Survival_Analysis_GROUP 1

Diana Nduku - 665419, Chesia Anyika - 665567, Zak

2024-06-08

```
#libraries used
library(readxl)
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.0     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.5.0     ✔ tibble    3.2.1
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force all conflicts to become errors
```

```
library(survival)
library(summarytools)
```

```
##
## Attaching package: 'summarytools'
##
## The following object is masked from 'package:tibble':
##
##     view
```

```
library(ggfortify)
library(survival)
library(survminer)
```

```
## Loading required package: ggpubr
##
## Attaching package: 'survminer'
##
## The following object is masked from 'package:survival':
##
##     myeloma
```

```
st_options(use.x11 = FALSE)
```

Question 1

The “Patient Data” dataset requires re-coding of several columns to enhance the analysis of patient responses. The columns to be modified include `Pleasure_Doingthings`, `Depressed`, `Sleep`, `Energy`, `Appetite`, `Bad_About_Myself`, `Concentration`, `Speak_Slowly`, and `Thoughts`. These will be re-coded to quantify the level of difficulty patients experience as follows:

- 0 for *Not difficult*
- 1 for *Somewhat difficult*
- 2 for *Very difficult*
- 3 for *Extremely difficult.*

This systematic re-coding aims to standardize the entries, thereby simplifying the data analysis process and ensuring consistency across the data-set.

1.1 Data Overview

Load the “Patient Data” data-set and examine the structure of key columns for our analysis.

```
#import the dataset
library(readxl)
patient_data <- read_excel("Patient Data.xlsx")
```

```
## New names:
## • `` -> `...8`
## • `Current_Condition` -> `Current_Condition...18`
## • `Current_Condition` -> `Current_Condition...65`
## • `Kidney_Condition` -> `Kidney_Condition...72`
## • `Kidney_Condition` -> `Kidney_Condition...107`
```

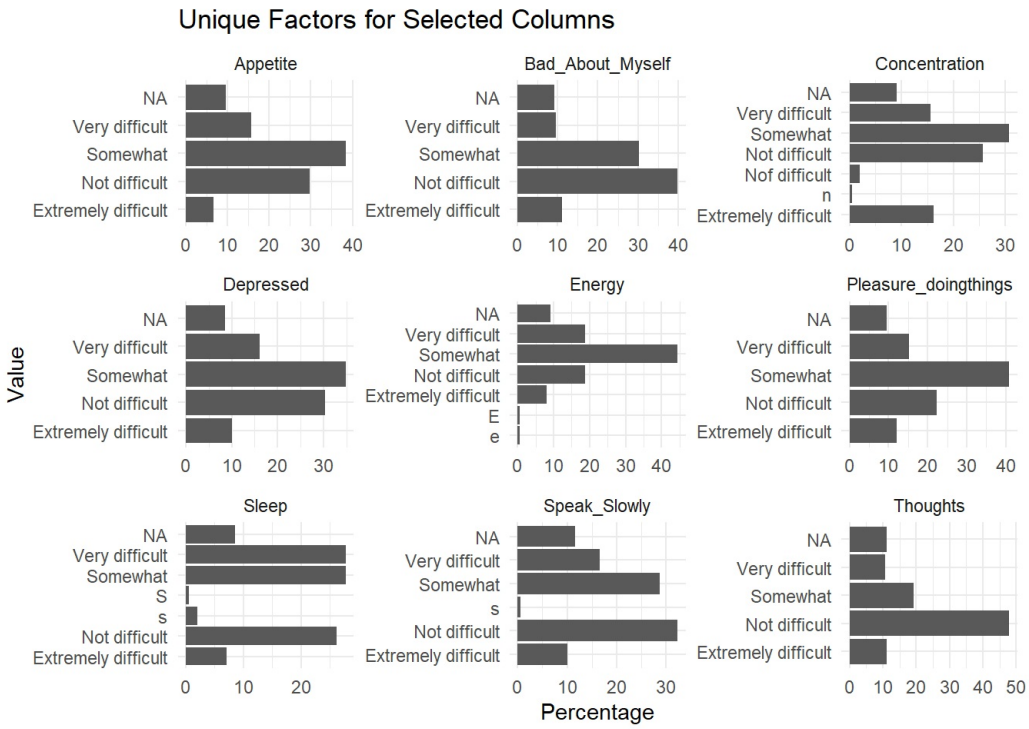
```
#subset the data and view the results
mental_health <- patient_data %>% select('Pleasure_doingthings',
                                         'Depressed',
                                         'Sleep',
                                         'Energy',
                                         'Appetite',
                                         'Bad_About_Myself',
                                         'Concentration',
                                         'Speak_Slowly',
                                         'Thoughts'
                                         ); mental_health
```

```
## # A tibble: 198 × 9
##    Pleasure_doingthings Depressed         Sleep Energy Appetite Bad_About_Myself
##    <chr>                <chr>             <chr> <chr>  <chr>    <chr>
##  1 Somewhat             Not difficult     Some… Somew… Not dif… Not difficult
##  2 Very difficult       Not difficult     Not … Not d… Somewhat Not difficult
##  3 Extremely difficult  Very difficult    Not … Extre… Not dif… Not difficult
##  4 Somewhat             Not difficult     Very… Very … Somewhat Not difficult
##  5 Somewhat             Not difficult     Very… Somew… Somewhat Very difficult
##  6 Extremely difficult  Somewhat          Very… Extre… Somewhat Very difficult
##  7 Somewhat             Extremely diffic… Very… Very … Somewhat Somewhat
##  8 Extremely difficult  Not difficult     Not … Not d… Extreme… Not difficult
##  9 Somewhat             Not difficult     Some… Somew… Not dif… Not difficult
## 10 Not difficult        Not difficult     Some… Somew… Somewhat Not difficult
## # i 188 more rows
## # i 3 more variables: Concentration <chr>, Speak_Slowly <chr>, Thoughts <chr>
```

We then found the unique factors of the columns of interest and visualized them using bar plots.

```
##TRANSFORM DATA-FRAME FOR PLOTTING
#create dataframe in long format
long_data <- mental_health %>%
  pivot_longer(cols = everything(), names_to = "Variable", values_to = "Value")

##PLOT BARPLOTS
#plot barplots
ggplot(long_data, aes(x = Value, y = (..count..)/sum(..count..))) +
  geom_bar(aes(y = ..prop.. * 100, group = 1), stat = "count") +
  facet_wrap(~ Variable, scales = "free") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1)) +
  labs(title = "Unique Factors for Selected Columns",
       x = "Value",
       y = "Percentage") +
  theme_minimal() + coord_flip()
```

```
## Warning: The dot-dot notation (`..prop..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(prop)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Interpretation:

The plot reveals data entry errors in several mental health variables, where unexpected values such as `e`, `E`, `n`, `s`, and `S` appear. Specifically, the "Energy" variable contains `e` and `E`, "Speak_Slowly" contains `s` and `S`, and "Concentration" contains `n`. These values likely result from typos or coding mistakes and should be corrected to align with existing categories or treated as missing data to ensure accurate analysis and interpretation.

1.2 Data Preprocessing

To address the data entry errors identified in the mental health variables, we implement a cleaning process by replacing the erroneous values with their correct categories.

The bar plots showed that the variables were not standardized and contained not only the four expected categories—Not difficult, Somewhat, Very difficult, and Extremely difficult—but also single-letter responses like 'n', 's', and 'e', as well as NA entries. To standardize the data, we recoded these values as follows:

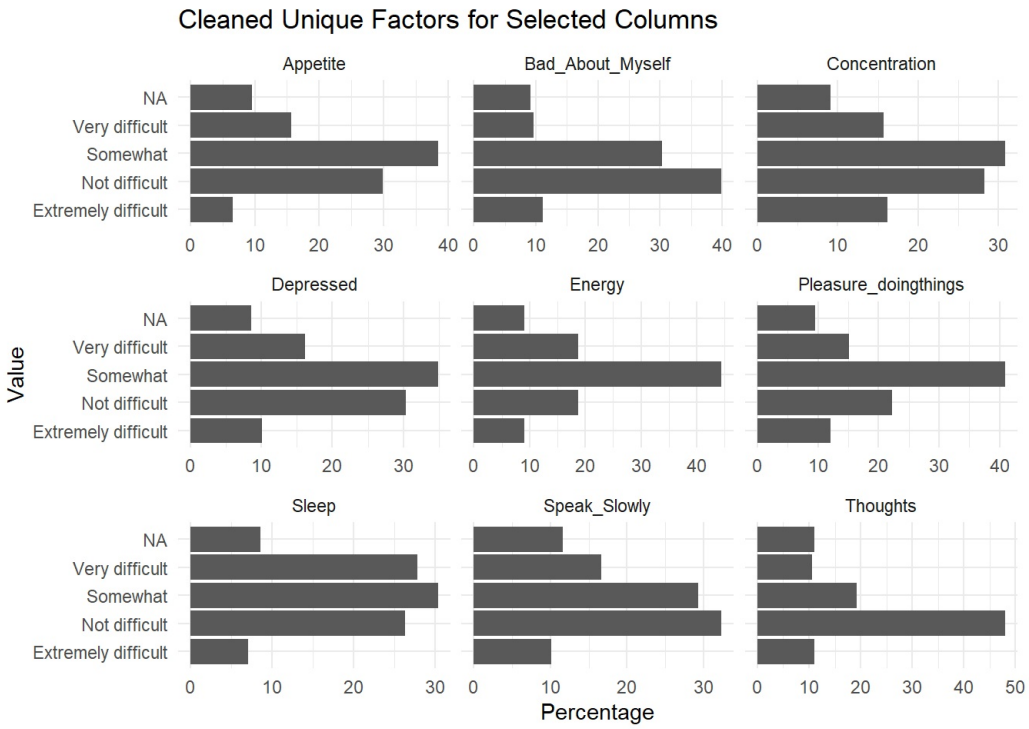
- `s` and `S` were recoded as "Somewhat"
- `e` and `E` were recoded as "Extremely difficult"
- `n` and `Nof difficult` were recoded as "Not difficult"
- NA values were left unchanged

This cleaning process ensured that the dataset was consistent and ready for accurate analysis.

```
#recode erroneous entries
mental_health<- mental_health %>%
  mutate(across(everything(), ~recode(.,
                                      'e' = "Extremely difficult",
                                      'E' = "Extremely difficult",
                                      'n' = "Not difficult",
                                      'Nof difficult' = 'Not difficult',
                                      's' = "Somewhat",
                                      'S' = "Somewhat")))
```

After implementing the data cleaning process, the updated bar plots no longer contain the previously identified errors. The variables are now contain only the expected categories: Not difficult, Somewhat, Very difficult, and Extremely difficult. Additionally, NA values remain unchanged

```
#create dataframe in long format
long_data <- mental_health %>%
  pivot_longer(cols = everything(), names_to = "Variable", values_to = "Value")

#plot barplots
ggplot(long_data, aes(x = Value, y = (..count..)/sum(..count..))) +
  geom_bar(aes(y = ..prop.. * 100, group = 1), stat = "count") +
  facet_wrap(~ Variable, scales = "free_x") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1)) +
  labs(title = "Cleaned Unique Factors for Selected Columns",
       x = "Value",
       y = "Percentage") +
  theme_minimal() + coord_flip()
```



We recoded categorical responses into numerical values to facilitate numerical analysis. Specifically, we converted the responses as follows: `Not difficult` to 0, `Somewhat` to 1, `Very difficult` to 2, and `Extremely difficult` to 3. Any other values were set to NA to ensure the columns remain numeric.

```
#recode variables
mental_health_recoded <- mental_health %>%
  mutate(across(everything(), ~ case_when(
    . == 'Not difficult' ~ 0,
    . == 'Somewhat' ~ 1,
    . == 'Very difficult' ~ 2,
    . == 'Extremely difficult' ~ 3,
    TRUE ~ NA_real_  # Ensure any other values are set to NA
  )))

#view results
head(mental_health_recoded)
```

```
## # A tibble: 6 × 9
##   Pleasure_doingthings Depressed Sleep Energy Appetite Bad_About_Myself
##                  <dbl>     <dbl> <dbl>  <dbl>    <dbl>            <dbl>
## 1                    1         0     1      1        0                0
## 2                    2         0     0      0        1                0
## 3                    3         2     0      3        0                0
## 4                    1         0     2      2        1                0
## 5                    1         0     2      1        1                2
## 6                    3         1     2      3        1                2
## # ℹ 3 more variables: Concentration <dbl>, Speak_Slowly <dbl>, Thoughts <dbl>
```

1.2.1 Missing Data Proportion

To better understand the quality and completeness of the data-set, we calculated the number and proportion of missing values for each variable. This information is crucial for deciding how to handle missing data in subsequent analyses. For instance, columns with higher proportions of missing data, such as **Speak_Slowly** (11.616%) and **Thoughts** (11.111%), may require special attention, such as data imputation.

