Survey Markdown

Group 16

2023-06-11

Contents


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

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library("tidyverse")
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.1     v tidyr     1.3.0
## v readr     2.1.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
#install.packages("gt")
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
library(gt)
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```r
library(readr)
```

The Impact of Online Learning on Technical Education During the Pandemic

```r
data <- read.csv("group_16.csv", header= TRUE, sep=";")
# preprocessing

# Split the string into parts
split_parts <- strsplit(data$Antwort.1, "; ")
split_parts2 <- strsplit(data$Antwort.2, "; ")
split_parts3 <- strsplit(data$Antwort.3, "; ")

# Create new columns based on split parts
new_columns <- t(sapply(split_parts, function(x) {
  parts <- strsplit(x, ": ")
  setNames(as.numeric(sapply(parts, function(y) y[2])), sapply(parts, function(y) paste0("Antwort1", y[
}))

new_columns2 <- t(sapply(split_parts2, function(x) {
  parts <- strsplit(x, ": ")
  setNames(as.numeric(sapply(parts, function(y) y[2])), sapply(parts, function(y) paste0("Antwort2", y[
}))

new_columns3 <- t(sapply(split_parts3, function(x) {
  parts <- strsplit(x, ": ")
  setNames(sapply(parts, function(y) y[2]), sapply(parts, function(y) paste0("Antwort3", y[1])))
}))
```

```r
# Combine the new columns with the original data frame
data_p <- cbind(data[c(1,2,3)], new_columns, new_columns2, new_columns3)

# Rename the columns
colnames(data_p) <- c("gender", "age", "academic.program",
                      "test.prep.dur", "test.prep.bef","test.prep.af","ects.fail.dur","semesters.dur",
                      "ects.fail.bef","semesters.bef","ects.fail.af","semesters.af",
                      "source.dur","source.bef","source.af")


# Replace values based on condition
data_p$academic.program <- ifelse(grepl("Data Science", data_p$academic.program), "Data Science / MSc /

data_p$gender = as.factor(data_p$gender)
data_p$academic.program = as.factor(data_p$academic.program)
data_p$source.dur = as.factor(data_p$source.dur)
data_p$source.bef = as.factor(data_p$source.bef)
data_p$source.af = as.factor(data_p$source.af)

data_p[c(5,38), c("ects.fail.dur", "ects.fail.bef", "ects.fail.af",
                  "semesters.dur", "semesters.bef", "semesters.af")] <- NA

#remove NaN
data_p = na.omit(data_p)
```

Exploration

What was the average time required for test preparation during and before/after the pandemic (in hours)

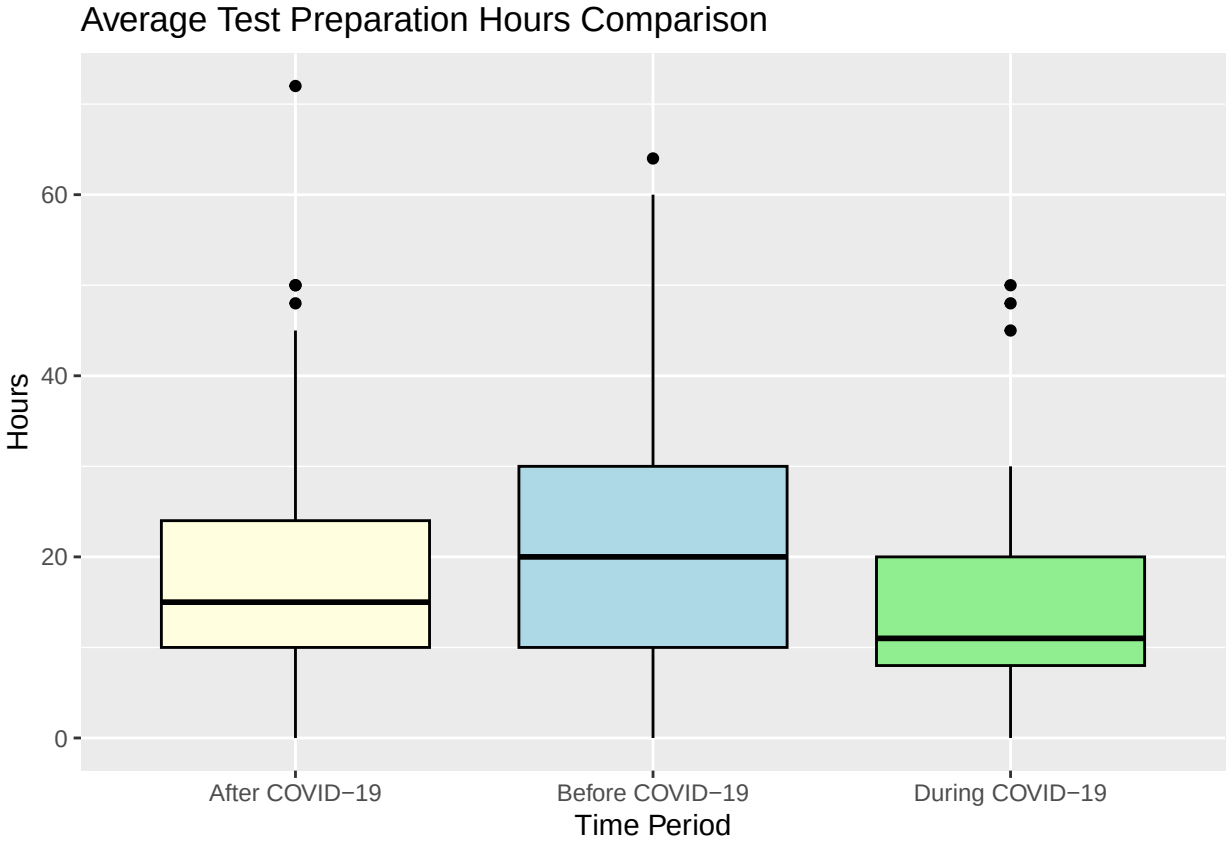
1. `pivot_longer`: This function is used to convert the data frame `data_p` from wide format to long format. It takes three columns (`test.prep.bef`, `test.prep.dur`, `test.prep.af`) and pivots them into two columns: `Period` and `Hours`. The `Period` column contains the names of the original three columns, and the `Hours` column contains the corresponding values.
2. `group_by` and `summarize`: These functions are used together to calculate statistics for each time period (`Before`, `During`, `After`). The `group_by` function groups the data by the `Period` column, and the `summarize` function calculates three statistics: `Mean` (average), `Median`, and `SD` (standard deviation) of the `Hours` column within each time period.
3. `print`: This function is used to print the formatted statistics to the console.
4. `ggplot`: This function is used to create a boxplot visualization. It takes the `data_p` data frame (excluding the 21st row) as input.
5. `geom_boxplot`: These functions are used to add boxplots to the plot for each time period. They specify the aesthetics (`aes`) mapping the x-axis to the time period (`"Before"`, `"During"`, `"After"`) and the y-axis to the corresponding columns (`test.prep.bef`, `test.prep.dur`, `test.prep.af`). The `fill` parameter sets the colors for the boxplots, and the `color` parameter sets the color for the outline.
6. `labs`: This function is used to set the labels for the x-axis, y-axis, and title of the plot.
7. `scale_x_discrete`: This function is used to set custom labels for the x-axis, mapping the original time period names to more descriptive labels.

