Introduction to Research Data Analysis

R Handout

Myo Minn Oo

2022-04-09

Contents

1 Introduction to R in RStudio

1.1 RStudio

Create a new project in RStudio

- Go to **File**
- Choose **New Project**
- Choose **New Directory** » **New Project**
- Type in **Directory Name**
- Choose **Directory** you want to save
- Click **Create Project**

Why do this?

- proper project management
- no directory set up required
- good practice

Let's practice!

1.2 Packages (modules)

- Many packages in R to add/use functionality of your interest
  - **tidyverse** » data management and processing
  - **magrittr** » facilitate R code workflow **%>%**
  - **readxl** » read excel files
  - **janitor** » clean variable names and tabulate data
  - **rmarkdown** » create documents and reports
  - **flextable** » create publication-ready tables
  - **flexDashboard** » for dashboard creation

```
packages_required <- c("tidyverse", "magrittr", "janitor", "readxl",
    "gtsummary", "flextable")
are_packages_installed <- packages_required %in% installed.packages()

if (!all(are_packages_installed)) {
    packages_required <- packages_required[!are_packages_installed]
    install.packages(packages_required)
}
```

1.3 Functions

- R is powerful because of functions.
- To use a function, 2 parts
  - name » to call the function
  - arguments » input + instructions
    - mandatory
    - optional

```
mean(x = input)
```

`x` is mandatory to feed into the function.

```
mean(x = input, na.rm = TRUE)
```

`na.rm` is optional, and used when you want to remove missing values from calcultion.

1.4 %>%

- pipe operator
- from `magrittr` package
- create workflows for writing R codes

```
## It pushes the output from left hand side as input to the right hand side.
left hand side %>% right hand side
```

Here is an example.

```
mtcars %>%
    summarize(avg_speed = mean(mpg))
```

```
##   avg_speed
## 1  20.09062
```

This code chunk works in two stages:

1. we push a dataset `mtcars` from the left hand side of `%>%` as input to the right hand side.
2. here, we use `summarize` function from `tidyverse` package. The argument is in the form of `variable_name = what you want to do`.

If you use R's default code, you will have to write as follow which gives the same result.

```
mean(mtcars$mpg)
```

```
## [1] 20.09062
```

Where does `mtcars` come from? `mtcars` is a built-in dataset that comes with R.

1.5 Use codes as template

- don't remember these codes by heart
- use codes that work as templates
- learn how to copy and paste codes

For example, we can replace `mpg` with other variables in `mtcars`.

```
mtcars %>%
    summarize(avg_weight = mean(wt))
```

```
##   avg_weight
## 1    3.21725
```

You can add more variables.

```
mtcars %>%
    summarize(avg_speed = mean(mpg),
              sd_speed = sd(mpg),
              avg_weight = mean(wt),
              sd_weight = sd(wt))
```

```
##   avg_speed sd_speed avg_weight sd_weight
## 1  20.09062 6.026948    3.21725 0.9784574
```

1.6 Help

So how do you know what to write?

Use `?function_name` to read its help page. But, it is mostly technical and hard to understand because nerds write them for nerds.

```
?mean
?sd
?`%>%`
?mtcars
```

1.7 Exercises

- use a function called `str` to display all variable names in `mtcars`.
  - how many variables and observations does `mtcars` have?
- use the remaining variables to summarize their means and standard deviations.

1.7.1 Answers

```
?str
str(mtcars)
```

2 SARS-COV-2 data - PNGIMR

The raw data `png_covid19_2021.xls` received in MS excel format is already processed and saved as `covid.RData`.

R scripts used for data management are stored under the folder `scripts`. if you want to examine the codes in detail, open `main.R` under `scripts` along with `00_setup.R` and `01_data_process.R`.

- `main.R` compiles the other two scripts.
- `00_setup.R` provides necessary setup to run all R scripts.
- `01_data_process.R` is the file where all data management processes happen.

2.1 Clean your workspace

Before you start a new session, use the following code to clean your workspace.

```
rm(list = ls())
```

2.2 Data import

The following codes show how to import excel files into R. We use `read_excel()` function from the `readxl` package.

```
covid <- readxl::read_excel("data/png_covid19_2021.xls")
```

For the purpose of this workshop, we will use `covid.RData` which was already created for you.

```
load("data/covid.RData")
```

2.3 Data dictionary

Name of the data: `covid_processed`

No	Variable Name	Description
1	rt_pcr_pos_neg	Result of RT-PCR
2	patient_age	Age in years
3	patient_sex	Sex of patient (Male or Female)
4	p_province	Province (EHP or Other)
5	symptom_status	Symptom Status (Yes or No)
6	case_contact	History of case contact
7	vaccine_status	Vaccination Status
8	dose_num	Number of vaccine doses
9	travel_hist	Travel History (Yes or No)
10	symp_number	Number of symptoms
11	time_onset_test	Time in days from onset of symptoms to a COVID-19 test

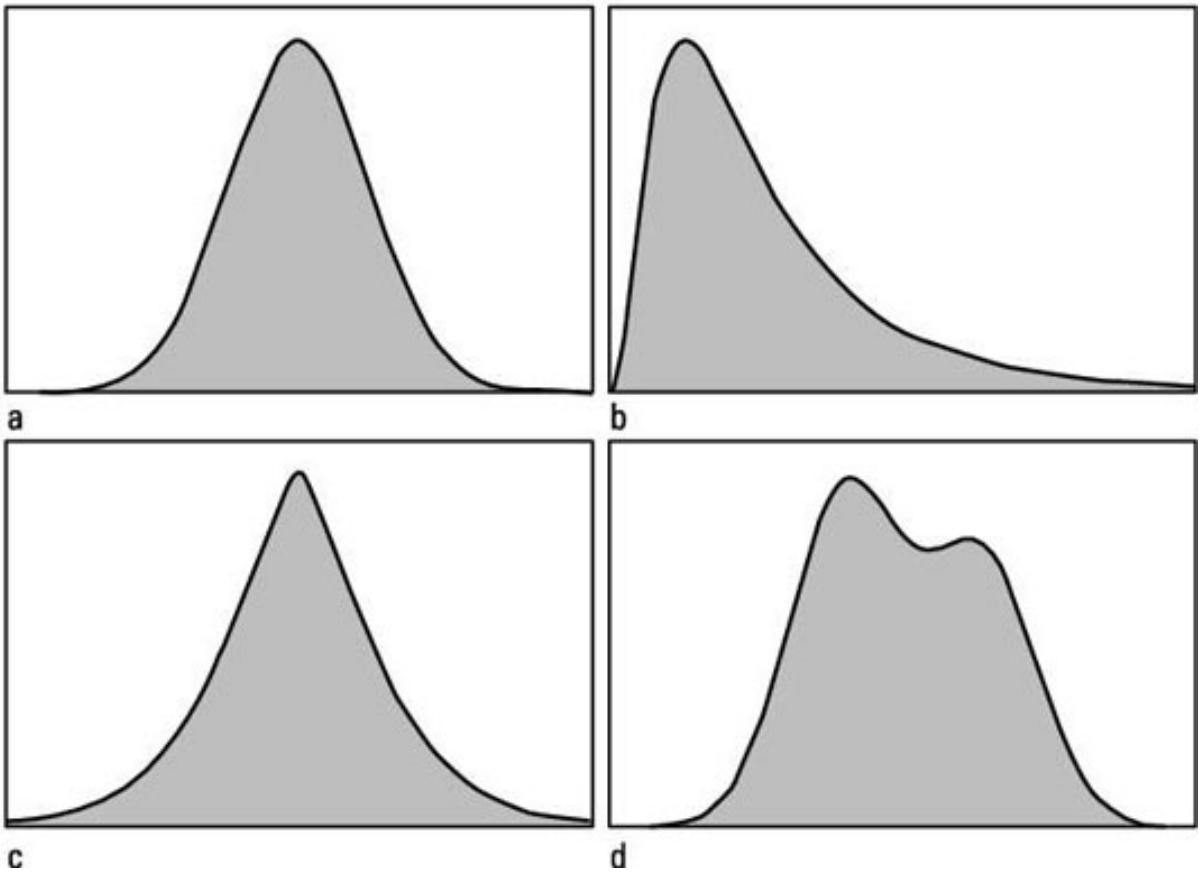
Now we are good to summarize our data!

3 Descriptive Statistics

3.1 Numerical summary

- mean and standard deviations
- median and interquartile range
- minimum and maximum

Distributions



a. normal distribution
b. right-skewed because tail is on the right side. If tail is on the left side, it's called left-skewed distribution.
c. normal distribution with a narrow peak
d. bimodal distribution

If your data follows normal distribution, use mean and standard deviation. Otherwise, use median and interquartile range.

3.1.1 Exercises

- Summarize `patient_age` and `time_onset_test`.
- Which numerical summary measures should we use for `time_onset_test`.
- In your free time, try `dose_num` and `symp_number`.

Tips: use an optional argument `na.rm = TRUE` because some variables contains missing values.

3.1.2 Answers

```
## import data if not done yet.
load("data/covid.RData")

## load packages
library(magrittr)
library(tidyverse)

## summarizing covid-19 data
covid_processed %>%
    summarise(mean_age = mean(patient_age, na.rm = TRUE),
              sd_age = sd(patient_age, na.rm = TRUE),
              mean_time = mean(time_onset_test, na.rm = TRUE),
              sd_time = sd(time_onset_test, na.rm = TRUE))
```

```
## # A tibble: 1 x 4
##   mean_age sd_age mean_time sd_time
##      <dbl>  <dbl>     <dbl>   <dbl>
## 1     33.1   13.7      7.97    29.3
```

- For `patient_age`, sd value is less than mean value. It seems like a normal distributioh.
- For `time_onset_test`, sd value is greater than mean value, suggesting a skewed distribution. We must use median and interquartile range for a robust summary measure.

```
## summarizing covid-19 data
covid_processed %>%
    summarise(mean_time = mean(time_onset_test, na.rm = TRUE),
              sd_time = sd(time_onset_test, na.rm = TRUE),
              median_time = median(time_onset_test, na.rm = TRUE),
              q1_time = quantile(time_onset_test, probs = 0.25, na.rm = TRUE),
              q3_time = quantile(time_onset_test, probs = 0.75, na.rm = TRUE))
```

```
## # A tibble: 1 x 5
##   mean_time sd_time median_time q1_time q3_time
##       <dbl>   <dbl>       <dbl>   <dbl>   <dbl>
## 1      7.97    29.3           4       2       9
```

As you can see, mean value is quite far right from the median value. This is a right-skewed distribution.

