

Analysis_notebook

Loading libraries

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.6      v purrr  0.3.4
## v tibble  3.1.7      v dplyr  1.0.9
## v tidyr   1.2.0      v stringr 1.4.0
## v readr   2.1.2      v forcats 0.5.1
```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

```
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(ggplot2)
library(tidyr)
library(stringr)
library(janitor)
```

```
##
```

```
## Attaching package: 'janitor'
```

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

```
##
```

```
##      chisq.test, fisher.test
```

Loading and Viewing the first 6 rows of the table

```
activity_sleep_weight <- read_csv("activity_sleep_weight_daily_joined_08_05_2022_v02.csv")
```

```
## Rows: 410 Columns: 15
```

```
## -- Column specification -----
```

```
## Delimiter: ","
```

```
## dbl  (13): id, total_steps, total_distance, sedentary_minutes, calories, tot...
```

```
## lgl   (1): is_manual_report
```

```
## date  (1): activity_date
```

```
##
```

```
## i Use `spec()` to retrieve the full column specification for this data.
```

```
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(activity_sleep_weight)
```

```
## # A tibble: 6 x 15
```

```
##       id activity_date total_steps total_distance sedentary_minutes calories
##   <dbl> <date>          <dbl>          <dbl>          <dbl>    <dbl>
## 1 1503960366 2016-04-12         13162           8.5             728     1985
## 2 1503960366 2016-04-13         10735           6.97            776     1797
## 3 1503960366 2016-04-15          9762           6.28            726     1745
## 4 1503960366 2016-04-16        12669           8.16            773     1863
```

```
## 5 1503960366 2016-04-17          9705          6.48          539          1728
## 6 1503960366 2016-04-19          15506          9.88          775          2035
## # ... with 9 more variables: total_sleep_records <dbl>,
## #   total_minutes_asleep <dbl>, total_time_in_bed <dbl>,
## #   total_hours_asleep <dbl>, total_hours_in_bed <dbl>, weight_kg <dbl>,
## #   weight_pounds <dbl>, bmi <dbl>, is_manual_report <lgl>
```

checking table structure to know * number of columns and rows * datatype for each variable (column)

```
str(activity_sleep_weight)
```

```
## spec_tbl_df [410 x 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
## $ id : num [1:410] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
## $ activity_date : Date[1:410], format: "2016-04-12" "2016-04-13" ...
## $ total_steps : num [1:410] 13162 10735 9762 12669 9705 ...
## $ total_distance : num [1:410] 8.5 6.97 6.28 8.16 6.48 ...
## $ sedentary_minutes : num [1:410] 728 776 726 773 539 775 818 838 732 709 ...
## $ calories : num [1:410] 1985 1797 1745 1863 1728 ...
## $ total_sleep_records : num [1:410] 1 2 1 2 1 1 1 1 1 1 ...
## $ total_minutes_asleep: num [1:410] 327 384 412 340 700 304 360 325 361 430 ...
## $ total_time_in_bed : num [1:410] 346 407 442 367 712 320 377 364 384 449 ...
## $ total_hours_asleep : num [1:410] 5.45 6.4 6.87 5.67 11.67 ...
## $ total_hours_in_bed : num [1:410] 5.77 6.78 7.37 6.12 11.87 ...
## $ weight_kg : num [1:410] NA NA NA NA NA NA NA NA NA NA ...
## $ weight_pounds : num [1:410] NA NA NA NA NA NA NA NA NA NA ...
## $ bmi : num [1:410] NA NA NA NA NA NA NA NA NA NA ...
## $ is_manual_report : logi [1:410] NA NA NA NA NA NA ...
## - attr(*, "spec")=
## .. cols(
## .. id = col_double(),
## .. activity_date = col_date(format = ""),
## .. total_steps = col_double(),
## .. total_distance = col_double(),
## .. sedentary_minutes = col_double(),
## .. calories = col_double(),
## .. total_sleep_records = col_double(),
## .. total_minutes_asleep = col_double(),
## .. total_time_in_bed = col_double(),
## .. total_hours_asleep = col_double(),
## .. total_hours_in_bed = col_double(),
## .. weight_kg = col_double(),
## .. weight_pounds = col_double(),
## .. bmi = col_double(),
## .. is_manual_report = col_logical()
## .. )
## - attr(*, "problems")=<externalptr>
```

Checking summary statistics