```r
#print out statistics
# Convert the data frame to long format
```

```r
data_long <- data_p %>%
  pivot_longer(cols = c(test.prep.bef,test.prep.dur,test.prep.af), names_to = "Period", values_to = "Hou

# Calculate statistics for each time period
statistics <- data_long %>%
  group_by(Period) %>%
  summarize(
    Mean = mean(Hours),
    Median = median(Hours),
    SD = sd(Hours)
  )

# Print the formatted statistics
print(statistics)
```

```
## # A tibble: 3 x 4
##   Period         Mean Median    SD
##   <chr>         <dbl>  <dbl> <dbl>
## 1 test.prep.af   27.7     15  50.0
## 2 test.prep.bef  30.8     20  49.6
## 3 test.prep.dur  23.6     12  49.7
```

```r
# Create a boxplot for each column
# we exclude row 21 because of unrealistic values
ggplot(data_p[-19,]) +
  geom_boxplot(aes(x = "Before", y = test.prep.bef), fill = "lightblue", color = "black") +
  geom_boxplot(aes(x = "During", y = test.prep.dur), fill = "lightgreen", color = "black") +
  geom_boxplot(aes(x = "After", y = test.prep.af), fill = "lightyellow", color = "black") +
  labs(x = "Time Period", y = "Hours", title = "Average Test Preparation Hours Comparison") +
  scale_x_discrete(labels = c("Before" = "Before COVID-19", "During" = "During COVID-19", "After" = "Af
```



Based on the graph, we observe that the average test preparation before COVID was the highest, with an average of 29.4 hours per test. The median for this period was also the highest at 20 hours, indicating a wider range of preparation times. Notably, 25% of the students required more than 30 hours of learning time per test during this period.

During COVID, there was a significant decrease in preparation time, which suggests the possibility of more efficient study practices (assuming successful exams during that period). This observation is interesting considering that the university was closed and access to learning materials was limited.

After the pandemic, we can see a median learning time of 15 hours. This trend indicates that the shift to online lectures and 24/7 availability of uploaded materials contributed to increased studying efficiency during the COVID pandemic.

How many ECTS/semester did you fail on the first try during/before/after the pandemic period? (Example Answer: During the pandemic I did 4 Semesters and failed 6 ECTS)

1. `mutate`: This function is used multiple times to create new columns in the `data_p` data frame. The `mutate` function calculates and assigns values to the new columns `performance_dur`, `performance_bef`, and `performance_af`. These columns are derived by dividing the corresponding columns (`ects.fail.dur`, `ects.fail.bef`, `ects.fail.af`) by the corresponding columns (`semesters.dur`, `semesters.bef`, `semesters.af`).