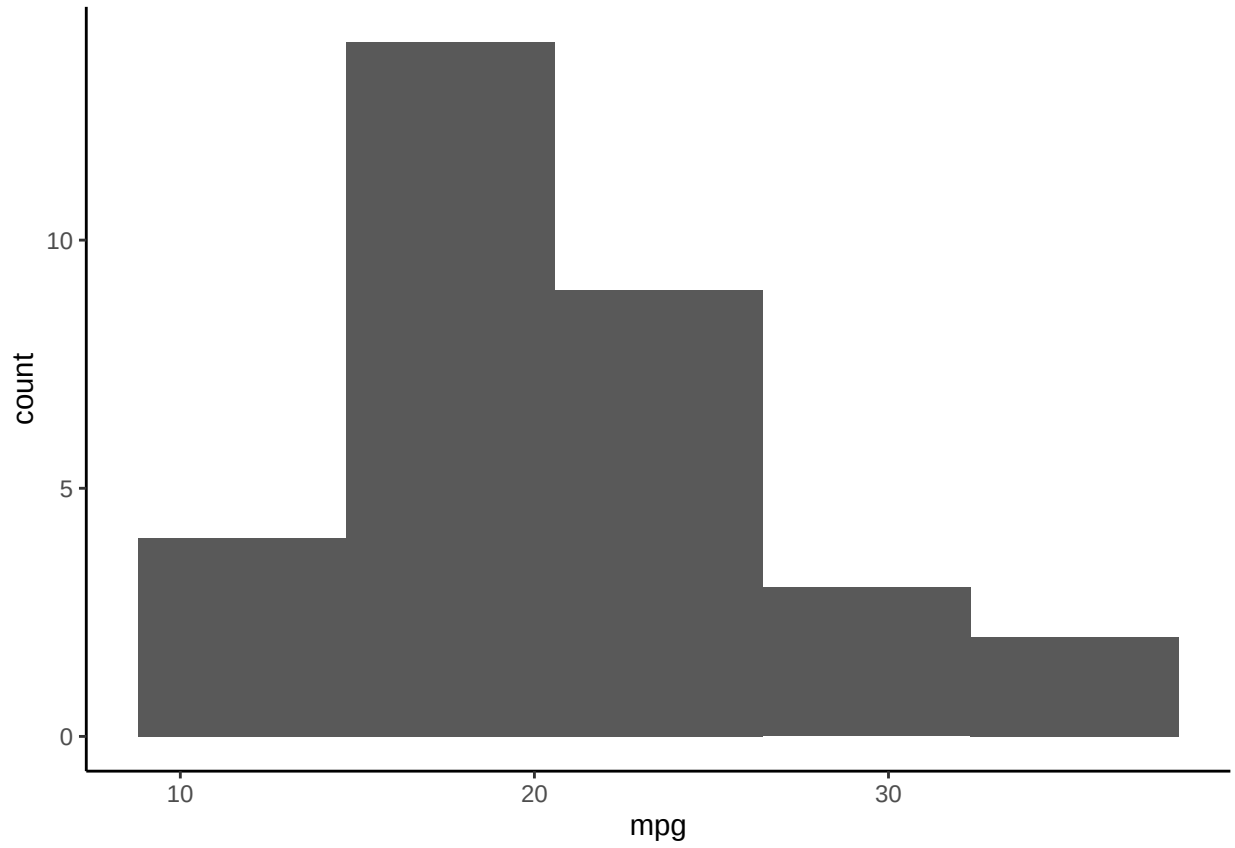
3.2 Visualization of numerical data

- histogram
- boxplot
- density plot
- dot plot
- steam and leaf plot

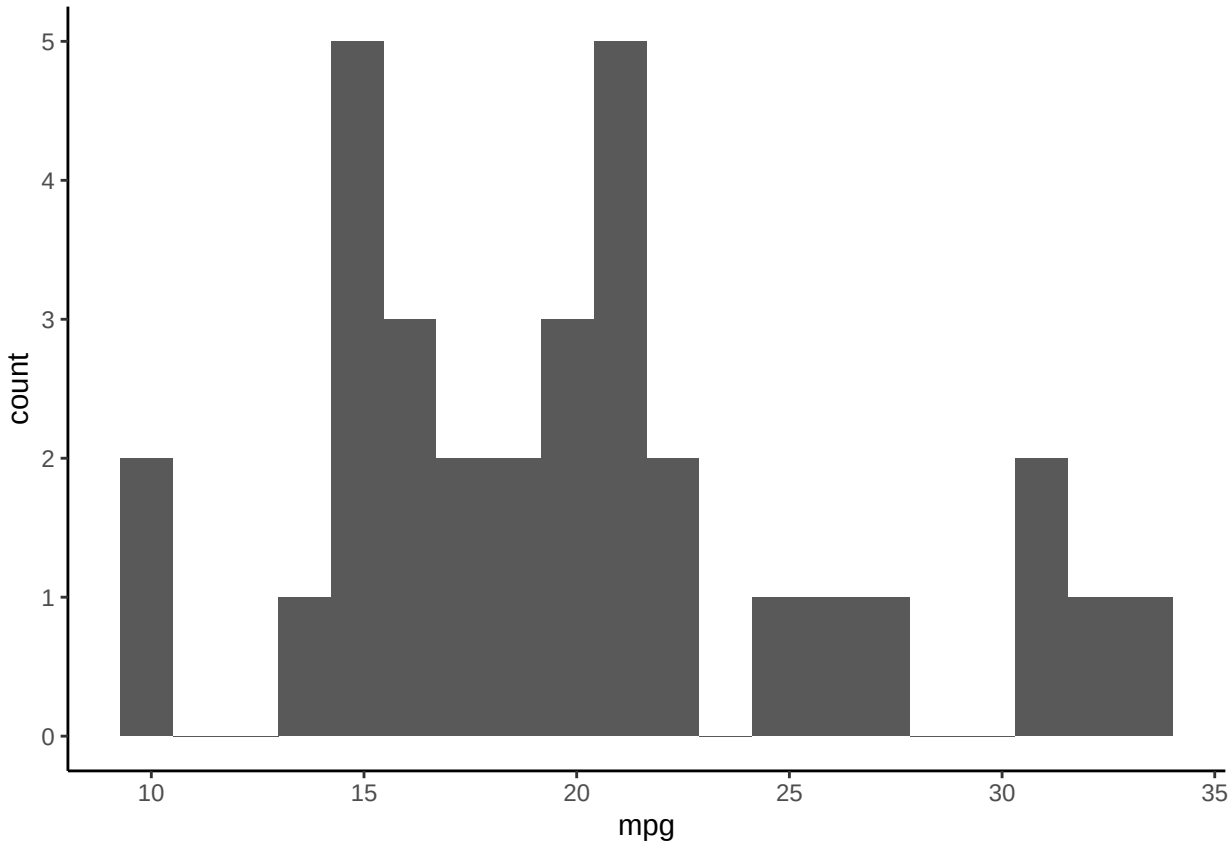
3.2.1 Histogram

- Histograms are barplots without gaps between bars.
- bin width is important to shape the distribution.

```
mtcars %>%
    ggplot(aes(mpg)) +
    geom_histogram(bins = 5) +
    theme_classic()
```



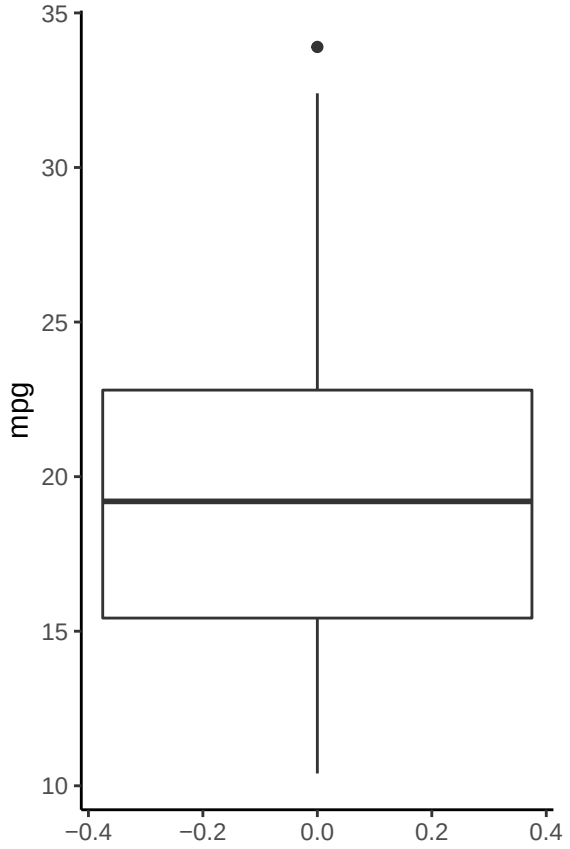

```
mtcars %>%
    ggplot(aes(mpg)) +
    geom_histogram(bins = 20) +
    theme_classic()
```



3.2.2 Boxplot

Boxplot shows median, interquartile range, lower and upper whiskers (limits), minimum and maximum values.

```
mtcars %>%
    ggplot(aes(mpg)) +
    geom_boxplot() +
    coord_flip() +
    theme_classic()
```

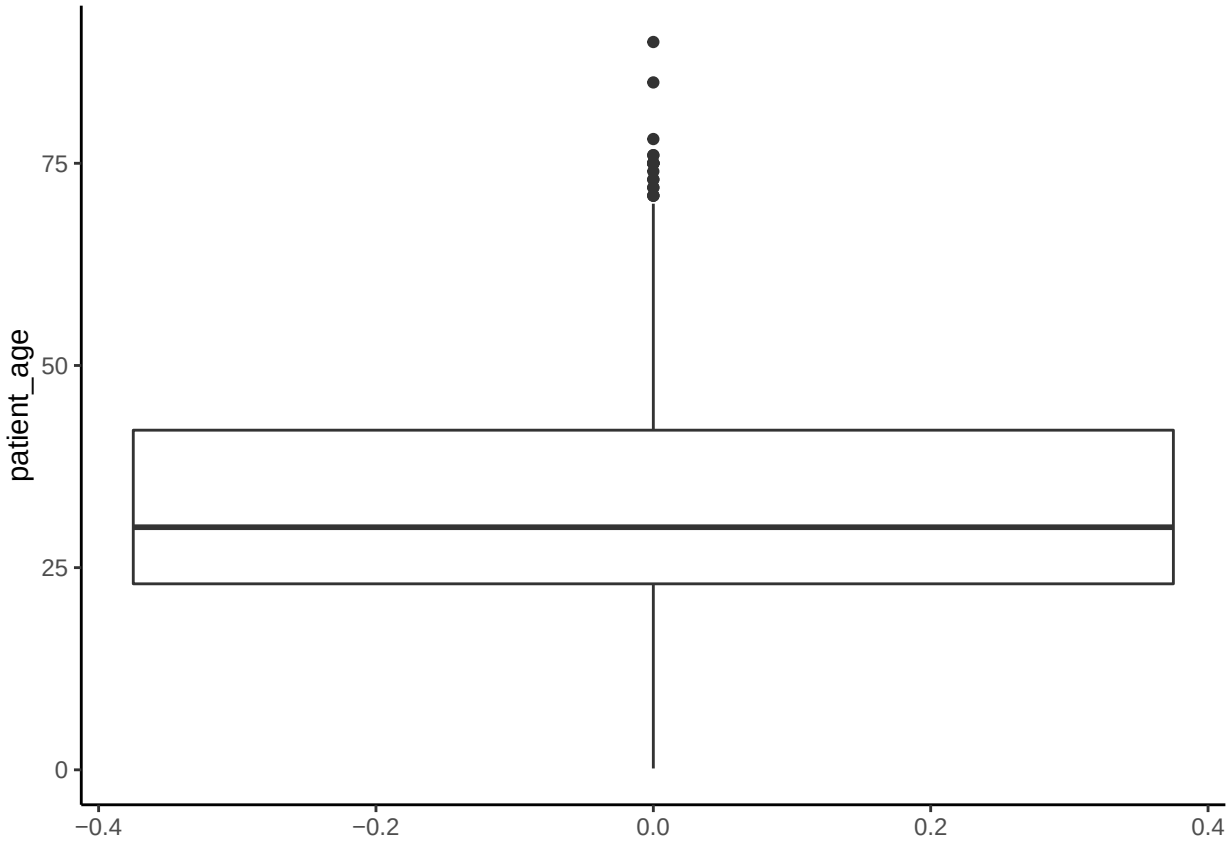


3.2.3 Exercises

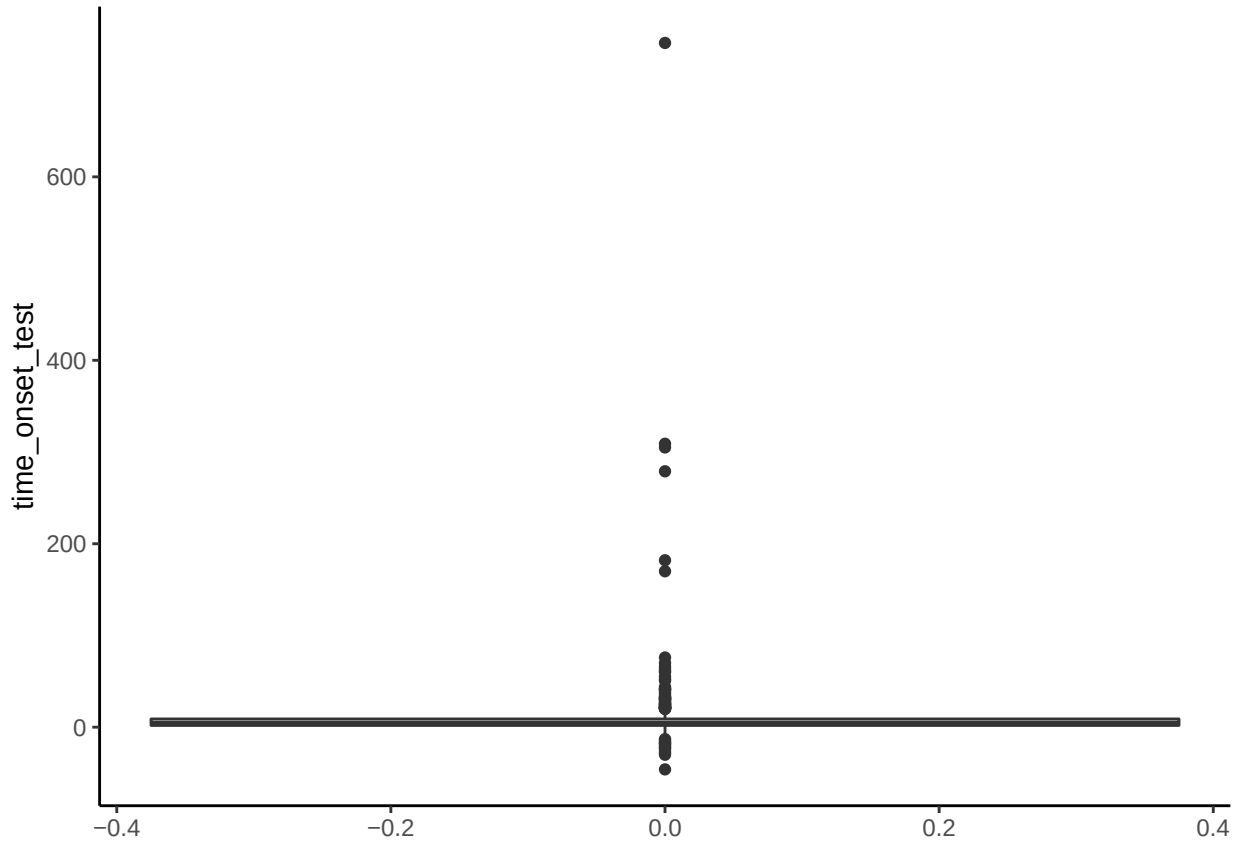
- Create histograms for `patient_age` and `time_onset_test`.
- What do you notice with histogram for `time_onset_test`?
- How would you deal with outliers?

3.2.4 Answers

```
covid_processed %>%
    ggplot(aes(patient_age)) +
    geom_boxplot() +
    coord_flip() +
    theme_classic()
```

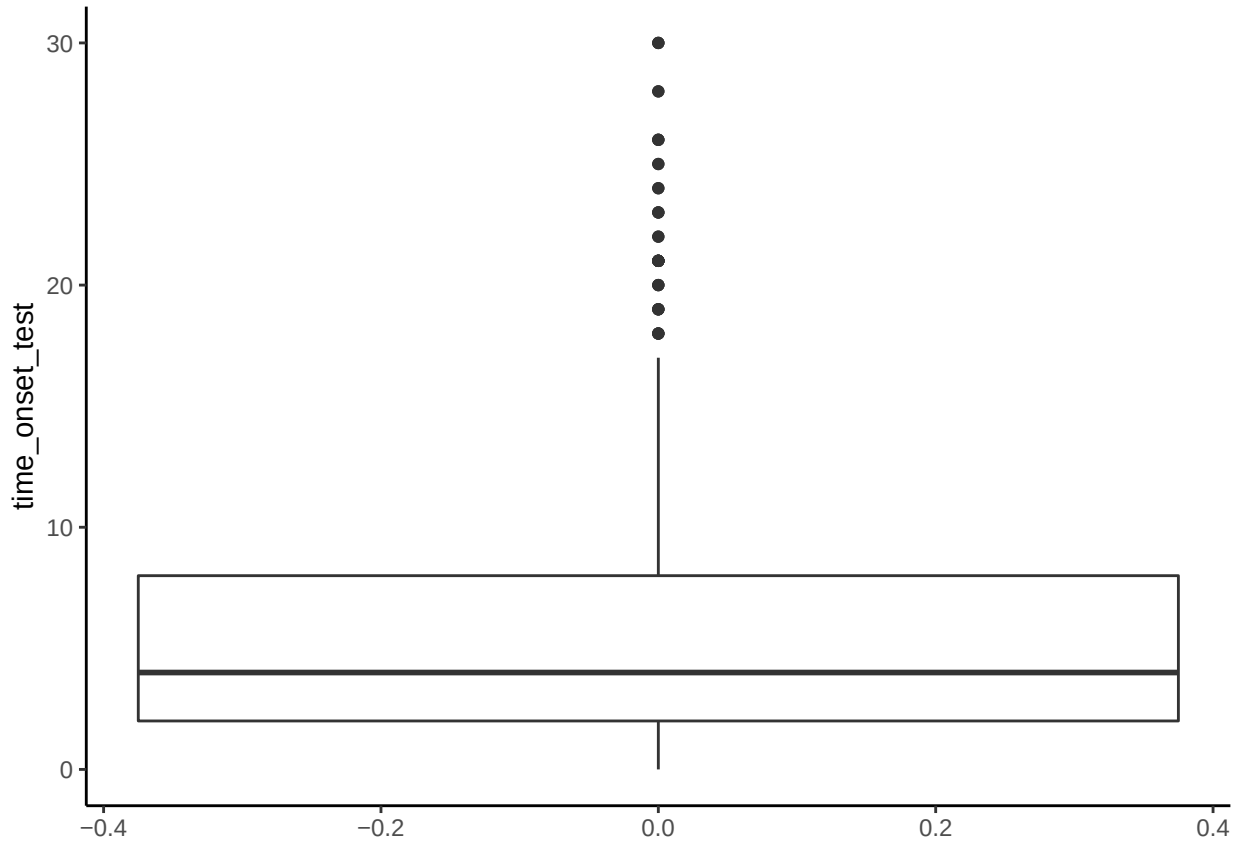



```
covid_processed %>%
    ggplot(aes(time_onset_test)) +
    geom_boxplot() +
    coord_flip() +
    theme_classic()
```