```
#compute count and proportion of missing values
count_missing_values <- function(data) {
  missing_counts <- sapply(data, function(x) sum(is.na(x)))
  missing_proportion <- sapply(data, function(x) round(mean(is.na(x)) * 100, 3))
  result <- data.frame(
    colname = names(missing_counts),
    missing_count = missing_counts,
    missing_proportion = missing_proportion,
    row.names = NULL
  )
  return(result)
}

#view results
count_missing_values(mental_health_recoded)
```

```
##                colname missing_count missing_proportion
## 1 Pleasure_doingthings            19              9.596
## 2            Depressed            17              8.586
## 3                Sleep            17              8.586
## 4               Energy            18              9.091
## 5             Appetite            19              9.596
## 6     Bad_About_Myself            18              9.091
## 7        Concentration            18              9.091
## 8         Speak_Slowly            23             11.616
## 9             Thoughts            22             11.111
```

Imputation

After identifying the missing values in the data-set, we proceeded to impute these missing values with the mode of their respective columns. This method was chosen because it efficiently preserves the categorical nature of the data by replacing missing values with the most frequently occurring category, ensuring the integrity of the variable's distribution and avoiding the introduction of bias that could arise from more complex imputation techniques.

The table confirms that all columns now have zero missing values, with both the count and proportion of missing values reduced to 0. This ensures that the data-set is complete and ready for further analysis without the need for additional handling of missing data

```
# Define a function to impute missing values with the mode
impute_mode <- function(x) {
  mode_value <- names(sort(table(x), decreasing = TRUE))[1]
  x[is.na(x)] <- mode_value
  return(x)
}

# Apply the function across columns - ensure result is a numeric dataframe
mental_health_imputed <- lapply(mental_health_recoded, impute_mode)
mental_health_imputed <- as.data.frame(lapply(mental_health_imputed, as.numeric))

#view results
count_missing_values(mental_health_imputed)
```

```
##                colname missing_count missing_proportion
## 1 Pleasure_doingthings             0                  0
## 2            Depressed             0                  0
## 3                Sleep             0                  0
## 4               Energy             0                  0
## 5             Appetite             0                  0
## 6     Bad_About_Myself             0                  0
## 7        Concentration             0                  0
## 8         Speak_Slowly             0                  0
## 9             Thoughts             0                  0
```

1.3 Data Transformation

1.3.1 Introduce a sum Variable

To enhance the analysis of the data-set, we introduced a new variable, `sum_coded` , representing the sum of the scores from seven mental health variables. This sum variable provides an overall score for each patient's mental health evaluation. We categorized these scores into different levels of depression severity based on the following value ranges:

Value Range	Factor
0-4	None-Minimal
5-9	Mild depression
10-14	Moderate depression
15-19	Moderately Severe
20-27	Severe depression

```
#compute rowsums and create new variable
row_sum <-  rowSums(mental_health_imputed)
mental_health <- mental_health_imputed %>% mutate(sum_coded = row_sum)

#view results
mental_health$sum_coded
```

```
##   [1]  5  6 14  8  7 18 16  6 11  4 17 19 16  6  7  3  6 18  9  2 18 11 25  8 21
##  [26] 22 17 27 14  4 17  6  5  6  5 15 22 24  0 16 22 11 27  0 10 10  5  4  5  7
##  [51] 19 10  1 10  4 11  3  4  5 10 10  8  0  1  4 10  9  2  4  9 10  0  7  7  8
##  [76]  4  3  6  5 19 10  8 10  7  7  8 12 10 10 18  4  6  8  5 10  4  8  6  2  3
## [101] 10 15  0 14  9  0  7  3  0  7  8  7 16 11 19  9  9  6  5  5 21 10 12 14 19
## [126]  4  6 18  8 17  8  5  7  1  4  0  4  7  4 12 14  5  7 15  9 16  8 12 14 12
## [151] 14  6  4  8  6  6  6  6 15 10  6  0 15 19  6  5 24  0  7  2 10 15 12 23 27
## [176]  3 12  7  0 12  7  4 18 11  4 17  0 27 10  6  6  6  6  6  6  6  6  6
```

1.3.2 Categorizing Sum Variable

We categorize the `sum_coded` variable into different levels of depression severity.

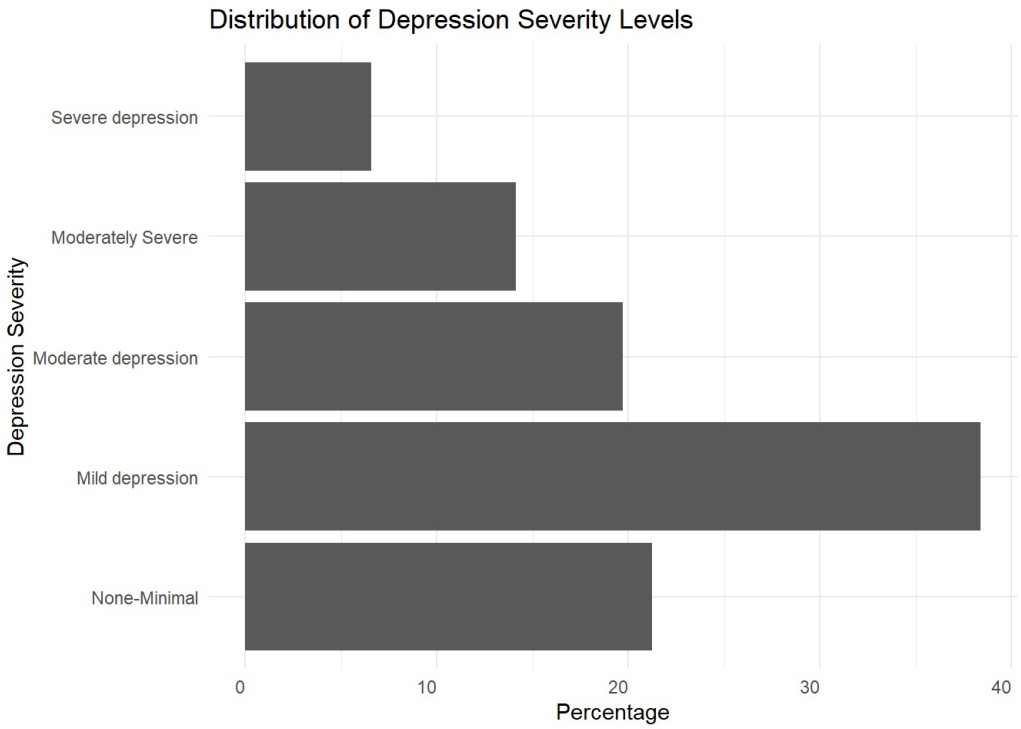
```
#categorize sum variable
mental_health <- mental_health %>%
  mutate(depression_severity = case_when(
    sum_coded >= 0 & sum_coded <= 4 ~ "None-Minimal",
    sum_coded >= 5 & sum_coded <= 9 ~ "Mild depression",
    sum_coded >= 10 & sum_coded <= 14 ~ "Moderate depression",
    sum_coded >= 15 & sum_coded <= 19 ~ "Moderately Severe",
    sum_coded >= 20 & sum_coded <= 27 ~ "Severe depression"
  ))

#convert to factor variable and specify levels
mental_health$depression_severity <- factor(mental_health$depression_severity, levels = c("None-Minimal", "Mild d
epression", "Moderate depression", "Moderately Severe", "Severe depression"))

#view results
head(mental_health %>% select(sum_coded, depression_severity))
```

```
##   sum_coded depression_severity
## 1         5     Mild depression
## 2         6     Mild depression
## 3        14 Moderate depression
## 4         8     Mild depression
## 5         7     Mild depression
## 6        18   Moderately Severe
```

1.3.3 Distribution of Depression Severity Levels

The bar plot reveals that the majority of individuals fall into the *None-Minimal and Mild depression categories, indicating that most have little to no depression symptoms*. Moderate depression is also relatively common, while Moderately Severe and Severe depression are less frequent.

```
#visualise barplot
ggplot(mental_health, aes(x = depression_severity, y = (..count..)/sum(..count..))) +
  geom_bar(aes(y = ..prop.. * 100, group = 1), stat = "count") +
  theme_minimal() +
  labs(title = "Distribution of Depression Severity Levels",
       x = "Depression Severity",
       y = "Percentage") +
  theme(axis.text.x = element_text(hjust = 1)) + coord_flip()
```



1.3.4 Comparison of Depression Severity with Residential Status

We then compared the `depression_severity` variable to the `Facility_type` variable. The `Facility_type` variable has two responses, which we will re-code as follows:

- `Resident` as In-patient
- `Non Resident` as Homecare

We first prepared the `Facility_Type` variable for analysis by examining its unique values and standardizing the responses to remove any errors. We found that the `Facility_Type` variable has the following unique values:

- `Resident`
- `Non Resident`
- `non Resident`
- `NA`

```
#Add Facility_Type Variable
mental_health$Facility_Type <- patient_data$Facilty_Type

#View unique values
unique(mental_health$Facility_Type)
```

```
## [1] "Non Resident" "non Resident" "Resident"     NA
```

We converted all `non Resident` responses to `Non Resident` , for standardization, and recoded the variables.

```
#Standardise responses
mental_health <- mental_health %>%
  mutate(Facility_Type = ifelse(Facility_Type == "non Resident", "Non Resident", Facility_Type))

#recode responses
mental_health <- mental_health %>%
  mutate(Facility_Type = recode(Facility_Type,
                                "Non Resident" = "Homecare",
                                "Resident" = "In-patient"))

#View results
head(mental_health$Facility_Type)
```

```
## [1] "Homecare" "Homecare" "Homecare" "Homecare" "Homecare" "Homecare"
```

We then ran a chi-square test to determine whether there is a statistically significant relationship between the two variables. We used simulated p-values due to low value counts. The p-value obtained is 0.02249 is less than the conventional significance level of 0.05. Thus there is a statistically significant association between depression severity and residential status.

```
#omitting the NA values
mental_health_na_omitted <- na.omit(mental_health)

#create table
table(mental_health_na_omitted$depression_severity,
          mental_health_na_omitted$Facility_Type)
```

```
##
##                       Homecare In-patient
##   None-Minimal              33          9
##   Mild depression           62          5
##   Moderate depression       33          6
##   Moderately Severe         28          0
##   Severe depression         13          0
```

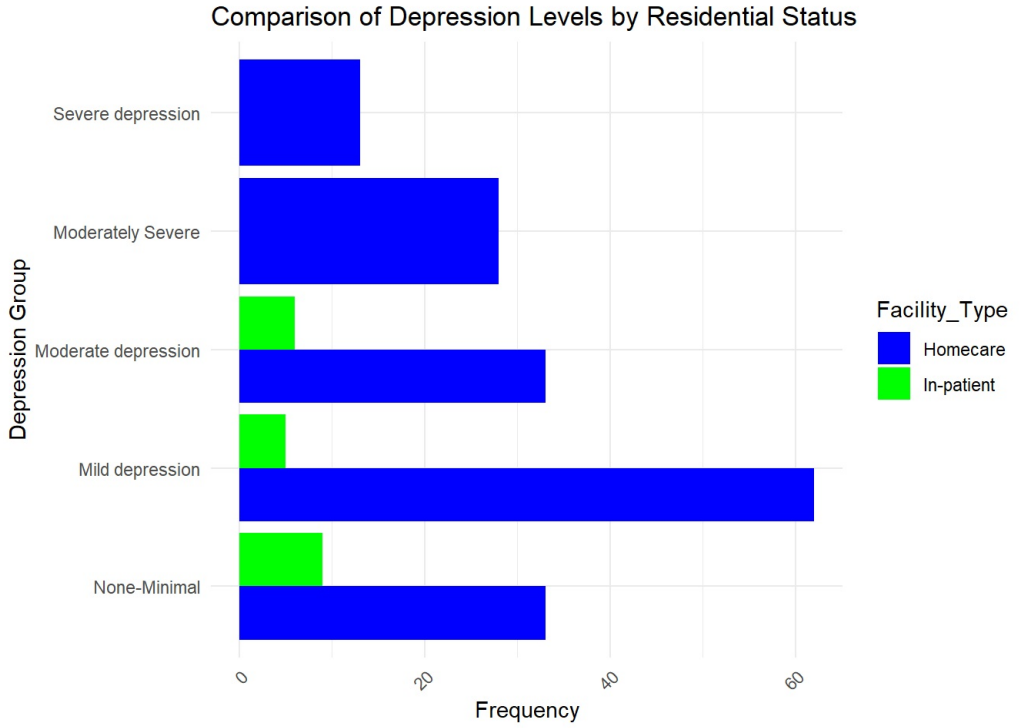
```
#chi-square test
chisq.test(mental_health_na_omitted$depression_severity,
          mental_health_na_omitted$Facility_Type, simulate.p.value = TRUE, B = 2000)
```

```
##
##  Pearson's Chi-squared test with simulated p-value (based on 2000
##  replicates)
##
## data:  mental_health_na_omitted$depression_severity and mental_health_na_omitted$Facility_Type
## X-squared = 11.714, df = NA, p-value = 0.01749
```

We then plotted a side-by-side bar-plot of depression severity levels in relation to residential status of the patients. The bar-plot shows that In-patients have non-minimal to moderate depression, with most In-patients having non-minimal depression. Homecare patients range from non-minimal to severe depression, with most homecare patients having mild depression. Thus *most patients regardless of residential status tend to have lower levels of depression.*

When comparing the two categories, homecare patients entirely occupy the moderately severe and severe depression categories, *suggesting that homecare patients have more severe depression levels than Inpatients.*

```
#plot barplot
ggplot(mental_health_na_omitted, aes(x = depression_severity, fill = Facility_Type)) +
  geom_bar(position = "dodge") +
  labs(title = "Comparison of Depression Levels by Residential Status",
       x = "Depression Group",
       y = "Frequency") +
  scale_fill_manual(values = c("Homecare" = "blue", "In-patient" = "green")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  coord_flip()
```



1.4 Survival Analysis

1.4.1 Preprocess

Let us examine the unique values in the `Date_Ended` column in the `patient_data` data-frame.

