mean schooling & gni per capita of both male and female.
4. For each gender, sort the life expectancy of all countries from the highest to the lowest, and print the top country.
5. For each gender, print the 15 countries with the highest life expectancy.
6. How many countries have both female and male life expectancy higher than 80?
7. Show the countries listed in question Q1.6.

2.1.1 Solution 1.1

```
q1_1 <- length(unique(real_data_GDI$country))
paste(q1_1, "countries in the data")
```

```
## [1] "191 countries in the data"
```

```
variables <- names(real_data_GDI)
print("Missing Values by column")
```

```
## [1] "Missing Values by column"
```

```
q1_1.2 <- real_data_GDI %>%
  summarise_all(list(function (x) sum(is.na(x))))
print(t(q1_1.2))
```

```
##                          [,1]
## country                     0
## female_life_expectancy      0
## male_life_expectancy        0
## female_mean_schooling      12
## male_mean_schooling        12
## female_gni_per_capita      13
## male_gni_per_capita        13
```

2.1.2 Solution 1.2

```
q1_2.df <- real_data_GDI %>%
  filter(complete.cases(.))
```

```r
q1_2 <- unique(q1_2.df$country)
paste(length(q1_2), "countries in the data")
```

```
## [1] "174 countries in the data"
```

2.1.3 Solution 1.3

```r
female.cols <- names(real_data_GDI)[startsWith(names(real_data_GDI), "female")]
male.cols <- names(real_data_GDI)[startsWith(names(real_data_GDI), "male")]
functions <- list(min=min, max=max)
```

```r
q1_2.df %>%
  select(all_of(male.cols)) %>%
  summarise_all(list(min=min, max=max))
```

2.1.3.1 Male Stats

```
##   male_life_expectancy_min male_mean_schooling_min male_gni_per_capita_min
## 1                     50.4                     2.2                     797
##   male_life_expectancy_max male_mean_schooling_max male_gni_per_capita_max
## 1                     83.2                    14.3                  105348
```

```r
q1_2.df %>%
  select(all_of(female.cols)) %>%
  summarise_all(list(min=min, max=max))
```

2.1.3.2 Female Stats

```
##   female_life_expectancy_min female_mean_schooling_min
## 1                       53.1                       1.3
##   female_gni_per_capita_min female_life_expectancy_max
## 1                       176                       88.3
##   female_mean_schooling_max female_gni_per_capita_max
## 1                      13.9                     75094
```

2.1.4 Solution 1.4

```r
sort.and.top <- function(df, index, col, top=5){
  return (df %>%
    select(all_of(c(index, col))) %>%
    arrange_at(col, desc) %>%
    head(n=top)
    )
}
```

```r
sort.and.top(q1_2.df, "country", "male_life_expectancy", 1)
```

2.1.4.1 Male top life expectancy

```
##     country male_life_expectancy
## 1 Australia                 83.2
```

```r
sort.and.top(q1_2.df, "country", "female_life_expectancy", 1)
```

2.1.4.2 Female top life expectancy

```
##                  country female_life_expectancy
## 1 Hong Kong, China (SAR)                   88.3
```

2.1.5 Solution 1.5

```r
sort.and.top(q1_2.df, "country", "male_life_expectancy", 15)
```

2.1.5.1 Male top 15 life expectancy

```
##                   country male_life_expectancy
## 1               Australia                 83.2
## 2  Hong Kong, China (SAR)                 82.7
## 3             Switzerland                 82.0
## 4                   Japan                 81.8
## 5                  Norway                 81.6
## 6                   Malta                 81.4
## 7                 Iceland                 81.2
## 8                  Sweden                 81.1
## 9                  Canada                 80.6
## 10            New Zealand                 80.6
## 11              Singapore                 80.6
## 12                  Italy                 80.5
## 13    Korea (Republic of)                 80.4
## 14             Luxembourg                 80.4
## 15                Ireland                 80.2
```

```r
sort.and.top(q1_2.df, "country", "female_life_expectancy", 15)
```

2.1.5.2 Female top 15 life expectancy

```
##                   country female_life_expectancy
## 1  Hong Kong, China (SAR)                   88.3
## 2                   Japan                   87.7
## 3     Korea (Republic of)                   86.8
## 4                   Malta                   86.1
## 5             Switzerland                   85.9
## 6               Australia                   85.8
## 7                   Spain                   85.8
## 8                  France                   85.5
## 9                   Italy                   85.1
## 10                 Norway                   84.9
## 11              Singapore                   84.9
## 12                 Sweden                   84.9
## 13             Luxembourg                   84.8
## 14                 Canada                   84.7
## 15                Finland                   84.7
```

2.1.6 Solution 1.6

```
q1_6.df <- q1_2.df[q1_2.df$female_life_expectancy > 80 & q1_2.df$male_life_expectancy > 80, ] %>%
  select(country)
```

2.1.7 Solution 1.7

```
print(q1_6.df)
```

```
##                    country
## 7                Australia
## 30                  Canada
## 68  Hong Kong, China (SAR)
## 70                 Iceland
## 75                 Ireland
## 76                  Israel
## 77                   Italy
## 80                   Japan
## 84     Korea (Republic of)
## 94              Luxembourg
## 100                  Malta
## 113            New Zealand
## 118                 Norway
## 141              Singapore
## 146                  Spain
## 150                 Sweden
## 151            Switzerland
```

2.2 Question 2

In this question, we use the dataset that was created in Q1.2 (the dataset without the missing values).

1. Define a new categorical variable flife_cat in the following way: Re-code the variable female_life_expectancy into three categories:

   female_life_expectancy <60: Low.
   female_life_expectancy 60-80: Medium.
   female_life_expectancy >80: High.

Count how many countries are included in each category.

2. Produce the pie plot and the barplot in a figure with one row of two panels, as presented in Figure 2.1.
3. Define a new dataset in which you include the countries for which female are classified with low life expectancy. Sort the data by male life expectancy and print the top 3 countries.
4. For the dataset in Q2.3, calculate the mean and standard deviation of male life expectancy and produce the output below.

2.2.1 Solution 2.1

```
categorize.life_exp <- function(row) {
  if (row < 60) return("Low")
  if (row <= 80) return("Medium")
  else return("High")
}
q2_1.df <- q1_2.df %>%
  mutate(flife_cat = factor(mapply(categorize.life_exp, female_life_expectancy), levels = c("Low", "Med
  group_by(flife_cat) %>%
  mutate(N = n()) %>%
  ungroup()

q2_1.2.df <- q2_1.df %>%
  select(flife_cat, N) %>%
  distinct()
```

2.2.2 Solution 2.2

```
p1 <- ggplot(q2_1.2.df, aes(x="", y=N, fill=flife_cat, label=N)) +
  geom_col(width=1, color="black") +
  geom_text(position = position_stack(vjust=0.5)) +
  coord_polar(theta="y", direction = 1) +
  labs(fill="Category") +
  theme_void()

p2 <- ggplot(q2_1.2.df, aes(x=flife_cat, y=N, fill=flife_cat, label=N)) +
  geom_bar(stat="identity", color="black") +
  geom_text(vjust=-0.25) +
  labs(fill="Category") +
  theme_void()

grid.arrange(p1, p2, ncol=2)
```

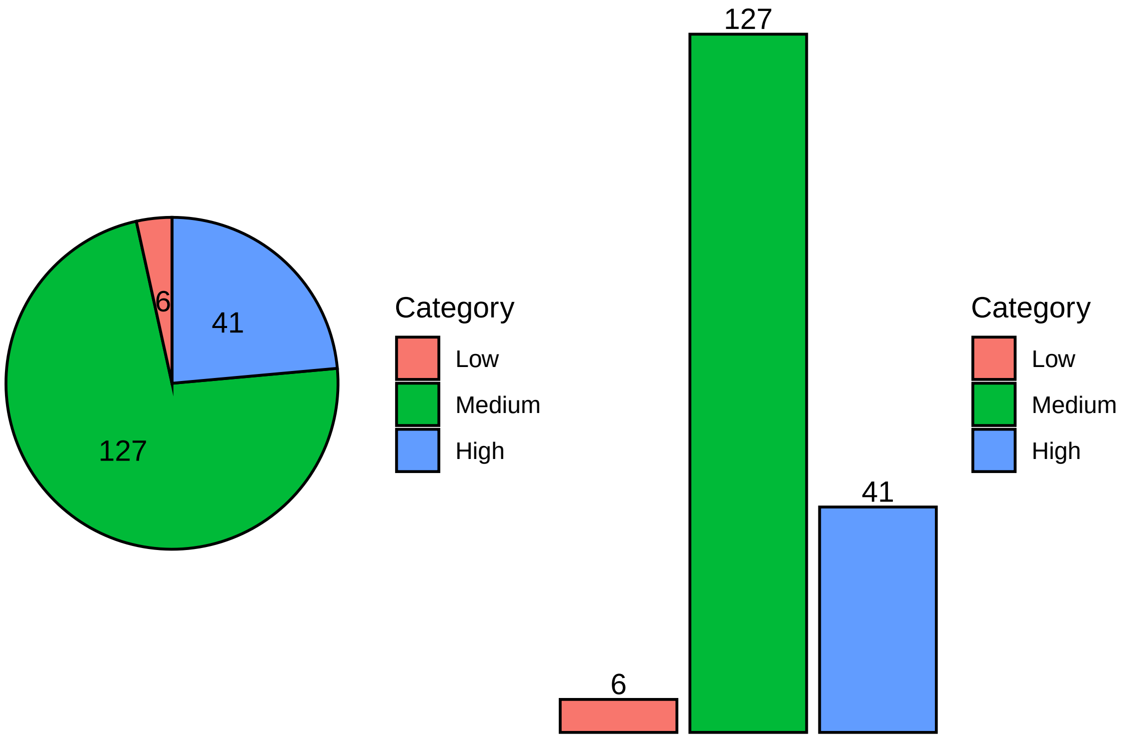


Figure 2.1

2.2.3 Solution 2.3

```
q2_3.df <- q2_1.df %>%
  filter(flife_cat=="Low")
res <- q2_3.df %>%
  select(country, male_life_expectancy) %>%
  arrange(-male_life_expectancy) %>%
  slice_head(n=3)
print(res)
```

```
## # A tibble: 3 x 2
##   country     male_life_expectancy
##   <chr>                      <dbl>
## 1 Ivory Coast                 57.4
## 2 South Sudan                 53.4
## 3 Nigeria                     52.3
```

2.2.4 Solution 2.4

4. For the dataset in Q2.3, calculate the mean and standard deviation of male life expectancy and produce the output below.

```
q2_3.df %>%
  summarise(mean_expectancy=mean(male_life_expectancy), sd_expectancy=sd(male_life_expectancy)) %>%
  select(mean_expectancy, sd_expectancy)
```

```
## # A tibble: 1 x 2
##   mean_expectancy sd_expectancy
##             <dbl>         <dbl>
## 1            52.6          2.56
```

2.3 Question 3

In this question we use the real_data_GDI dataset without the missing values.

1. Create a new data frame for countries with male life expectancy is higher than 53. How many countries are included in the new data set?
2. For the new dataset, calculate a 95% confidence interval for the female life expectancy using a standard normal distribution. Note that you need to program the formula for the confidence interval by yourself.
3. Write a function that receives a numerical vector and produces as a numerical output 95% confidence interval and the mean of numerical vector. Inside your function, use the R function t.test() to calculate the confidence interval and the mean. Apply this function to female life expectancy in the new data defined in Q3.1.
4. Use the R package interpretCI (and the meanCI() function) to calculate the confidence interval for the female life expectancy using a standard normal distribution in the new dataset defined in Q3.1.

2.3.1 Solution 3.1

```
q3_1.dt <- q1_2.df %>%
  filter(male_life_expectancy > 53)
paste(nrow(q3_1.dt))
```

```
## [1] "170"
```

2.3.2 Solution 3.2

```
data <- q3_1.dt$female_life_expectancy
n <- nrow(q3_1.dt)
u <- mean(data)
z <- qnorm(0.975)
ci <- c(lower=u - z * sd(data / sqrt(n)), upper=u + z * sd(data / sqrt(n)), mean=u)
ci
```

```
##    lower    upper     mean
## 73.35136 75.59570 74.47353
```

2.3.3 Solution 3.3

```
q3_3.f <- function(x) {
  x.test <- t.test(x, conf.level=0.95)
  return(c(x.test$conf.int, x.test$estimate))
}
q3_3.f(q3_1.dt$female_life_expectancy)
```

```
##                       mean of x
##  73.34326  75.60380  74.47353
```

2.3.4 Solution 3.4

```
q3_4 <- interpretCI::meanCI(q3_1.dt$female_life_expectancy)$result
res <- c(lower=q3_4$lower, upper=q3_4$upper, mean=q3_4$m)
res
```

```
##    lower    upper     mean
## 73.34326 75.60380 74.47353
```

2.4 Question 4

In this question, we use the real_data_GDI dataset without the missing values.

1. Produce the scatter plot in Figure 4.1. Note that the countries that are identified on the plot are all classified with low female life expectancy.
2. Produce the scatter plot in Figure 4.2.
3. Calculate the correlation between the variables female_mean_schooling and female_life_expectancy using the R function cor.test.
4. Fit a linear regression model which includes the mean schooling for female as predictor and the life expectancy for female as dependent variable. Print only coefficients panel (coefficients, standard error, t values and p values).
5. Produce a scatter plot of the female_mean_schooling vs female_life_expectancy, and add a regression line as shown in Figure 4.3.