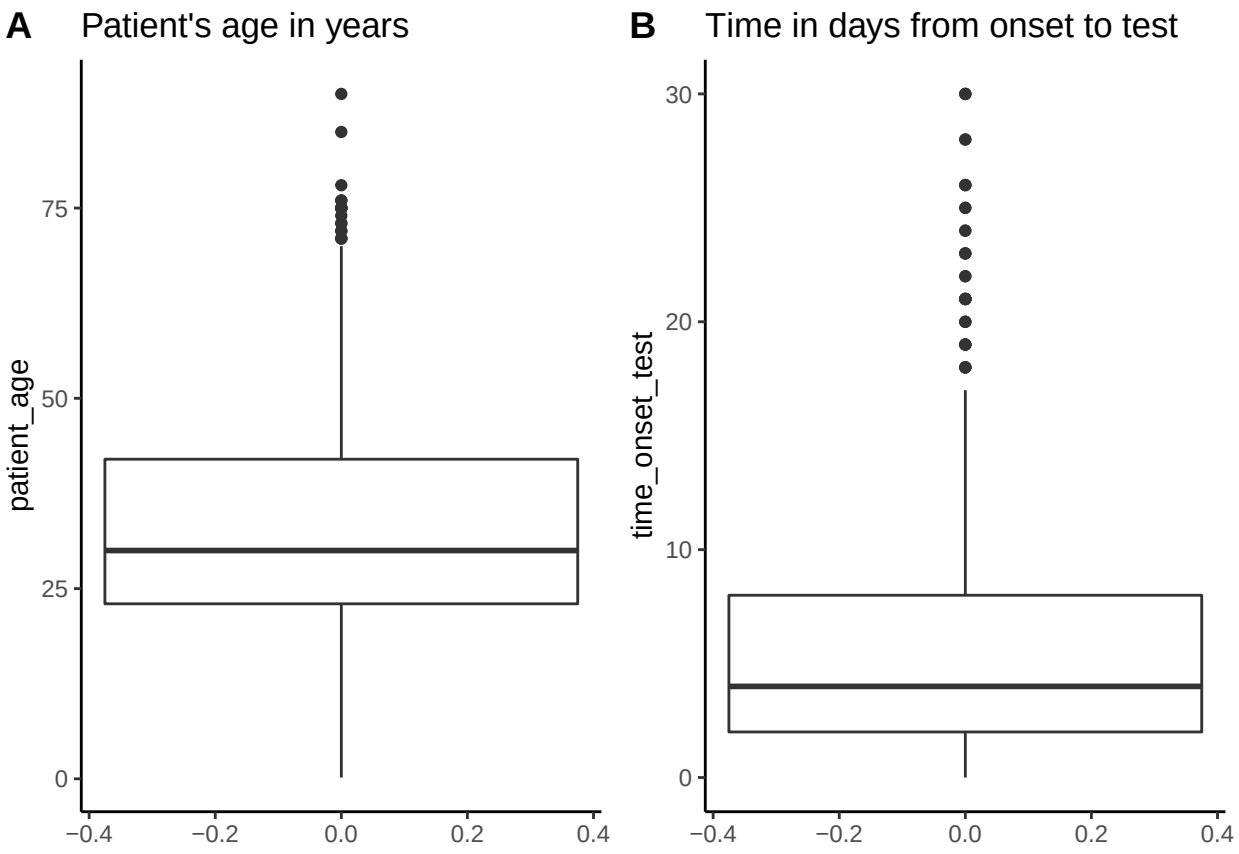
In the second histogram, there are several outlier values. On a closer look, negative time in days are not possible and for a covid-19 test, symptoms that occured more than a month ago might not be relevant for our study.

```
## remove time_onset_test with negative values or values more than 30 days
covid_processed %>%
    filter(time_onset_test >= 0 & time_onset_test <= 30) %>%
    ggplot(aes(time_onset_test)) +
    geom_boxplot() +
    coord_flip() +
    theme_classic()
```



You can combine two graphs using `plot_grid()` function from `cowplot` package.

```
## Commented out, we don't call this
library(cowplot)

## histogram for patient_age
plot_age <- covid_processed %>%
    ggplot(aes(patient_age)) +
    geom_boxplot() +
    ggtitle("Patient's age in years") +
    coord_flip() +
    theme_classic()

## histogram for time_onset_test
plot_time <- covid_processed %>%
    filter(time_onset_test >= 0 & time_onset_test <= 30) %>%
    ggplot(aes(time_onset_test)) +
    geom_boxplot() +
    ggtitle("Time in days from onset to test") +
    coord_flip() +
    theme_classic()

## combine two graphs
plot_grid(plot_age, plot_time, labels = "AUTO")
```



3.3 Tabulation of categorical data

- frequency tabulation

Let's use `tbl_summary` function from `gtsummary` package.

Here is the frequency tabulation for

```
covid_processed %>%
    tabyl(patient_sex) %>%
    adorn_totals("row") %>%
    adorn_pct_formatting()
```

```
##  patient_sex    n percent valid_percent
##       Female 1689   43.4%         43.5%
##         Male 2194   56.4%         56.5%
##         <NA>    5    0.1%             -
##        Total 3888  100.0%        100.0%
```

3.3.1 Exercises

- Try tabulating the other categorical variables in `covid_processed`.

3.3.2 Answers

Individual tabulations of all categorical variables will be skipped. Instead, a short version using `lapply` is shown below. Using `lapply` is advanced R topic and out of scope for this workshop.

```
covid_processed %>%
    select(rt_pcr_pos_neg, patient_sex:symp_number) %>%
    lapply(tabyl)
```

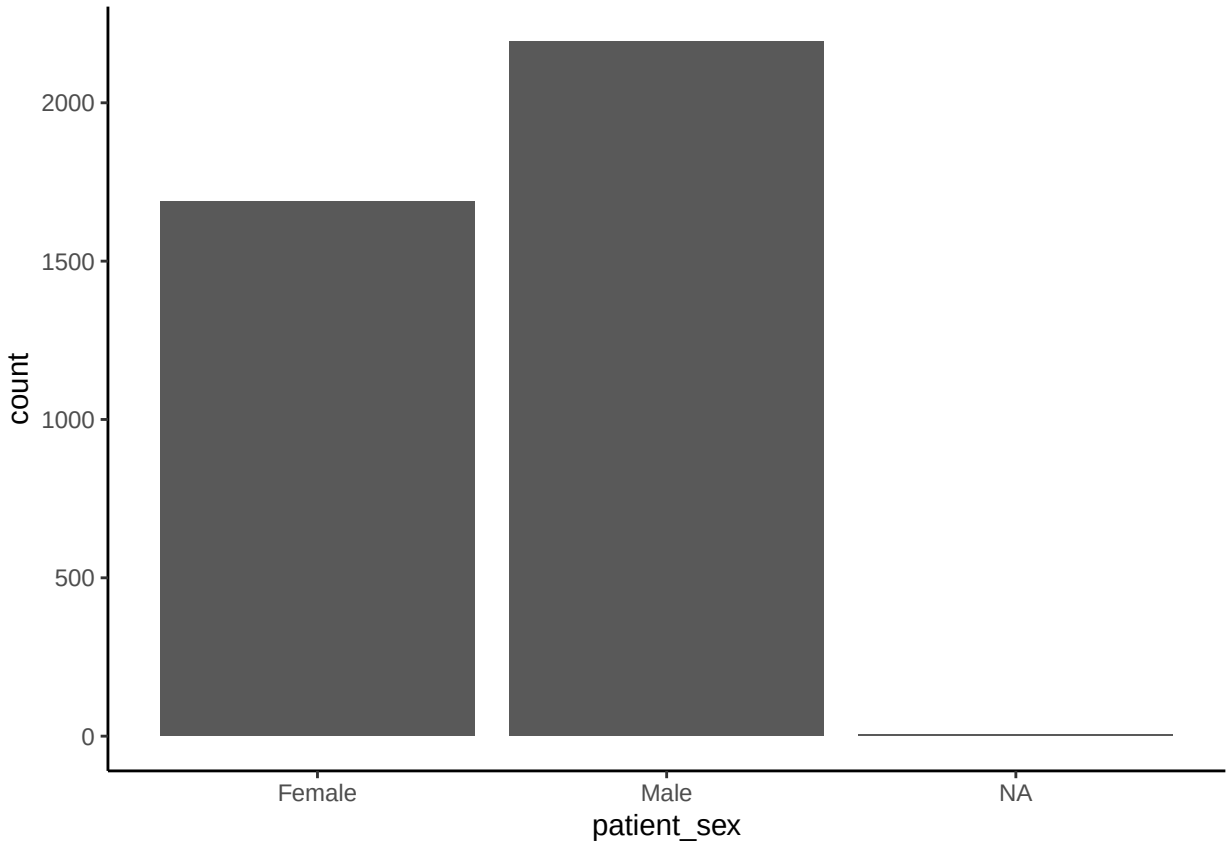
```
## $rt_pcr_pos_neg
##  X[[i]]    n   percent
##       0 3119 0.8022119
##       1  769 0.1977881
##
## $patient_sex
##  X[[i]]    n     percent valid_percent
##  Female 1689 0.434413580      0.434973
##    Male 2194 0.564300412      0.565027
##    <NA>    5 0.001286008            NA
##
## $p_province
##  X[[i]]    n   percent
##     EHP 2907 0.7476852
##   Other  981 0.2523148
##
## $symptom_status
##  X[[i]]    n   percent
##      No 2300 0.5915638
##     Yes 1588 0.4084362
##
## $symp_number
##  X[[i]]    n    percent
##       0 2389 0.61445473
##       1  568 0.14609053
##       2  488 0.12551440
##       3  293 0.07536008
##       4  108 0.02777778
##       5   42 0.01080247
```

3.4 Barplots

- Use horizontal barplot if there are more than five categories.

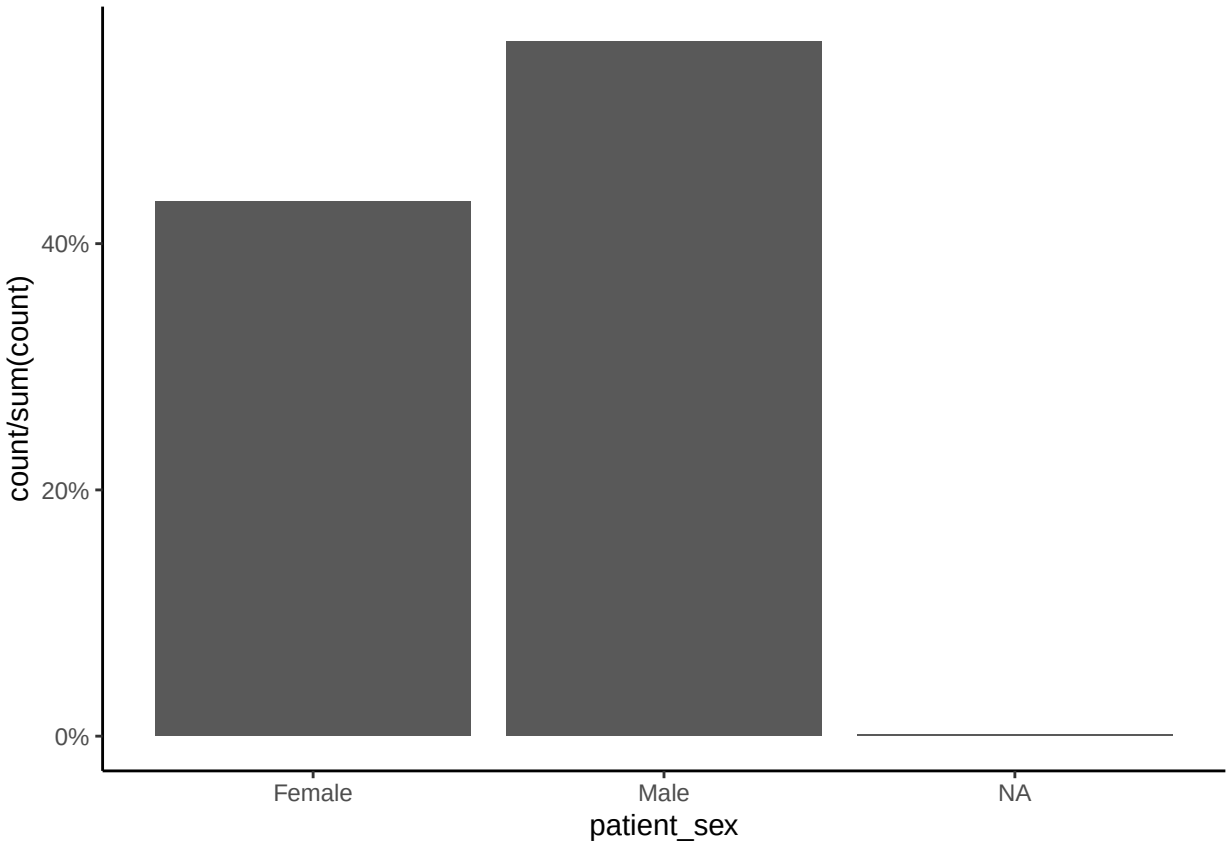
Here is a barplot of patient's sex displaying counts.

```
covid_processed %>%
    ggplot(aes(patient_sex)) +
    geom_bar() +
    theme_classic()
```



Here is a barplot of patient's sex displaying percentage.

```
covid_processed %>%
    ggplot(aes(patient_sex)) +
    geom_bar(aes(y = ..count.. / sum(..count..))) +
    scale_y_continuous(labels=scales::percent) + # this add percent sign to the axis
    theme_classic()
```


40%
20%
0%
count/sum(count)
Female
Male
NA
patient_sex

3.5 Creating Table 1

It is usually a daunting process to create publication-ready tables in any software. R is no exception.