```
summary(activity_sleep_weight)
```

```
##           id           activity_date           total_steps           total_distance
## Min.      :1.504e+09   Min.      :2016-04-12   Min.      : 17   Min.      : 0.010
## 1st Qu.:3.977e+09   1st Qu.:2016-04-19   1st Qu.: 5189   1st Qu.: 3.592
## Median :4.703e+09   Median :2016-04-27   Median : 8913   Median : 6.270
## Mean    :4.995e+09   Mean    :2016-04-26   Mean    : 8515   Mean    : 6.012
```

```
## 3rd Qu.:6.962e+09 3rd Qu.:2016-05-04 3rd Qu.:11370 3rd Qu.: 8.005
## Max. :8.792e+09 Max. :2016-05-12 Max. :22770 Max. :17.540
##
## sedentary_minutes calories total_sleep_records total_minutes_asleep
## Min. : 0.0 Min. : 257 Min. :1.00 Min. : 58.0
## 1st Qu.: 631.2 1st Qu.:1841 1st Qu.:1.00 1st Qu.:361.0
## Median : 717.0 Median :2207 Median :1.00 Median :432.5
## Mean : 712.1 Mean :2389 Mean :1.12 Mean :419.2
## 3rd Qu.: 782.8 3rd Qu.:2920 3rd Qu.:1.00 3rd Qu.:490.0
## Max. :1265.0 Max. :4900 Max. :3.00 Max. :796.0
##
## total_time_in_bed total_hours_asleep total_hours_in_bed weight_kg
## Min. : 61.0 Min. : 0.970 Min. : 1.020 Min. : 52.60
## 1st Qu.:403.8 1st Qu.: 6.020 1st Qu.: 6.732 1st Qu.: 61.20
## Median :463.0 Median : 7.210 Median : 7.720 Median : 61.50
## Mean :458.5 Mean : 6.987 Mean : 7.641 Mean : 64.17
## 3rd Qu.:526.0 3rd Qu.: 8.170 3rd Qu.: 8.770 3rd Qu.: 61.90
## Max. :961.0 Max. :13.270 Max. :16.020 Max. :133.50
## NA's :375
## weight_pounds bmi is_manual_report
## Min. :116.0 Min. :22.65 Mode :logical
## 1st Qu.:134.9 1st Qu.:23.89 FALSE:2
## Median :135.6 Median :24.00 TRUE :33
## Mean :141.5 Mean :24.83 NA's :375
## 3rd Qu.:136.5 3rd Qu.:24.17
## Max. :294.3 Max. :47.54
## NA's :375 NA's :375
```

Total number of users in the fitbit dataset are 34 users, however ONLY 24 users have records and 10 users have empty records (all data is 0) so users that have at least one not empty record are considered Active users users that have empty records are considered Dormant users

```
active_dormant <- data.frame(users_activity = c("Active", "Dormant"),
                             users_count = c(24,10),
                             users_percent = c(round((24/34)*100),round((10/34)*100)),
                             users_percent_lbl = c(str_c(round((24/34)*100),"%"),str_c(round((10/34)*100),"%")))
active_dormant
```

```
## users_activity users_count users_percent users_percent_lbl
## 1 Active 24 71 71%
## 2 Dormant 10 29 29%
```

Creating a summary table

```
table_summary <- activity_sleep_weight %>% group_by(id) %>%
  summarise(days_count = n(), avg_steps = mean(total_steps), avg_sleep = mean(total_hours_asleep))
table_summary
```

```
## # A tibble: 24 x 4
## id days_count avg_steps avg_sleep
## <dbl> <int> <dbl> <dbl>
## 1 1503960366 25 12406. 6.01
## 2 1644430081 4 7968. 4.90
## 3 1844505072 3 3477 10.9
```

```
## 4 1927972279      5      1490      6.95
## 5 2026352035     28     5619.      8.44
## 6 2320127002      1     5079      1.02
## 7 2347167796     15     8533.      7.45
## 8 3977333714     28    11218      4.90
## 9 4020332650      8     6597.      5.83
## 10 4319703577    26     7125.      7.94
## # ... with 14 more rows
```

Creating dataframe that contains data about Active users and how many days they use their devices.