2.4.1 Solution 4.1

```
ggplot(q2_1.df, aes(female_mean_schooling, female_life_expectancy, color=flife_cat)) +
  geom_point() +
  labs(color="Category") +
  geom_text_repel(aes(label=ifelse(flife_cat == "Low", country, "")), color="black", vjust=0.2, size=2.
```

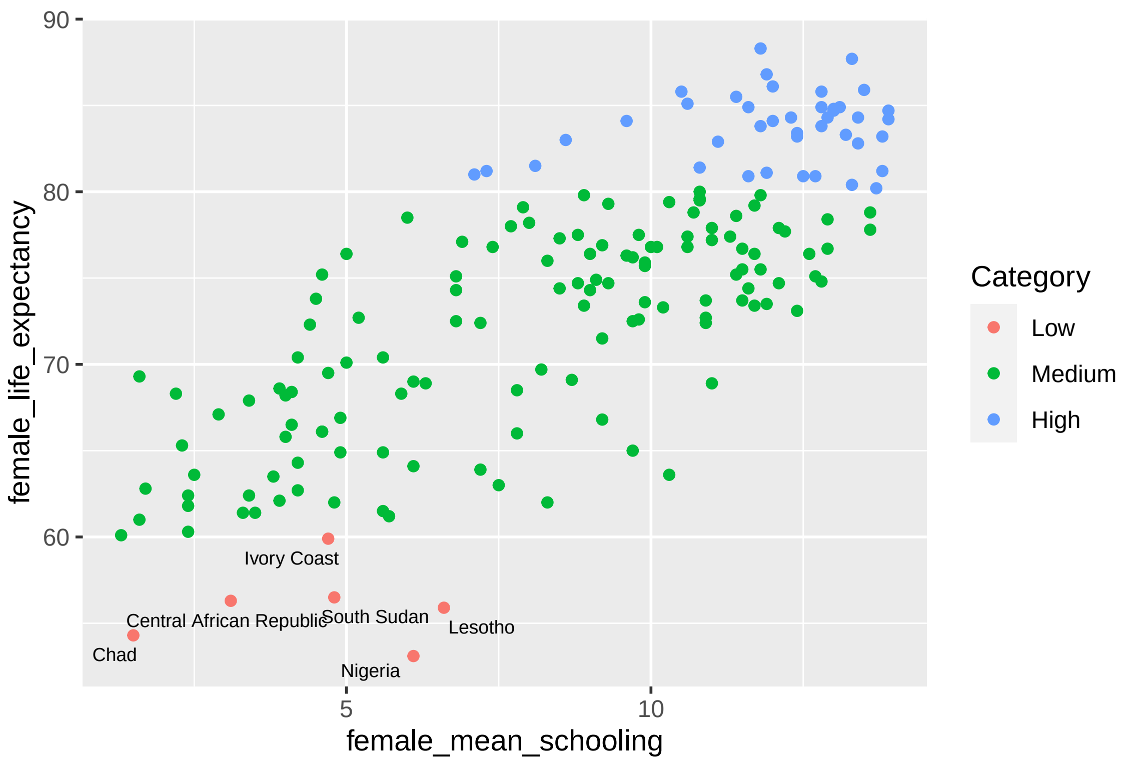


Figure 4.1

2.4.2 Solution 4.2

```
ggplot(q2_1.df, aes(female_mean_schooling, female_life_expectancy)) +
  geom_point(aes(color=flife_cat, size=female_gni_per_capita, alpha=0.4)) +
  labs(color="Category", size="gni per capita") +
  guides(alpha="none")
```

2.4.3 Solution 4.3

```
cor <- cor.test(q2_1.df$female_mean_schooling, q2_1.df$female_life_expectancy)
cor$estimate
```

```
##       cor
## 0.7812331
```

2.4.4 Solution 4.4

```
m4_4 <- lm(female_life_expectancy ~ female_mean_schooling, data=q2_1.df)
summary(m4_4)$coefficients
```

```
##                        Estimate Std. Error  t value      Pr(>|t|)
## (Intercept)           58.642943  1.0101189 58.05549 6.451493e-115
## female_mean_schooling  1.760213  0.1072436 16.41322  4.893158e-37
```

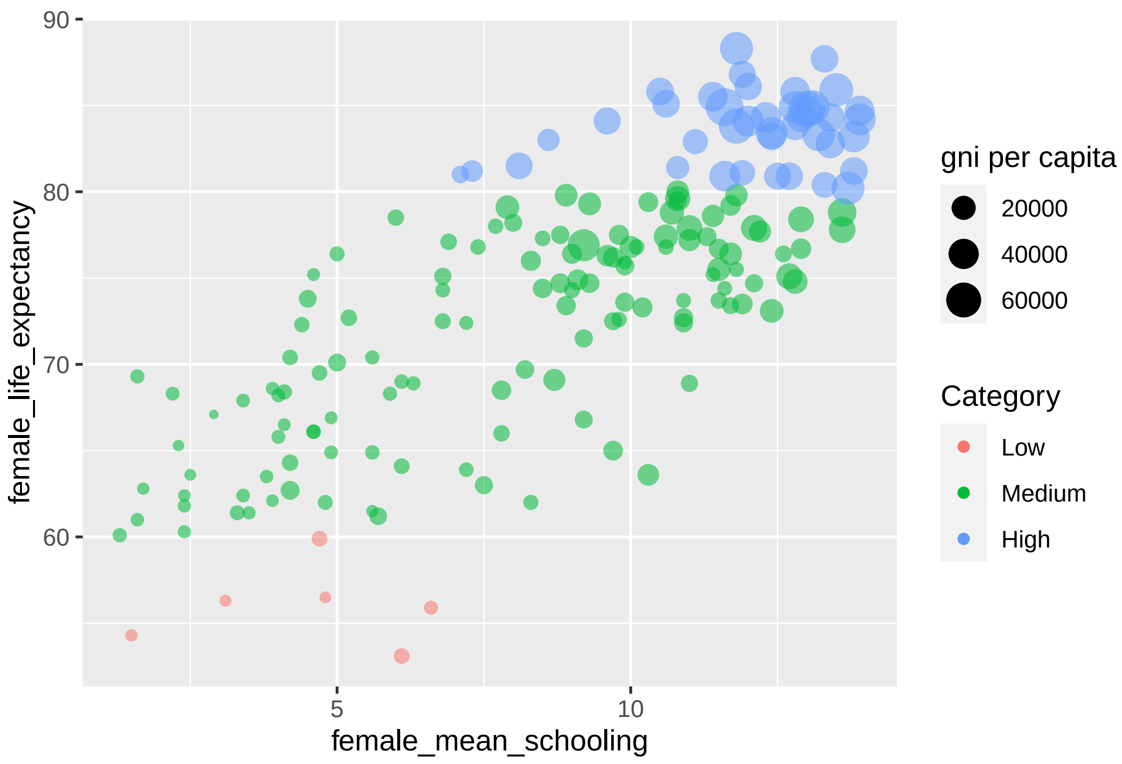


Figure 4.2

2.4.5 Solution 4.5

```
ggplot(q2_1.df, aes(female_mean_schooling, female_life_expectancy)) +
  geom_point() +
  geom_smooth(method=lm, se=FALSE, formula = y ~ x) +
  theme_bw()
```

/newpage

3 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

```
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"
```

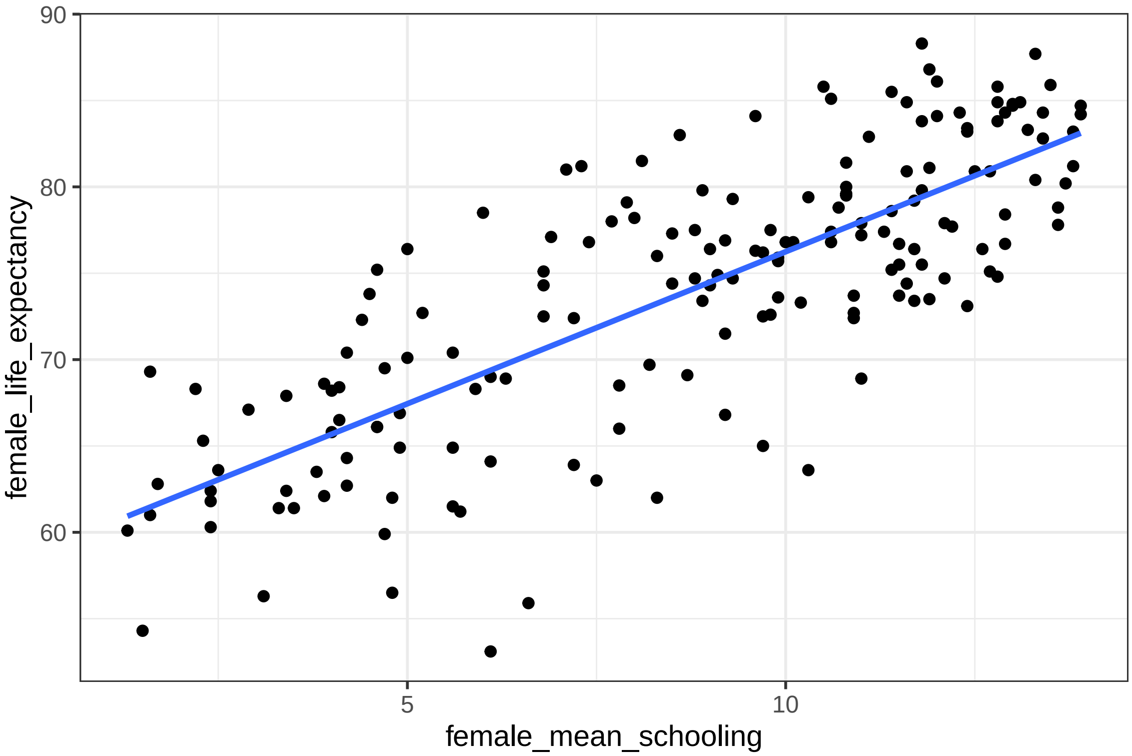


Figure 4.3

```
## [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"
```

3.1 Question 5

1. Remove the missing values from the data. How many observations remain in the data?

2. For the rest of question 5 we use the flying data without the missing values. Produce the data frame shown below, which shows the number of respondents for each age and gender category.
3. Produce the box plot in Figure 5.1.
4. Use a barplot to visualize the distribution of the gender across the factor levels of the age as shown in Figure 5.2.
5. Produce plot in Figure 5.3.

3.1.1 Solution 5.1

```
flying.df <- data.frame(flying[complete.cases(flying), ])
nrow(flying.df)
```

```
## [1] 677
```

3.1.2 Solution 5.2

```
flying.df %>%
  select(age, gender) %>%
  group_by(age, gender) %>%
  summarize(n=n(), .groups="keep")
```

```
## # A tibble: 8 x 3
## # Groups:   age, gender [8]
##   age   gender     n
##   <chr> <chr>  <int>
## 1 18-29 Female    75
## 2 18-29 Male      62
## 3 30-44 Female    78
## 4 30-44 Male      95
## 5 45-60 Female    95
## 6 45-60 Male     108
## 7 > 60  Female    87
## 8 > 60  Male      77
```

3.1.3 Solution 5.3

```
ggplot(flying.df, aes(gender, height, fill=gender)) +
  geom_boxplot() +
  geom_jitter(size=0.1) +
  labs(fill="Gender")
```

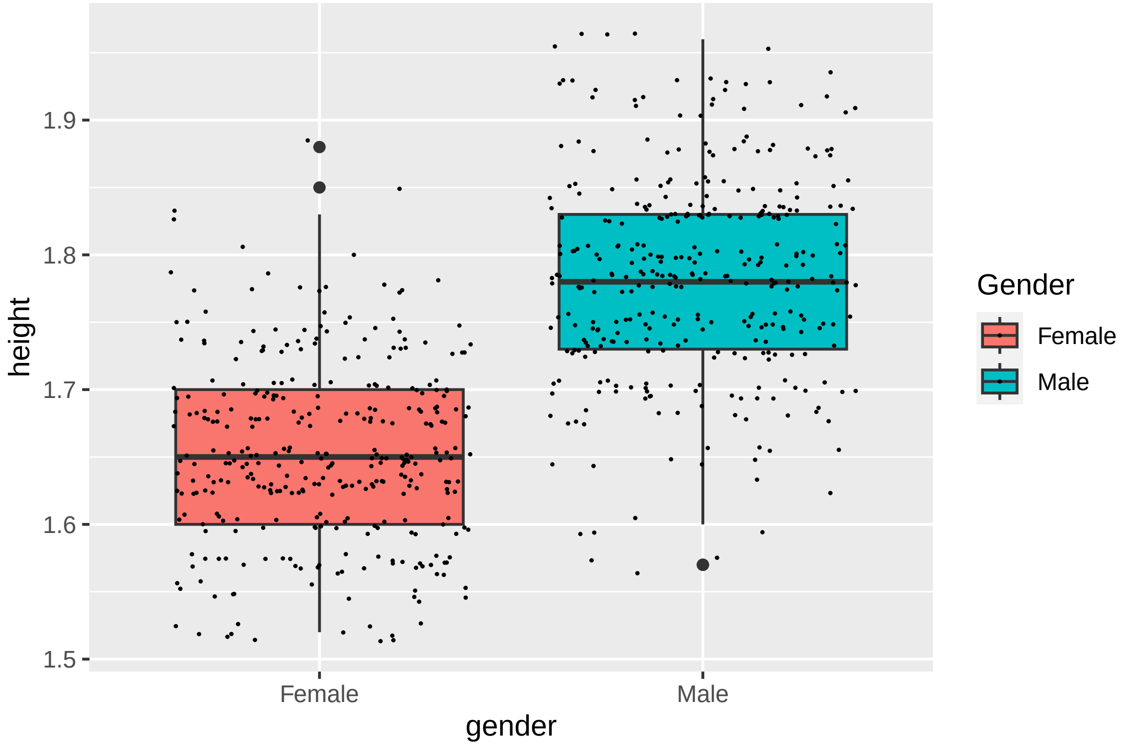


Figure 5.1

3.1.4 Solution 5.4

```
ggplot(flying.df, aes(age, fill=gender)) +
  geom_bar(position="dodge")
```

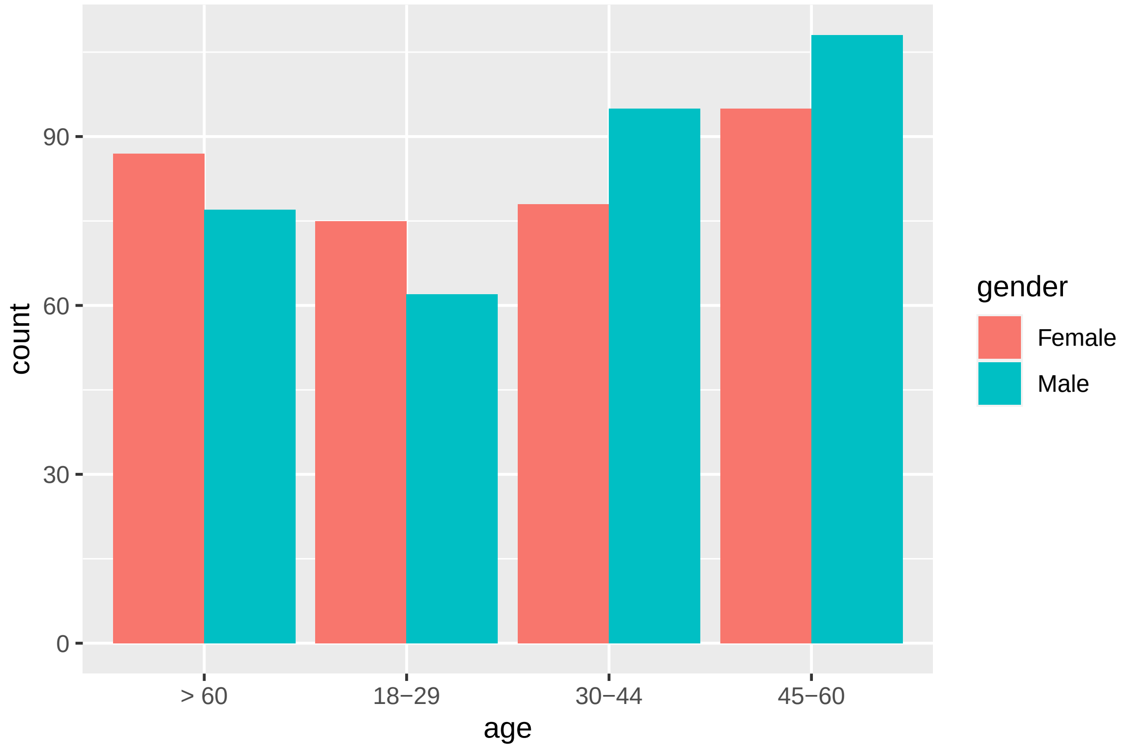


Figure 5.2

3.1.5 Solution 5.5

```
degrees.levels <- c("Less than high school degree",
             "High school degree",
             "Some college or Associate degree",
             "Bachelor degree",
             "Graduate degree"
             )
age.levels <- c(
  "18-29",
  "30-44",
  "45-60",
  "> 60"
)
flying.df <- flying.df %>%
  mutate(
    education = factor(education, levels=degrees.levels),
    age = factor(age, levels=age.levels)
    )
ggplot(flying.df, aes(x = age, fill = gender)) +
  geom_bar(position = "dodge") +
  facet_wrap(~education, nrow=1, labeller = label_wrap_gen()) +
  theme_bw() +
  labs(x = "Age") +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5))
```