```
#view unique values
unique(patient_data$Date_Ended)
```

```
##  [1] "Ongoing"   "45410"     "45383"     "45317"     NA          "45627"
##  [7] "45443"     "45322"     "45323"     "45419"     "45333"     "45376"
## [13] "45065"     "45174"     "44805"     "45257"     "45138"     "45200"
## [19] "45347"     "45214"     "44682"     "45170"     "45386"     "45375"
## [25] "45290"     "45275"     "45371"     "45270"     "45427"     "45381"
## [31] "44972"     "44910"     "45153"     "45122"     "45031"     "44941"
## [37] "45415"     "45421"     "45428"     "45378"     "45434"     "45422"
## [43] "45413"     "45395"     "45394"     "45245"     "04/12/923" "45295"
## [49] "45366"     "45411"     "45426"     "45393"
```

The dates are encoded in the Excel Serial Number Format for dates, which requires us to change the dates to from an integer to the `POSIXct` date-time format. Furthermore, there is an 'Ongoing' entry, which represents the date `04/06/2024` . There is also an error `04/12/923` which should be converted to an NA value.

```
#Convert all 'ongoing' to appropriate date
patient_data$Date_Ended <- ifelse(patient_data$Date_Ended == 'Ongoing', "2024-06-04", patient_data$Date_Ended)

#convert error to an NA value
patient_data$Date_Ended <- ifelse(patient_data$Date_Ended == '04/12/923', NA, patient_data$Date_Ended)

#convert dates to POSIXct
date <- patient_data$Date_Ended
date_b = date[!is.na(date) & date != "2024-06-04"]
date[!is.na(date) & date != "2024-06-04"] = as.character(as.Date(as.numeric(date_b), origin = "1899-12-30"))

#add dates to original dataframe, view results
patient_data$Date_Ended <- date
patient_data$Date_Ended <- as.POSIXct(patient_data$Date_Ended)

unique(patient_data$Date_Ended)
```

```
##  [1] "2024-06-04 EAT" "2024-04-28 EAT" "2024-04-01 EAT" "2024-01-26 EAT"
##  [5] NA               "2024-12-01 EAT" "2024-05-31 EAT" "2024-01-31 EAT"
##  [9] "2024-02-01 EAT" "2024-05-07 EAT" "2024-02-11 EAT" "2024-03-25 EAT"
## [13] "2023-05-19 EAT" "2023-09-05 EAT" "2022-09-01 EAT" "2023-11-27 EAT"
## [17] "2023-07-31 EAT" "2023-10-01 EAT" "2024-02-25 EAT" "2023-10-15 EAT"
## [21] "2022-05-01 EAT" "2023-09-01 EAT" "2024-04-04 EAT" "2024-03-24 EAT"
## [25] "2023-12-30 EAT" "2023-12-15 EAT" "2024-03-20 EAT" "2023-12-10 EAT"
## [29] "2024-05-15 EAT" "2024-03-30 EAT" "2023-02-15 EAT" "2022-12-15 EAT"
## [33] "2023-08-15 EAT" "2023-07-15 EAT" "2023-04-15 EAT" "2023-01-15 EAT"
## [37] "2024-05-03 EAT" "2024-05-09 EAT" "2024-05-16 EAT" "2024-03-27 EAT"
## [41] "2024-05-22 EAT" "2024-05-10 EAT" "2024-05-01 EAT" "2024-04-13 EAT"
## [45] "2024-04-12 EAT" "2023-11-15 EAT" "2024-01-04 EAT" "2024-03-15 EAT"
## [49] "2024-04-29 EAT" "2024-05-14 EAT" "2024-04-11 EAT"
```

Let us also ensure that the `Date_Started` variable is in the same `POSIXct` format as the `Date_Ended` variable.

```
#view format of date_started variable
class(patient_data$Date_Started)
```

```
## [1] "POSIXct" "POSIXt"
```

Let us then rename an unspecified column to `Event` for clarity, and selects key columns `Date_Started` , `Date_Ended` , `Event` , `Marital` , `Length_Separated` etc to streamline the data-set for survival analysis.

```
#rename ...8 column to event
patient_data <- patient_data %>% rename(Event = '...8')

#create subset of variables of interest
subset_data <- patient_data %>% select(Date_Started, Date_Ended, Event, Marital, Length_Separated, Reason_Caregiv
er)

subset_data$Facility_Type <- mental_health$Facility_Type ; subset_data
```

```
## # A tibble: 198 × 7
##    Date_Started        Date_Ended          Event Marital Length_Separated
##    <dttm>              <dttm>              <chr> <chr>   <chr>
##  1 2024-04-16 00:00:00 2024-06-04 00:00:00 <NA>  Widowed More than 5 years
##  2 2022-12-06 00:00:00 2024-06-04 00:00:00 <NA>  Widowed More than 5 years
##  3 2024-01-29 00:00:00 2024-06-04 00:00:00 <NA>  Widowed More than 5 years
##  4 2024-04-01 00:00:00 2024-04-28 00:00:00 Died  Married NA
##  5 2024-04-01 00:00:00 2024-06-04 00:00:00 <NA>  <NA>    <NA>
##  6 2024-03-01 00:00:00 2024-06-04 00:00:00 <NA>  Widowed More than 5 years
##  7 2024-03-01 00:00:00 2024-06-04 00:00:00 <NA>  Married <NA>
##  8 2024-01-01 00:00:00 2024-06-04 00:00:00 <NA>  Married <NA>
##  9 2024-04-01 00:00:00 2024-06-04 00:00:00 <NA>  Married <NA>
## 10 2023-12-01 00:00:00 2024-06-04 00:00:00 <NA>  Married <NA>
## # i 188 more rows
## # i 2 more variables: Reason_Caregiver <chr>, Facility_Type <chr>
```

Add new columns to indicate the status of an event (1 for `Ended` , 0 otherwise) and calculates the survival time in days between `Date_Started` and `Date_Ended` .

```
#add status column
subset_data <- subset_data %>%
  mutate(status = ifelse(Event == "Died", 1, 0),
         survival_time = as.numeric(difftime(Date_Ended, Date_Started, units = "days"))) %>%
  mutate(status = ifelse(is.na(status), 0, status))

#view results
subset_data[7:8]
```

```
## # A tibble: 198 × 2
##    Facility_Type status
##    <chr>          <dbl>
##  1 Homecare           0
##  2 Homecare           0
##  3 Homecare           0
##  4 Homecare           1
##  5 Homecare           0
##  6 Homecare           0
##  7 Homecare           0
##  8 Homecare           0
##  9 Homecare           0
## 10 Homecare           0
## # i 188 more rows
```

We then checked for any negative survival times, and found four columns with negative survival times.

```
#check for negative survival times
neg.times <- subset_data[subset_data$survival_time < 0 & !is.na(subset_data$survival_time), ]

#view results
neg.times[8]
```

```
## # A tibble: 4 × 1
##   status
##    <dbl>
## 1      0
## 2      0
## 3      1
## 4      0
```

We dealt with these values by getting the absolute values of these entries, assuming an error in inputting the beginning and ending dates. There are no more negative values.

```
# Convert negative values to positive equivalents, excluding NA values
subset_data$survival_time <- ifelse(!is.na(subset_data$survival_time) & subset_data$survival_time < 0, abs(subset
_data$survival_time),
                                    subset_data$survival_time)

#check for negative values
subset_data[subset_data$survival_time < 0 & !is.na(subset_data$survival_time), ]#
```

```
## # A tibble: 0 × 9
## # i 9 variables: Date_Started <dttm>, Date_Ended <dttm>, Event <chr>,
## #   Marital <chr>, Length_Separated <chr>, Reason_Caregiver <chr>,
## #   Facility_Type <chr>, status <dbl>, survival_time <dbl>
```

1.4.2 Compute Survival Function (Kaplan Meier)

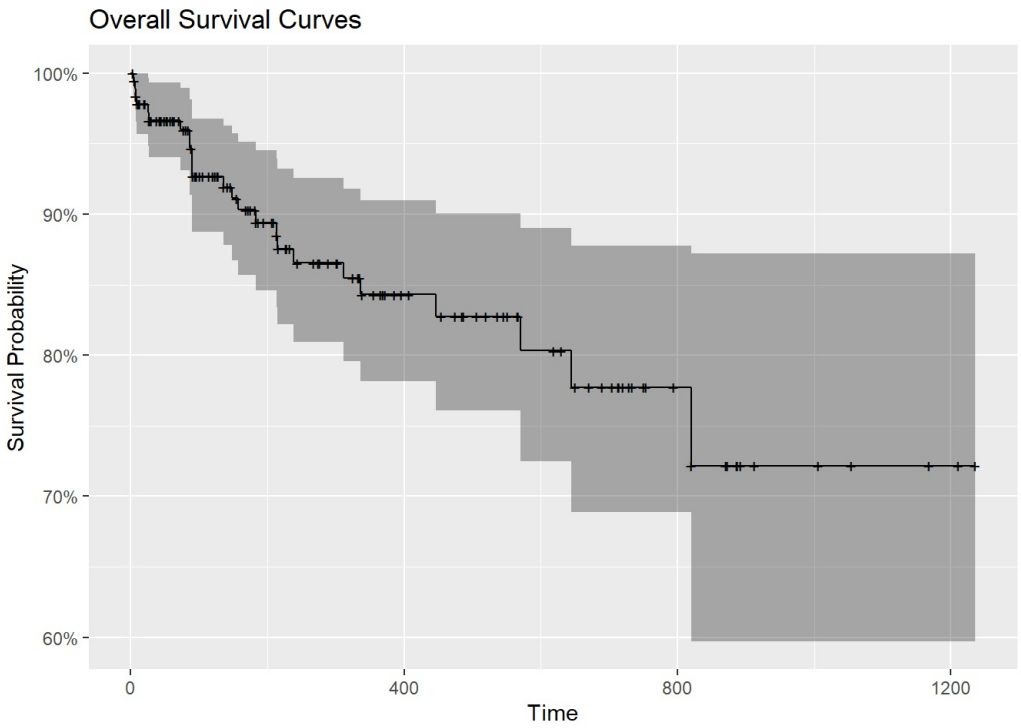
Overall Survival Curve

The survival curve from the data-set shows that survival probability starts at 100% and experiences significant declines early on, indicating a higher event rate at the beginning. As time progresses beyond approximately 200 days, the curve stabilizes, suggesting fewer events occur as time advances. The presence of markers along the curve indicates censored data, where some events have not been observed by the study's end. This pattern suggests an initial vulnerability period followed by a plateau in event likelihood, typical in survival analyses used to understand event timing and risk factors in various fields.