2. `summarise`: This function is used to calculate summary statistics for the `data_p` data frame. It computes the mean of the `performance_dur`, `performance_bef`, and `performance_af` columns while excluding any non-finite values (`is.finite`) and removing any missing values (`na.rm = TRUE`). The resulting statistics are assigned to the `stats` data frame.
3. `data.frame`: This function is used to create a new data frame called `performance`. It combines the performance metrics for all periods (during, before, after) into a single data frame with two

columns: `period` and `performance`. The `period` column is created using the `rep` function to repeat the time period labels, and the `performance` column combines the corresponding performance values from `data_p`. The argument `stringsAsFactors = FALSE` ensures that character vectors are not automatically converted to factors.

4. `geom_histogram`: These functions are used to create histograms for each period using the `ggplot` package. They specify the aesthetics (`aes`) mapping the x-axis to the respective performance columns (`performance_bef`, `performance_dur`, `performance_af`). The `binwidth` parameter determines the width of the bins in the histogram. The `fill` parameter sets the fill color of the histogram bars, and the `alpha` parameter controls the transparency. The `color` parameter sets the color of the outline of the bars.
5. `labs`: This function is used to set the x-axis and y-axis labels for the histograms.
6. `ggtitle`: This function is used to set the title of each histogram.
7. `theme_minimal`: This function is used to set a minimalistic theme for the histograms.
8. `geom_vline`: These functions are used to add vertical dashed lines to each histogram, indicating the mean value of the respective performance column. The `xintercept` aesthetic specifies the x-coordinate of the vertical line.
9. `grid.arrange`: This function is used to arrange the histograms in subplots. It takes the individual histograms (`hist_bef`, `hist_dur`, `hist_af`) as arguments and arranges them in a single row (`nrow = 1`). The `bottom` parameter is used to add a bottom label to the subplot.

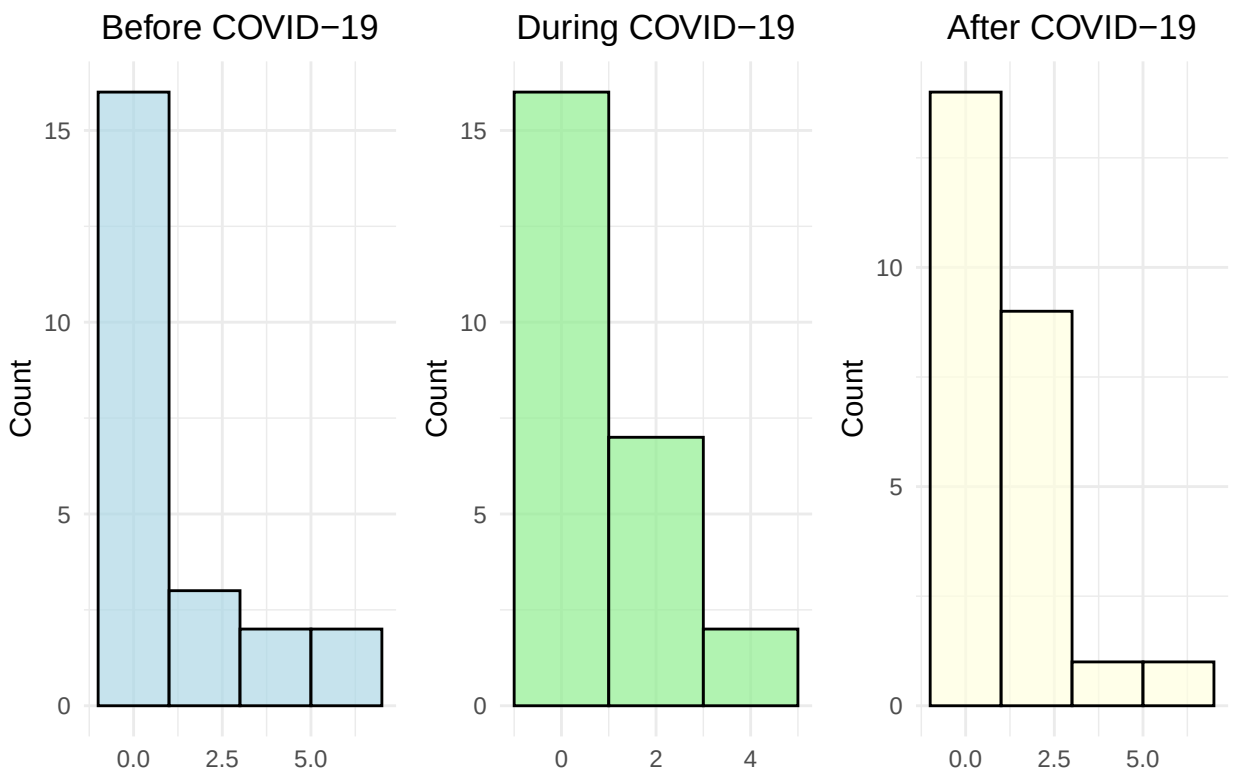
```
#head(data_p[c(7:12)])
data_p <- data_p %>%
  mutate(performance_dur = ects.fail.dur / semesters.dur) %>%
  mutate(performance_bef = ects.fail.bef / semesters.bef) %>%
  mutate(performance_af = ects.fail.af / semesters.af)