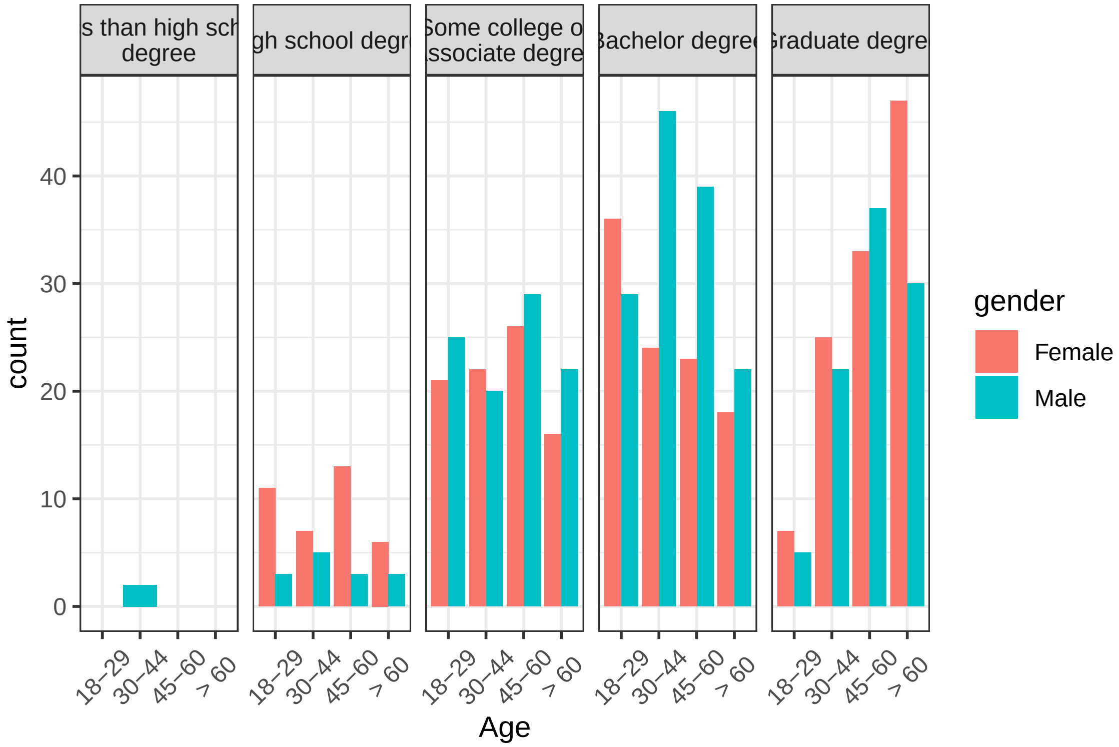


Figure 5.3

3.2 Question 6

In this question, we use the flying data without the missing values.

1. Produce the data frame below.
2. Produce the plot in Figure 6.1.
3. Count the distribution of the respondents' answers (from each gender and age group) to the question "is it rude to bring a baby on a plane?".
4. Produce plot in Figure 6.2.
5. Produce the plot in Figure 6.3 which shows the distribution of the male respondents' answers to 5 questions:

- "in general, is it rude to bring a baby on a plane?" (baby_on_plane)
- "is it rude to ask someone to switch seats with you in order to be closer to family?" (switch_for_family)
- is it rude to move to an unsold seat on a plane?" (moving_to_unsold_seat)
- "generally speaking, is it rude to say more than a few words to the stranger sitting next to you on a plane?"
  (talking_to_seatmate)
- "is it rude to wake a passenger up if you are trying to walk around?" (wake_passenger_walk)

3.2.1 Solution 6.1

```
q6_1.df <- flying.df %>%
  select(gender, baby_on_plane) %>%
  group_by(gender, baby_on_plane) %>%
  summarize(n=n(), .groups = "drop_last") %>%
  mutate(percentage = n / sum(n) * 100)
q6_1.df
```

```
## # A tibble: 6 x 4
## # Groups:   gender [2]
##   gender baby_on_plane           n percentage
##   <chr>  <chr>               <int>      <dbl>
## 1 Female No, not at all rude   255      76.1
## 2 Female Yes, somewhat rude     58      17.3
## 3 Female Yes, very rude         22       6.57
## 4 Male   No, not at all rude   214      62.6
## 5 Male   Yes, somewhat rude     89      26.0
## 6 Male   Yes, very rude         39      11.4
```

3.2.2 Solution 6.2

```
ggplot(q6_1.df, aes(gender, percentage, fill=baby_on_plane, label=sprintf("%.1f %%", percentage))) +
  geom_bar(stat="identity") +
  geom_text(position = position_stack(vjust=0.5)) +
  theme_bw()
```

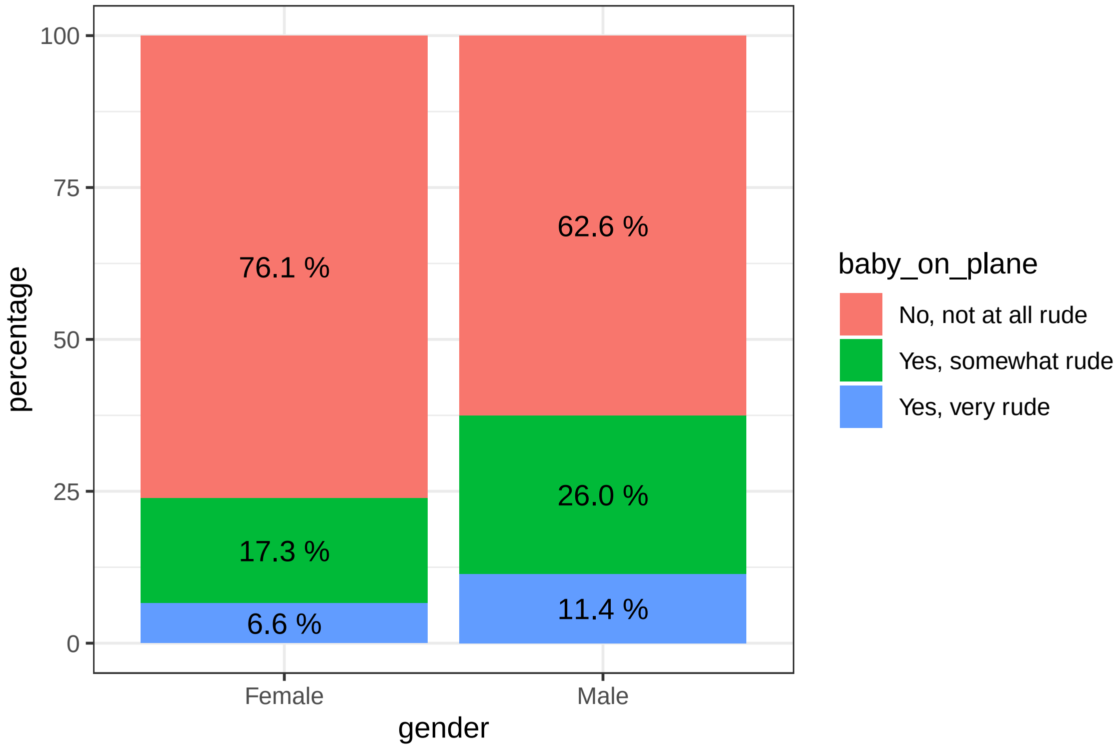


Figure 6.1

3.2.3 Solution 6.3

```
q6_3.df <- flying.df %>%
  select(age, gender, baby_on_plane) %>%
  group_by(age, gender, baby_on_plane) %>%
  summarize(n=n(), .groups = "drop_last") %>%
  mutate(percentage=n/sum(n) * 100)
q6_3.df
```

```
## # A tibble: 24 x 5
## # Groups:   age, gender [8]
##    age   gender baby_on_plane          n percentage
##    <fct> <chr>  <chr>              <int>      <dbl>
##  1 18-29 Female No, not at all rude    47      62.7
##  2 18-29 Female Yes, somewhat rude     23      30.7
##  3 18-29 Female Yes, very rude          5       6.67
##  4 18-29 Male   No, not at all rude    35      56.5
##  5 18-29 Male   Yes, somewhat rude     19      30.6
##  6 18-29 Male   Yes, very rude          8      12.9
##  7 30-44 Female No, not at all rude    62      79.5
##  8 30-44 Female Yes, somewhat rude     11      14.1
##  9 30-44 Female Yes, very rude          5       6.41
## 10 30-44 Male   No, not at all rude    50      52.6
## # i 14 more rows
```

3.2.4 Solution 6.4

```
# https://ggplot2.tidyverse.org/reference/labellers.html
ggplot(q6_3.df, aes(percentage,y="", fill=baby_on_plane)) +
  geom_col(position = "stack") +
  coord_polar(theta="x", direction = 1) +
  facet_grid(gender~age, labeller = label_both) +
  geom_label_repel(aes(label=sprintf("%.2f %%", percentage)), size=2.5, nudge_y = 0.8, show.legend = F)
  theme(axis.text.x = element_blank(), axis.title.x.bottom = element_blank())
```

3.2.5 Solution 6.5

```
answer_levels = c(
  "Yes, very rude",
  "Yes, somewhat rude",
  "No, not at all rude"
)
q6_5.df <- flying.df %>%
  select(respondent_id,
         switch_for_family,
         talking_to_seatmate,
         moving_to_unsold_seat,
         baby_on_plane,
         wake_passenger_walk,
         ) %>%
  gather("activity", "Response", -respondent_id) %>%
  group_by(activity, Response) %>%
  summarise(n=n(), .groups = "drop_last") %>%
```

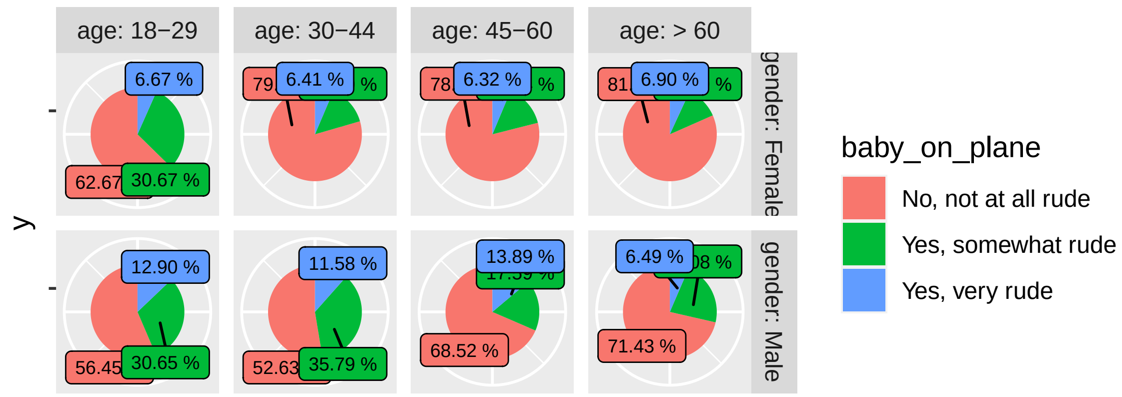


Figure 6.2

```
  mutate(percentage=n/sum(n) * 100) %>%
  mutate(Response = ifelse(Response == "No, not rude at all", "No, not at all rude", Response)) %>%
  mutate(Response = factor(Response, levels=answer_levels))
q6_5.manual.plot <- ggplot(q6_5.df, aes(x=percentage, y=activity, fill=Response, )) +
   geom_bar(stat="identity", width=0.5) +
   theme_minimal() +
   theme(legend.position="bottom") +
   scale_fill_manual(values = c("#5ab4ac", "#e5e5e5", "#d8b365")) +
   labs(x="Percentage", y="Behavior", fill="Response") +
   geom_text(aes(label=ifelse(Response == "Yes, somewhat rude", sprintf("%.0f %%", percentage) , "")),
             position = position_stack(vjust=0.5)) +
   geom_vline(xintercept = 0)
```

```
answer_levels = c(
  "No, not at all rude",
  "Yes, somewhat rude",
  "Yes, very rude"
)
q6_5likert.df <- flying.df %>%
  select(
         wake_passenger_walk,
         baby_on_plane,
         moving_to_unsold_seat,
         talking_to_seatmate,
         switch_for_family,
         ) %>%
```

```r
  mutate(moving_to_unsold_seat = ifelse(moving_to_unsold_seat == "No, not rude at all", "No, not at all
  mutate_all(function(x) factor(x, levels=answer_levels))
p <- likert(data.frame(q6_5likert.df))
a <- likert.bar.plot(p, legend.position = "bottom", text.size = 3,
                     order = F,
                     group.order = c(
                       "wake_passenger_walk",
                       "baby_on_plane",
                       "talking_to_seatmate",
                       "moving_to_unsold_seat",
                       "switch_for_family"
                       ),
                     ordered = F)
plot(a)
```