```
table(subset_data$status)
```

```
##
##   0   1
## 173  25
```

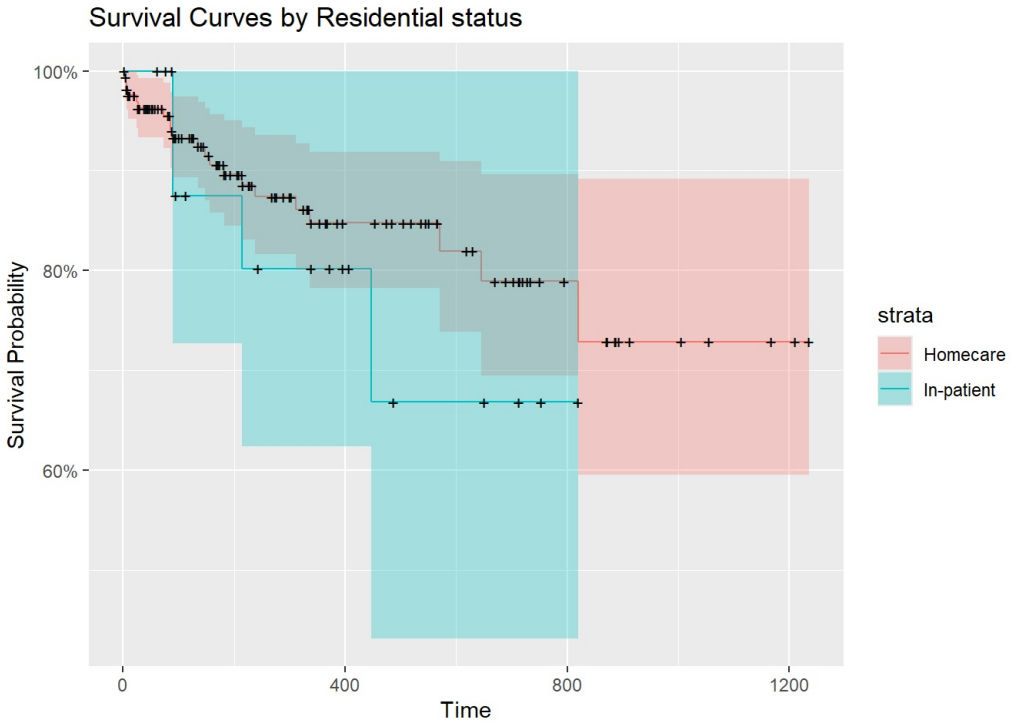
```
surv_fit <- survfit(Surv(survival_time, status) ~ 1, data=subset_data)
autoplot(surv_fit) +  labs(x = "Time", y = "Survival Probability", title = "Overall Survival Curves")
```



Survival Curve by Residential Status

The survival curve graph shows that both the "Homecare" and "In-patient" groups start with a 100% survival probability. Over time, the "Homecare" group, represented by a blue line, maintains higher survival probabilities, with a slower decline compared to the "In-patient" group, represented by a red line. For instance, at the 400-day mark, the survival probability for the "Homecare" group is around 80%, while the "In-patient" group drops to approximately 60%. This indicates that individuals in home care have better survival outcomes over time compared to those who are in-patients.

```
#survival curve
surv_fit2 <- survfit(Surv(survival_time, status) ~ Facility_Type, data=subset_data)

autoplot(surv_fit2) + labs(x = "Time", y = "Survival Probability", title = "Survival Curves by Residential status
")
```



Log-Rank Test

The log-rank test is commonly used in survival analysis to compare the survival distributions of two or more groups (strata) to determine if there are statistically significant differences between them.

In the below Log-Rank analysis with 185 total observations (13 excluded due to missing data), the chi-squared statistic is 0.6 with 1 degree of freedom, resulting in a p-value of 0.5, which is greater than the conventional significance level of 0.05. This p-value suggests that there is no statistically significant difference in survival between the groups Homecare and In-patient, as the observed differences in survival times are not likely to be due to chance. Therefore, based on these results, we do not reject the null hypothesis that the survival distributions across different facility types are similar

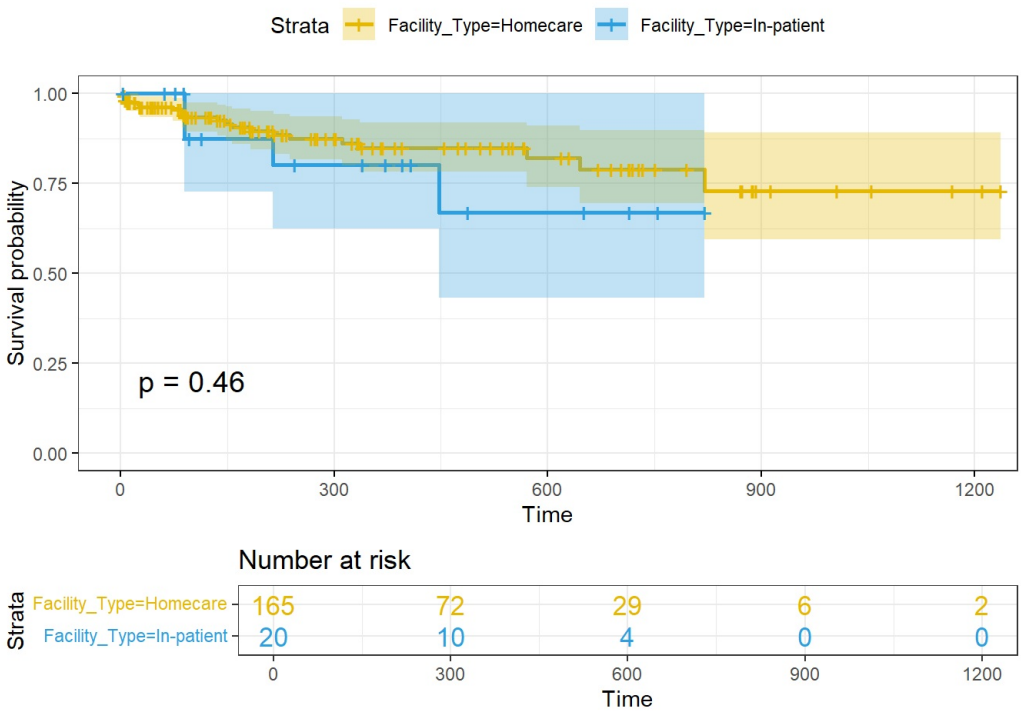
```
surv_diff <- survdiff(Surv(survival_time, status) ~ Facility_Type, data=subset_data); surv_diff
```

```
## Call:
## survdiff(formula = Surv(survival_time, status) ~ Facility_Type,
##     data = subset_data)
##
## n=185, 13 observations deleted due to missingness.
##
##                            N Observed Expected (O-E)^2/E (O-E)^2/V
## Facility_Type=Homecare   165       21    22.18    0.0623     0.553
## Facility_Type=In-patient  20        4     2.82    0.4893     0.553
##
##  Chisq= 0.6  on 1 degrees of freedom, p= 0.5
```

Log Rank Test on Kaplan Meier Curve

In the below curve, the p-value of 0.46 is greater than the conventional significance level of 0.05, thus this indicates no statistically significant difference between the two groups' survival curves. While both groups start with high survival probabilities, the "Homecare" group maintains slightly higher probabilities over time. The number at risk decreases from 165 to 60 for "Homecare" and from 200 to 90 for "In-patient." *This suggests that residential status does not significantly impact survival outcomes based on this data*.

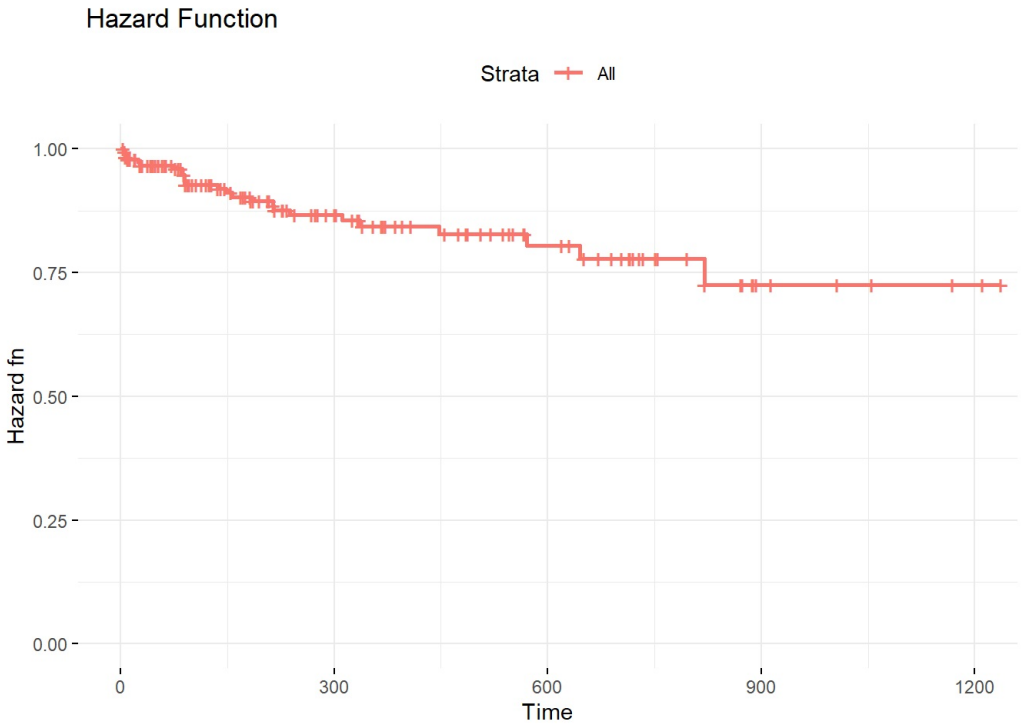
```
#run log rank test
ggsurvplot(surv_fit2,
           pval = TRUE, conf.int = TRUE,
           risk.table = TRUE,
           risk.table.col = "strata",
           ggtheme = theme_bw(),
           palette = c("#E7B800", "#2E9FDF"))
```



1.4.3 Compute Hazard Function

The plot represents the cumulative hazard function over 1236 time units, showing the total hazard accumulated over time. Initially, the hazard increases steeply, indicating a high risk of the event occurring early on. After about 300 time units, the curve flattens, suggesting a decrease in the event rate as time progresses. The consistent error bars across the plot reflect uniform precision in the hazard estimates. This type of analysis is essential in fields like medical research to understand risk dynamics and plan interventions.

```
# Compute the hazard function using the Nelson-Aalen estimator
cum_haz <- survfit(coxph(Surv(survival_time, status) ~ 1, data = subset_data), type = "aalen")

# Plotting the cumulative hazard function
ggsurvplot(cum_haz, data = subset_data, conf.int = FALSE,
           ggtheme = theme_minimal(),
           xlab = "Time", ylab = "Hazard fn",
           title = " Hazard Function")
```



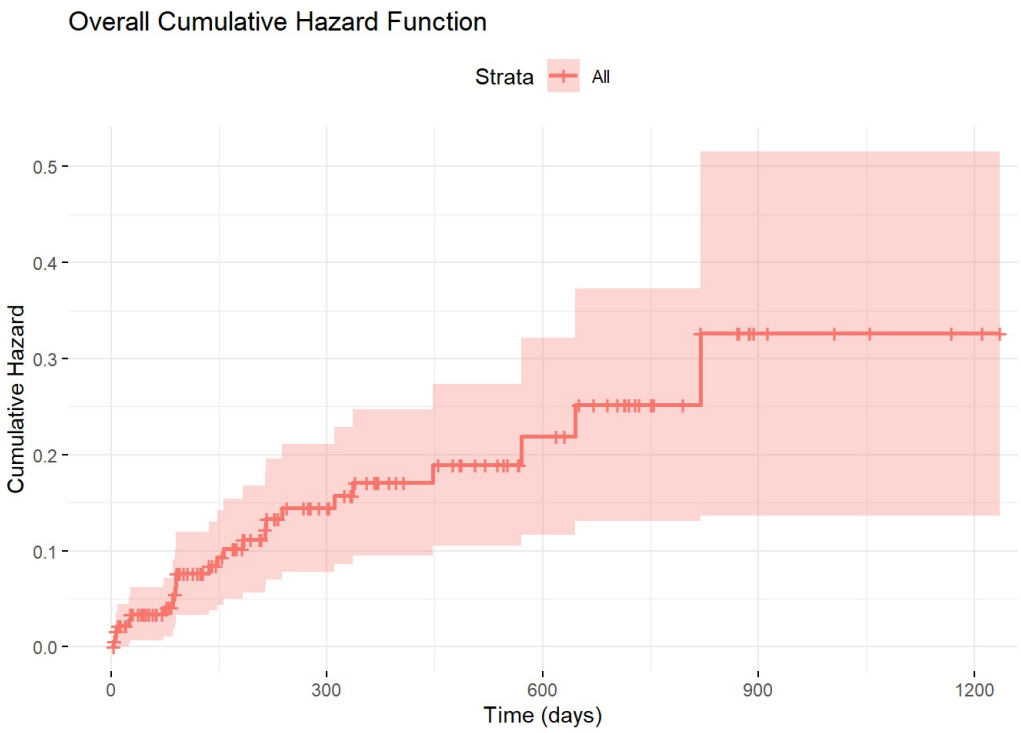
1.4.4 Compute Cumulative Hazard Function

Overall Cumulative Hazard Function

The `Cumulative Hazard Function` graph displays the cumulative hazard over time for a study, showing a significant initial increase within the first 200 days, followed by a plateau. This pattern indicates a high risk of the event occurring early in the period, which stabilizes as time progresses. Such insights are crucial in contexts like healthcare or mechanical system maintenance, where early detection and intervention can

significantly mitigate risks. The graph effectively outlines how risks accumulate, aiding in strategic planning and informed decision-making based on risk timing.

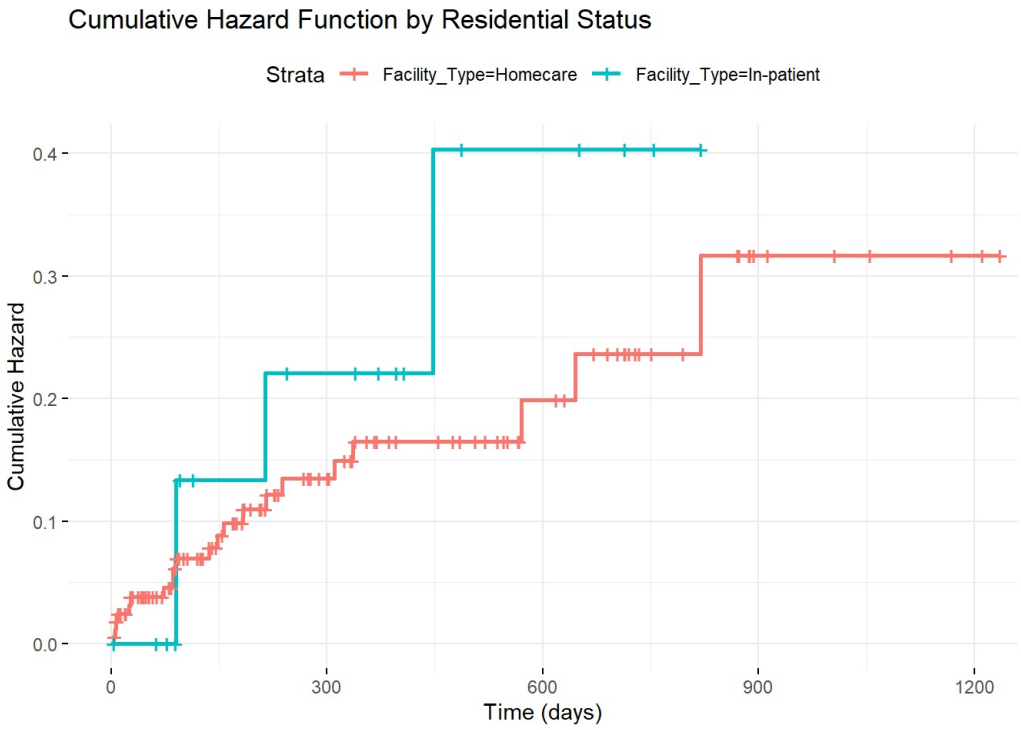
```
# Plot the cumulative hazard function
ggsurvplot(surv_fit,
           fun = "cumhaz",
           xlab = "Time (days)",
           ylab = "Cumulative Hazard",
           ggtheme = theme_minimal(),
           title = "Overall Cumulative Hazard Function")
```



Cumulative Hazard Function by Residential Status

This graph shows the cumulative hazard function over 1200 days for two types of facilities. The cumulative hazard, representing the accumulated risk of an event over time, increases in steps for both facility types. The red line (Facility_Type=0) shows a gradual increase, while the blue line (Facility_Type=1) has sharper rises followed by periods of stability. Both lines plateau around 900 days, indicating no significant additional risk of the event occurring after this point. This analysis helps compare how different conditions influence risk over time in various contexts.