#print out statistics
stats <- data_p %>%
  summarise(
    Performance_Dur = mean(performance_dur[is.finite(performance_dur)], na.rm = TRUE),
    Performance_Bef = mean(performance_bef[is.finite(performance_bef)], na.rm = TRUE),
    Performance_Af = mean(performance_af[is.finite(performance_af)], na.rm = TRUE)
  )

print(stats)
```

```
##   Performance_Dur Performance_Bef Performance_Af
## 1        1.042667        1.184783       1.159333
```

```
# Combine the performance metrics for all periods into a single data frame
performance <- data.frame(
  period = rep(c("During", "Before", "After"), each = nrow(data_p)),
  performance = c(data_p$performance_dur, data_p$performance_bef, data_p$performance_af),
  stringsAsFactors = FALSE
)

# Create separate histograms for each period
hist_bef <- ggplot(data_p, aes(x = performance_bef)) +
  geom_histogram(binwidth = 2, fill = "lightblue", alpha = 0.7, color="black") +
  labs(x = "", y = "Count") +
  ggtitle("Before COVID-19") +
  theme_minimal() +
```

```
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_vline(aes(xintercept = mean(performance_bef)), color = "black", linetype = "dashed")

hist_dur <- ggplot(data_p, aes(x = performance_dur)) +
  geom_histogram(binwidth = 2, fill = "lightgreen", alpha = 0.7, color="black") +
  labs(x = "", y = "Count") +
  ggtitle("During COVID-19") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_vline(aes(xintercept = mean(performance_dur)), color = "black", linetype = "dashed")

hist_af <- ggplot(data_p, aes(x = performance_af)) +
  geom_histogram(binwidth = 2, fill = "lightyellow", alpha = 0.7, color="black") +
  labs(x = "", y = "Count") +
  ggtitle("After COVID-19") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_vline(aes(xintercept = mean(performance_af)), color = "black", linetype = "dashed")


# Arrange the histograms in subplots
subplot <- grid.arrange(hist_bef, hist_dur, hist_af, nrow = 1,
                        bottom="failed ECTS / attended Semester Ratio")
```

```
## Warning: Removed 12 rows containing non-finite values (`stat_bin()`).
## Warning: Removed 35 rows containing missing values (`geom_vline()`).
## Warning: Removed 10 rows containing non-finite values (`stat_bin()`).
## Warning: Removed 35 rows containing missing values (`geom_vline()`).
## Warning: Removed 10 rows containing non-finite values (`stat_bin()`).
## Warning: Removed 35 rows containing missing values (`geom_vline()`).
```



failed ECTS / attended Semester Ratio

```
# Display the subplots
subplot
```

```
## TableGrob (2 x 3) "arrange": 4 grobs
##   z     cells    name                grob
## 1 1 (1-1,1-1) arrange      gtable[layout]
## 2 2 (1-1,2-2) arrange      gtable[layout]
## 3 3 (1-1,3-3) arrange      gtable[layout]
## 4 4 (2-2,1-3) arrange text[GRID.text.180]
```

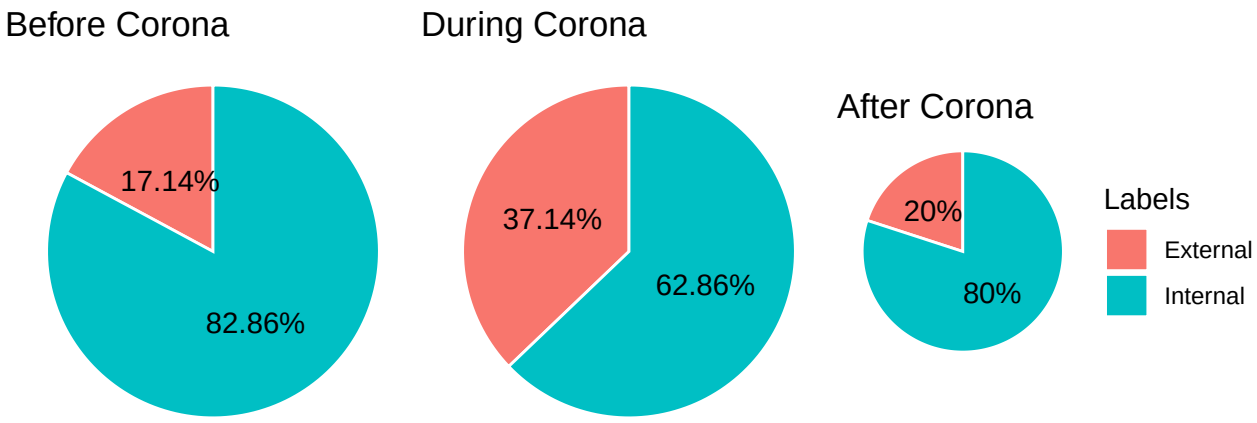
In this plot, we analyze the performance of students in three periods. We examine the ratio between failed semesters and attended semesters for each student in each period. If a student did not study before COVID, their score would be NaN and therefore not included in the graph. Thus, we only observe data from students who studied during a particular period. A score of zero indicates that no ECTS (European Credit Transfer and Accumulation System) were failed during the attended semesters.

The average performance ratio before COVID was 1.13, during COVID it was 1.0, and after COVID it was 1.11. The performance during COVID was the best. However, we notice an improvement in performance when comparing the data from before and after COVID. It is important to note that the chart depicting the data from before COVID is not as well-populated as the others. This is because only a few students started their studies during COVID. Nonetheless, we observe that over five students achieved a performance ratio of about 2, and we also see a few students with a similar performance ratio after COVID.

What was your main source of learning for Test preparation during/before/after the pandemic?