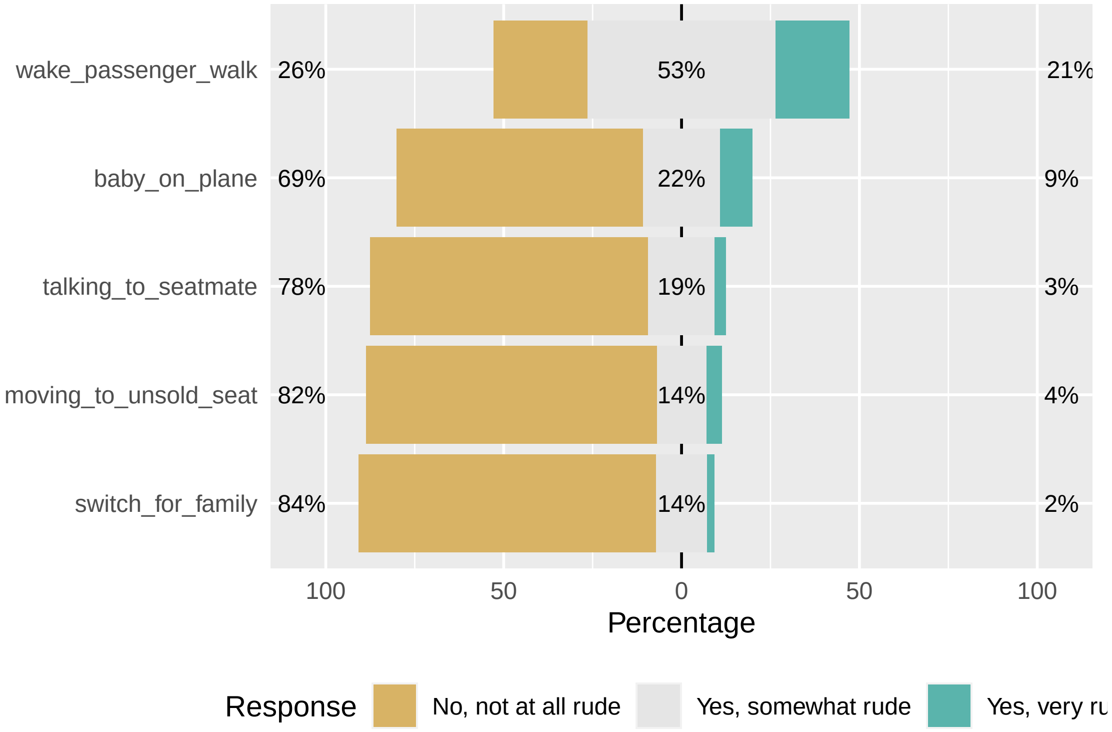


Figure 6.3

3.3 Question 7

In this question we focus on the flying data without missing values.

1. We focus on the variables gender and baby_on_plane. Produce the $2X3$ table shown below.
2. Use a chi-square test to test the hypothesis gender and baby_on_plane are independent.
3. Define an R object for the test statistic, plot the density plot of the test statistic under the null hypothesis and add the line for the observed test statistic.

3.3.1 Solution 7.1

```
q7_1.table <- flying.df %>%
  select(gender, baby_on_plane) %>%
  table()
q7_1.table
```

```
##         baby_on_plane
## gender   No, not at all rude Yes, somewhat rude Yes, very rude
##   Female                 255                 58             22
##   Male                   214                 89             39
```

3.3.2 Solution 7.2

```
q7_2.test <- chisq.test(flying.df$gender, flying.df$baby_on_plane)
```

3.3.3 Solution 7.3

```
q7_3.test.stat <- q7_2.test$statistic
q7_3.test.df <- q7_2.test$parameter
q7_3.test.density <- dchisq(q7_3.test.stat, q7_3.test.df)
label = sprintf("Statistic = %.2f with density %.4f", q7_3.test.stat, q7_3.test.density)
ggplot() +
  stat_function(fun=dchisq, args=list(df=q7_3.test.df), color="gray") +
  xlim(q7_3.test.stat - q7_3.test.stat / 2, q7_3.test.stat + q7_3.test.stat / 2) +
  geom_vline(xintercept = q7_3.test.stat, color="black") +
  labs(x="Test statistic", y="Density") +
  geom_text(aes(x=q7_3.test.stat, y=q7_3.test.density, label=label), hjust=-0, vjust=-1)
```

3.4 Question 8

Prepare a presentation of 5-10 slides using R markdown about the connection between the gender and the variable baby_on_plane. Make sure that your presentation includes:

- A Title slide.
- At least one slide with text.
- At least one slide with a figure
- At least one slide with text and a figure.

Please note that you **WILL NOT** be asked to give the presentation and you **WILL NOT** be asked questions about the presentation. Your aim in this question is to demonstrate that you know how to use R markdown to make a presentation about your analysis. More details how to make a presentation using R markdown: https://rmarkdown.rstudio.com/lesson-11.html.

/newpage # Part 3: 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.

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

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

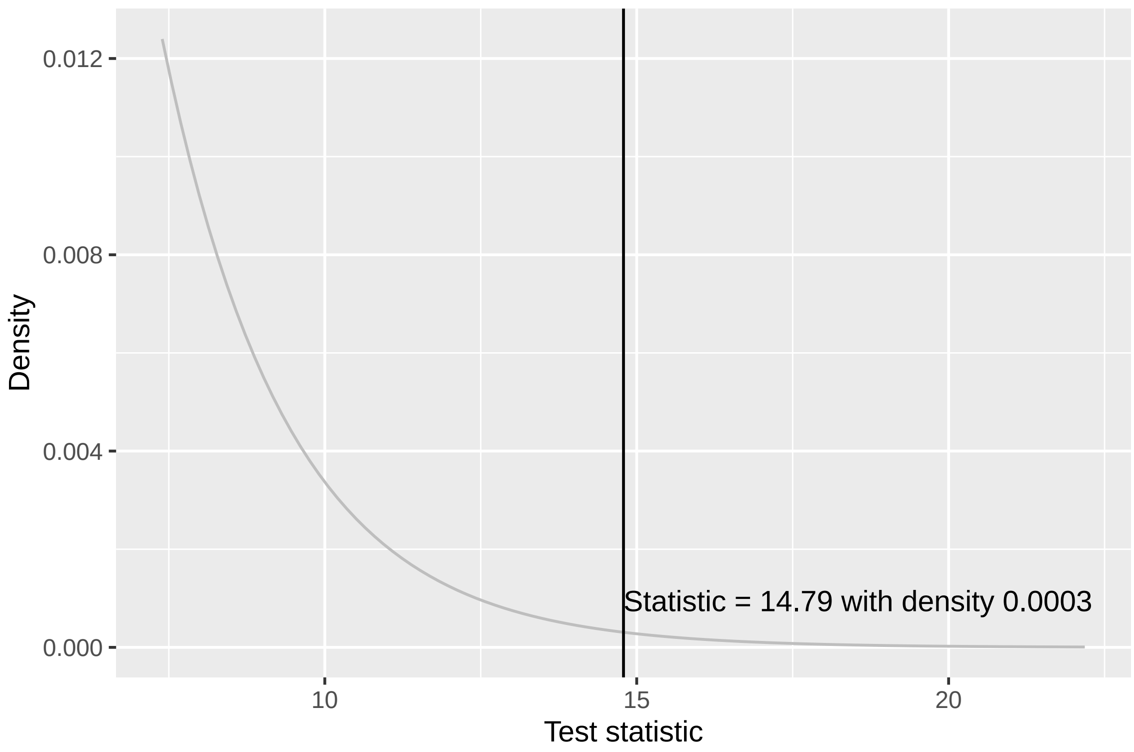


Figure 7.1

3.5 Question 9

For the unemp dataset,

1. How many observations are included in the dataset? How many states are included in this dataset?
2. How many counties there are in NY?
3. Create a new data frame named unemp_NY for NY state. Produce the following output for the variable rate:

3.5.1 Solution 9.1

```
q9_1.n <- nrow(unemp)
q9_1.n
```

```
## [1] 3218
```

```
length(unique(unemp$state))
```

```
## [1] 52
```

3.5.2 Solution 9.2

```
q9_2.counties.ny <- unemp %>%
  filter(state == "NY") %>%
  select(county) %>%
  unique()
nrow(q9_2.counties.ny)
```

```
## [1] 62
```

3.5.3 Solution 9.3

```
unemp_NY <- unemp %>%
  filter(state == "NY")
unemp_NY %>%
  select(rate) %>%
  summarize(min_rate_NY = min(rate),
          max_rate_NY = max(rate),
          mean_rate_NY = mean(rate),
          )
```

```
##   min_rate_NY max_rate_NY mean_rate_NY
## 1         5.6        13.3     8.009677
```

3.6 Question 10

Create a new data frame named sub_unemp, which includes data of 3 states: GA, TX and VA.

```
sub_unemp <- unemp %>%
  filter(state %in% c("GA", "TX", "VA"))
```

1. How many observations are included in the new data frame?
2. Produce Figure 10.1 presented below.
3. Save Figure 10.1, produced in Q10.2, as a png file and include it in the zip file of your solution.
4. Conduct a t-test to test the hypothesis that the unemployment rate in states TX and VA is equal against a two-sided alternative. What is the value of the test statistic? How many observations were included in the analysis?
5. Create a new R object that contains the upper and lower limit of the 95% confidence interval for the mean difference. DO NOT use xxx<-c(-0.2592,0.6928).
6. Test if the variance of the unemployment rate in the two states is equal.
7. If needed, adjust your analysis in Q10.4 according to the result obtained in Q10.6.

3.6.1 Solution 10.1

```
nrow(sub_unemp)
```

```
## [1] 547
```

3.6.2 Solution 10.2

```
p1 <- ggplot(sub_unemp, aes(x=state, y=rate, color = state)) +
  geom_boxplot() +
  theme(legend.position = "bottom")
p2 <- ggplot(sub_unemp, aes(x=state, y=rate, color = state)) +
  geom_violin() +
  theme(legend.position = "bottom")
layout <- matrix(c(1, 1, 2, 2, NA, 3, 3, NA), nrow = 2, byrow = TRUE)
p3 <- ggplot(sub_unemp, aes(x=rate, fill=state)) +
  geom_density(alpha=0.7) +
  theme(legend.position = "bottom")
p <- grid.arrange(p1, p2, p3, nrow=2, layout_matrix=layout)
```

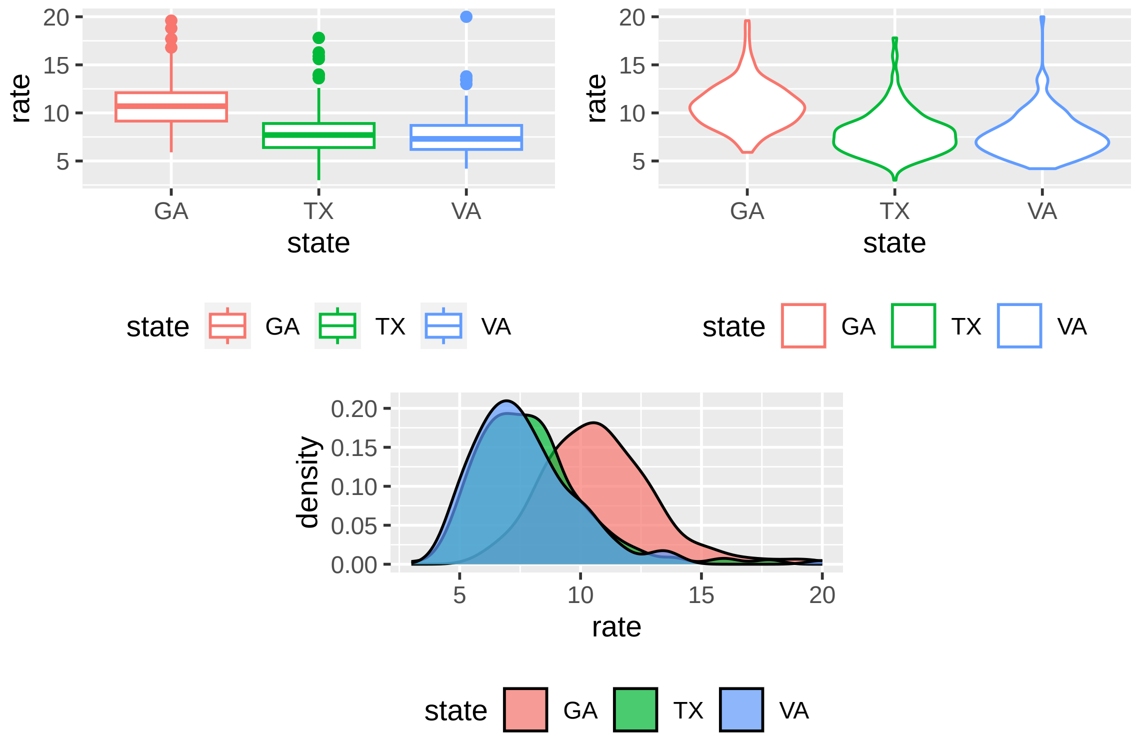


Figure 10.1

3.6.3 Solution 10.3

```
ggsave("q10_3.png", p)
```

3.6.4 Solution 10.4

4. Conduct a t-test to test the hypothesis that the unemployment rate in states TX and VA is equal against a two-sided alternative. What is the value of the test statistic? How many observations were included in the analysis?

```
q10_4.df <- sub_unemp %>%
  filter(state %in% c("TX", "VA")) %>%
  select(rate, state)
q10_4.test <- t.test(rate ~ state, data = q10_4.df, alternative = "two.sided")
q10_4.test
```

```
##
##  Welch Two Sample t-test
##
## data:  rate by state
## t = 0.89662, df = 274.16, p-value = 0.3707
## alternative hypothesis: true difference in means between group TX and group VA is not equal to 0
## 95 percent confidence interval:
##  -0.2592371  0.6928721
## sample estimates:
## mean in group TX mean in group VA
##         7.936220         7.719403
```

3.6.5 Solution 10.5

```
q10_5.ci <- q10_4.test$conf.int
q10_5.ci
```

```
## [1] -0.2592371  0.6928721
## attr(,"conf.level")
## [1] 0.95
```

3.6.6 Solution 10.6

```
q10_6.test <- var.test(rate ~ state, data = q10_4.df, alternative = "two.sided")
q10_6.test
```

```
##
##  F test to compare two variances
##
## data:  rate by state
## F = 1.0293, num df = 253, denom df = 133, p-value = 0.8618
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.7587649 1.3768044
## sample estimates:
## ratio of variances
##           1.029254
```

```
# Manually
s1_square <- var(sub_unemp %>%
                  filter(state == "TX") %>%
                  select(rate))
s2_square <- var(sub_unemp %>%
                  filter(state == "VA") %>%
                  select(rate))
f.test <- max(s1_square, s2_square)/min(s1_square, s2_square)
df <- c(nrow(sub_unemp %>%
              filter(state == "TX") %>%
              select(rate)) - 1,
        nrow(sub_unemp %>%
              filter(state == "VA") %>%
              select(rate)) - 1)
q10_6.test <- 2 * pf(f.test, df[1], df[2], lower.tail = FALSE)
q10_6.test
```

```
## [1] 0.86178
```

/newpage # Part 4: the pigs data In this part, the questions are focused on the pigs dataset which is a part of the emmeans R package. To access the data you need to install the package. More information can be found by help(pigs). You can use the code below to access the data.

```
library(emmeans)
data(pigs)
names(pigs)
```

```
## [1] "source"  "percent" "conc"
```

3.7 Question 11

In this question, we use the pigs dataset without the missing values.