We will use and `gtsummary` package to facilitate this process. It is a fully developed package and will take times to use its functions with ease.

```
covid_processed %>%
    tbl_summary()
```

```
## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in code chunk header.
```

Characteristic	**N = 3,888**
patient_age	30 (23, 42)
Unknown	93
patient_sex	
Female	1,689 (43%)
Male	2,194 (57%)
Unknown	5
p_province	
EHP	2,907 (75%)
Other	981 (25%)
symptom_status	1,588 (41%)
symp_number	
0	2,389 (61%)
1	568 (15%)
2	488 (13%)
3	293 (7.5%)
4	108 (2.8%)
5	42 (1.1%)
vaccine_status	712 (21%)
Unknown	441
dose_num	
0	2,735 (82%)
1	199 (5.9%)
2	413 (12%)
3	1 (<0.1%)
Unknown	540
case_contact	751 (90%)
Unknown	3,051
travel_hist	789 (39%)
Unknown	1,886
rt_pcr_pos_neg	769 (20%)
time_onset_test	4 (2, 9)
Unknown	2,788

We can add options to customize the table.

```r
covid_processed %>%
    tbl_summary(
        statistic = list(
            time_onset_test ~ "{median} ({p25}, {p75})" # "{mean} ({sd})"
        ),
        digits = all_continuous() ~ 1,
        label = list(
            patient_age = "Age in years",
            patient_sex = "Sex",
            p_province = "Province",
            symptom_status = "Symptomatic",
            symp_number = "Number of symptoms",
            vaccine_status = "Vaccination status",
            dose_num = "Number of doses received",
            case_contact = "History of contact with case",
            travel_hist = "Travel history",
            rt_pcr_pos_neg = "RT-PCR",
            time_onset_test = "Time in days from onset to test"
        ),
        missing = "ifany", ## set to "no" to remove missing values
        missing_text = "(Missing)")
```

```
## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in code chunk header.
```

Characteristic	**N = 3,888**
Age in years	30.0 (23.0, 42.0)
(Missing)	93
Sex	
Female	1,689 (43%)
Male	2,194 (57%)
(Missing)	5
Province	
EHP	2,907 (75%)
Other	981 (25%)
Symptomatic	1,588 (41%)
Number of symptoms	
0	2,389 (61%)
1	568 (15%)
2	488 (13%)
3	293 (7.5%)
4	108 (2.8%)
5	42 (1.1%)
Vaccination status	712 (21%)
(Missing)	441
Number of doses received	
0	2,735 (82%)
1	199 (5.9%)
2	413 (12%)
3	1 (<0.1%)

Characteristic	N = 3,888
(Missing)	540
History of contact with case	751 (90%)
(Missing)	3,051
Travel history	789 (39%)
(Missing)	1,886
RT-PCR	769 (20%)
Time in days from onset to test	4.0 (2.0, 9.0)
(Missing)	2,788

For more details, check **gtsummary** webpage here: https://www.danieldsjoberg.com/gtsummary/articles/tbl_summary.html

4 Relationship between two variables

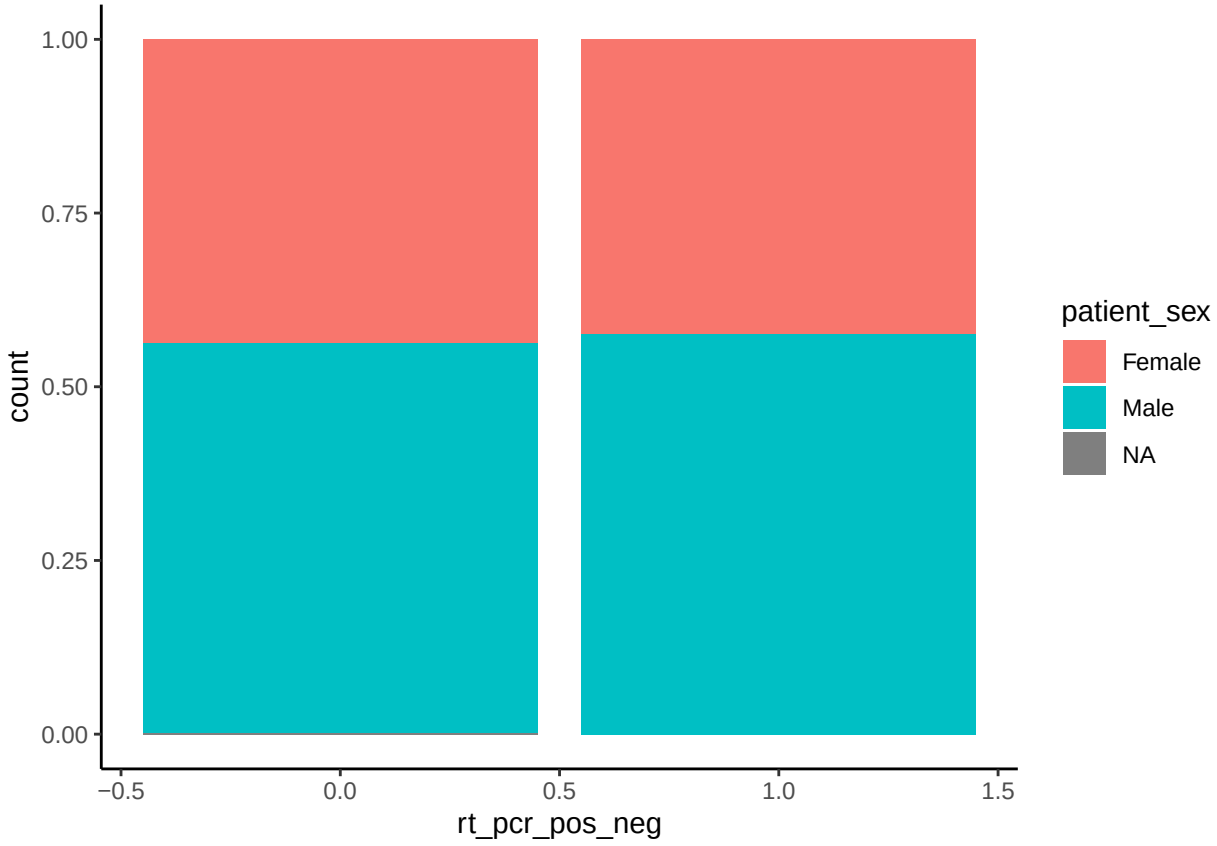
- categorical ~ categorical » cross-tabulation (contigency table)
- categorical ~ numerical » grouped (stratified) summary measures
- numerical ~ numerical » pearson's correlation (**r**)

4.1 categorical ~ categorical

```
covid_processed %>%
    tabyl(patient_sex, rt_pcr_pos_neg) %>%
    adorn_totals(c("row", "col")) %>%
    adorn_percentages("row") %>%
    adorn_pct_formatting(digits = 1, affix_sign = FALSE) %>%
    adorn_ns("front")
```

```
##  patient_sex            0          1        Total
##       Female 1363  (80.7) 326 (19.3) 1689 (100.0)
##         Male 1751  (79.8) 443 (20.2) 2194 (100.0)
##         <NA>    5 (100.0)   0  (0.0)    5 (100.0)
##        Total 3119  (80.2) 769 (19.8) 3888 (100.0)
```

```
covid_processed %>%
    ggplot(aes(rt_pcr_pos_neg, fill = patient_sex)) +
    geom_bar(position = "fill") +
    theme_classic()
```



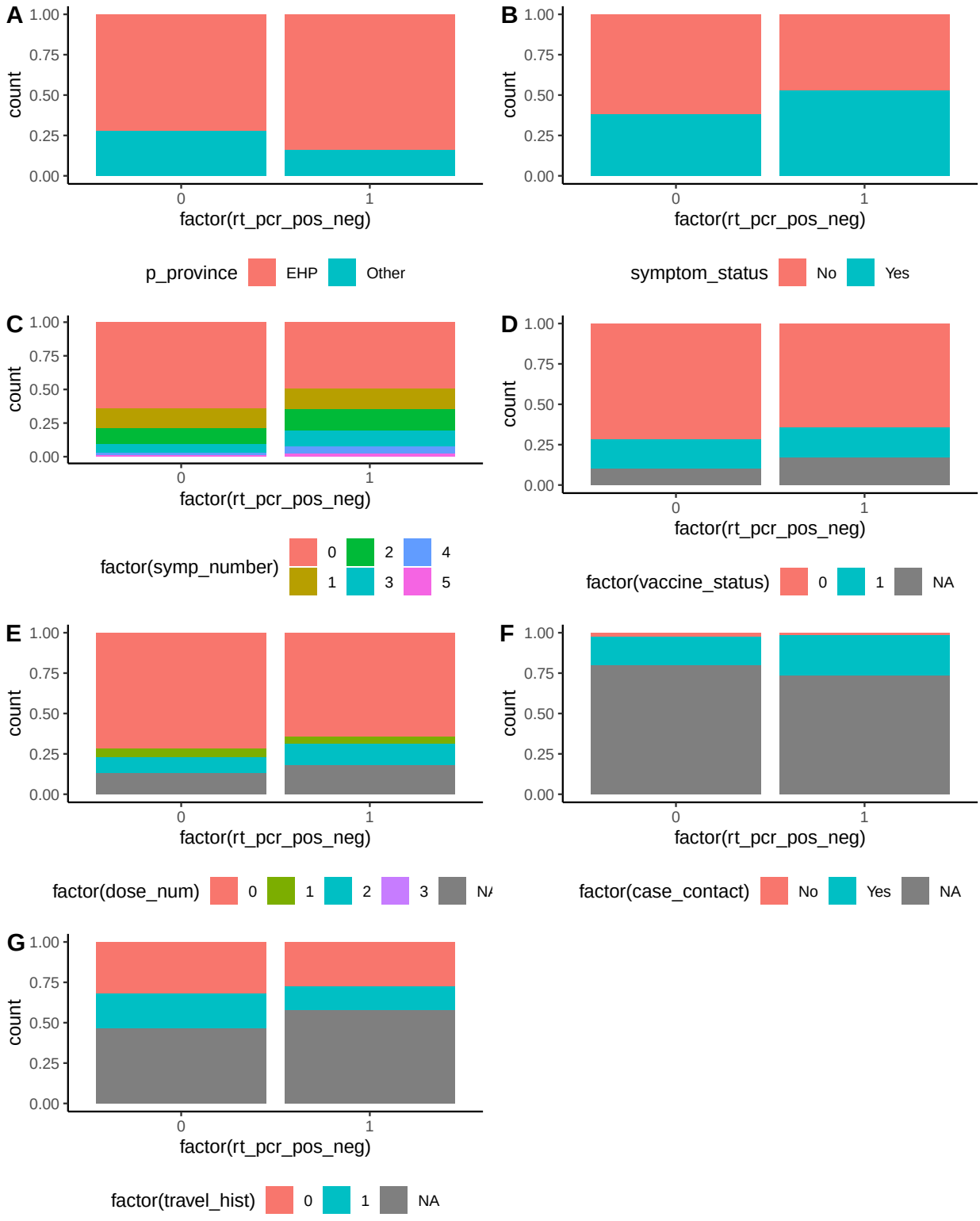
4.1.1 Exercises

- Check bivariate analysis between RT-PRC positivity and other categorical variables.

4.1.2 Answers

We will use `plot_grid` from `cowplot` package to minimize page numbers. Here we need to change data type of `rt_pcr_pos_neg` to factor, just to tell R to treat it like categorical data. We will also do this to other variables that contain numeric values.

```
plot_grid(
    covid_processed %>%
        ggplot(aes(factor(rt_pcr_pos_neg), fill = p_province)) +
        geom_bar(position = "fill") +
        theme_classic() +
        theme(legend.position = "bottom"),

    covid_processed %>%
        ggplot(aes(factor(rt_pcr_pos_neg), fill = symptom_status)) +
        geom_bar(position = "fill") +
        theme_classic() +
        theme(legend.position = "bottom"),

    covid_processed %>%
        ggplot(aes(factor(rt_pcr_pos_neg), fill = factor(symp_number))) +
        geom_bar(position = "fill") +
        theme_classic() +
        theme(legend.position = "bottom"),

    covid_processed %>%
        ggplot(aes(factor(rt_pcr_pos_neg), fill = factor(vaccine_status))) +
        geom_bar(position = "fill") +
        theme_classic() +
        theme(legend.position = "bottom"),

    covid_processed %>%
        ggplot(aes(factor(rt_pcr_pos_neg), fill = factor(dose_num))) +
        geom_bar(position = "fill") +
        theme_classic() +
        theme(legend.position = "bottom"),

    covid_processed %>%
        ggplot(aes(factor(rt_pcr_pos_neg), fill = factor(case_contact))) +
        geom_bar(position = "fill") +
        theme_classic() +
        theme(legend.position = "bottom"),

    covid_processed %>%
        ggplot(aes(factor(rt_pcr_pos_neg), fill = factor(travel_hist))) +
        geom_bar(position = "fill") +
        theme_classic() +
        theme(legend.position = "bottom"),

    labels = "AUTO",
    ncol = 2
)
```


A
B
C
D
E
F
G
count
1.00
0.75
0.50
0.25
0.00
0
1
factor(rt_pcr_pos_neg)
p_province
EHP
Other
symptom_status
No
Yes
factor(symp_number)
0
1
2
3
4
5
factor(vaccine_status)
NA
factor(dose_num)
factor(case_contact)
factor(travel_hist)