```
# Calculate the frequency of each label
label_counts <- table(data_p$source.bef)
label_counts2 <- table(data_p$source.dur)
```

```r
label_counts3 <- table(data_p$source.af)

# Convert counts to proportions
label_proportions <- prop.table(label_counts)
label_proportions2 <- prop.table(label_counts2)
label_proportions3 <- prop.table(label_counts3)

# Create a data frame for plotting
plot_data <- data.frame(labels = names(label_proportions),
                        proportions = label_proportions)
plot_data2 <- data.frame(labels = names(label_proportions2),
                        proportions = label_proportions2)
plot_data3 <- data.frame(labels = names(label_proportions3),
                        proportions = label_proportions3)

# Create a pie plot using ggplot2
pie_plot <- ggplot(plot_data, aes(x = "", y = proportions.Freq, fill = labels)) +
  geom_bar(stat = "identity", width = 1, color="white") +
  coord_polar("y", start = 0) +
  #scale_fill_discrete(name = "Labels") +
  labs(title = "Before Corona") +
  geom_text(aes(label = paste0(round(proportions.Freq * 100, 2), "%")), position = position_stack(vjust
  theme_void()+
  theme(legend.position = "none")

pie_plot2 <- ggplot(plot_data2, aes(x = "", y = proportions.Freq, fill = labels)) +
  geom_bar(stat = "identity", width = 1, color="white") +
  coord_polar("y", start = 0) +
  #scale_fill_discrete(name = "Labels") +
  geom_text(aes(label = paste0(round(proportions.Freq * 100, 2), "%")), position = position_stack(vjust
  labs(title = "During Corona") +
  theme_void()+
  theme(legend.position = "none")

pie_plot3 <- ggplot(plot_data3, aes(x = "", y = proportions.Freq, fill = labels)) +
  geom_bar(stat = "identity", width = 1, color="white") +
  coord_polar("y", start = 0) +
  scale_fill_discrete(name = "Labels") +
  geom_text(aes(label = paste0(round(proportions.Freq * 100, 2), "%")), position = position_stack(vjust
  labs(title = "After Corona") +
  theme_void()
  #theme(legend.position = "none")

# Arrange the histograms in subplots
subplot <- grid.arrange(pie_plot,pie_plot2,pie_plot3, nrow = 1,
                        bottom="Proportion of Learning Source")
```



Proportion of Learning Source

```
# Display the subplots
subplot
```

```
## TableGrob (2 x 3) "arrange": 4 grobs
##   z     cells    name                grob
## 1 1 (1-1,1-1) arrange      gtable[layout]
## 2 2 (1-1,2-2) arrange      gtable[layout]
## 3 3 (1-1,3-3) arrange      gtable[layout]
## 4 4 (2-2,1-3) arrange text[GRID.text.282]
```

In this pie chart, we observe a noticeable rise in the demand for external lecture data, such as YouTube videos or online papers. Before COVID, only 17.14% of students primarily consumed external learning material not provided by the university. However, during COVID, this percentage increased significantly to over 37.14%. Even after COVID, a substantial portion, 20%, continued to rely on external sources.

This increase in demand for external material, coupled with the observed improvement in study performance and the perceived time efficiency during the COVID period, suggests that these external resources were more beneficial to students than the lecture materials provided by professors.

Descriptive Inference Summary

Preparation Time

To show the differences of preparation time between before, during and after COVID, blocks (intervals) of <=10, <=20, etc. hours are created. To emphasize the differences between before during and after covid, difference tables are created showing how the behaviour for students changed compared to before COVID.