1. Add to the pigs data a category variable percent_class that takes the value of "high" when the percent is 18, "moderate" when the percent is 12 or 15, and "low" when the percent is 9. What is the proportion of the pigs for which percent_class is high? Produce the plot presented in Figure 11.1.
2. For the new data, compute summary statistics (count, mean, sd) of the concentration of free plasma leucine (the variable conc) by the variable percent_class.
3. Use the function aov() to fit a one-way ANOVA model in which the concentration of free plasma leucine (the variable conc) is the independent variable and the protein percentage in the diet (percent_class) is the factor.
4. Print the ANOVA table for the model.
5. Create a new R object, F.value, that contains the value of the F test statistics. DO NOT use F.value=1.858.
6. Produce the diagnostic plots (qq normal plot for residuals and histogram for residuals) presented in Figure 11.2 and 11.3 below.

3.7.1 Solution 11.1

```r
levels = c("low", "moderate", "high")
pigs.df <- pigs[complete.cases(pigs), ]
q11_1.f <- function(x) {
  if (x == 18) return ("high")
  else if (x == 12 || x == 15) return ("moderate")
  else if (x == 9) return ("low")
}
pigs.df <- pigs.df %>%
  mutate(percent_class = factor(lapply(percent, q11_1.f), levels = levels))

q11_1.ratio <- nrow(pigs.df[pigs.df$percent_class == "high",])/nrow(pigs.df)
q11_1.ratio
```

```
## [1] 0.1724138
```

```r
q11_1.df <- pigs.df %>%
  group_by(source, percent_class) %>%
  summarize(n = n(), .groups = "drop_last") %>%
  mutate(percentage_by_source = 100 * n/sum(n))

ggplot(q11_1.df, aes("", percentage_by_source, fill=percent_class, label=
                       ifelse(source=="skim", sprintf("%.2f %%", percentage_by_source),
                              sprintf("%.0f %%", percentage_by_source)))) +
  geom_col() +
  coord_polar(theta = "y", direction = 1) +
  theme_grey() +
  scale_fill_grey() +
  geom_label_repel(size = 4, show.legend = F, position = position_stack(vjust=0.5), color="white") +
  facet_wrap(~source, labeller = label_both) +
  theme(axis.text.x = element_blank(),
        axis.title.x.bottom = element_blank(),
        axis.title.y.left = element_blank(),
        panel.grid = element_blank()
        )
```

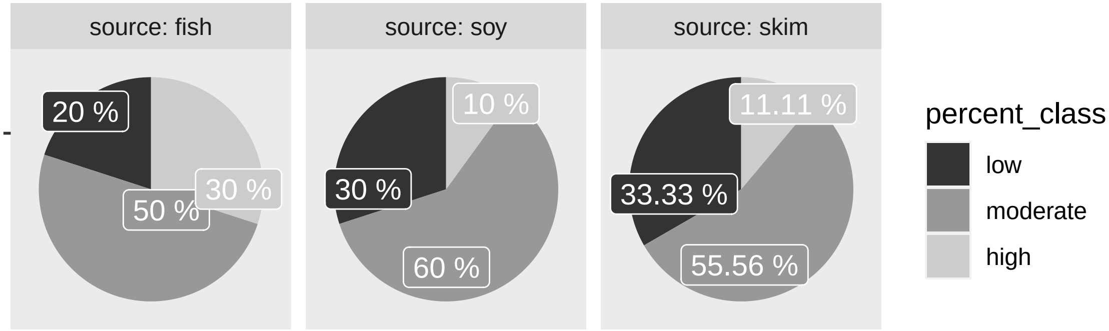


Figure 11.1

3.7.2 Solution 11.2

```
pigs.df %>%
  group_by(percent_class) %>%
  summarize(count = n(), mean = mean(conc), sd = sd(conc))
```

```
## # A tibble: 3 x 4
##   percent_class count  mean    sd
##   <fct>         <int> <dbl> <dbl>
## 1 low               8  32.7  5.74
## 2 moderate         16  38.9  7.81
## 3 high              5  39.9 12.1
```

3.7.3 Solution 11.3

```
q11_3.model <- aov(conc ~ percent_class, data = pigs.df)
```

3.7.4 Solution 11.4

```
summary(q11_3.model)
```

```
##                 Df  Sum Sq Mean Sq F value Pr(>F)
## percent_class  2   246.8   123.4   1.858  0.176
## Residuals      26 1726.4    66.4
```

3.7.5 Solution 11.5

```
q11_5.f <- summary(q11_3.model)[[1]][["F value"]][[1]]
q11_5.f
```

```
## [1] 1.858447
```

3.7.6 Solution 11.6

```
ggplot(q11_3.model, aes(sample = residuals(q11_3.model))) +
  stat_qq() +
  stat_qq_line() +
  labs(x = "Theoretical Quantiles", y = "Sample Quantiles")
```

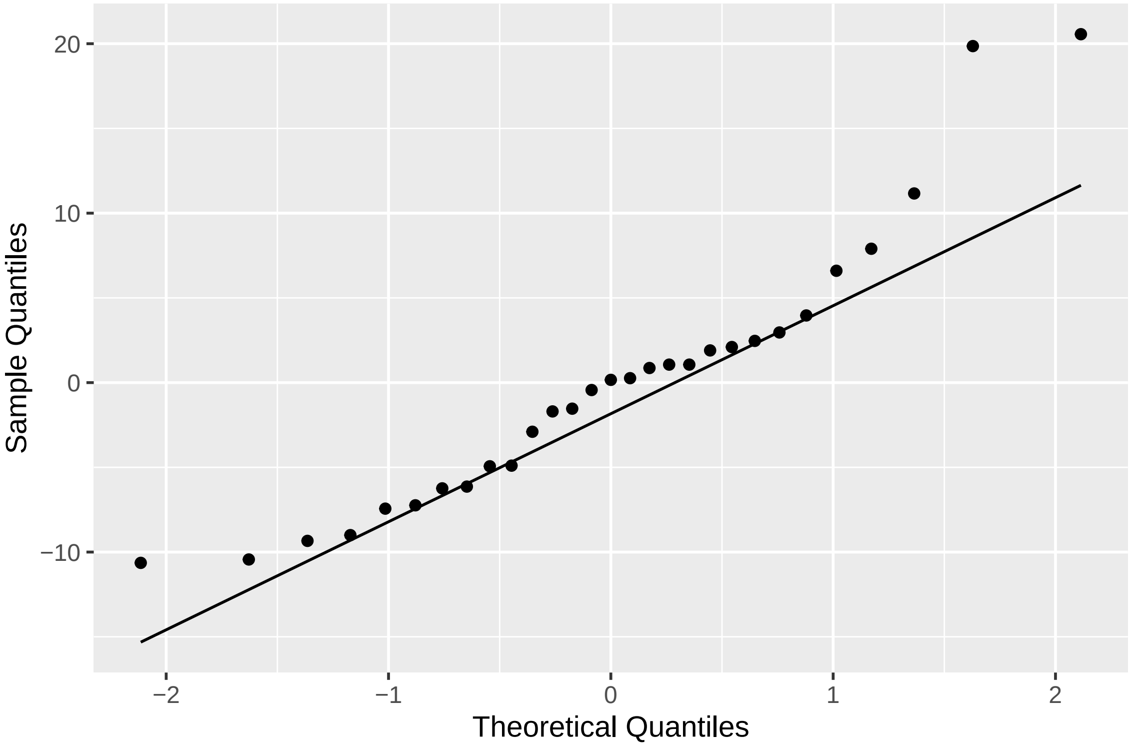


Figure 11.2

```
ggplot(q11_3.model, aes(x = residuals(q11_3.model))) +
  geom_histogram(bins = 10, color = "black", fill = "lightblue") +
  labs(x = "Residuals", y = "Frequency")
```

/newpage # 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"
```

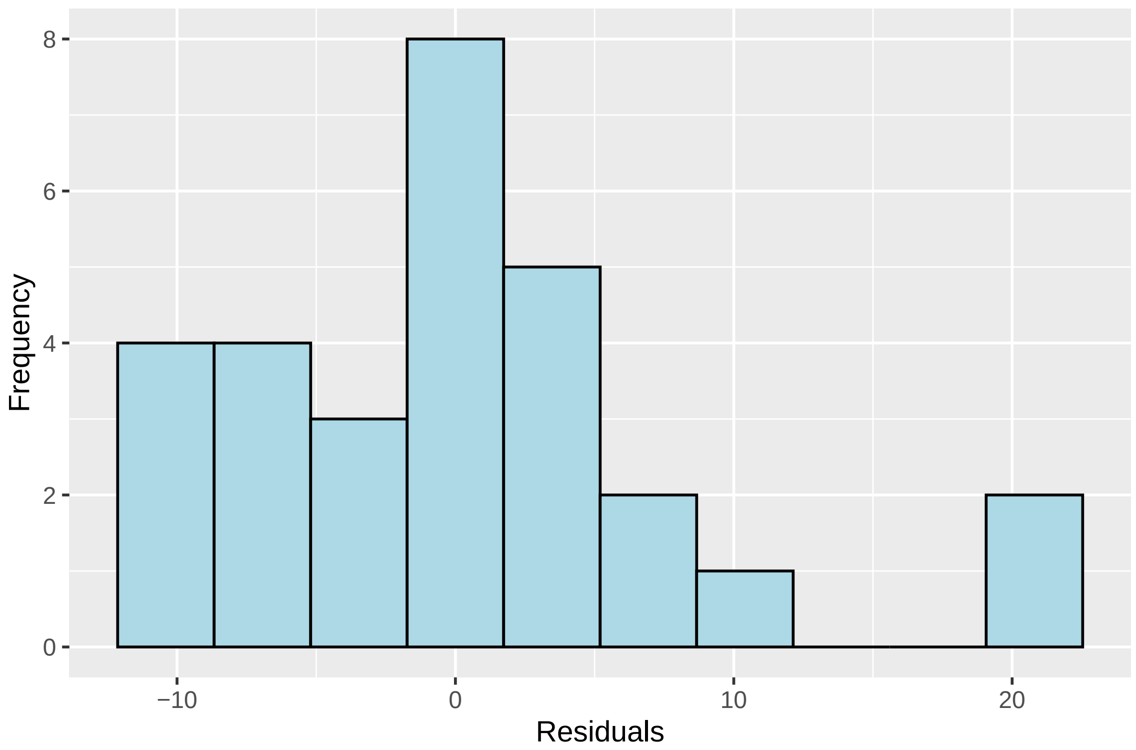


Figure 11.3

3.8 Question 12

In this question we use the fish dataset WITH the missing values.

1. Produce a frequency table for the number fish for each species.
2. Observation 14 has a missing value in variable Weight. Remove this observation from the data and create a new dataset, fish2. Use the new dataset to create a bar plot for the weight by species as shown in Figure 12.1.
3. For the new dataset created in Q12.2, produce Figure 12.2, a scatter plot for Width vs. Weight by Species.

3.8.1 Solution 12.1

```
q12_1.df <- fish %>%
  group_by(Species) %>%
  summarize(n = n())
q12_1.df
```

```
## # A tibble: 7 x 2
##   Species     n
##     <int> <int>
## 1       1    35
## 2       2     6
## 3       3    20
## 4       4    11
## 5       5    14
## 6       6    17
## 7       7    56
```

3.8.2 Solution 12.2

```r
fish2 <- fish[-14, ] # Or better fish2 <- fish[!is.na(fish$Weight), ]
ggplot(fish2, aes(x = Species, y = Weight, fill = as.factor(Species))) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("lightblue", "orange", "green", "blue", "red", "yellow", "pink")) +
  labs(x = "Species", y = "Weight")
```

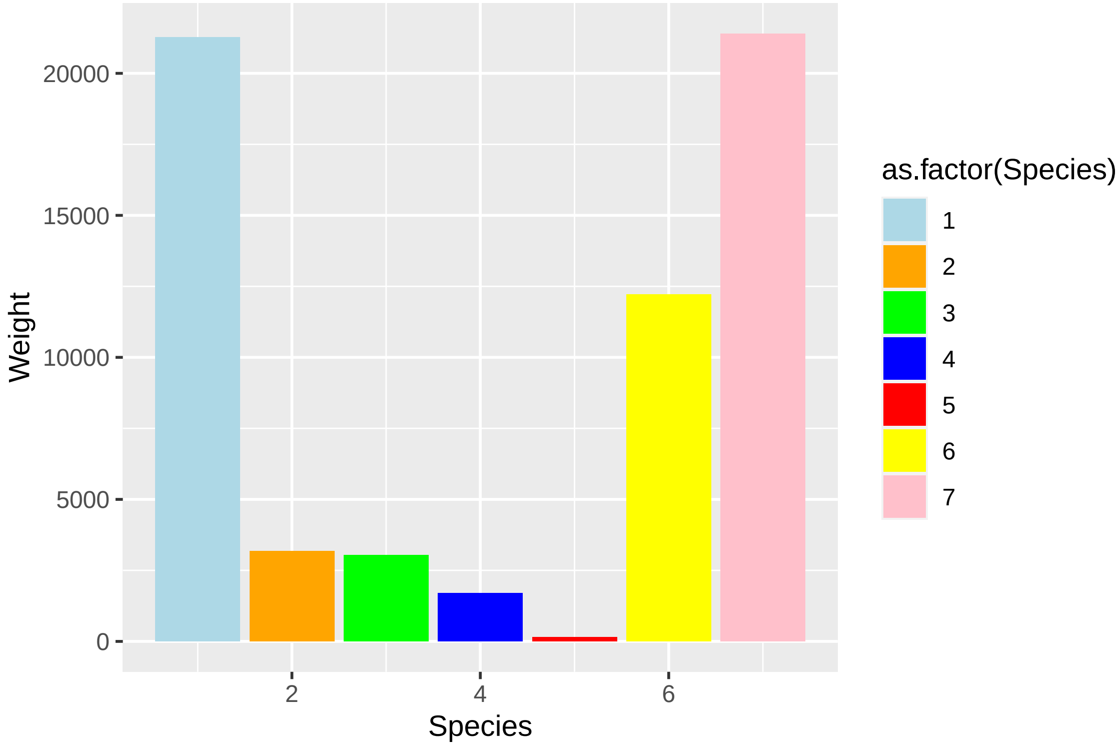


Figure 12.1

3.8.3 Solution 12.3