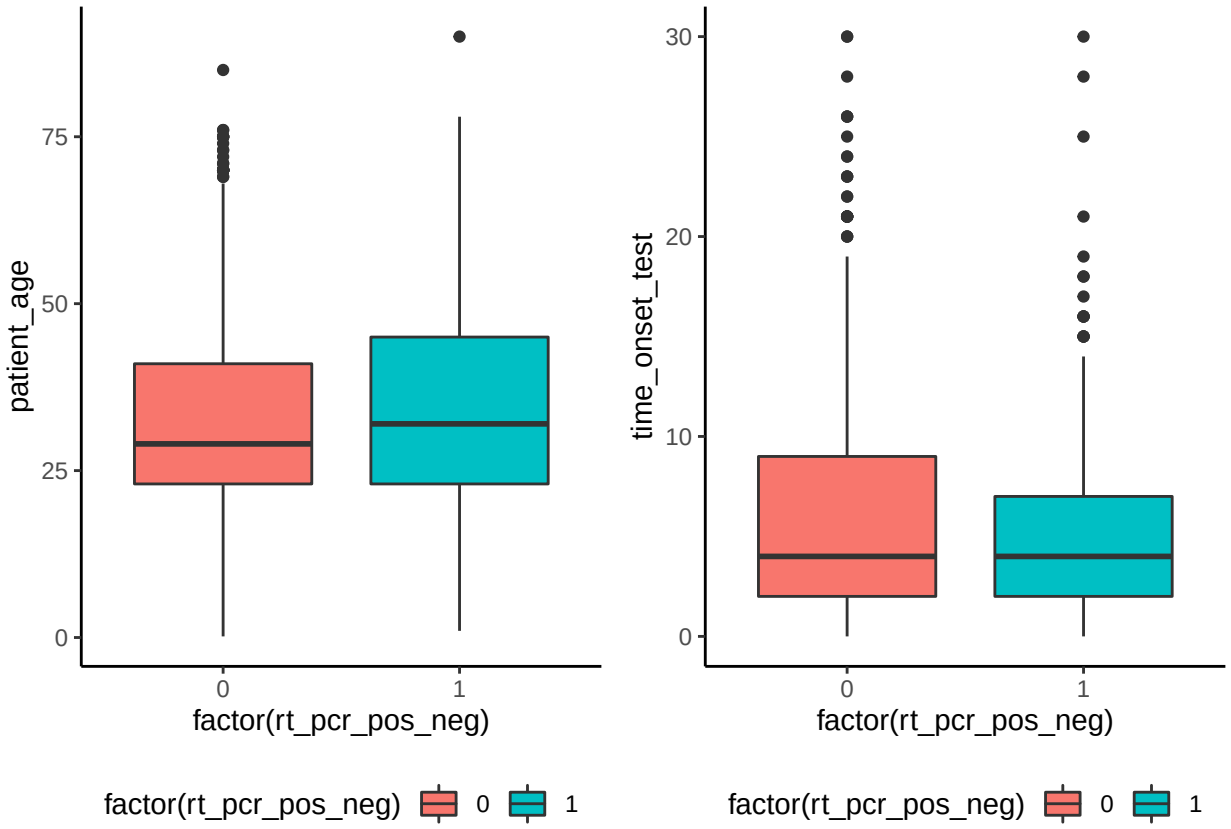
4.2 categorical ~ numerical

We have only a handful of numerical variables. Since our main outcome is RT-PCR, we will find summary measures grouped by `rt_pcr_pos_neg`.

```
covid_processed %>%
    group_by(rt_pcr_pos_neg) %>%
    summarize(mean_age = mean(patient_age, na.rm = TRUE),
              sd_age = sd(patient_age, na.rm = TRUE),
              median_time = mean(time_onset_test, na.rm = TRUE),
              q1_time = quantile(time_onset_test, probs = 0.25, na.rm = TRUE),
              q3_time = quantile(time_onset_test, probs = 0.75, na.rm = TRUE))
```

```
## # A tibble: 2 x 6
##   rt_pcr_pos_neg mean_age sd_age median_time q1_time q3_time
##            <dbl>    <dbl>  <dbl>       <dbl>   <dbl>   <dbl>
## 1              0     32.8   13.5        8.84       2       9
## 2              1     34.7   14.7        5.65       2       7
```

```
plot_grid(
    covid_processed %>%
        ggplot(aes(factor(rt_pcr_pos_neg), patient_age, fill = factor(rt_pcr_pos_neg))) +
        geom_boxplot() +
        theme_classic() +
        theme(legend.position = "bottom"),

    covid_processed %>%
        filter(time_onset_test >= 0 & time_onset_test <= 30) %>%
        ggplot(aes(factor(rt_pcr_pos_neg), time_onset_test, fill = factor(rt_pcr_pos_neg))) +
        geom_boxplot() +
        theme_classic() +
        theme(legend.position = "bottom")
)
```

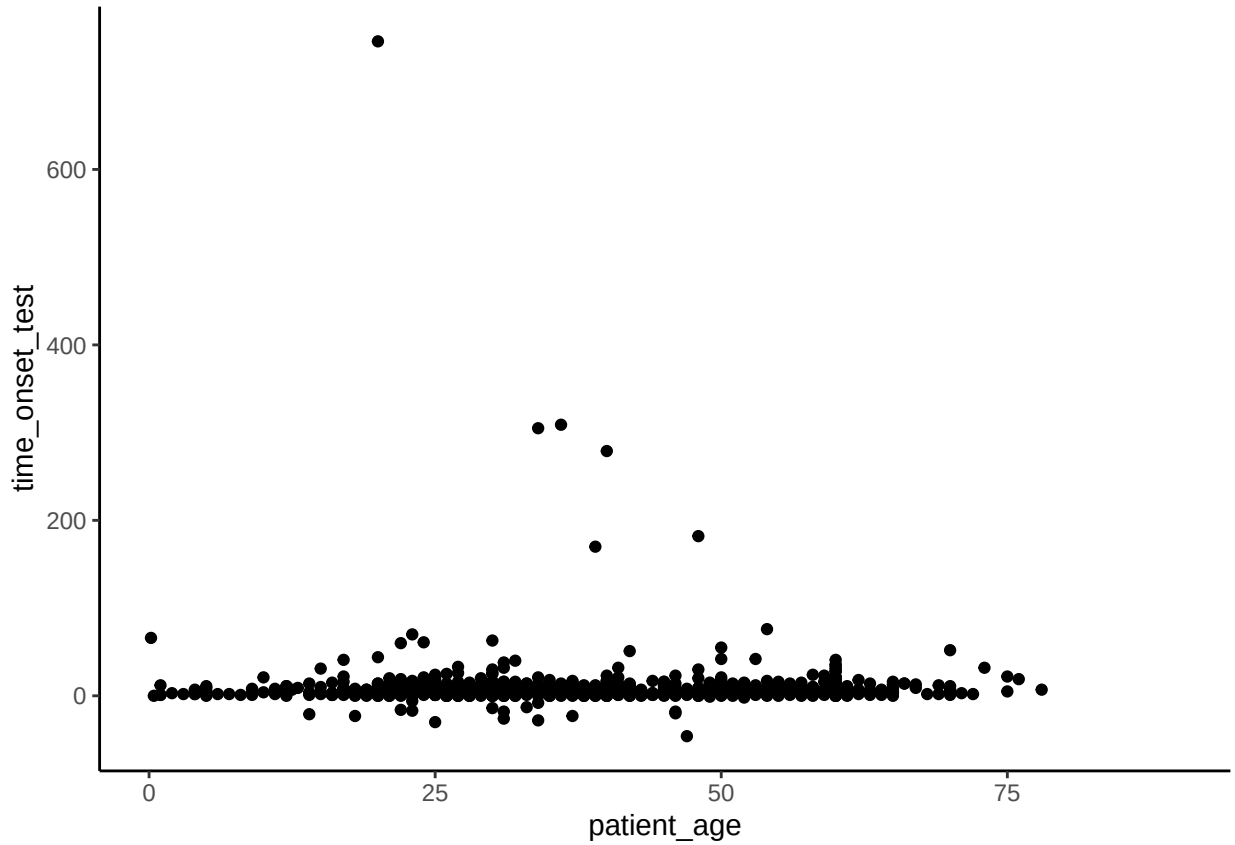


4.3 numerical ~ numerical

```
covid_processed %>%
    summarise(correlation = cor(patient_age, time_onset_test, use = "complete.obs"))
```

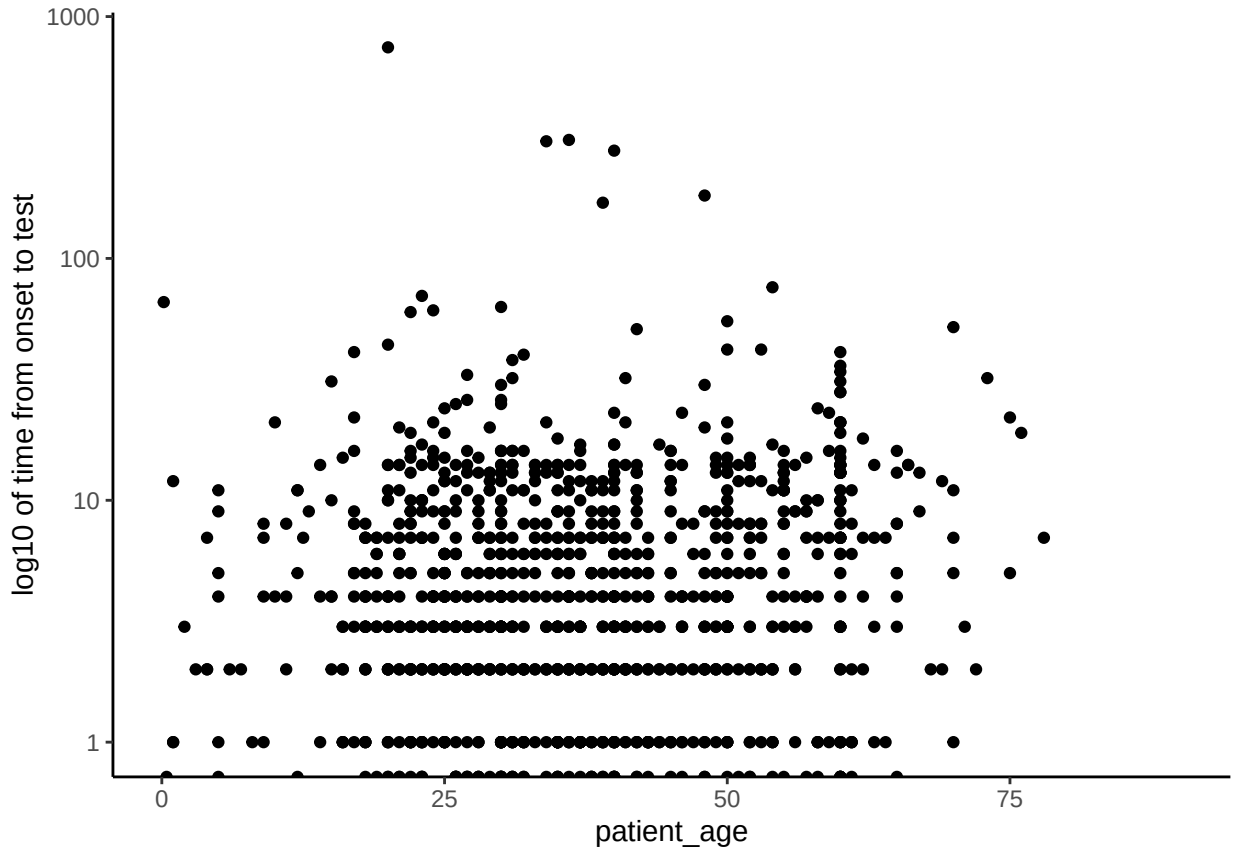
```
## # A tibble: 1 x 1
##   correlation
##         <dbl>
## 1     0.00402
```

```
covid_processed %>%
    ggplot(aes(patient_age, time_onset_test)) +
    geom_point() +
    theme_classic()
```



We notice that `time_onset_test` has a skewed distribution. Since correlation depends on the assumption of linear association, let's transform this variable by converting to log scale.

```
covid_processed %>%
    ggplot(aes(patient_age, time_onset_test)) +
    geom_point() +
    scale_y_log10() +
    ylab("log10 of time from onset to test") +
    theme_classic()
```



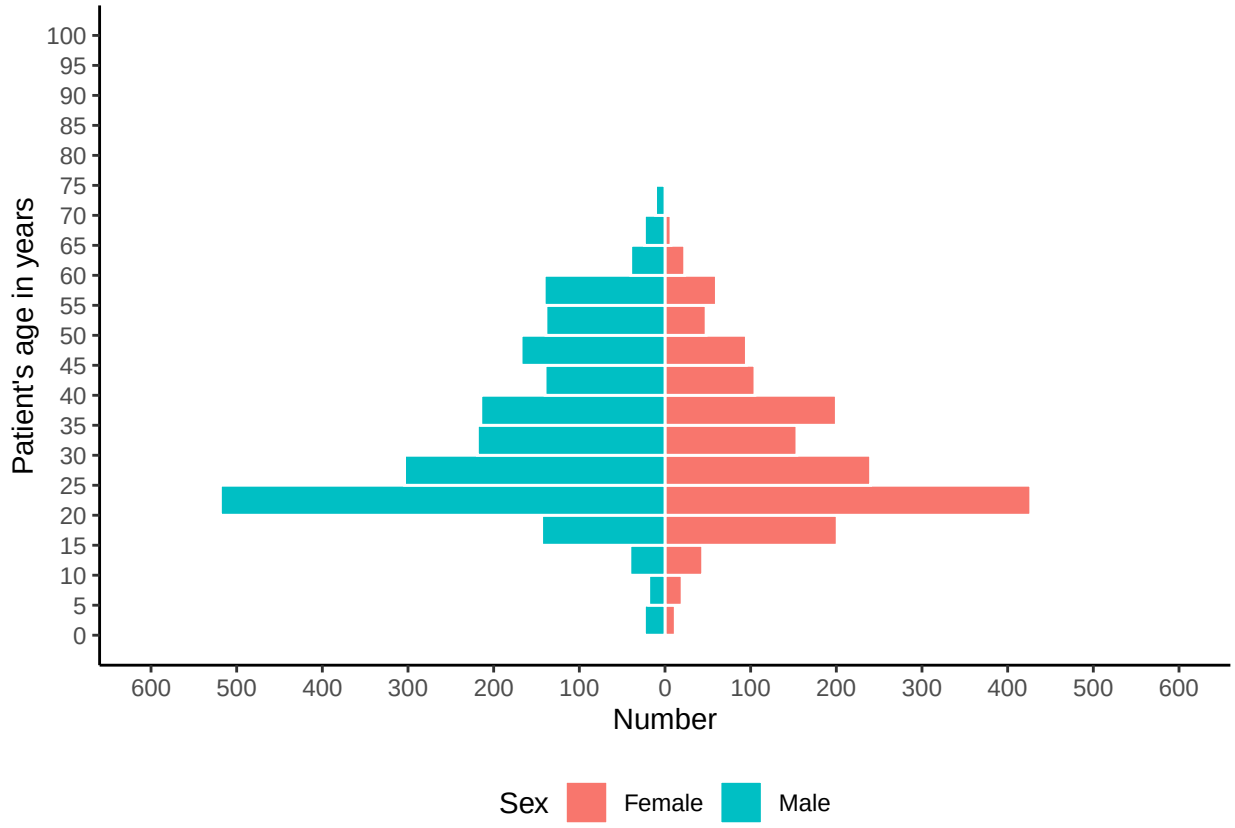
4.4 Population Pyramid graph

Another useful way of visualizating demographic data is to create a population pyramid. This graph can compare different age distributions across male and female.