```
intervals = c(0,10, 20, 30, 40, 50, 60, 70, 80, Inf)

summary_palette = c("white", "darkblue")

cnt_bef <- data_p %>%
       select(test.prep.bef) %>%
        group_by(interval_bef = cut(test.prep.bef, breaks = intervals, right=FALSE)) %>%
        count() %>%
```

```
        spread(interval_bef, n)

cnt_dur <- data_p %>%
        select(test.prep.dur) %>%
        group_by(interval_dur = cut(test.prep.dur, breaks = intervals, right=FALSE)) %>%
        count() %>%
        spread(interval_dur, n)

cnt_af <- data_p %>%
        select(test.prep.af) %>%
        group_by(interval_af = cut(test.prep.af, breaks = intervals, right=FALSE)) %>%
        count() %>%
        spread(interval_af, n)



inferential_summary_preparation <- bind_rows(cnt_bef, cnt_dur, cnt_af) %>%
        mutate(index = c("Before COVID", "During COVID", "After COVID")) %>%
        replace(is.na(.), 0) %>%
        select(index, "[0,10)", "[10,20)", "[20,30)", "[30,40)", "[40,50)", "[50,60)", "[60,70)", "[70,8

inferential_summary_preparation %>%
        gt() %>%
        tab_header(
                title = md("**Preparation Time**"),
                subtitle = md("*Amount of students who prepared for the exam for x hours*")
        ) %>%
        data_color(
                columns = 2:10,
                fn = col_numeric(
                  palette = summary_palette,
                  domain = c(0, 13)
                )
        )
```

Preparation Time

Amount of students who prepared for the exam for x hours

index	[0,10)	[10,20)	[20,30)	[30,40)	[40,50)	[50,60)	[60,70)	[70,80)	[80,Inf)
Before COVID	8	8	7	4	4	1	2	0	1
During COVID	11	13	5	2	2	1	0	0	1
After COVID	8	11	10	0	2	2	0	1	1

The amount of people who prepared for the exam for >= 30 hours decreased during covid considerably from 12 to 6 and stayed down after covid at 6. This could be caused by making videos available during covid, which stayed available also after covid. If the difference is significant will be shown by a hypothesis test.

```
intervals_diff = c(-Inf, -20, -10, -5, 0, 1, 5, 10, 20, Inf)

# Difference in preparation time
prep_diff <- data_p %>%
        select(test.prep.bef, test.prep.dur, test.prep.af) %>%
        mutate(diff_dur = test.prep.dur - test.prep.bef,
```

```
                diff_af = test.prep.af - test.prep.bef) %>%
        select(diff_dur, diff_af)
# prep_diff

cnt_diff_dur <- prep_diff %>%
        select(diff_dur) %>%
        group_by(interval_dur = cut(diff_dur, breaks = intervals_diff, right=FALSE)) %>%
        count() %>%
        spread(interval_dur, n)

cnt_diff_af <- prep_diff %>%
        select(diff_af) %>%
        group_by(interval_af = cut(diff_af, breaks = intervals_diff, right=FALSE)) %>%
        count() %>%
        spread(interval_af, n)


inferential_summary_preparation_diff <- bind_rows(cnt_diff_dur, cnt_diff_af) %>%
        mutate(index = c("Difference During COVID", "Difference After COVID")) %>%
        replace(is.na(.), 0) %>%
        select(index, everything())

inferential_summary_preparation_diff %>%
        gt() %>%
        tab_header(
                title = md("**Preparation Time Difference**"),
                subtitle = md("*Amount of students who prepared x hours more/less for an exam for during
        ) %>%
        data_color(
                columns = 2:9,
                fn = col_numeric(
                  palette = summary_palette,
                  domain = c(0, 17)
                )
        )
```

Preparation Time Difference

Amount of students who prepared x hours more/less for an exam for during and after COVID, base before COVID

index	[-Inf,-20)	[-20,-10)	[-10,-5)	[-5,0)	[0,1)	[5,10)	[10,20)	[20, Inf)
Difference During COVID	5	3	4	8	12	1	2	0
Difference After COVID	2	1	5	7	16	1	2	1

The differences show the already suspected shift of less preparation time during COVID even better. Those benefits seem to have stayed after COVID in a reduced extend for some people, while some have gone back to their old values.

Failure Rates

The Failure Rates show how many ECTS have been failed per semester. The summary tables will show how the failure rates differed during and after COVID compared to before.

```r
# Inferential summary for performance
intervals = c(0, 1, 2, 3, 4, 5, 6, 7, Inf)

summary_performance <- data_p %>%
        select(performance_bef, performance_dur, performance_af) %>%
        mutate(perf_diff_dur = performance_dur - performance_bef,
               perf_diff_af = performance_af - performance_bef)


perf_cnt_bef <- summary_performance %>%
        select(performance_bef) %>%
        group_by(interval_bef = cut(performance_bef, breaks = intervals, right=FALSE)) %>%
        count() %>%
        spread(interval_bef, n)

perf_cnt_dur <- summary_performance %>%
        select(performance_dur) %>%
        group_by(interval_dur = cut(performance_dur, breaks = intervals, right=FALSE)) %>%
        count() %>%
        spread(interval_dur, n)

perf_cnt_af <- summary_performance %>%
        select(performance_af) %>%
        group_by(interval_af = cut(performance_af, breaks = intervals, right=FALSE)) %>%
        count() %>%
        spread(interval_af, n)


inferential_summary_perf <- bind_rows(perf_cnt_bef, perf_cnt_dur, perf_cnt_af) %>%
        mutate(index = c("Before COVID", "During COVID", "After COVID")) %>%
        replace(is.na(.), 0) %>%
        select(index, "[0,1)", "[1,2)", "[2,3)", "[3,4)", "[4,5)", "[5,6)", "[6,7)", "[7,Inf)", everythi