```r
ggplot(fish2, aes(x = Weight, y = Width)) +
  geom_point(size = 1) +
  facet_wrap(~Species, ncol = 3, scales = "free")
```

3.9 Question 13

For the new dataset defined in Q12.2.

1. Use a for loop to calculate the correlation between Weight and Width for each species. This implies that for each step in the for loop another species will be selected and the correlation between Weight and Width will be calculated and printed.
2. Produce the following output WITHOUT using a for loop. Note that the variable Correlation is the correlation between Weight and Width.

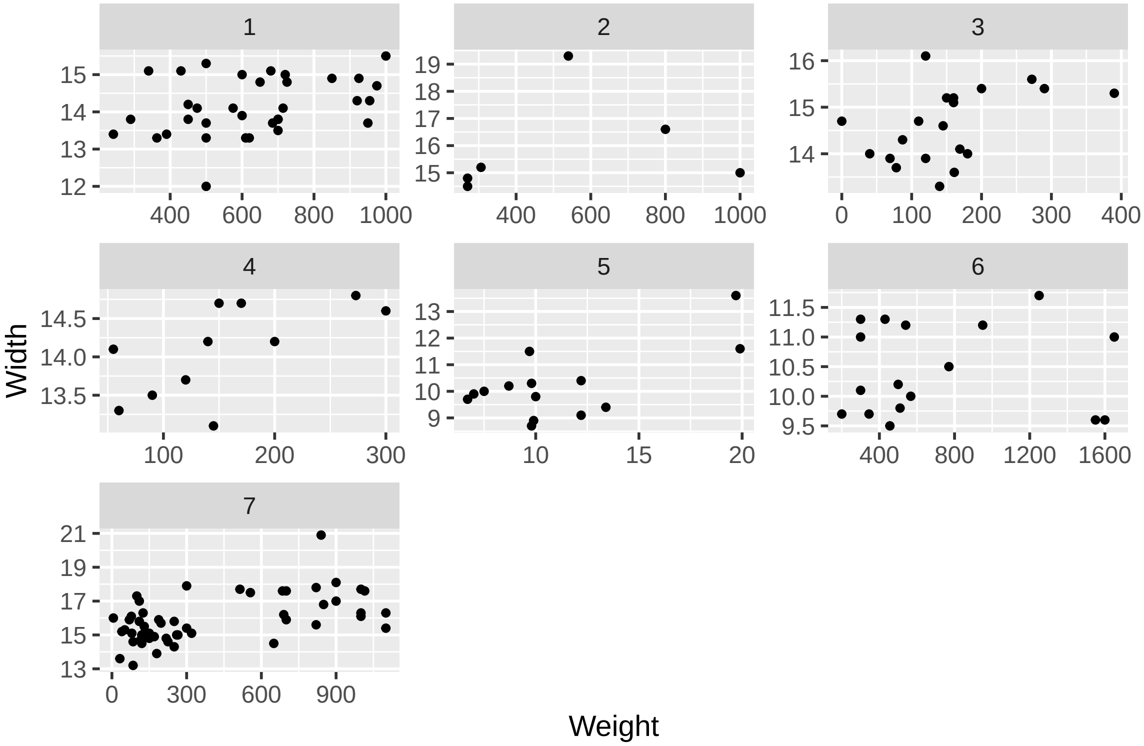


Figure 12.2

3.9.1 Solution 13.1

```
# ini
q13.1.vector <- data.frame(Species = character(), Correlation = numeric())
for (i in unique(fish2$Species)) {
  # append a new record to the dataset
  cor.i <- cor(fish2$Weight[fish2$Species == i], fish2$Width[fish2$Species == i])
  r.i <- data.frame(Species = i, Correlation = cor.i)
  q13.1.vector <- rbind(q13.1.vector, r.i)
}
q13.1.vector
```

```
##   Species Correlation
## 1       1  0.34444076
## 2       2  0.21565048
## 3       3  0.44890097
## 4       4  0.63814822
## 5       5  0.64815801
## 6       6  0.04297192
## 7       7  0.56262710
```

3.9.2 Solution 13.2

```
q13.2.vector <- fish2 %>%
  group_by(Species) %>%
  summarize(Correlation = cor(Weight, Width)) %>%
  mutate(Correlation = round(Correlation, 3))
q13.2.vector
```

```
## # A tibble: 7 x 2
##   Species Correlation
##     <int>       <dbl>
## 1       1       0.344
## 2       2       0.216
## 3       3       0.449
## 4       4       0.638
## 5       5       0.648
## 6       6       0.043
## 7       7       0.563
```

3.10 Question 14

1. For the dataset created in Q12.2 calculate the mean for the variables Weight, Length1 and Height by species and produce the data frame below.
2. Save the table that you produce in Q14.1 as an excel file and add this excel file to the zip file with your solutions.

3.10.1 Solution 14.1

```
q14.1.df <- fish2 %>%
  group_by(Species) %>%
  summarize(
    avg_w = round(mean(Weight), 0),
    avg_L1 = round(mean(Length1), 1),
    avg_h = round(mean(Height), 1)
  )
q14.1.df
```

```
## # A tibble: 7 x 4
##   Species avg_w avg_L1 avg_h
##     <int> <dbl>  <dbl> <dbl>
## 1       1   626   30.3  39.6
## 2       2   531   28.8  29.2
## 3       3   152   20.6  26.7
## 4       4   155   18.7  39.3
## 5       5    11   11.3  16.9
## 6       6   719   42.5  15.8
## 7       7   382   25.7  26.3
```

3.10.2 Solution 14.2

```
write.xlsx(q14.1.df, "q14.1.xlsx")
```

/newpage # Part 6: the msleep data

In this part we use the data msleep which is a part of the ggplot2 R package. To access the data you need to install the package. More information can be found in https://github.com/tidyverse/ggplot2/blob/main/data-raw/msleep.csv. You can use the code below to access the data.

```
library(dplyr)
data("msleep", package = "ggplot2")
head(msleep, 5)
```

```
## # A tibble: 5 x 11
##   name    genus vore  order conservation sleep_total sleep_rem sleep_cycle awake
##   <chr>   <chr> <chr> <chr> <chr>              <dbl>     <dbl>       <dbl> <dbl>
## 1 Cheetah Acin~ carni Carn~ lc                  12.1      NA        NA      11.9
## 2 Owl mo~ Aotus omni  Prim~ <NA>                17         1.8      NA       7
## 3 Mounta~ Aplo~ herbi Rode~ nt                  14.4       2.4      NA       9.6
## 4 Greate~ Blar~ omni  Sori~ lc                  14.9       2.3       0.133   9.1
## 5 Cow     Bos   herbi Arti~ domesticated         4         0.7       0.667  20
## # i 2 more variables: brainwt <dbl>, bodywt <dbl>
```

3.11 Question 15

1. How many observations and variables are included in the data?
2. Create a summary table of average sleep time (the variable sleep_total) for each level of the variable order, sorted in descending order of average sleep time.

3.11.1 Solution 15.1

```
nrow(msleep)
```

```
## [1] 83
```

3.11.2 Solution 15.2

```
q15.2.df <- msleep %>%
  group_by(order) %>%
  summarize(avg_sleep = round(mean(sleep_total), 2)) %>%
  arrange(desc(avg_sleep))
q15.2.df
```

```
## # A tibble: 19 x 2
##    order           avg_sleep
##    <chr>               <dbl>
##  1 Chiroptera          19.8
##  2 Didelphimorphia     18.7
##  3 Cingulata           17.8
##  4 Afrosoricida        15.6
##  5 Pilosa              14.4
##  6 Rodentia            12.5
##  7 Diprotodontia       12.4
##  8 Soricomorpha        11.1
##  9 Primates            10.5
## 10 Erinaceomorpha      10.2
## 11 Carnivora           10.1
## 12 Scandentia           8.9
## 13 Monotremata          8.6
## 14 Lagomorpha           8.4
## 15 Hyracoidea           5.67
## 16 Artiodactyla         4.52
## 17 Cetacea              4.5
## 18 Proboscidea           3.6
## 19 Perissodactyla        3.47
```

3.12 Question 16

1. For the msleep dataset, produce Figure 16.1 to visualize the relationship between the bodywt and brainwt variables. Add a regression line to the plot. As shown in Figure 16.1, take key note on the color.
2. Identify the outlying observations for which the body weight (the variable bodywt) is higher than 2000. Add the value of the body weight to the figure (inside the frame) as shown in Figure 16.2.
3. Create a new data frame without the two outlying observations. Produce the scatterplot shown in Figure 16.3.

3.12.1 Solution 16.1

```
q16_1.df <- msleep %>%
  filter(!is.na(bodywt) & !is.na(brainwt)) %>%
  mutate(pred = lm(brainwt ~ bodywt)$fitted.values)

ggplot(q16_1.df, aes(x = bodywt, color = bodywt)) +
  geom_point(aes(y = brainwt)) +
  geom_line(aes(y = pred)) +
  labs(x = "Body Weight", y = "Brain weight") +
  labs(title = "Scatter Plot of Body Weight and Brain Weight", color="Body Weight") +
  theme_bw()
```

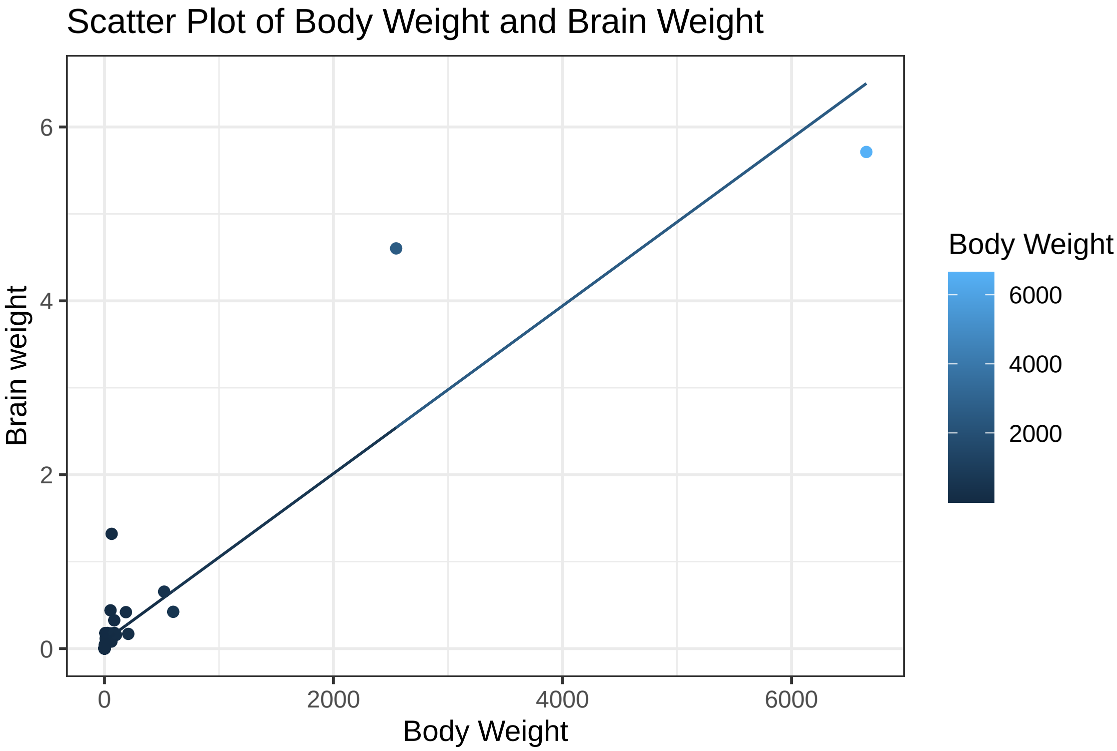


Figure 16.1

3.12.2 Solution 16.2

```
q16_2.df <- q16_1.df %>%
  filter(bodywt > 2000)

ggplot(q16_1.df, aes(x = bodywt, color = bodywt)) +
  geom_point(aes(y = brainwt)) +
  geom_line(aes(y = pred)) +
  labs(x = "Body weight", y = "Brain weight") +
  labs(title = "Scatter Plot of Body Weight and Brain Weight", color="Body Weight") +
  geom_text(data = q16_2.df, aes(x = bodywt, y=brainwt, label = bodywt), hjust = 0.5, vjust = -0.5) +
  theme_bw()
```

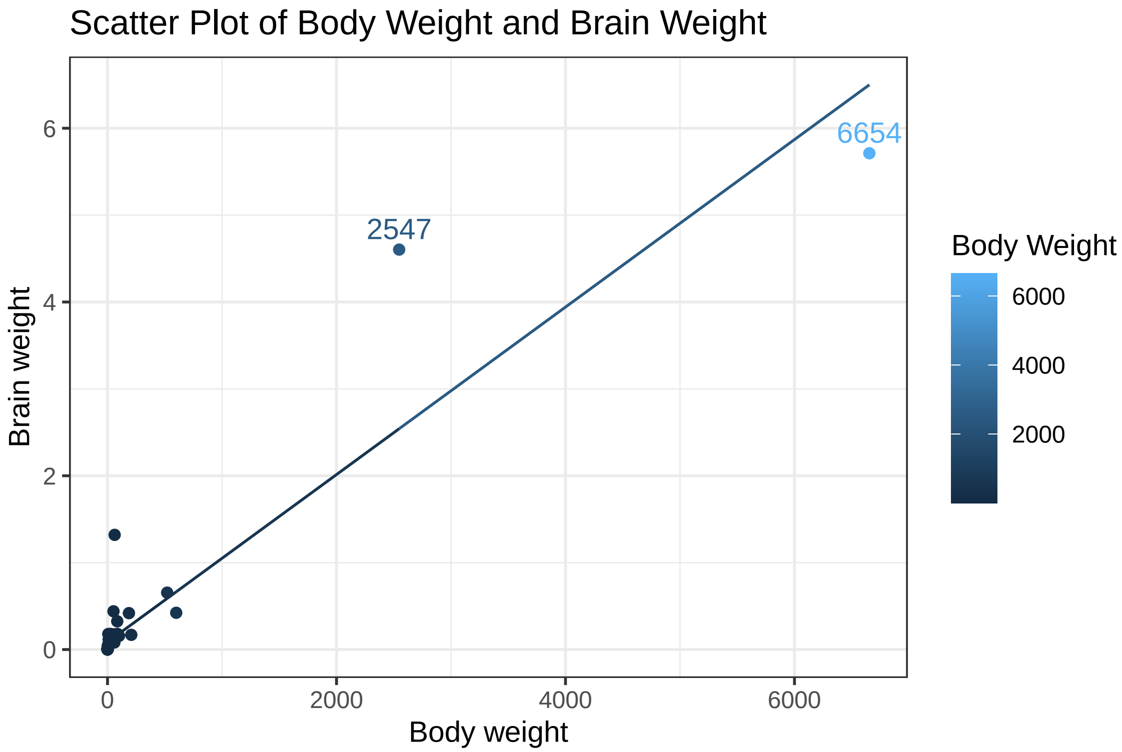


Figure 16.2

3.12.3 Solution 16.3