- age in intervals on y-axis
- sex in x-axis

```
covid_processed %>%
    ggplot(aes(patient_age, fill = patient_sex)) +
    # female histogram
    geom_histogram(data = covid_processed %>% filter(patient_sex == "Female"),
                   breaks = seq(0, 100, 5),
                   colour = "white") +
    # male histogram
    geom_histogram(data = covid_processed %>% filter(patient_sex == "Male"),
                   breaks = seq(0, 100, 5),
                   mapping = aes(y = ..count..*(-1)),
                   colour = "white") +
    ylab("Number") +
    xlab("Patient's age in years") +
    # adjust counts-axis scale
    scale_y_continuous(limits = c(-600, 600),
                       breaks = seq(-600, 600, 100),
```

```
                    labels = abs(seq(-600, 600, 100))) +
  scale_x_continuous(limits = c(0, 100),
                     breaks = seq(0, 100, 5),
                     labels = abs(seq(0, 100, 5))) +
  # flip the X and Y axes
  coord_flip() +
  scale_fill_discrete(name = "Sex") +
  theme_classic() +
  theme(legend.position = "bottom")
```



4.5 Creating another version of Table 1 stratified by outcome variable

Our outcome of interest here is `rt_pcr_pos_neg`. So let's stratify all variables and see what we can make sense of it.

For this purpose, we use `tbl_summary` function from the `gtsummary` package. Here we add a few more options to make the table look nice.

```
covid_processed %>%
    ## convert to readable values for rt_pcr_pos_neg
    mutate(rtpcr = ifelse(rt_pcr_pos_neg == 1, "Yes", "No")) %>%
    tbl_summary(
        by = rtpcr,
        statistic = list(
            time_onset_test ~ "{median} ({p25}, {p75})" # "{mean} ({sd})"
        ),
        digits = all_continuous() ~ 1,
        label = list(
            patient_age = "Age in years",
            patient_sex = "Sex",
            p_province = "Province",
            symptom_status = "Symptomatic",
            symp_number = "Number of symptoms",
            vaccine_status = "Vaccination status",
            dose_num = "Number of doses received",
            case_contact = "History of contact with case",
            travel_hist = "Travel history",
            rt_pcr_pos_neg = "RT-PCR",
            time_onset_test = "Time in days from onset to test"
        ),
        missing = "ifany", ## set to "no" to remove missing values
        missing_text = "(Missing)") %>%
    modify_header(all_stat_cols() ~ "**{level}**<br>, N =  {n} ({style_percent(p)}%)") %>%
    add_n() %>%
    bold_labels() %>%
    modify_spanning_header(all_stat_cols() ~ "**RT-PCR Positivity**")
```

```
## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in code chunk header.
```

Characteristic	**N**	**No**, N = 3119 (80%)	**Yes**, N = 769 (20%)
Age in years	3,795	29.0 (23.0, 41.0)	32.0 (23.0, 45.0)
(Missing)		68	25
Sex	3,883		
Female		1,363 (44%)	326 (42%)
Male		1,751 (56%)	443 (58%)
(Missing)		5	0
Province	3,888		
EHP		2,259 (72%)	648 (84%)
Other		860 (28%)	121 (16%)
Symptomatic	3,888	1,181 (38%)	407 (53%)

Characteristic	**N**	**No**, N = 3119 (80%)	**Yes**, N = 769 (20%)
Number of symptoms	3,888		
0		2,010 (64%)	379 (49%)
1		447 (14%)	121 (16%)
2		370 (12%)	118 (15%)
3		203 (6.5%)	90 (12%)
4		63 (2.0%)	45 (5.9%)
5		26 (0.8%)	16 (2.1%)
Vaccination status	3,447	569 (20%)	143 (22%)
(Missing)		309	132
Number of doses received	3,348		
0		2,241 (83%)	494 (78%)
1		166 (6.1%)	33 (5.2%)
2		308 (11%)	105 (17%)
3		1 (<0.1%)	0 (0%)
(Missing)		403	137
History of contact with case	837	559 (88%)	192 (94%)
(Missing)		2,486	565
Travel history	2,002	676 (40%)	113 (35%)
(Missing)		1,441	445
RT-PCR	3,888	0 (0%)	769 (100%)
Time in days from onset to test	1,100	4.0 (2.0, 9.0)	4.0 (2.0, 7.0)
(Missing)		2,318	470

5 Inferential Statistics

5.1 Adding p-values to Table 1

```
covid_processed %>%
    ## convert to readable values for rt_pcr_pos_neg
    mutate(rtpcr = ifelse(rt_pcr_pos_neg == 1, "Yes", "No")) %>%
    tbl_summary(
        by = rtpcr,
        statistic = list(
            time_onset_test ~ "{median} ({p25}, {p75})" # "{mean} ({sd})"
        ),
        digits = all_continuous() ~ 1,
        label = list(
            patient_age = "Age in years",
            patient_sex = "Sex",
            p_province = "Province",
            symptom_status = "Symptomatic",
            symp_number = "Number of symptoms",
            vaccine_status = "Vaccination status",
            dose_num = "Number of doses received",
            case_contact = "History of contact with case",
            travel_hist = "Travel history",
            rt_pcr_pos_neg = "RT-PCR",
            time_onset_test = "Time in days from onset to test"
        ),
        missing = "ifany", ## set to "no" to remove missing values
        missing_text = "(Missing)") %>%
    modify_header(all_stat_cols() ~ "**{level}**<br>, N =  {n} ({style_percent(p)}%)") %>%
    add_p() %>%
    add_n() %>%
    bold_labels() %>%
    modify_spanning_header(all_stat_cols() ~ "**RT-PCR Positivity**")
```

```
## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in code chunk header.
```

Characteristic	**N**	**No**, N = 3119 (80%)	**Yes**, N = 769 (20%)	**p-value**
Age in years	3,795	29.0 (23.0, 41.0)	32.0 (23.0, 45.0)	0.001
(Missing)		68	25	
Sex	3,883			0.5
Female		1,363 (44%)	326 (42%)	
Male		1,751 (56%)	443 (58%)	
(Missing)		5	0	
Province	3,888			<0.001
EHP		2,259 (72%)	648 (84%)	
Other		860 (28%)	121 (16%)	
Symptomatic	3,888	1,181 (38%)	407 (53%)	<0.001
Number of symptoms	3,888			<0.001

Characteristic	N	No, N = 3119 (80%)	Yes, N = 769 (20%)	p-value
0		2,010 (64%)	379 (49%)	
1		447 (14%)	121 (16%)	
2		370 (12%)	118 (15%)	
3		203 (6.5%)	90 (12%)	
4		63 (2.0%)	45 (5.9%)	
5		26 (0.8%)	16 (2.1%)	
Vaccination status	3,447	569 (20%)	143 (22%)	0.2
(Missing)		309	132	
Number of doses received	3,348			0.003
0		2,241 (83%)	494 (78%)	
1		166 (6.1%)	33 (5.2%)	
2		308 (11%)	105 (17%)	
3		1 (<0.1%)	0 (0%)	
(Missing)		403	137	
History of contact with case	837	559 (88%)	192 (94%)	0.018
(Missing)		2,486	565	
Travel history	2,002	676 (40%)	113 (35%)	0.068
(Missing)		1,441	445	
RT-PCR	3,888	0 (0%)	769 (100%)	<0.001
Time in days from onset to test	1,100	4.0 (2.0, 9.0)	4.0 (2.0, 7.0)	0.070
(Missing)		2,318	470	

5.2 Linear regression

While our main outcome is RT-PCR positivity, we will use `time_onset_test` for the purpose of demonstrating linear regression. Let's see what factors predict time from onset of symptoms to testing.

5.2.1 Running a simple linear model

Let's start with a simple linear regression which contains an outcome and only one predictor `patient_age`.

```
m1 <- lm(time_onset_test ~ patient_age, data = covid_processed)
m1
```

```
##
## Call:
## lm(formula = time_onset_test ~ patient_age, data = covid_processed)
##
## Coefficients:
## (Intercept)  patient_age
##    7.733737     0.008351
```

As you can see, time from onset to testing slowly increases at a rate of 0.008351 day with one year increment in patient's age.

Let's see how well this model fits.

```
summary(m1)
```

```
##
## Call:
## lm(formula = time_onset_test ~ patient_age, data = covid_processed)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -54.13  -6.08  -4.04   0.80 738.10
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.733737   2.517531   3.072  0.00218 **
## patient_age 0.008351   0.063714   0.131  0.89575
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 29.75 on 1063 degrees of freedom
##   (2823 observations deleted due to missingness)
## Multiple R-squared:  1.616e-05,  Adjusted R-squared:  -0.0009246
## F-statistic: 0.01718 on 1 and 1063 DF,  p-value: 0.8957
```

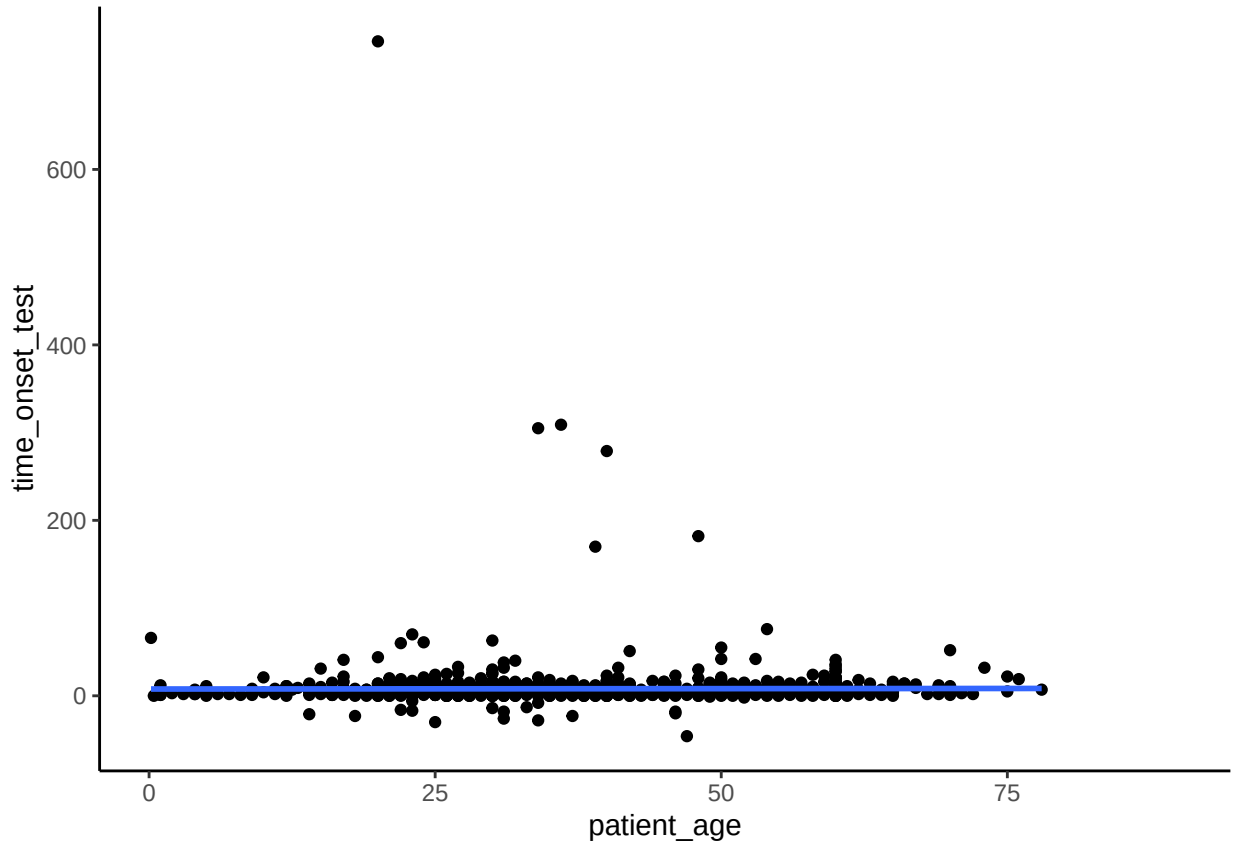
It seems this model with age as predictor is not doing well.

5.2.2 Visualizing linear relationships

Let's plot this relationship.

```
covid_processed %>%
    ggplot(aes(patient_age, time_onset_test)) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE) +
    theme_classic()
```

```
## `geom_smooth()` using formula 'y ~ x'
```

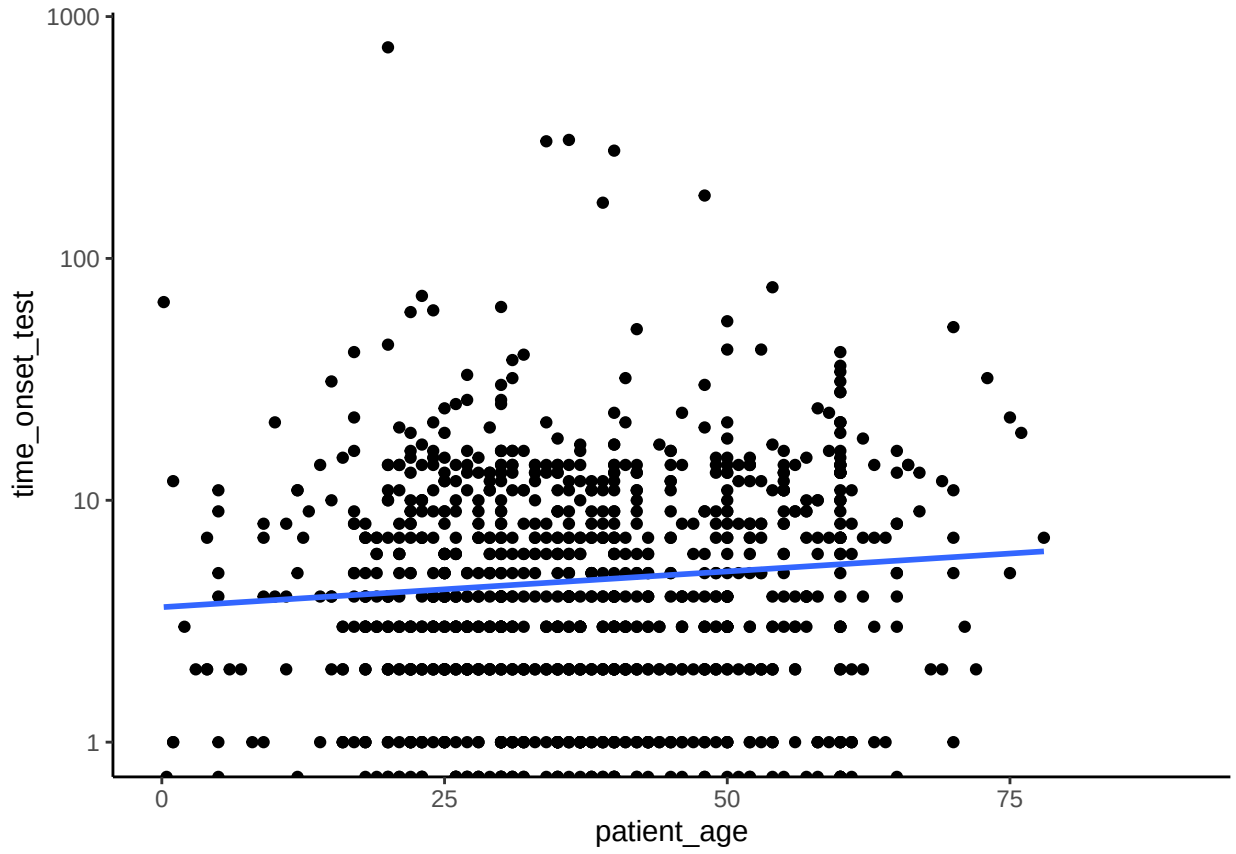


You can see clearly from this graph that the relationship is not linear, especially because time_onset_test has a skewed distribution.