```
# Plot the cumulative hazard function
ggsurvplot(surv_fit2,
           fun = "cumhaz",
           xlab = "Time (days)",
           ylab = "Cumulative Hazard",
           ggtheme = theme_minimal(),
           title = "Cumulative Hazard Function by Residential Status")
```



Question 2

2.1 Data Overview

In this question we're going to analyze the impact of marital status and compare its survival curves, and re-coded the `Marital Status` variable from the patient data, specifically focusing on the question, which asks, "What is your current marital status?" We categorized the responses into two groups: "Separated" and "Not Separated." Using the following rubric, we re-coded the variables:

- Single - Not Separated
- Married - Not Separated
- Divorced - Separated
- Widowed - Separated
- Separated - Separated

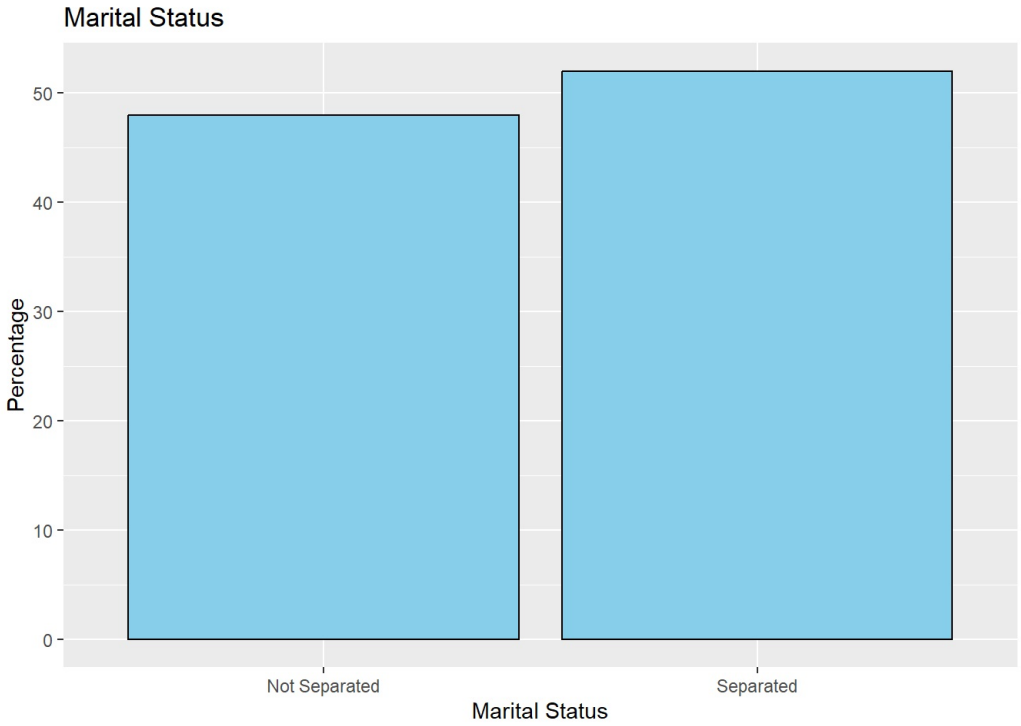
```
#recode variables
subset_data <- subset_data %>%
  mutate(Marital = ifelse(Marital %in% c("Separated", "Widowed", "Divorced"),
                          "Separated", "Not Separated"))

#view results
subset_data
```

```
## # A tibble: 198 × 9
##    Date_Started        Date_Ended          Event Marital       Length_Separated
##    <dttm>              <dttm>              <chr> <chr>         <chr>
##  1 2024-04-16 00:00:00 2024-06-04 00:00:00 <NA>  Separated     More than 5 years
##  2 2022-12-06 00:00:00 2024-06-04 00:00:00 <NA>  Separated     More than 5 years
##  3 2024-01-29 00:00:00 2024-06-04 00:00:00 <NA>  Separated     More than 5 years
##  4 2024-04-01 00:00:00 2024-04-28 00:00:00 Died  Not Separated NA
##  5 2024-04-01 00:00:00 2024-06-04 00:00:00 <NA>  Not Separated <NA>
##  6 2024-03-01 00:00:00 2024-06-04 00:00:00 <NA>  Separated     More than 5 years
##  7 2024-03-01 00:00:00 2024-06-04 00:00:00 <NA>  Not Separated <NA>
##  8 2024-01-01 00:00:00 2024-06-04 00:00:00 <NA>  Not Separated <NA>
##  9 2024-04-01 00:00:00 2024-06-04 00:00:00 <NA>  Not Separated <NA>
## 10 2023-12-01 00:00:00 2024-06-04 00:00:00 <NA>  Not Separated <NA>
## # i 188 more rows
## # i 4 more variables: Reason_Caregiver <chr>, Facility_Type <chr>,
## #   status <dbl>, survival_time <dbl>
```

2.2 Distribution of Marital Status

The bar chart illustrates the distribution of two categories of marital status: "Separated" and "Not Separated." The "Separated" category, which includes those who are divorced, widowed, or legally separated, shows a higher count, nearly reaching 100 individuals. In contrast, the "Not Separated" category, which encompasses single and married individuals, has a significantly lower count, around 70 individuals. This suggests that in the sampled population, a larger number of individuals fall into the "Separated" category compared to the "Not Separated" group.

```
# Create a barplot
ggplot(subset_data, aes(x = Marital, y = (..count..)/sum(..count..))) +
  geom_bar(aes(y = ..prop.. * 100, group = 1), fill = "skyblue", color = "black") +
  labs(title = "Marital Status",
       x = "Marital Status",
       y = "Percentage")
```



2.2.1 Comparison of Marital Status by Residential Status

We ran a chi-square test to determine if there is a statistically significant relationship between the Marital Status and Residential status of patients. The P-value obtained of 0.4472 is greater than the conventional significance level of 0.05. Thus we fail to reject the null hypothesis that there is no significant association between the two variables.

```
#smaller subset
subset_data2 <- subset_data %>%
  select(Marital, Facility_Type)

#omitt NA values
subset_data2 <- na.omit(subset_data2)

#create table
table(subset_data2$Marital, subset_data2$Facility_Type)
```

```
##
##                Homecare In-patient
##   Not Separated       79          7
##   Separated           90         13
```

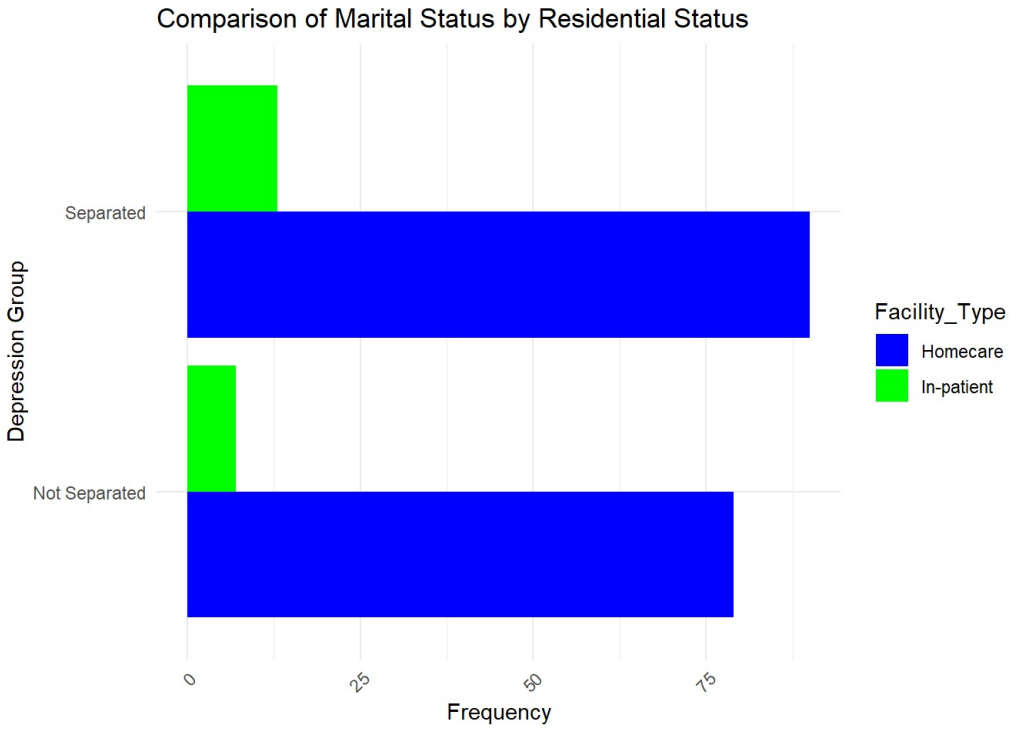
```
#chi test
chisq.test(subset_data2$Marital, subset_data2$Facility_Type)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  subset_data2$Marital and subset_data2$Facility_Type
## X-squared = 0.57764, df = 1, p-value = 0.4472
```

We then plotted a side-by-side bar-plot of the two variables, however the relationships shown are not statistically significant.

The bar-plot shows that for both Separated and Not Separated patients, majority are Homecare patients. Furthermore, for both Homecare and In-patients majority of patients are Separated rather than Not Separated.

```
#plot barplot
ggplot(subset_data2, aes(x = Marital, fill = Facility_Type)) +
  geom_bar(position = "dodge", na.rm = TRUE) +
  labs(title = "Comparison of Marital Status by Residential Status",
       x = "Depression Group",
       y = "Frequency") +
  scale_fill_manual(values = c("Homecare" = "blue", "In-patient" = "green")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  coord_flip()
```



2.2.2 Comparison of Depression Severity and Marital Satatus

The ordinal variable created in Q1 is `Depression_severity` . Let us compare this to the `Marital` variable using a **chi-square test**, to determine if there is an association between depression state and relationship status.

The below output of the chi-square test shows a P-Value of 0.00129, which is less than the chosen significance level of 0.05. Thus, there is a statistically significant association between depression severity and marital status.