```
q16_3.df <- msleep %>%
  filter(!is.na(bodywt) & !is.na(brainwt)) %>%
  filter(bodywt < 2000) %>%
  mutate(pred = lm(brainwt ~ bodywt)$fitted.values)

ggplot(q16_3.df, aes(x = bodywt, )) +
  geom_point(aes(y = brainwt, color = bodywt)) +
  geom_smooth(aes(y = brainwt), method = "lm", se = T, formula = y ~ x, linewidth=0) +
  geom_line(aes(y = pred, color = bodywt)) +
  labs(x = "Body weight", y = "Brain weight") +
  labs(title = "Scatter Plot of Body Weight and Brain Weight", color="Body Weight") +
  xlim(0, 1000) +
  theme_bw()
```

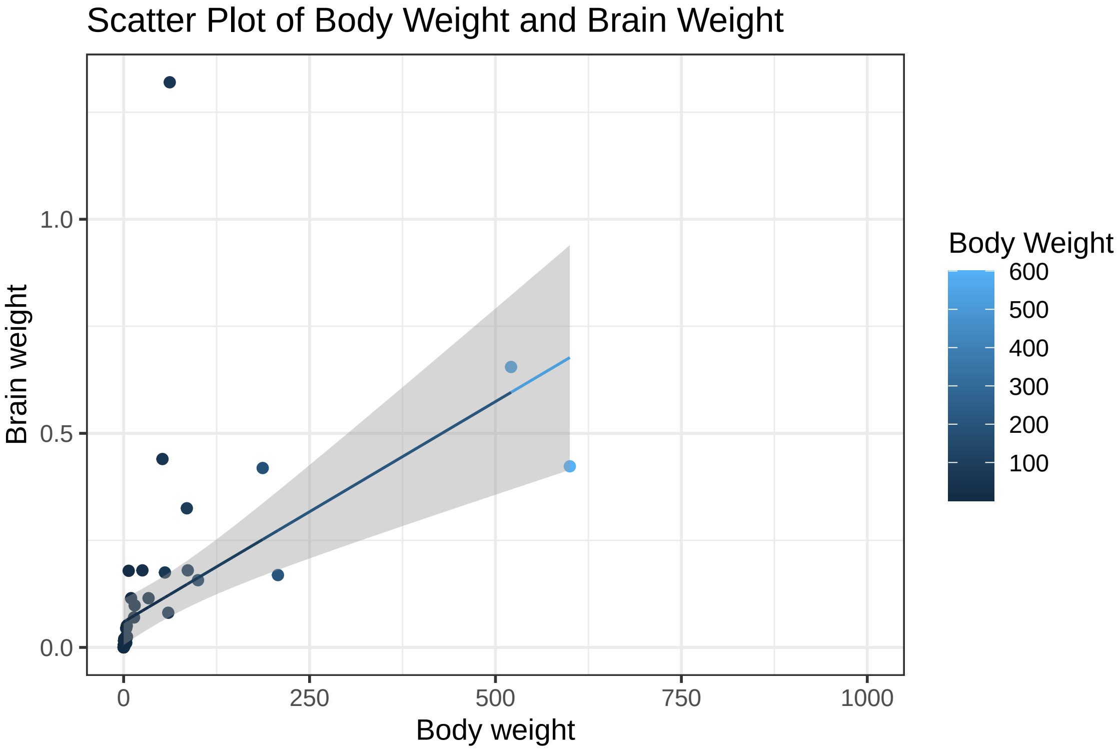


Figure 16.3

/newpage # Part 7: the ChickWeight data

In this part we use the ChickWeight data which is a part of the R datasets. To access the data you need to install the package. More information can be found in https://www.rdocumentation.org/packages/datasets/versions/3.6.2/topics/ChickWeight. You can use the code below to access the data.

```
data(ChickWeight)
head(ChickWeight)
```

```
## Grouped Data: weight ~ Time | Chick
##   weight Time Chick Diet
## 1     42    0     1    1
## 2     51    2     1    1
## 3     59    4     1    1
## 4     64    6     1    1
## 5     76    8     1    1
## 6     93   10     1    1
```

3.13 Question 17

1. Write a function that receives a dataset and a variable as an input and output returns the mean, median, and standard deviation of the variable rounded to 2 decimal places. Apply this function to the ChickWeight data and the variable weight.

2. In the output below, both numerical and graphical output were produced using the user function my.analysis(). The function receives as an input: (1) a dataset name, (2) the column number of variable

1 (a numerical variable) and (3) the column number of variable 2 (a factor). Note that both variable 1 and variable 2 are a part of the dataset. For the analysis in this question we use the ChickWeight dataset at time 0 (so only observations at time 0 are included). The output was produce using the following code: my.analysis(ChickWeight0,1,4).
The dataset ChickWeight0 contains the observations that were measured at time 0. Based on the output below, your task in the question is to write the function my.analysis() and to produce the output using the code above. Note that your function should produce an identical output.

3.13.1 Solution 17.1

```
q17_1.f <- function(df, variable) {
  mean <- round(mean(df[[variable]], na.rm = T), 2)
  median <- round(median(df[[variable]], na.rm = T), 2)
  sd <- round(sd(df[[variable]], na.rm = T), 2)
  return(c(mean, median, sd))
}
q17_1.f(ChickWeight, "weight")
```

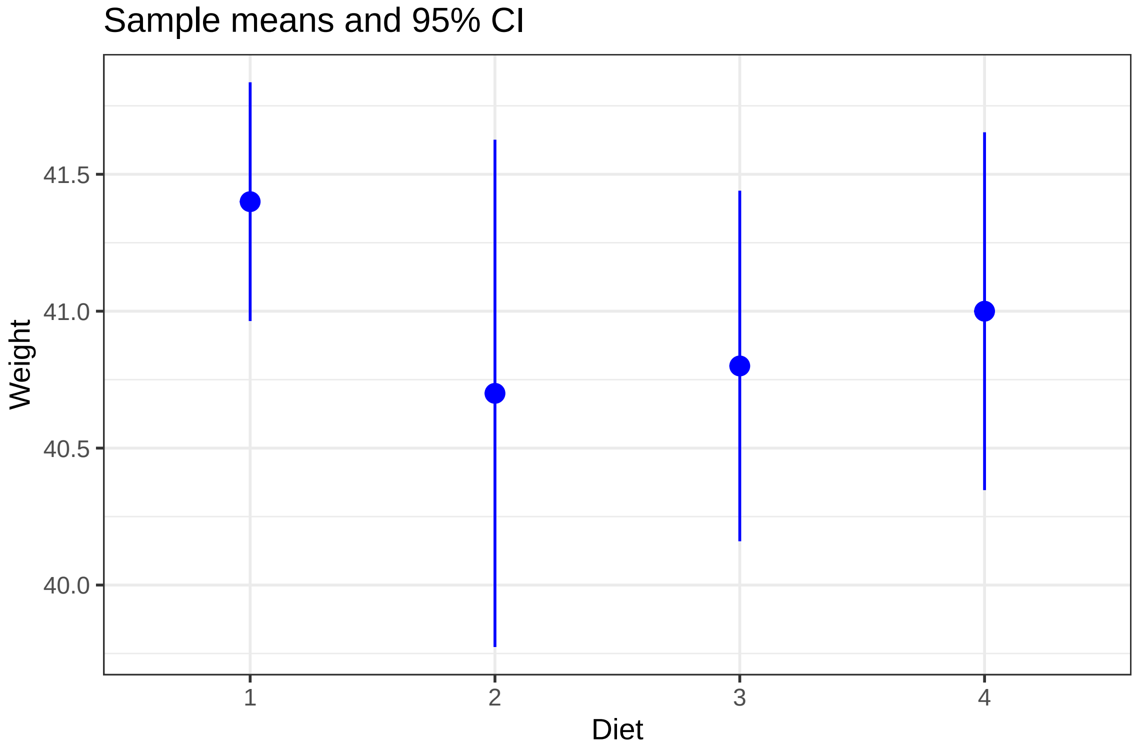
```
## [1] 121.82 103.00  71.07
```

3.13.2 Solution 17.2

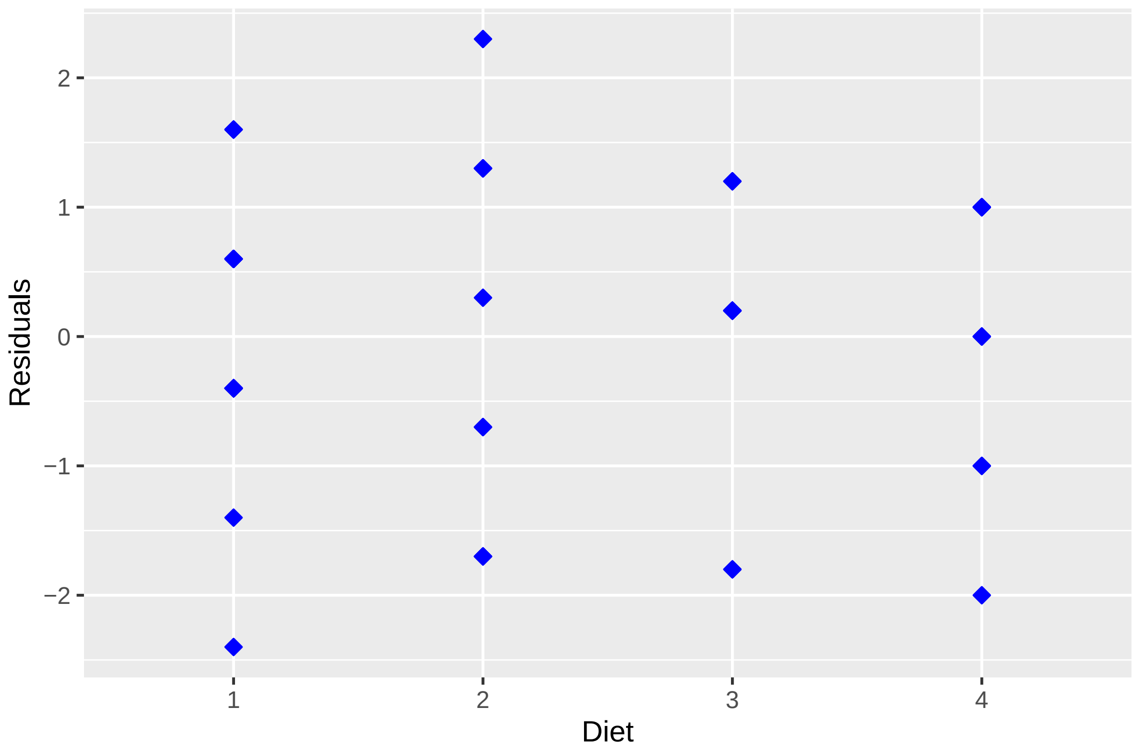
```
my.analysis <- function(df, variable1, variable2) {
  var1.name <- names(df)[variable1]
  var2.name <- names(df)[variable2]
  temp.df <- df %>%
    mutate_at(var2.name, as.factor) %>%
    mutate_at(var1.name, as.numeric) %>%
    group_by_at(var2.name) %>%
    summarise(mean = mean(.data[[var1.name]], na.rm = T),
              SD = sd(.data[[var1.name]], na.rm = T),
              n = n())
  print("Summary statistics")
  print(temp.df)
  var2 <- var2.name
  model <- aov(df[, var1.name] ~ df[, var2])
  print("ANOVA table")
  print(summary(model))
  print("Sample mean and 95% CI")
  temp.df <- temp.df %>%
    mutate(lb = mean - 1.96 * SD / sqrt(n),
           ub = mean + 1.96 * SD / sqrt(n))
  p <- ggplot(temp.df, aes(x = factor(Diet), y = mean)) +
    geom_point(color = "blue", size = 3) +
    geom_linerange(aes(ymin = lb, ymax = ub), color = "blue") +
    labs(y = "Weight", x = "Diet", title = "Sample means and 95% CI") +
    theme_bw()
  print(p)
  print("Plot of the Residuals")
  p <- ggplot() +
    geom_point(aes(x=model$model[, 2], y =residuals(model)), color = "blue", shape=23, fill="blue", siz
    labs(x=var2.name, y="Residuals")
  print(p)
}
```

```r
ChickWeight0 <- ChickWeight %>%
  filter(Time == 0)
my.analysis(ChickWeight0, 1, 4)
```

```
## [1] "Summary statistics"
## # A tibble: 4 x 4
##   Diet   mean    SD     n
##   <fct> <dbl> <dbl> <int>
## 1 1      41.4 0.995    20
## 2 2      40.7 1.49     10
## 3 3      40.8 1.03     10
## 4 4      41   1.05     10
## [1] "ANOVA table"
##             Df Sum Sq Mean Sq F value Pr(>F)
## df[, var2]   3   4.32   1.440   1.132  0.346
## Residuals   46  58.50   1.272
## [1] "Sample mean and 95% CI"
```



```
## [1] "Plot of the Residuals"
```



3.14 Question 18

1. Create the boxplot shown in Figure 18.1 to visualize the distribution of weights for each time point in the ChickWeight dataset. Color the boxplot based on the Time variable.
2. Create an **interactive** boxplot, shown in Figure 18.2, to visualize the distribution of weights for each time point in the ChickWeight dataset. Color the boxplot based on the Time variable. **DO NOT** include this figure in the PDF document for your answers but ONLY in the HTML document.

3.14.1 Solution 18.1

```
q18_1.plot <- ggplot(ChickWeight, aes(x = as.factor(Time), y = weight)) +
  geom_boxplot(aes(fill = factor(Time)), outlier.shape = NA) +
  labs(title = "Boxplot of Weight by Time") +
  theme_bw()
print(q18_1.plot)
```

3.14.2 Solution 18.2

/newpage # Part 8: the Quakes data

In this part we use the data quakes which is a part of the R datasets collection. Use help() to get more information about the data. More information can be found in https://www.rdocumentation.org/packages/datasets/versions/3.6.2/topics/quakes. You can use the code below to access the data.