Let's see if we can make this work by converting it to log scale.

```
covid_processed %>%
    ggplot(aes(patient_age, time_onset_test)) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE) +
    scale_y_log10() +
    theme_classic()
```

```
## `geom_smooth()` using formula 'y ~ x'
```

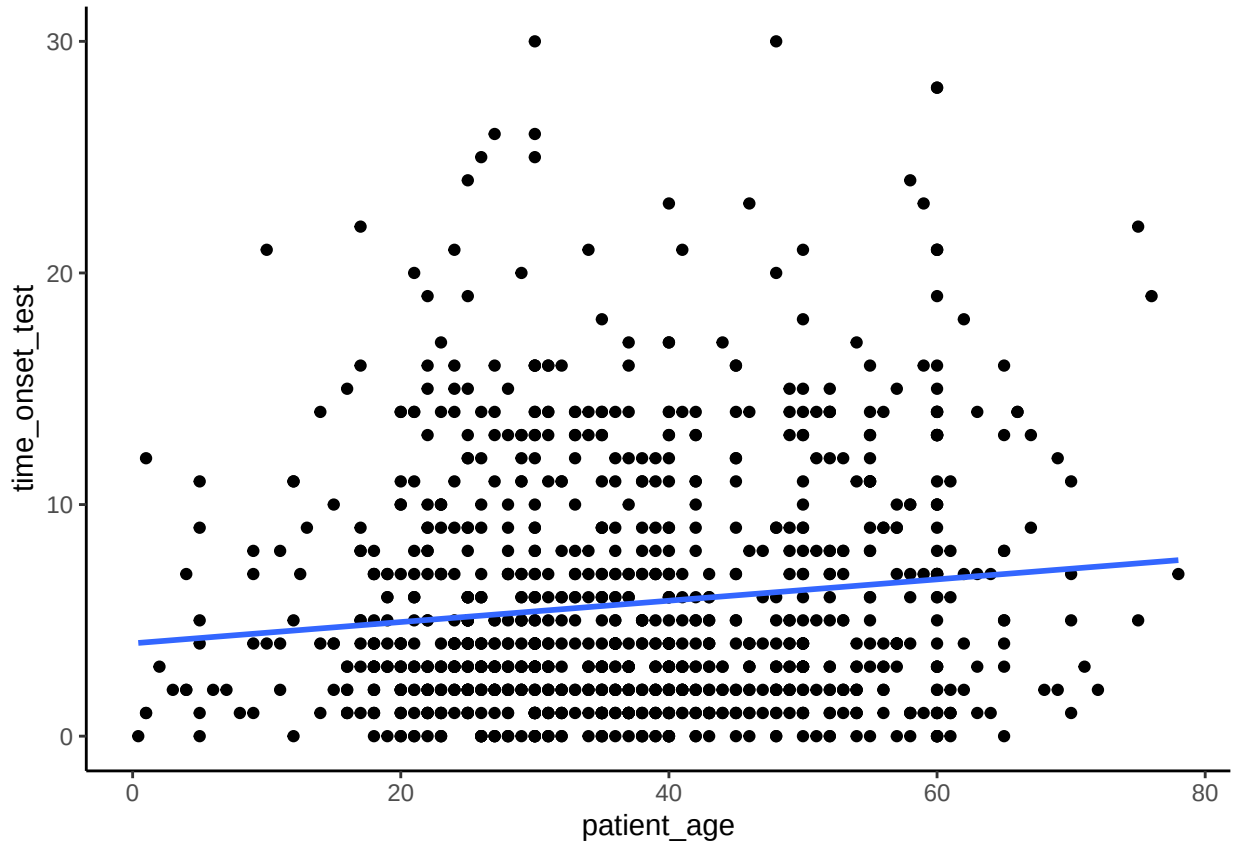


It still doesn't fit well.

We have already discussed that time_onset_test has invalid or irrelevant values. We can try the regression by removing these values.

```
covid_processed %>%
    filter(time_onset_test >= 0 & time_onset_test <= 30) %>%
    ggplot(aes(patient_age, time_onset_test)) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE) +
    theme_classic()
```

```
## `geom_smooth()` using formula 'y ~ x'
```



It seems like there is a really weak association between them but still the model is not that good.

Let's check some values of model fitness.

```
m2 <- covid_processed %>%
    filter(time_onset_test >= 0 & time_onset_test <= 30) %>%
    lm(time_onset_test ~ patient_age, data = .)
summary(m2)
```

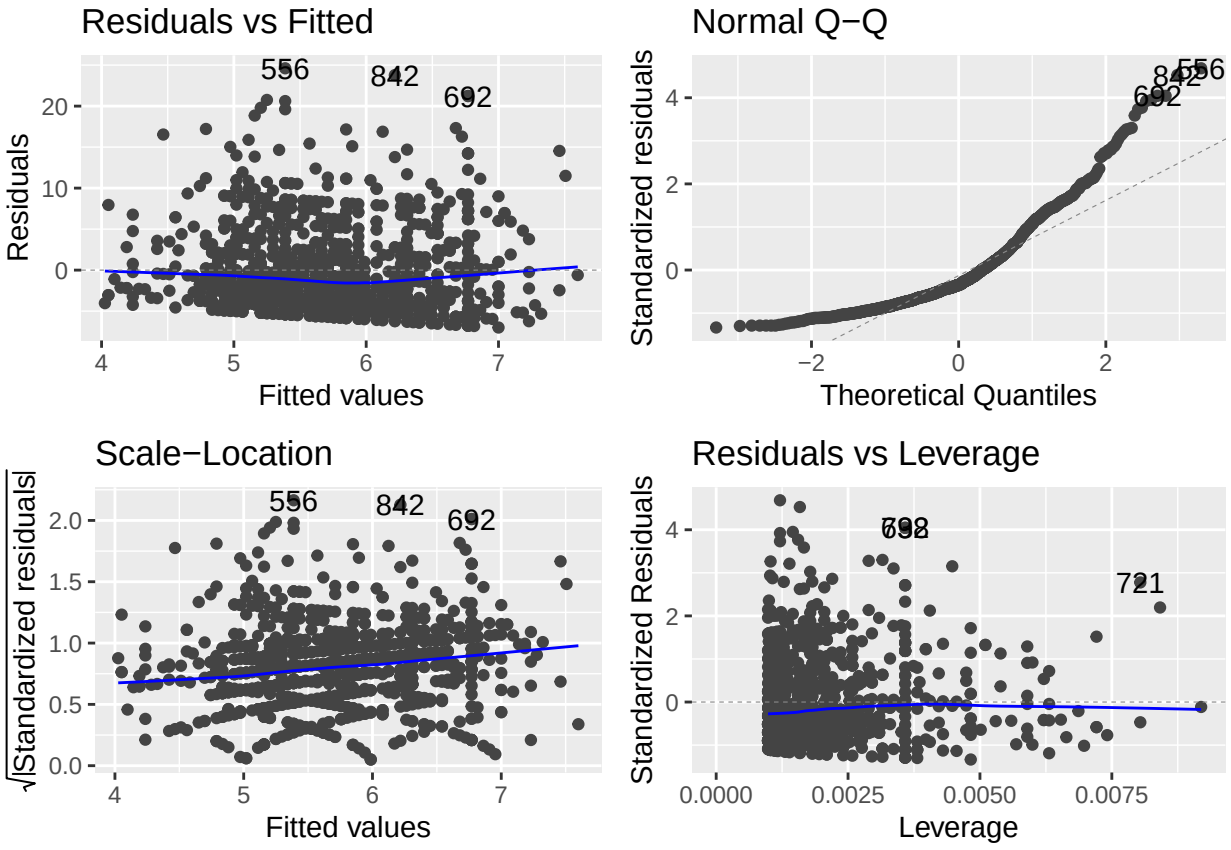
```
##
## Call:
## lm(formula = time_onset_test ~ patient_age, data = .)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.000 -3.770 -1.835  2.414 24.612
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.00595    0.45717   8.762  < 2e-16 ***
## patient_age  0.04606    0.01157   3.980 7.38e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.257 on 1015 degrees of freedom
##   (35 observations deleted due to missingness)
```

```
## Multiple R-squared:  0.01537,    Adjusted R-squared:  0.0144
## F-statistic: 15.84 on 1 and 1015 DF,  p-value: 7.385e-05
```

Now, we are getting something. WIth one year increase in age, it takes additional 0.04 day (almost one hour) longer. But only 1.44 % (R-squared value) is explained by `patient_age`.

Let's do the `autoplot` from `ggfortify` package.

```
library(ggfortify)
autoplot(m2)
```



- Figure 1: Residuals versus Fitted
  - although residual values are stable across fitted values, they are not normally distributed.
- Figure 2: Normal Q-Q plot
  - outcome is not normally distributed.
- Figure 3: Scale-Location
  - seems like a minor heteroskedascity issue (equal variance)
- Figure 4: Residuals versus leverage
  - several leverage and influential points

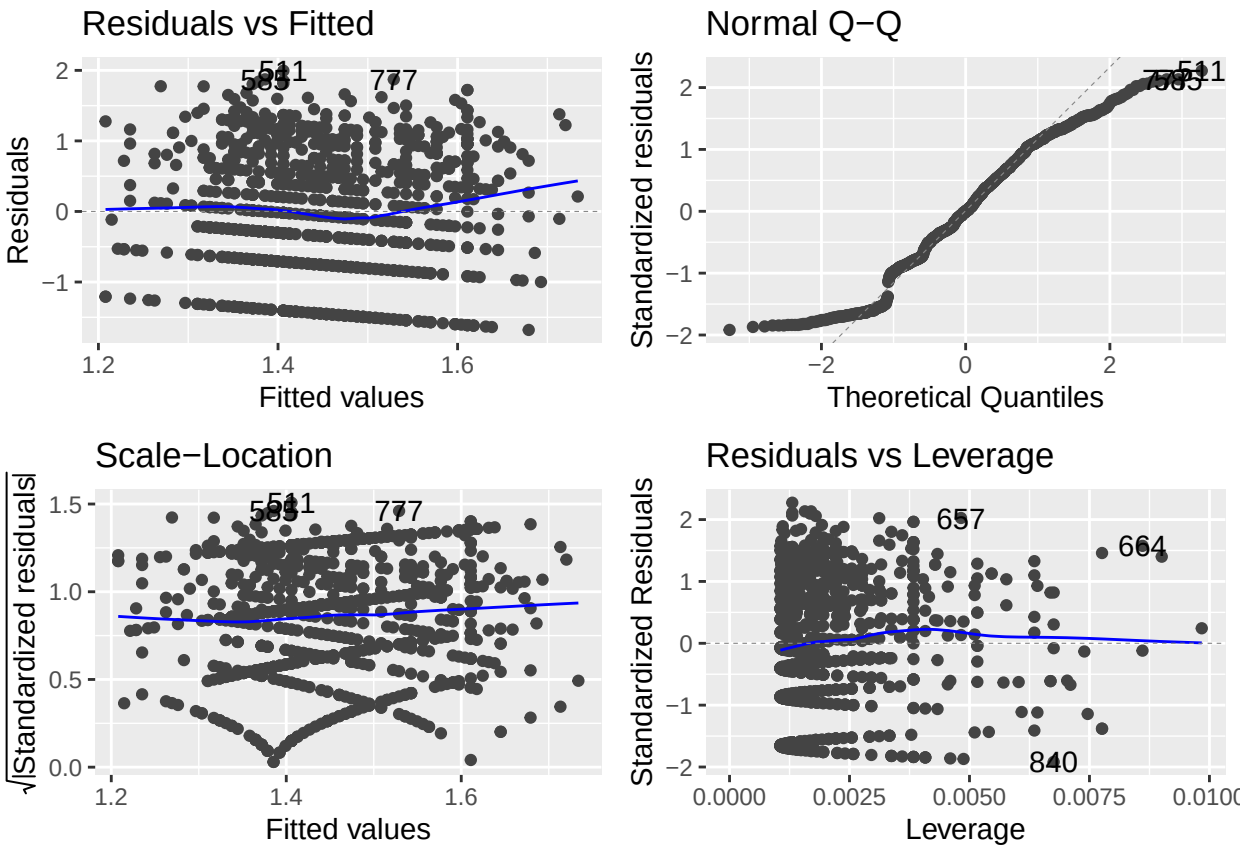
All these points indicate our linear model `m2` is still a poor fit.

Let's do log transformation for the second time. To convert on log scale, we need to be careful with zero. So we remove time of zero from this dataset as well.

```r
m3 <- covid_processed %>%
    ## remove zero values here
    filter(time_onset_test > 0 & time_onset_test <= 30) %>%
    lm(log(time_onset_test) ~ patient_age, data = .)
summary(m3)
```

```
##
## Call:
## lm(formula = log(time_onset_test) ~ patient_age, data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.67922 -0.67870 -0.00035  0.69509  1.99519
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.201094   0.079254   15.15  < 2e-16 ***
## patient_age 0.006830   0.002003    3.41 0.000677 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8784 on 942 degrees of freedom
##   (29 observations deleted due to missingness)
## Multiple R-squared:  0.01219,    Adjusted R-squared:  0.01115
## F-statistic: 11.63 on 1 and 942 DF,  p-value: 0.0006769
```

```r
autoplot(m3)
```



In terms of model fit, it's not doing any better. But if you look at the residuals values or Figure 1, it pretty much center around 0 and less skewed now. But you can see patterns in these figures, still meaning a poor fit.