```
#add depression_severity variable to subset_data
subset_data$depression_severity <- mental_health$depression_severity

#tabulate both depression_severity and marital
table(subset_data$depression_severity)
```

```
##
##        None-Minimal     Mild depression Moderate depression   Moderately Severe
##                  42                  76                  39                  28
##   Severe depression
##                  13
```

```
table(subset_data$Marital)
```

```
##
## Not Separated     Separated
##            95           103
```

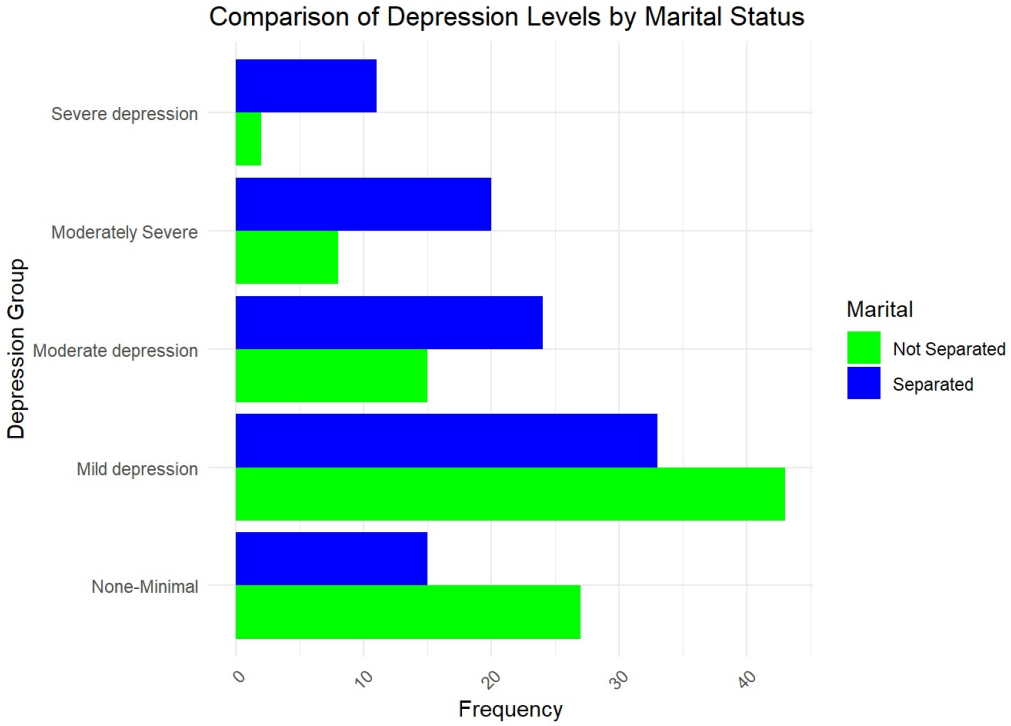
```
#run chi-square test
chisq.test(subset_data$depression_severity, subset_data$Marital)
```

```
##
##  Pearson's Chi-squared test
##
## data:  subset_data$depression_severity and subset_data$Marital
## X-squared = 17.901, df = 4, p-value = 0.00129
```

We will create a side-by-side bar chart to compare the depression severity levels and `separated` and `not separated` marital status groups.

The bar-plot reveals a distinct pattern where individuals classified under the lower levels of depression, (`None-Minimal` and `Mild depression`) predominantly belong to the `Not Separated` marital status group. Conversely, for those categorized under the `Severe depression` , `Moderately Severe` , and `Moderate depression` groups there is a noticeable prevalence of individuals classified as `Separated` . This trend suggests a correlation between higher levels of depression and marital separation.

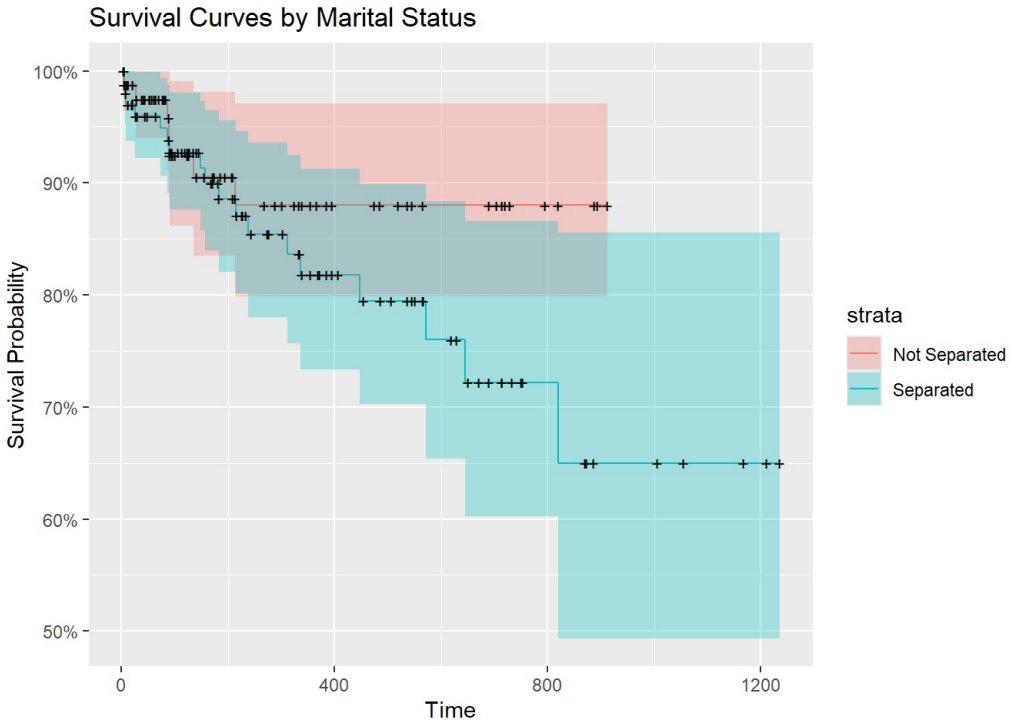
```
# Create a bar plot
ggplot(subset_data, aes(x = depression_severity, fill = Marital)) +
  geom_bar(position = "dodge") +
  labs(title = "Comparison of Depression Levels by Marital Status",
       x = "Depression Group",
       y = "Frequency") +
  scale_fill_manual(values = c("Separated" = "blue", "Not Separated" = "green")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) + coord_flip()
```



2.2.3 Survival Analysis

The plot “Survival Curves by Marital Status” illustrates the survival probabilities of individuals based on their marital status, distinguishing between those who are not separated (Stratum 0, in red) and those who are separated (Stratum 1, in blue). It shows that individuals who are not separated have consistently higher survival probabilities over time compared to their separated counterparts. The survival rates for both groups decline as time progresses, typical in survival analysis, with a more pronounced decline observed in the separated group. This suggests a higher mortality rate among separated individuals. The narrower confidence intervals for the not separated group indicate more precise survival estimates, whereas the broader intervals for the separated group suggest greater uncertainty in their survival predictions. The graph also highlights specific times where there are significant drops in survival, particularly for the separated group, which points to moments when their risk of death is notably higher. This visualization underscores the impact of marital separation on survival, suggesting that marital status is a significant factor in mortality studies and may have implications for healthcare and social support interventions.

```
#plot survival function
surv_fit3 <- survfit(Surv(survival_time, status) ~ Marital , data=subset_data)

autoplot(surv_fit3) + labs(x = "Time", y = "Survival Probability", title = "Survival Curves by Marital Status")
```



Log-Rank Test

The output from the log rank test indicates the results of a statistical test comparing survival distributions between two groups categorized by marital status (`Marital`). In this analysis with 185 total observations (13 excluded due to missing data), the chi-squared statistic is 1.4 with 1 degree of freedom, resulting in a p-value of 0.2, which is greater than the conventional significance level of 0.05. This p-value suggests that there is no statistically significant difference in survival between the groups (`Marital=Not Separated` and `Marital=Separated`), as the observed differences in survival times are not statistically significant at the conventional significance level of 0.05. *Therefore, based on these results, we do not have sufficient evidence to reject the null hypothesis that the survival distributions across different marital statuses are similar.*

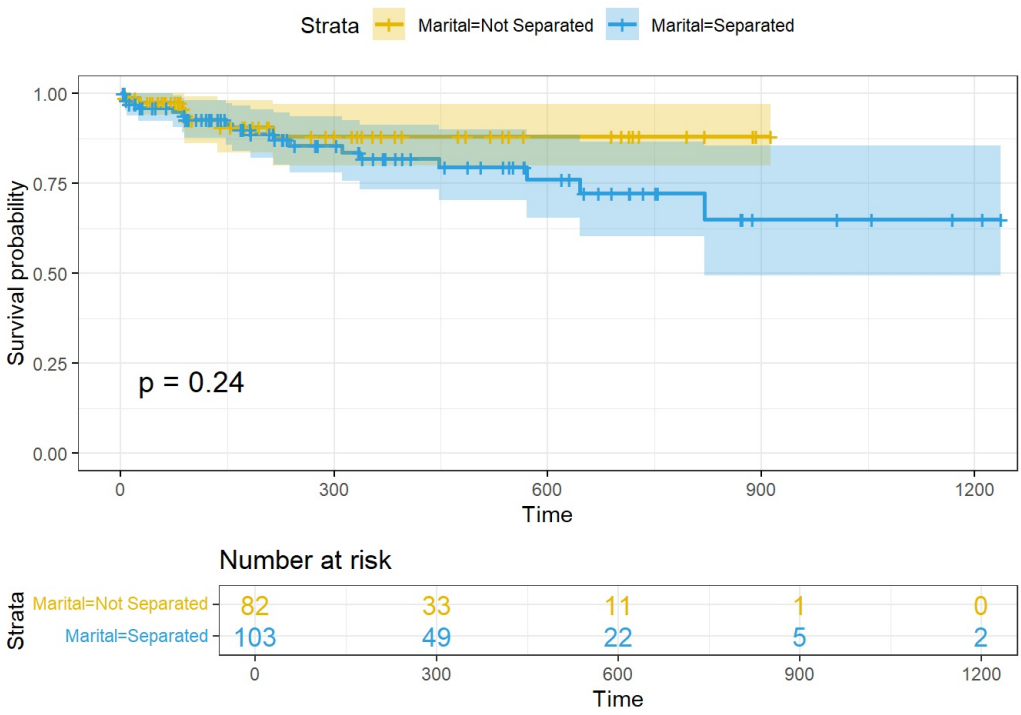
```
surv_diff <- survdiff(Surv(survival_time, status) ~ Marital , data=subset_data); surv_diff
```

```
## Call:
## survdiff(formula = Surv(survival_time, status) ~ Marital, data = subset_data)
##
## n=185, 13 observations deleted due to missingness.
##
##                          N Observed Expected (O-E)^2/E (O-E)^2/V
## Marital=Not Separated  82        7     9.89     0.846      1.41
## Marital=Separated     103       18    15.11     0.554      1.41
##
##  Chisq= 1.4  on 1 degrees of freedom, p= 0.2
```

Log-Rank Test on Kaplan Meier Curve

The log rank test displayed in the Kaplan-Meier survival curve compares two groups: "Marital-Not Separated" and "Marital-Separated." The survival probabilities for both groups are plotted over time, with the "Marital-Not Separated" group shown in yellow and the "Marital-Separated" group in blue. At the start (time zero), there are 103 individuals in the "Not Separated" group and 83 in the "Separated" group. The p-value of 0.24 indicates no statistically significant difference in survival between the two groups, as it is greater than the conventional significance level of 0.05. This conclusion is further supported by the overlapping confidence intervals for both groups throughout the observed period.

```
ggsurvplot(surv_fit3,
pval = TRUE, conf.int = TRUE,
risk.table = TRUE,
risk.table.col = "strata",
ggtheme = theme_bw(),
palette = c("#E7B800", "#2E9FDF"))
```



2.3 Distribution of Length Separated Variable

We will create a bar chart for the `Length_Separated` variable in column L, which answers the question: "If widowed/divorced/separated, for how long have you been widowed/divorced/separated?" The response categories are:

- Last six months – `<6m`
- More than six months but less than one year – `6m < 1`
- One year to below three years – `1<3`
- Three years to below five years – `3<5`
- More than five years – `>5`

Let us examine the `Length-Separated` variable for any discrepancies in categorization.