```
library(datasets)
data("quakes")
head(quakes)
```

```
##      lat   long depth mag stations
```

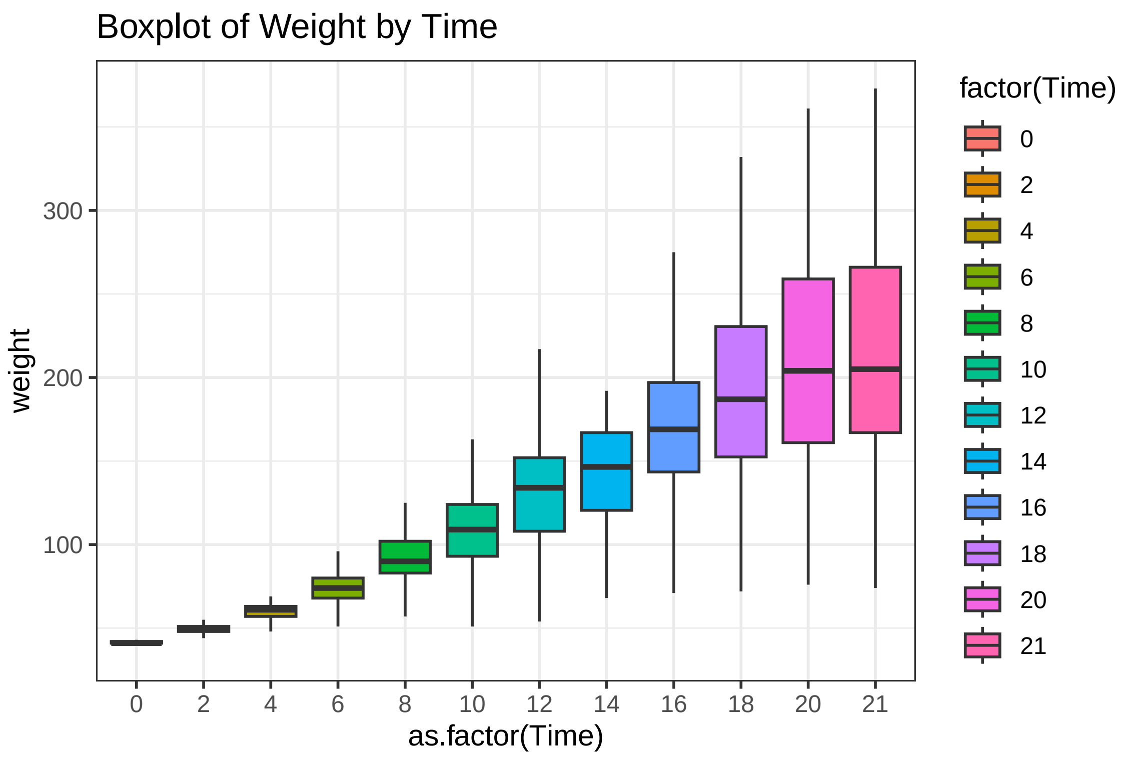


Figure 18.1

```
## 1 -20.42 181.62   562 4.8       41
## 2 -20.62 181.03   650 4.2       15
## 3 -26.00 184.10    42 5.4       43
## 4 -17.97 181.66   626 4.1       19
## 5 -20.42 181.96   649 4.0       11
## 6 -19.68 184.31   195 4.0       12
```

3.15 Question 19

1. Create the 3D scatter plot presented in Figure 19.1 to illustrate the relationship between latitude (lat), longitude (long), and depth (depth) of earthquakes in the quakes dataset.
2. Create an interactive 3D scatter plot, shown in Figure 19.2, to illustrate the relationship between latitude (`lat`), longitude (`long`), and depth (`depth`) of earthquakes in the `quakes` dataset. **DO NOT** include this figure in the PDF document for your answers but ONLY in the HTML document.

3.15.1 Solution 19.1

```
scatter3D(quakes$long, quakes$lat, quakes$depth, colvar = quakes$mag, pch = 16, ticktype = "detailed",
          col = viridis(n=100), size = 0.5, bty = "u", col.axis = "gray", col.grid = "gray",
          xlab = "\nX", ylab = "\nY", zlab = "\nZ", cex.axis = 0.6)
```

3.15.2 Solution 19.2

3.16 Question 20

1. Calculate the mean Richter Magnitude (the variable mag) by the station (the variable stations).

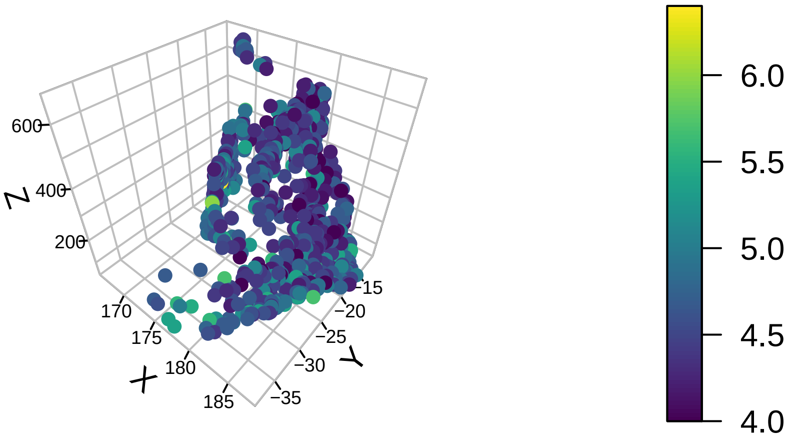


Figure 19.1

2. Create a new dataset which contains the observations from the top 25 stations with the highest Richter Magnitude. How many observations are included?
3. For the new data, define a new variable that is equal to the ratio between Richter Magnitude and the depth, i.e.,
$$ratio = \frac{Richter Magnitude}{depth}$$
.
Sort the data according to the variable ratio.
4. Print the three stations with the highest mean ratio.
5. Create a new dataset for the stations with ratio higher than 0.099. For the new data, produce Figure 20.1.
6. For the dataset created in Q20.2 and Q20.3, create a new categorical variable (mat_cat) that takes the value of 0 if the Richter Magnitude (the variable mag) is below the overall mean and 1 otherwise. Produce Figure 20.2.

3.16.1 Solution 20.1

```
quakes %>%
  group_by(stations) %>%
  summarise(mean_mag = mean(mag))
```

```
## # A tibble: 102 x 2
##    stations mean_mag
##       <int>    <dbl>
##  1       10     4.23
##  2       11     4.23
##  3       12     4.20
##  4       13     4.33
##  5       14     4.28
##  6       15     4.28
##  7       16     4.27
##  8       17     4.35
##  9       18     4.44
## 10       19     4.38
## # i 92 more rows
```

3.16.2 Solution 20.2

```r
highest_25 <- quakes %>%
  group_by(stations) %>%
  summarise(max_mag = max(mag)) %>%
  arrange(desc(max_mag)) %>%
  head(25)

q20_2.df <- quakes %>%
  filter(stations %in% highest_25$stations) %>%
  arrange(desc(mag))
nrow(q20_2.df)
```

```
## [1] 52
```

3.16.3 Solution 20.3

```r
q20_3.df <- q20_2.df %>%
  mutate(ratio = mag / depth) %>%
  arrange(desc(ratio))
```

3.16.4 Solution 20.4

```r
q20_4.df <- q20_3.df %>%
  group_by(stations) %>%
  summarise(mean_ratio = mean(ratio)) %>%
  arrange(desc(mean_ratio)) %>%
  head(3)

print(q20_4.df$stations)
```

```
## [1]  76  83 123
```

3.16.5 Solution 20.5

```r
q20_5.df <- q20_3.df %>%
  filter(ratio > 0.099)
ggplot(q20_5.df, aes(x = ratio, y = depth)) +
  geom_point() +
  labs(title = "Boxplot of Ratio by Station") +
  theme_bw()
```

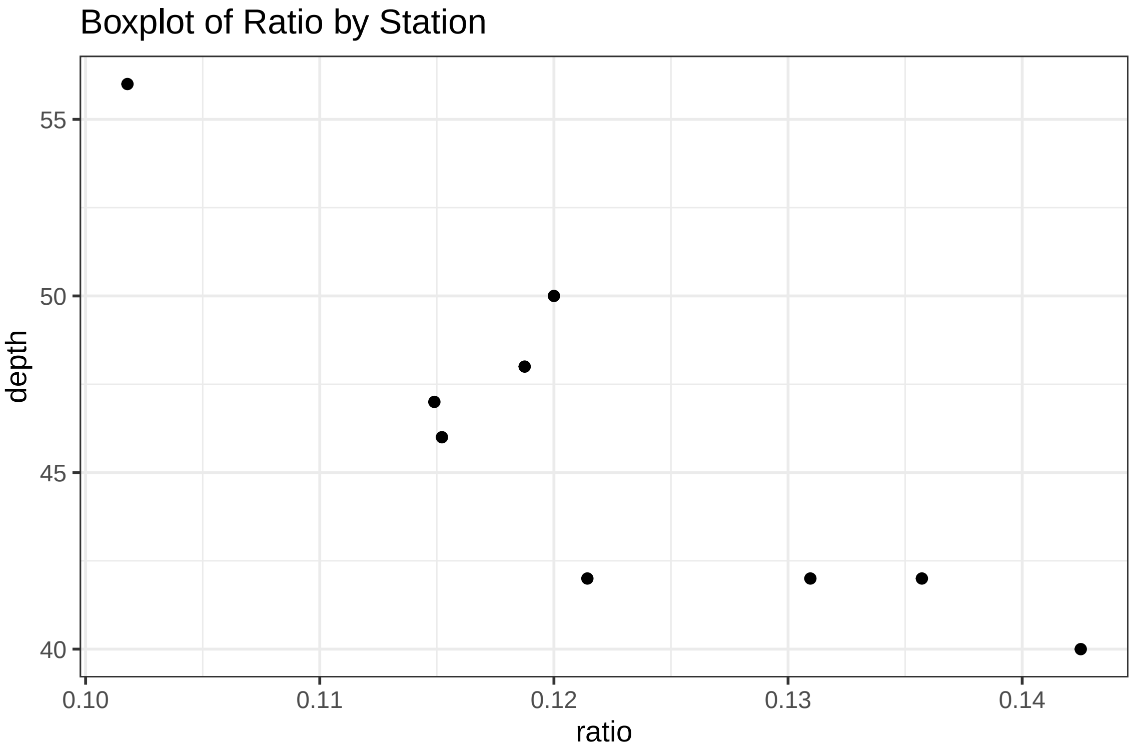


Figure 20.1

3.16.6 Solution 20.6

```
q20_6.df <- q20_3.df %>%
  mutate(mat_cat = ifelse(mag > mean(mag), 1, 0))
ggplot(q20_6.df, aes(x = ratio, y = depth, color = as.factor(stations))) +
  geom_point() +
  labs(title = "Boxplot of Ratio by Station") +
  facet_wrap(~mat_cat, labeller = label_both) +
  theme_bw()
```

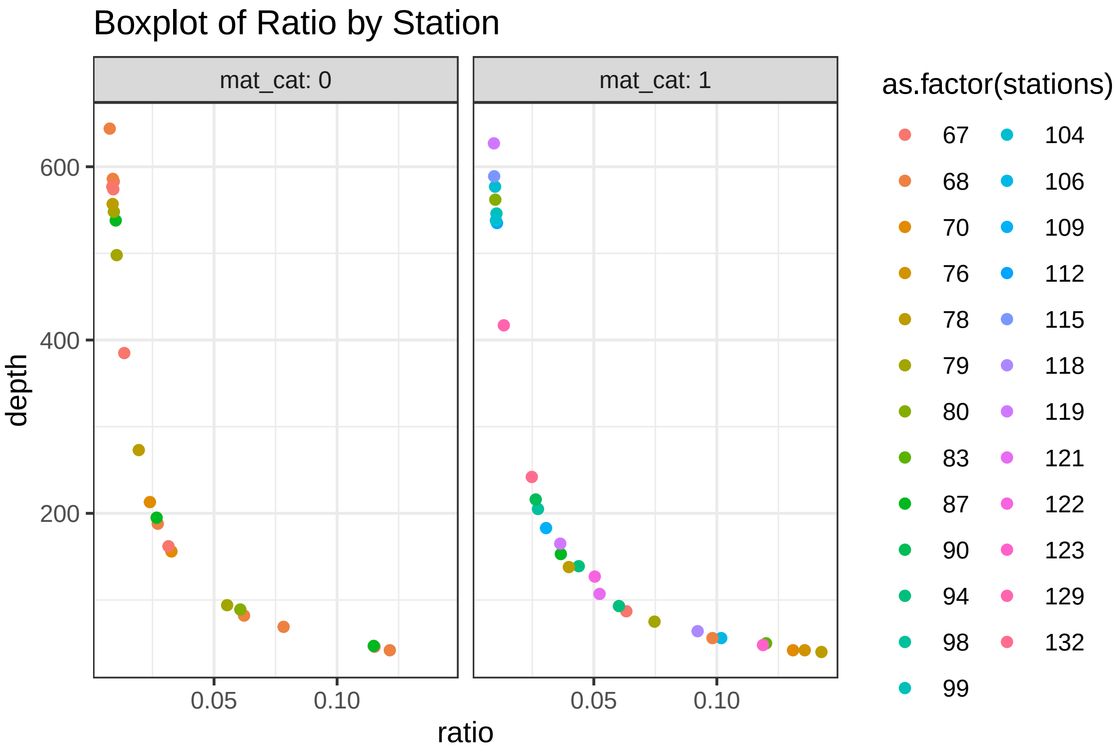

Boxplot of Ratio by Station
mat_cat: 0
mat_cat: 1
depth
600
400
200
ratio
0.05
0.10
0.05
0.10
as.factor(stations)
67
68
70
76
78
79
80
83
87
90
94
98
99
104
106
109
112
115
118
119
121
122
123
129
132

Figure 20.2