Things to consider

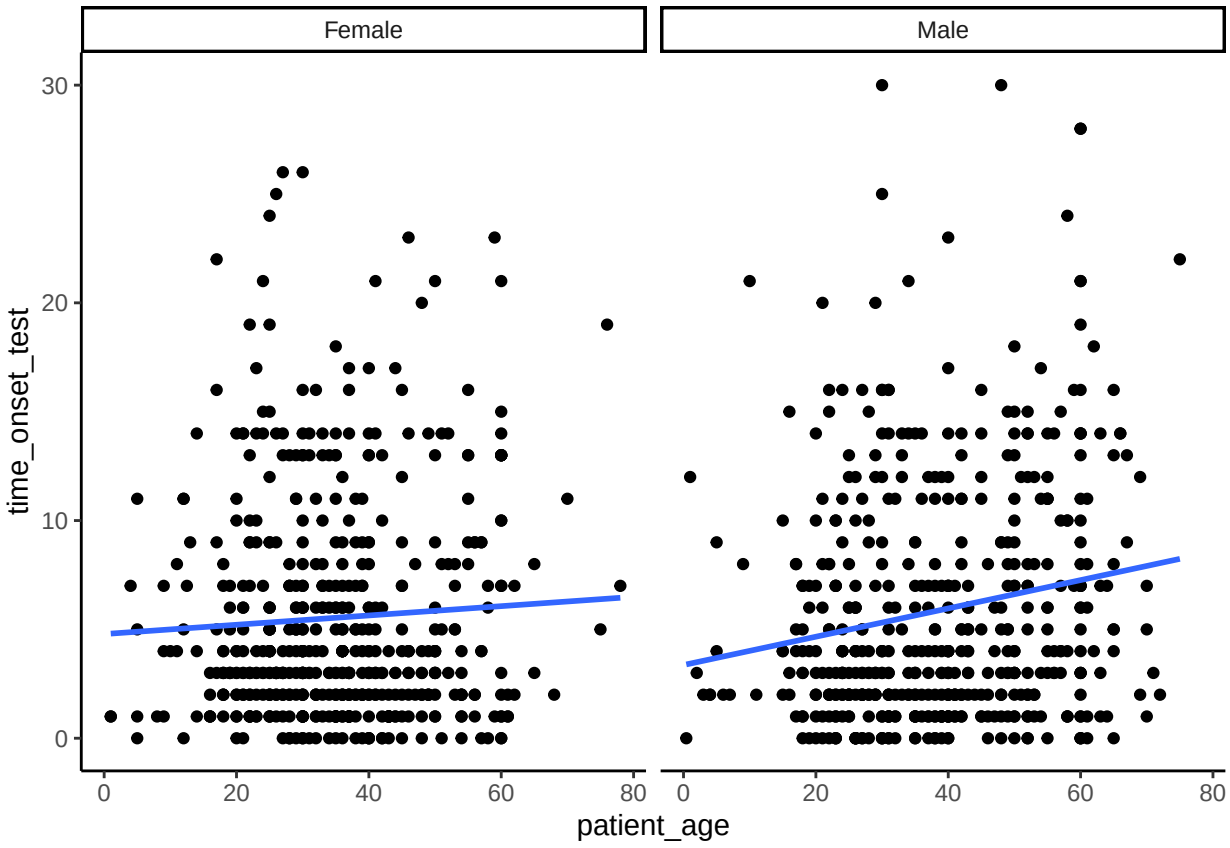
- removing observations reduces sample size hence power to reject null hypothesis.
- transforming data might complicate interpretation
- consider using other models

5.2.3 Adding a categorical variable

Let's add `patient_sex`. We remove missing values. In fact, if data quality is good, there shouldn't be missing values in `sex`.

```
covid_processed %>%
    filter(time_onset_test >= 0 & time_onset_test <= 30) %>%
    filter(!is.na(patient_sex)) %>%
    ggplot(aes(patient_age, time_onset_test)) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE) +
    facet_grid(cols = vars(patient_sex)) +
    theme_classic()
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Male older patients seem to be taking longer time.

Let's run the model.

```
m4 <- covid_processed %>%
    filter(time_onset_test > 0 & time_onset_test <= 30) %>%
    lm(log(time_onset_test) ~ patient_age + patient_sex, data = .)
summary(m4)
```

```
##
## Call:
## lm(formula = log(time_onset_test) ~ patient_age + patient_sex,
##     data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.69632 -0.68681 -0.00571  0.69625  1.96851
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.186547   0.081146  14.622  < 2e-16 ***
## patient_age      0.006591   0.002024   3.257  0.00117 **
## patient_sexMale 0.048418   0.057802   0.838  0.40244
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.8786 on 941 degrees of freedom
##   (29 observations deleted due to missingness)
## Multiple R-squared:  0.01293,    Adjusted R-squared:  0.01083
## F-statistic: 6.163 on 2 and 941 DF,  p-value: 0.002191
```

Adding `patient_sex` to the model seems to reduce the model performance (look at adjusted R-squared). To conclude, `patient_sex` is not a significant predictor for `time_onset_test`.

5.3 Logistic regression

```
logm1 <- glm(rt_pcr_pos_neg ~ patient_age, data = covid_processed,
             family = binomial)
logm1
```

```
##
## Call:  glm(formula = rt_pcr_pos_neg ~ patient_age, family = binomial,
##     data = covid_processed)
##
## Coefficients:
## (Intercept)  patient_age
##     -1.7585       0.0103
##
## Degrees of Freedom: 3794 Total (i.e. Null);  3793 Residual
##   (93 observations deleted due to missingness)
## Null Deviance:       3756
## Residual Deviance: 3744  AIC: 3748
```

```
summary(logm1)
```

```
##
## Call:
## glm(formula = rt_pcr_pos_neg ~ patient_age, family = binomial,
##     data = covid_processed)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8319  -0.6738  -0.6344  -0.6168   1.9538
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.758537   0.108050 -16.275  < 2e-16 ***
## patient_age  0.010297   0.002913   3.535 0.000409 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3756.1  on 3794  degrees of freedom
## Residual deviance: 3743.8  on 3793  degrees of freedom
##   (93 observations deleted due to missingness)
## AIC: 3747.8
```

```
##
## Number of Fisher Scoring iterations: 4
```

It seems like age is a good predictor of RT PCR positivity. But is it?

With one year increase, the log odds of being PCR positive rises by 0.010297. Let's convert this to odds ratio which we can comprehend more easily.

```
exp(0.010297)
```

```
## [1] 1.01035
```

So there is only 1% chance of being PCR positive with age increment. AIC value for this model is `3747.8`.

5.3.1 Add a categorical variable

```
logm2 <- glm(rt_pcr_pos_neg ~ patient_age + patient_sex, data = covid_processed,
             family = binomial)
summary(logm2)
```

```
##
## Call:
## glm(formula = rt_pcr_pos_neg ~ patient_age + patient_sex, family = binomial,
##     data = covid_processed)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8260  -0.6738  -0.6348  -0.6138   1.9478
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.76761    0.11395 -15.513  < 2e-16 ***
## patient_age      0.01021    0.00294   3.472 0.000516 ***
## patient_sexMale  0.02288    0.08348   0.274 0.783982
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3754.8  on 3791  degrees of freedom
## Residual deviance: 3742.4  on 3789  degrees of freedom
##   (96 observations deleted due to missingness)
## AIC: 3748.4
##
## Number of Fisher Scoring iterations: 4
```

The slope value of `patient_age` barely changes and it is highly significant. But adding patient_sex to the model increases AIC value which is not good. So it is statistically useless to the model.

Let's try `symptom_status`.

```r
logm3 <- glm(rt_pcr_pos_neg ~ symptom_status, data = covid_processed,
             family = binomial)
summary(logm3)
```

```
##
## Call:
## glm(formula = rt_pcr_pos_neg ~ symptom_status, family = binomial,
##     data = covid_processed)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7696  -0.7696  -0.5852  -0.5852   1.9230
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -1.67777    0.05726 -29.302  < 2e-16 ***
## symptom_statusYes  0.61246    0.08113   7.549 4.38e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3867.2  on 3887  degrees of freedom
## Residual deviance: 3810.1  on 3886  degrees of freedom
## AIC: 3814.1
##
## Number of Fisher Scoring iterations: 4
```

```r
exp(0.61246)
```

```
## [1] 1.844964
```

So patients who shows symptoms were 1.84 times more likely to have a PCR test positive. But AIC value increases to 3814.1.

5.3.2 Things to consider

- how do you know which variables to add in the model?
  - AIC, BIC, or likelihood ratio test
- interaction terms or any confounding variables
- how to handle missing values

6 Creating tables for regression models

We will use another function `tbl_regression` from the same `gtsummary` package.

6.1 Linear regression

```
lm_final <- covid_processed %>%
    lm(time_onset_test ~ patient_age + patient_sex + p_province +
           symptom_status + vaccine_status + case_contact, data = .)
summary(lm_final)
```

```
##
## Call:
## lm(formula = time_onset_test ~ patient_age + patient_sex + p_province +
##     symptom_status + vaccine_status + case_contact, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.194  -4.347  -1.594   1.929 171.880
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -6.62164   13.20675  -0.501    0.617
## patient_age        0.12417    0.08879   1.398    0.164
## patient_sexMale    2.80126    2.31608   1.209    0.228
## p_provinceOther   -1.79336    3.18116  -0.564    0.574
## symptom_statusYes  9.20256   11.23241   0.819    0.414
## vaccine_status    -3.51500    2.54759  -1.380    0.169
## case_contactYes   -1.22229    6.05279  -0.202    0.840
##
## Residual standard error: 15.7 on 186 degrees of freedom
##   (3695 observations deleted due to missingness)
## Multiple R-squared:  0.03522,    Adjusted R-squared:  0.004098
## F-statistic: 1.132 on 6 and 186 DF,  p-value: 0.3455
```

```
tbl_regression(lm_final,
                label = list(
            patient_age = "Age in years",
            patient_sex = "Sex",
            p_province = "Province",
            symptom_status = "Symptomatic",
            vaccine_status = "Vaccination status",
            case_contact = "History of contact with case"
               )) %>%
        bold_labels() %>%
        add_global_p() %>%  # add global p-value
        bold_p(t = 0.10) %>% # bold p-value
        italicize_levels()
```

```
## add_global_p: Global p-values for variable(s) `add_global_p(include
## = c("patient_age", "patient_sex", "p_province", "symptom_status",
```

```
## "vaccine_status", "case_contact"))‘ were calculated with
##   ‘car::Anova(x$model_obj, type = "III")‘
## Table printed with ‘knitr::kable()‘, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include ‘message = FALSE‘ in code chunk header.
```

Characteristic	Beta	95% CI	p-value
Age in years	0.12	-0.05, 0.30	0.2
Sex			0.2
Female			
Male	2.8	-1.8, 7.4	
Province			0.6
EHP			
Other	-1.8	-8.1, 4.5	
Symptomatic			0.4
No			
Yes	9.2	-13, 31	
Vaccination status	-3.5	-8.5, 1.5	0.2
History of contact with case			0.8
No			
Yes	-1.2	-13, 11	

No significant predictors in this model! This is expected.

6.2 Logistic regression

First we categorize `patient_age` for better interpretability.

```
covid_processed <- covid_processed %>%
    mutate(age_grp = case_when(
        patient_age < 18 ~ "18 years",
        patient_age >= 18 & patient_age < 60 ~ "18-59 years",
        patient_age >= 60 ~ "60+ years"
    ))
logm_final <- glm(rt_pcr_pos_neg ~ age_grp + p_province +
                      symptom_status + vaccine_status +
                      case_contact,
                  data = covid_processed,
                  family = binomial)
summary(logm_final)
```

```
##
## Call:
## glm(formula = rt_pcr_pos_neg ~ age_grp + p_province + symptom_status +
##     vaccine_status + case_contact, family = binomial, data = covid_processed)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9658  -0.7302  -0.6391  -0.2825   2.5460
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -0.1812     0.9855  -0.184 0.854134
## age_grp18-59 years   -0.0753     0.3690  -0.204 0.838304
## age_grp60+ years     -0.1362     0.6282  -0.217 0.828383
## p_provinceOther      -1.7166     0.4790  -3.584 0.000339 ***
## symptom_statusYes     0.6278     0.2099   2.990 0.002787 **
## vaccine_status       -0.2611     0.2318  -1.126 0.260051
## case_contactYes      -0.9671     0.9224  -1.048 0.294415
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 601.84  on 554  degrees of freedom
## Residual deviance: 573.19  on 548  degrees of freedom
##   (3333 observations deleted due to missingness)
## AIC: 587.19
##
## Number of Fisher Scoring iterations: 5
```

```
tbl_regression(logm_final,
               exponentiate = TRUE,
               label = list(
            age_grp = "Age categories",
            p_province = "Province",
            symptom_status = "Symptomatic",
```

```
            vaccine_status = "Vaccination status",
            case_contact = "History of contact with case",
            rt_pcr_pos_neg = "RT-PCR"
               )) %>%
        bold_labels() %>%
        add_global_p() %>%  # add global p-value
        bold_p(t = 0.10) %>% # bold p-value
        italicize_levels()
```

```
## add_global_p: Global p-values for variable(s) `add_global_p(include =
## c("age_grp", "p_province", "symptom_status", "vaccine_status", "case_contact"))`
## were calculated with
##   `car::Anova(x$model_obj, type = "III")`
## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in code chunk header.
```

Characteristic	OR	95% CI	p-value
Age categories			>0.9
18 years			
18-59 years	0.93	0.46, 1.98	
60+ years	0.87	0.24, 2.89	
Province			**<0.001**
EHP			
Other	0.18	0.06, 0.42	
Symptomatic			**0.003**
No			
Yes	1.87	1.24, 2.83	
Vaccination status	0.77	0.49, 1.21	0.3
History of contact with case			0.3
No			
Yes	0.38	0.07, 2.98	

Patients who resided outside EHP are less likely to test positive and those symptomatic patients were more likely to have a positive PCR test.

7 References

1. Datacamp. Career Track - Statistician with R. 2022
2. Various vignettes including janitor and gtsummary