```
#examine variable
unique(subset_data$Length_Separated)
```

```
## [1] "More than 5 years" "NA"                NA
## [4] "3<5"               ">5"                "1<3"
## [7] ">50,000"           "<6m"
```

The "More than 5 years" and ">50,000" categories need to be re-coded to a standardized ">5". This ensures consistency in our data analysis. Furthermore, we filter the data-set to include only those who are separated, as they are the only individuals we want to consider for the `Length_Separated` variable.

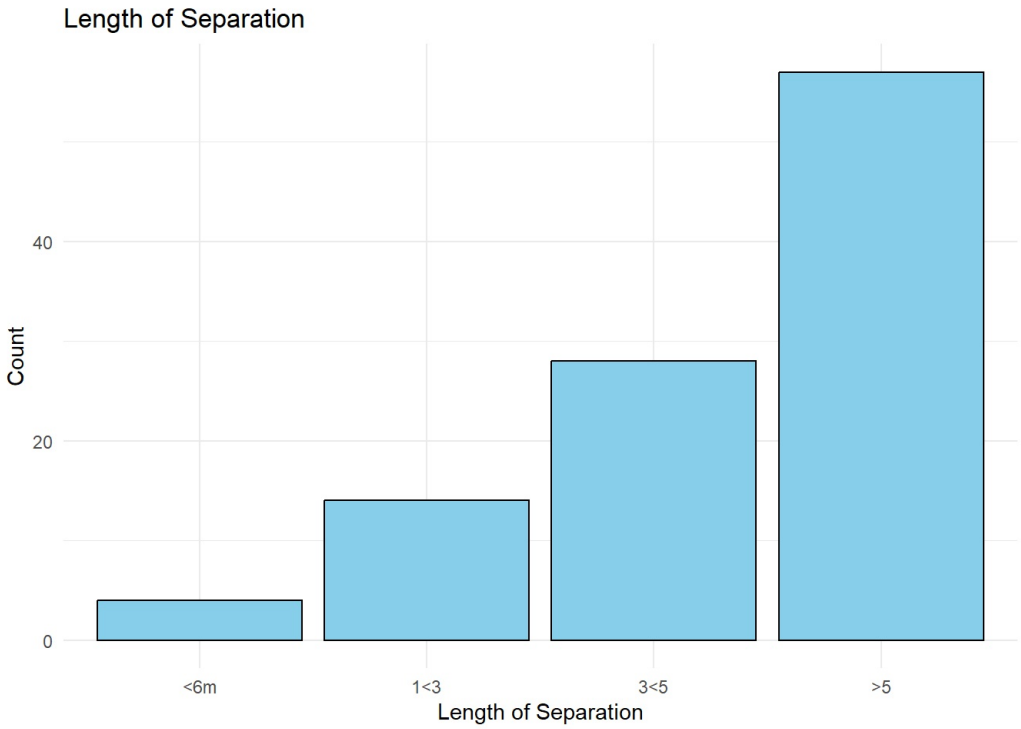
This prepares the data for creating a bar chart to visualize the distribution of separation durations.

```
#filter for only separated individuals
filtered_data <- subset_data %>%
  filter(Marital == "Separated")

#standaardise the >5 category
filtered_data <- filtered_data %>%
  mutate(Length_Separated = ifelse(Length_Separated %in% c("More than 5 years", ">50,000"), ">5", Length_Separate
d))
#view results
filtered_data
```

```
## # A tibble: 103 × 10
##    Date_Started        Date_Ended          Event Marital   Length_Separated
##    <dttm>              <dttm>              <chr> <chr>     <chr>
##  1 2024-04-16 00:00:00 2024-06-04 00:00:00 <NA>  Separated >5
##  2 2022-12-06 00:00:00 2024-06-04 00:00:00 <NA>  Separated >5
##  3 2024-01-29 00:00:00 2024-06-04 00:00:00 <NA>  Separated >5
##  4 2024-03-01 00:00:00 2024-06-04 00:00:00 <NA>  Separated >5
##  5 2024-04-01 00:00:00 2024-06-04 00:00:00 <NA>  Separated 3<5
##  6 2024-01-01 00:00:00 2024-04-01 00:00:00 <NA>  Separated >5
##  7 2024-02-01 00:00:00 2024-06-04 00:00:00 <NA>  Separated 1<3
##  8 2023-11-01 00:00:00 2024-01-26 00:00:00 Died  Separated 3<5
##  9 2022-12-01 00:00:00 2024-06-04 00:00:00 <NA>  Separated >5
## 10 2024-05-01 00:00:00 2024-05-31 00:00:00 <NA>  Separated 1<3
## # i 93 more rows
## # i 5 more variables: Reason_Caregiver <chr>, Facility_Type <chr>,
## #   status <dbl>, survival_time <dbl>, depression_severity <fct>
```

The bar chart illustrates the distribution of the length of time individuals have been separated. The highest count is in the “>5” category, indicating that the majority of individuals have been separated for more than five years. The “3<5” category follows, showing a significant number of individuals separated for three to five years. The “1<3” category has fewer individuals, and the “<6m” category has the least. This distribution suggests that longer separations are more common among the respondents in the data, with a substantial portion having been separated for extended periods.

```
# Convert Length_Separated to an ordered factor
filtered_data$Length_Separated <- factor(filtered_data$Length_Separated,
                                         levels = c("<6m", "1<3","3<5", ">5"), # Replace with actual levels in de
sired order
                                         ordered = TRUE)

# Create the bar plot
ggplot(filtered_data, aes(x = Length_Separated)) +
  geom_bar(fill = "skyblue", color = "black") +
  labs(title = "Length of Separation",
       x = "Length of Separation",
       y = "Count") +
  theme_minimal()
```



2.3.1 Comparison of Length Separated and Residential Status

We ran a chi-square test to determine if there is a significant association between the a patients length of separation and their residential status. We used a simulated p-value due to low value counts for the chi-square test. The P-value obtained of 0.9325 is greater than the conventional significance level of 0.05. Thus we fail to reject the null hypothesis that there is no significant association between the two variables.

```
#create table
table(filtered_data$Length_Separated, filtered_data$Facility_Type)
```

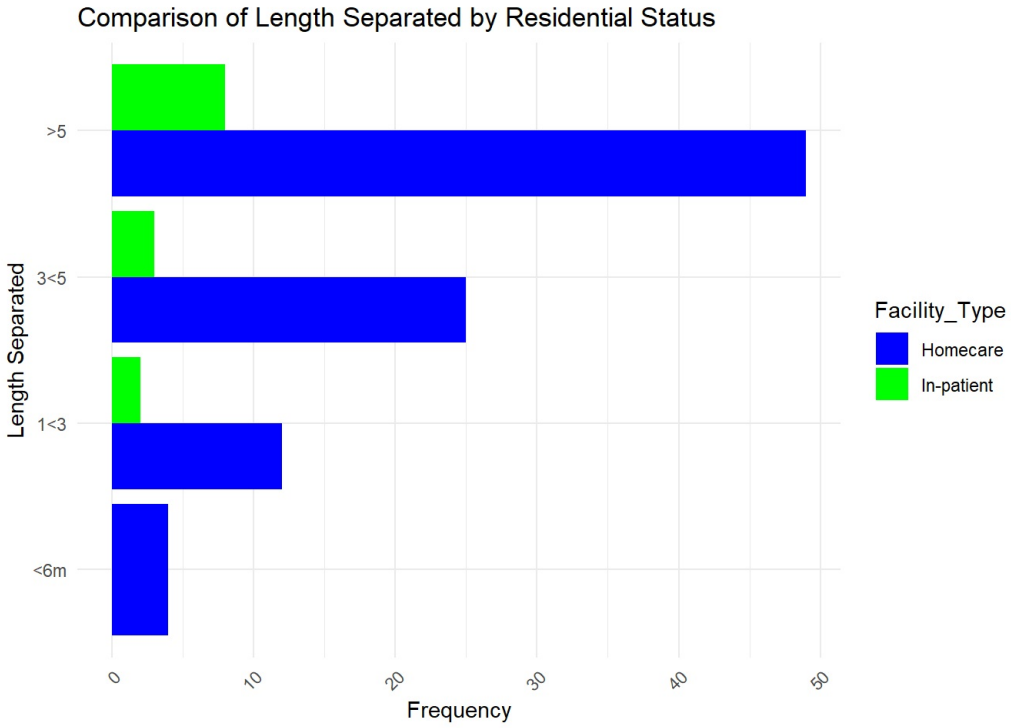
```
##
##       Homecare In-patient
##   <6m        4          0
##   1<3       12          2
##   3<5       25          3
##   >5        49          8
```

```
#chi test
chisq.test(filtered_data$Length_Separated, filtered_data$Facility_Type, simulate.p.value = TRUE, B = 2000)
```

```
##
##  Pearson's Chi-squared test with simulated p-value (based on 2000
##  replicates)
##
## data:  filtered_data$Length_Separated and filtered_data$Facility_Type
## X-squared = 0.80858, df = NA, p-value = 0.933
```

We then plotted a side-by-side bar-plot visualising the distribution of patient's length of separation, categorised by their residential status; however, any relationships observed are not significant. The bar-plot shows that only homecare patients have been separated for less than 6 months. Following the distribution of the length separation variable, majority of patients have been separated for more than 5 years, with the count of patients increasing with every longer-duration category. This is true for both in-patients and homecare patients.

```
#plot barplot
ggplot(filtered_data, aes(x = Length_Separated, fill = Facility_Type)) +
  geom_bar(position = "dodge", na.rm = TRUE) +
  labs(title = "Comparison of Length Separated by Residential Status",
       x = "Length Separated",
       y = "Frequency") +
  scale_fill_manual(values = c("Homecare" = "blue", "In-patient" = "green")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  coord_flip()
```



2.4 Summary for `Reason_Caregiver` Variable

We summarised the variable `Reason_Caregiver` in column BH, that is question 35, "What is the main reason that made you bring your patient to Care 360?". The variable has 36 unique responses, and thus can be summarised into larger categories for similar responses.

First we examined the unique values of the variable:

```
as.matrix(unique(patient_data$Reason_Caregiver))
```

```
##       [,1]
##  [1,] "Patient Condition"
##  [2,] "Chronic Condition"
##  [3,] "Assist"
##  [4,] "Recommended by a doctor"
##  [5,] "ompanionship and post care"
##  [6,] "Companionship"
##  [7,] "Preference"
##  [8,] "Close monitoring epilepsy"
##  [9,] "Adequate care"
## [10,] "Family engagement"
## [11,] NA
## [12,] "Patient discharged"
## [13,] "Recovery"
## [14,] "Patient lives alone"
## [15,] "Convinience"
## [16,] "Accessibility"
## [17,] "Care and support"
## [18,] "Monitoring"
## [19,] "Support"
## [20,] "Extra help"
## [21,] "Total Care"
## [22,] "Provide Care"
## [23,] "Peace of Mind"
## [24,] "Close monitoring"
## [25,] "Condition"
## [26,] "Better care"
## [27,] "Personalised Care"
## [28,] "Companionship and support"
## [29,] "Supervission"
## [30,] "Specialised care"
## [31,] "Management"
## [32,] "Companionship and care"
## [33,] "Loneliness"
## [34,] "Post surgery"
## [35,] "Post Hospital Care"
## [36,] "Post care"
## [37,] "Compassionate"
```

We then computed the counts and percentage of responses per category, for better understanding of the distribution of responses. This computation shows that the categories with the highest percentage responses are ‘Patient Condition’ (18.69%), ‘Care and support’ (12.63%), ‘Companionship’ (11.11%), ‘Convenience’ (7.58%) and ‘Family engagement’ (7.07).

```
# Summarize the Reason_Caregiver variable
reason_summary <- patient_data %>%
  group_by(Reason_Caregiver) %>%
  summarise(Count = n()) %>%
  arrange(desc(Count)) %>%
  mutate(Percentage = (Count / sum(Count)) * 100)

#omit NA values
reason_summary <- reason_summary[-6,]

# Display the summary
print(reason_summary)
```

```
## # A tibble: 36 × 3
##    Reason_Caregiver  Count Percentage
##    <chr>             <int>      <dbl>
##  1 Patient Condition    37      18.7
##  2 Care and support     25      12.6
##  3 Companionship        22      11.1
##  4 Convinience          15       7.58
##  5 Family engagement    14       7.07
##  6 Recovery              8       4.04
##  7 Total Care            8       4.04
##  8 Monitoring            6       3.03
##  9 Support               6       3.03
## 10 Assist                5       2.53
## # ℹ 26 more rows
```

As the variable has 36 unique responses, we can further summarise the categories into 7 main reasons, by merging the categories as follows:

Category	Values
Patient Condition	• Patient Care

	• Specialized Care • Condition • Recommended by a doctor • Chronic Condition
Care & Support	• Support • Assist • Provide Care • Extra help • Better Care • Personalised Care • Care and support • Specialised care
Companionship	• Loneliness • Companionship • Companionship and Care • Companionship and Support • Compassion • Companionship and support • Companionship and care • Compassionate
Convenience	• Family Engagement • Patient lives alone • Accessibility • Peace of Mind • Preference • Convinience • Peace of Mind • Family engagement
Recovery	• Post surgery management • Patient discharged • post care • post hospital care • Management • Post surgery • Post care • ompanionship and post care • Recovery • Post Hospital Care
Total Care	• Adequate care • Personalized care • Total Care • Better care
Monitoring	• Close monitoring • Epilepsy • Supervision

- Close monitoring epilepsy
- Monitoring
- Supervission

Let us summarize the variables as follows in a data-frame, getting the Total Count and Total Percentage distribution per category.

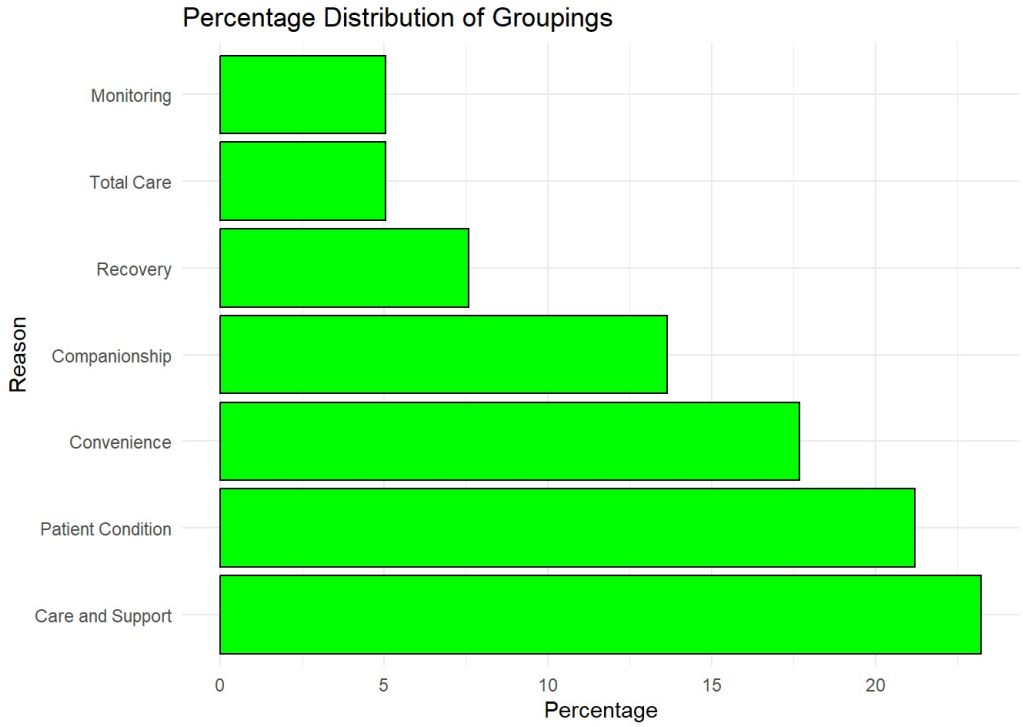
```
# Define the groups
Patient_Condition <- c("Patient Condition","Specialized Care", "Condition", "Recommended by a doctor", "Chronic C
ondition")