```
users_percent <- data.frame(days_range = c("1 Week", "2 Weeks", "3 Weeks", "4 Weeks"), user_count = c(0, 0, 0, 0))
```

initialize the counters

```
counter_1_wk <- 0
counter_2_wks <- 0
counter_3_wks <- 0
counter_4_wks <- 0
```

for loop to count number of users based on the number of days they used the device (table_summary) and associating it with the corresponding days range “Number of users that used the device for the specified days range”

```
for (i in table_summary$days_count){

  if (1 <= i & i <= 7){

    users_percent[1,]['days_range'] <- "1-7"
    counter_1_wk <- counter_1_wk + 1
    users_percent[1,]['user_count'] <- counter_1_wk
  }

  else if (8 <= i & i <= 14){
    users_percent[2,]['days_range'] <- "8-14"
    counter_2_wks <- counter_2_wks + 1
    users_percent[2,]['user_count'] <- counter_2_wks
  }

  else if (15 <= i & i <= 21){
    users_percent[3,]['days_range'] <- "15-21"
    counter_3_wks <- counter_3_wks + 1
    users_percent[3,]['user_count'] <- counter_3_wks
  }
  else if (22 <= i & i <= 31){
    users_percent[4,]['days_range'] <- "22-31"
    counter_4_wks <- counter_4_wks + 1
    users_percent[4,]['user_count'] <- counter_4_wks
  }
}
```

viewing the new table

```
users_percent
```

```
##   days_range user_count
## 1      1-7           8
```

```
## 2      8-14      1
## 3     15-21      3
## 4     22-31     12
```

adding new columns

user_percentage to calculate the percentage of users *user_percentage_lbl* to add % symbol to be the label column (to be used in the charts)

```
users_percent$user_percentage <- round((users_percent$user_count/24)*100,1)
users_percent$user_percentage_lbl <- str_c(users_percent$user_percentage,"%")
users_percent
```

```
##   days_range user_count user_percentage user_percentage_lbl
## 1      1-7         8         33.3         33.3%
## 2      8-14         1          4.2          4.2%
## 3     15-21         3         12.5         12.5%
## 4     22-31        12        50.0         50%
```

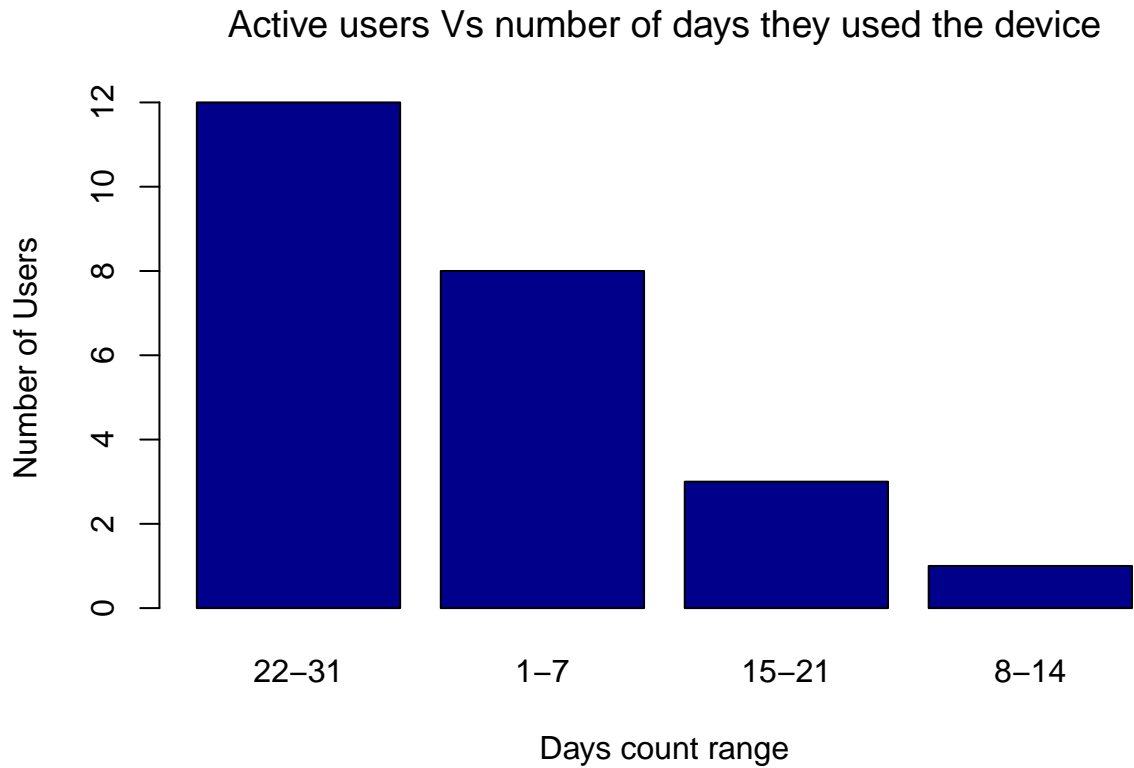
order *user_percentage* DESC