inferential_summary_perf %>%
    gt() %>%
    tab_header(
        title = md("**Failure Rate**"),
        subtitle = md("*Amount of Students who failed x ECTS*")
    ) %>%
    data_color(
        columns = 2:9,
        fn = col_numeric(
          palette = summary_palette,
          domain = c(0, 16)
        )
    )
```

Failure Rate

Amount of Students who failed x ECTS

index	[0,1)	[1,2)	[2,3)	[3,4)	[4,5)	[5,6)	[6,7)	[7,Inf)	<NA>
Before COVID	15	4	0	1	1	0	1	1	12

During COVID	14	8	1	0	1	1	0	0	10
After COVID	13	6	3	1	1	0	1	0	10

Failure rates didn't change much during COVID with a low downward shift of fewer people with zero failures even after COVID. This could be caused by simpler exams during COVID or simply by difficulties getting back to the old presence mode.

```
intervals_diff = c(-Inf, -5, -2, -1, 0, 1, 2, 5, Inf)

perf_cnt_diff_dur <- summary_performance %>%
        select(perf_diff_dur) %>%
        group_by(interval_dur = cut(perf_diff_dur, breaks = intervals_diff, right=FALSE)) %>%
        count() %>%
        spread(interval_dur, n)

perf_cnt_diff_af <- summary_performance %>%
        select(perf_diff_af) %>%
        group_by(interval_af = cut(perf_diff_af, breaks = intervals_diff, right=FALSE)) %>%
        count() %>%
        spread(interval_af, n)

inferential_summary_perf_diff <- bind_rows(perf_cnt_diff_dur, perf_cnt_diff_af) %>%
        replace(is.na(.), 0) %>%
        mutate(index = c("Difference During COVID", "Difference After COVID")) %>%
        select(index, "[-Inf,-5)", "[-5,-2)", everything())

inferential_summary_perf_diff %>%
    gt() %>%
    tab_header(
        title = md("**Failure Rate Difference**"),
        subtitle = md("*Amount of students with x different failed ECTS per semester for during and aft
    ) %>%
    data_color(
        columns = 2:9,
        fn = col_numeric(
          palette = summary_palette,
          domain = c(0, 13)
        )
    )
```

Failure Rate Difference

Amount of students with x different failed ECTS per semester for during and after COVID with base before COVID

index	[-Inf,-5)	[-5,-2)	[-2,-1)	[-1,0)	[0,1)	[1,2)	[2,5)	<NA>
Difference During COVID	3	0	2	3	8	5	1	13
Difference After COVID	2	1	2	4	9	0	5	12

The Failure Rate difference additionally show a relatively high fluctuation of failed ECTS, it seems that some students had difficulties adapting to the new situation, while others found it easier to study during COVID.

Learning Sources

A summary for the main learning sources will be shown and how they changed during and after COVID compared to before.

```
# Inferential summary for learning sources
summary_learningsources <- data_p %>%
        select(source.bef, source.dur, source.af)

source_cnt_bef <- summary_learningsources %>%
        select(source.bef) %>%
        group_by(source.bef) %>%
        count() %>%
        spread(source.bef, n)

source_cnt_dur <- summary_learningsources %>%
        select(source.dur) %>%
        group_by(source.dur) %>%
        count() %>%
        spread(source.dur, n)

source_cnt_af <- summary_learningsources %>%
        select(source.af) %>%
        group_by(source.af) %>%
        count() %>%
        spread(source.af, n)

inferential_summary_sources <- bind_rows(source_cnt_bef, source_cnt_dur, source_cnt_af) %>%
        mutate(index = c("Before COVID", "During COVID", "After COVID")) %>%
        select(index, External, Internal)

inferential_summary_sources %>%
    gt() %>%
        tab_header(
                title = md("**Learning Sources**"),
                subtitle = md("*Amount of students who used x learning source*")
        ) %>%
        data_color(
                columns = 2:3,
                fn = col_numeric(
                  palette = summary_palette,
                  domain = c(0, 30)
                )
        )
```

Learning Sources

Amount of students who used x learning source

index	External	Internal
Before COVID	6	29
During COVID	13	22
After COVID	7	28

As seen in the table the learning sources seem to have shifted from to more external sources during COVID,

going back to the old distribution after COVID. This could be caused by the lack of presence lectures and the need to study more on your own, as internal material supplied by the university (lectures, etc.) was missing.