Care_and_Support <- c("Support", "Assist", "Provide Care", "Extra help", "Better Care", "Personalised Care", "Car
e and support", "Specialised care")

Companionship <- c("Loneliness", "Companionship", "Companionship and Care", "Companionship and Support", "Compass
ion", "Companionship and support", "Companionship and care", "Compassionate")

Convenience <- c("Family Engagement", "Patient lives alone", "Accessibility", "Peace of Mind", "Preference", "Con
vinience", "Peace of Mind", "Family engagement")

Recovery <- c("Post surgery management", "Patient discharged", "post care", "post hospital care", "Management", "
Post surgery", "Post care", "ompanionship and post care", "Recovery", "Post Hospital Care")

Total_Care <- c("Adequate care", "Personalized care", "Total Care", "Better care")

Monitoring <- c("Close monitoring", "Epilepsy", "Supervision", "Close monitoring epilepsy", "Monitoring", "Superv
ission")

# Summarize by groups
group_summary <- reason_summary %>%
  mutate(Group = case_when(
    Reason_Caregiver %in% Patient_Condition ~ "Patient Condition",
    Reason_Caregiver %in% Care_and_Support ~ "Care and Support",
    Reason_Caregiver %in% Companionship ~ "Companionship",
    Reason_Caregiver %in% Convenience ~ "Convenience",
    Reason_Caregiver %in% Recovery ~ "Recovery",
    Reason_Caregiver %in% Total_Care ~ "Total Care",
    Reason_Caregiver %in% Monitoring ~ "Monitoring",
    TRUE ~ "Other"
  )) %>%
  group_by(Group) %>%
  summarise(
    Total_Count = sum(Count),
    Total_Percentage = sum(Percentage)
  ) %>%
  arrange(desc(Total_Count))

# Display the summary
group_summary
```

```
## # A tibble: 7 × 3
##   Group             Total_Count Total_Percentage
##   <chr>                   <int>            <dbl>
## 1 Care and Support           46            23.2
## 2 Patient Condition          42            21.2
## 3 Convenience                35            17.7
## 4 Companionship              27            13.6
## 5 Recovery                   15             7.58
## 6 Monitoring                 10             5.05
## 7 Total Care                 10             5.05
```

We then create a bar-plot to visualize the percentage distribution per category.

The bar plot shows that the greatest reason for bringing a patient to ‘Care360’ is for Care and Support (23.23%) followed by Patient Condition (21.21%), then Condition (17.67%) taking up the top 50% of responses. This is followed by Companionship(13.64%), Recovery(7.58%), Total Care(5.05%) then finally Monitoring (5.05%) which has the fewest percentage of responses.

```
# Create a bar plot for the distribution
ggplot(group_summary, aes(x = reorder(Group, -Total_Percentage), y = Total_Percentage)) +
  geom_bar(stat = "identity", fill = "green", color = "black") +
  coord_flip() + # Flip coordinates for better readability
  labs(title = "Percentage Distribution of Groupings",
       x = "Reason",
       y = "Percentage") +
  theme_minimal()
```



We then re-coded the `Reason_Caregiver` column appropriately, as follows:

```
# Combine all groups into a named vector
group_map <- c(
  setNames(rep("Patient_Condition", length(Patient_Condition)), Patient_Condition),
  setNames(rep("Care_and_Support", length(Care_and_Support)), Care_and_Support),
  setNames(rep("Companionship", length(Companionship)), Companionship),
  setNames(rep("Convenience", length(Convenience)), Convenience),
  setNames(rep("Recovery", length(Recovery)), Recovery),
  setNames(rep("Total_Care", length(Total_Care)), Total_Care),
  setNames(rep("Monitoring", length(Monitoring)), Monitoring)
)

# Replace responses with group names
subset_data <- subset_data %>%
  mutate(Reason_Caregiver = recode(Reason_Caregiver, !!!group_map))

#view results
unique(subset_data$Reason_Caregiver)
```

```
## [1] "Patient_Condition" "Care_and_Support"  "Recovery"
## [4] "Companionship"     "Convenience"       "Monitoring"
## [7] "Total_Care"        NA
```

2.4.1 Comparison of Reason_Caregiver and Residential Status

We ran a chi-square test to determine if there is a statistically significant relationship between a patient's reason for being brought to care 360, and their residential status. We used a simulated p-value due to low value counts for the chi-square test. The P-value obtained of 0.0004998 is less than the conventional significance level of 0.05. Thus we reject the null hypothesis that there is no significant relationship between the two variables. This result is subject to inaccuracies, as some values used int he chi-square test are less than 5.

```
#subset data
subset_data3 <- subset_data %>%
  select(Reason_Caregiver, Facility_Type)

#create table
table(subset_data3$Reason_Caregiver, subset_data3$Facility_Type)
```

```
##
##                     Homecare In-patient
##   Care_and_Support        46          0
##   Companionship           27          0
##   Convenience             28          7
##   Monitoring              10          0
##   Patient_Condition       34          8
##   Recovery                10          5
##   Total_Care              10          0
```

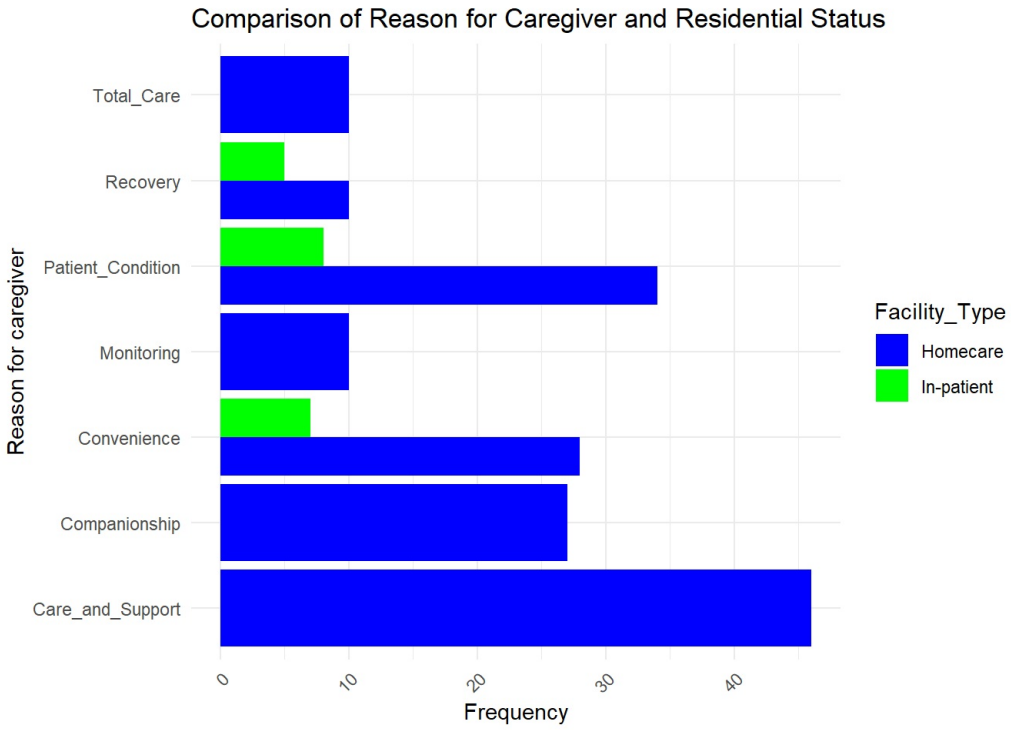
```
#chi test
chisq.test(subset_data3$Reason_Caregiver, subset_data3$Facility_Type, simulate.p.value = TRUE, B = 2000)
```

```
##
##  Pearson's Chi-squared test with simulated p-value (based on 2000
##  replicates)
##
## data:  subset_data3$Reason_Caregiver and subset_data3$Facility_Type
## X-squared = 25.185, df = NA, p-value = 0.001499
```

We then created a side-by-side bar-plot of patients reasons for coming to care360 and their residential status. The bar-plot shows that inpatients are mainly brought to care360 due to patient condition, followed by convenience then recovery. *These reasons– especially patient condition and recovery– tend to suggest a higher level of care required for these patients, which aligns with why they are admitted as inpatients*.

For homecare patients, their to reason for being brought to care360 is care and support, followed by patient condition, convenience, companionship, monitoring, recovery and total care, in that order. *The top four categories–especially care and support and companionship – suggest a need for extra care and attention, but not as high as inpatients.*

```
#remove NA values
subset_data3 <- na.omit(subset_data3)

# Create a bar plot
ggplot(subset_data3, aes(x = Reason_Caregiver, fill = Facility_Type)) +
  geom_bar(position = "dodge") +
  labs(title = "Comparison of Reason for Caregiver and Residential Status",
       x = "Reason for caregiver",
       y = "Frequency") +
  scale_fill_manual(values = c("Homecare" = "blue", "In-patient" = "green")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) + coord_flip()
```



2.5 Comparison of Sex and the `Facilty_Type` Variable

Preprocessing

The `Sex_Patient` variable indicates the sex assigned to the patient at birth, and has four possible responses:

- Female
- Male
- Other
- Prefer not to say

First, we created a subset with the two variables of interest then examined the unique variables of the sex patient variable, to check for errors.

```
#add desired column to the data
filtered_data2 <- data.frame(Facility_Type = subset_data$Facility_Type, Sex_Patient = patient_data$Sex_Patient)

#examine unique values for both
unique(filtered_data2$Facility_Type)
```

```
## [1] "Homecare"   "In-patient" NA
```

```
unique(filtered_data2$Sex_Patient)
```

```
## [1] "Female" "Male"   NA
```

There are no errors, thus we then ran a chi-square test to determine whether there is a statistically significant relationship between the sex of the patient and their residential status. The P-value obtained of 0.8973 is greater than the conventional significance level of 0.05. Thus, we fail to reject the null hypothesis that there is no significant association between the two variables.

```
#create table
table(filtered_data2$Facility_Type, filtered_data2$Sex_Patient)
```

```
##
##              Female Male
##   Homecare      102   66
##   In-patient     13    7
```

```
#run chi test
chisq.test(filtered_data2$Facility_Type,filtered_data2$Sex_Patient)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  filtered_data2$Facility_Type and filtered_data2$Sex_Patient
## X-squared = 0.016662, df = 1, p-value = 0.8973
```

We then created a side-by-side bar plot to compare the relationship between the residential status of a patient and their sex. The relationships portrayed are not statistically significant.

The Bar-plot shows that overall, there are more female patients of Care360 than male patients. Both in residents and non-residents category, female patients are the majority. Furthermore, there are more Non-resident patients of care360 than resident patients.

```
#omit NA values
filtered_data2 <- na.omit(filtered_data2)

# Create a bar plot
ggplot(filtered_data2, aes(x = Sex_Patient, fill = Facility_Type)) +
  geom_bar(position = "dodge") +
  labs(title = "Comparison of Sex and Residential Status",
       x = "Sex",
       y = "Frequency") +
  scale_fill_manual(values = c("Homecare" = "blue", "In-patient" = "green")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) + coord_flip()
```