```
sources_diff_dur <- summary_learningsources %>%
        group_by(source.bef, source.dur) %>%
        count() %>%
        mutate(from = source.bef,
               to = source.dur,
               during = n) %>%
        ungroup() %>%
        select(from, to, during)

sources_diff_af <- summary_learningsources %>%
        group_by(source.bef, source.af) %>%
        count() %>%
        mutate(from = source.bef,
               to = source.af,
               after = n) %>%
        ungroup() %>%
        select(from, to, after)

inferential_summary_sources_diff<- bind_cols(sources_diff_dur %>% select(during), sources_diff_af) %>%
        select(from, to, during, after)

inferential_summary_sources_diff %>%
    gt() %>%
        tab_header(
                title = md("**Learning Sources Difference**"),
                subtitle = md("*Amount of students who switched learning source during and after COVID w
        ) %>%
        data_color(
                columns = 3:4,
                fn = col_numeric(
                  palette = summary_palette,
                  domain = c(0, 25)
                )
        )
```

Learning Sources Difference

Amount of students who switched learning source during and after COVID with base before COVID

from	to	during	after
External	External	5	3
External	Internal	1	3
Internal	External	8	4
Internal	Internal	21	25

As already expected by the last table students either stayed on external learning sources, if they already preferred them pre COVID or were likely to switch to external sources during COVID otherwise with 8 students switching from internal sources to external ones compared to 1 student switching from external sources to internal ones during COVID. After COVID the situation normalized itself partly with some students staying on external sources.

Inference

Prep time

```
shapiro.test(data_p$test.prep.bef)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_p$test.prep.bef
## W = 0.43916, p-value = 1.984e-10
```

```
shapiro.test(data_p$test.prep.af)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_p$test.prep.af
## W = 0.40789, p-value = 9.543e-11
```

```
shapiro.test(data_p$test.prep.dur)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_p$test.prep.dur
## W = 0.35848, p-value = 3.163e-11
```

The tests tell us that none of the periods is normally distributed which we should keep in mind when choosing, what kind of test we should use. We can use the Kruskal-Wallis test to check for significant changes in medians between the three groups.

In the analysis of multiple groups, it's imperative to control for the multiple comparisons problem. This statistical issue arises when multiple pairwise tests are conducted simultaneously. With an increased number of tests, there's an elevated probability of a Type I error – falsely identifying a significant difference when there isn't one.

The Kruskal-Wallis test offers an effective solution to this problem. Rather than making pairwise comparisons, this test considers all groups simultaneously and evaluates the overall null hypothesis: the assertion that all groups originate from the same population, or equivalently, their medians are identical.

The primary advantage of the Kruskal-Wallis test is that it provides a holistic view of the dataset and determines if there's a significant difference in the medians of three or more independent groups. This approach avoids inflating the Type I error rate, as it's a single test. If the Kruskal-Wallis test result is significant, it implies that at least one group diverges from the others. Subsequent pairwise comparisons can then be conducted to identify specifically which groups exhibit a significant difference.

Applying the Kruskal-Wallis test before performing pairwise comparisons ensures that we only delve into the specifics if there's substantial evidence of a difference among the groups. This stepwise procedure effectively controls the Type I error rate and mitigates the risks associated with the multiple comparisons problem.

```
kruskal.test(list(data_p$test.prep.bef, data_p$test.prep.dur, data_p$test.prep.af))
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  list(data_p$test.prep.bef, data_p$test.prep.dur, data_p$test.prep.af)
## Kruskal-Wallis chi-squared = 3.3482, df = 2, p-value = 0.1875
```

Based on the Kruskal-Wallis test we conclude that there is no significant shift in the medians between the groups. So there are no significant differences between study times for each period

Performance

```
data_p_noinf <- subset(data_p, !is.infinite(data_p[, "performance_dur"]))
data_p_noinf <- subset(data_p_noinf, !is.infinite(data_p_noinf[, "performance_bef"]))
data_p_noinf <- subset(data_p_noinf, !is.infinite(data_p_noinf[, "performance_af"]))
data_p_noinf <- data_p_noinf[complete.cases(data_p_noinf[, c("performance_bef", "performance_dur", "per
```

```
shapiro.test(data_p_noinf$performance_dur)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_p_noinf$performance_dur
## W = 0.73784, p-value = 0.0001172
```

```
shapiro.test(data_p_noinf$performance_bef)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_p_noinf$performance_bef
## W = 0.63765, p-value = 7.418e-06
```

```
shapiro.test(data_p_noinf$performance_af)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_p_noinf$performance_af
## W = 0.80362, p-value = 0.0009789
```

Here we also see data that is not normally distributed so we can use the kruskal wallis test again.

```
kruskal.test(list(data_p_noinf$performance_bef, data_p_noinf$performance_af, data_p_noinf$performance_du
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  list(data_p_noinf$performance_bef, data_p_noinf$performance_af, data_p_noinf$performance_dur)
## Kruskal-Wallis chi-squared = 0.20625, df = 2, p-value = 0.902
```

The kruskal wallis rejects the hypothesis that there is a difference between the time periods (before, during, after covid).

Materials used

```
data_materials <- data_p %>%
    pivot_longer(
        cols = c("source.dur", "source.bef", "source.af"),
        names_to = "period",
        values_to = "source"
    )
```

```r
contingency_table <- table(data_materials$period, data_materials$source)
chisq.test(contingency_table)
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table
## X-squared = 4.3963, df = 2, p-value = 0.111
```

The p-value suggests that there is no association between the period and sources used for studying.