```
users_percent <- users_percent[order(users_percent$user_percentage, decreasing = TRUE),]
users_percent
```

```
##   days_range user_count user_percentage user_percentage_lbl
## 4     22-31         12        50.0         50%
## 1      1-7         8         33.3         33.3%
## 3     15-21         3         12.5         12.5%
## 2      8-14         1          4.2          4.2%
```

creating bar plot to show Active users in the specified days range

```
barplot(users_percent$user_count, names.arg = users_percent$days_range,
        xlab = "Days count range", ylab = "Number of Users",
        col = "darkblue", main = "Active users Vs number of days they used the device",
        font.main = 1 )
```



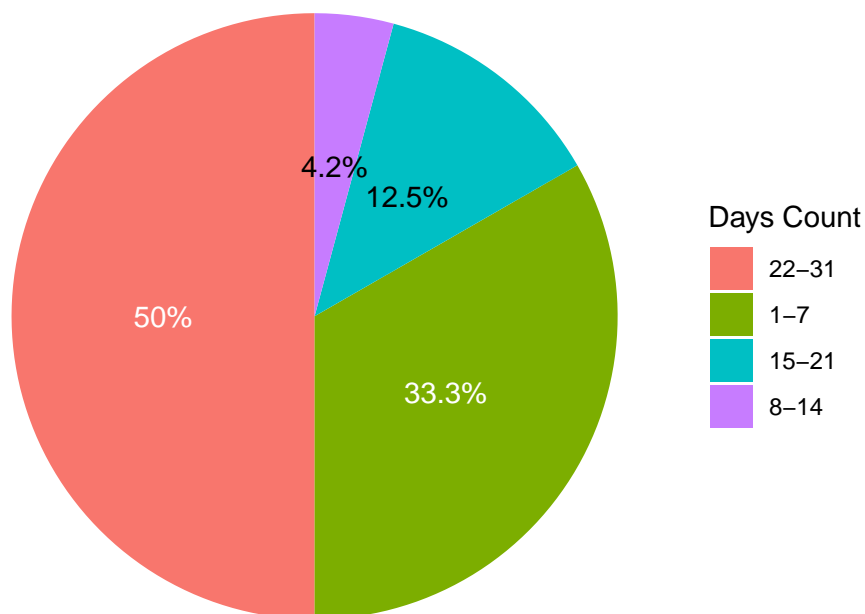
creating pie plot to show the percentage of users and how many days they used the device

```
#png(file="Piechart_users_days.png", width = 500, height = 500)
ggplot(users_percent, aes(x=1, y = user_percentage, fill= fct_inorder(days_range) )) +
  geom_col() +
  geom_text(aes(label= user_percentage_lbl,
                position= position_stack(vjust = .5),
                color = c("white","white",1,1)) +
  labs(title = "Users and the number of days they used the device",
        subtitle = "N=24",
        caption = "users used their device at least once ONLY COUNTS")+
  scale_fill_viridis_d()+
  scale_fill_discrete(name= "Days Count")+
  coord_polar(theta = "y")+
  theme_void() +
  theme( plot.title = element_text(hjust = .5, size = 9 ),
        plot.subtitle = element_text(hjust = 0.5, size = 8),
        plot.caption = element_text(hjust = 1, size = 8))
```

```
## Scale for 'fill' is already present. Adding another scale for 'fill', which
## will replace the existing scale.
```

Users and the number of days they used the device

N=24



users used their device at least once ONLY COUNTS

```
#dev.off()
```

number of active users per day

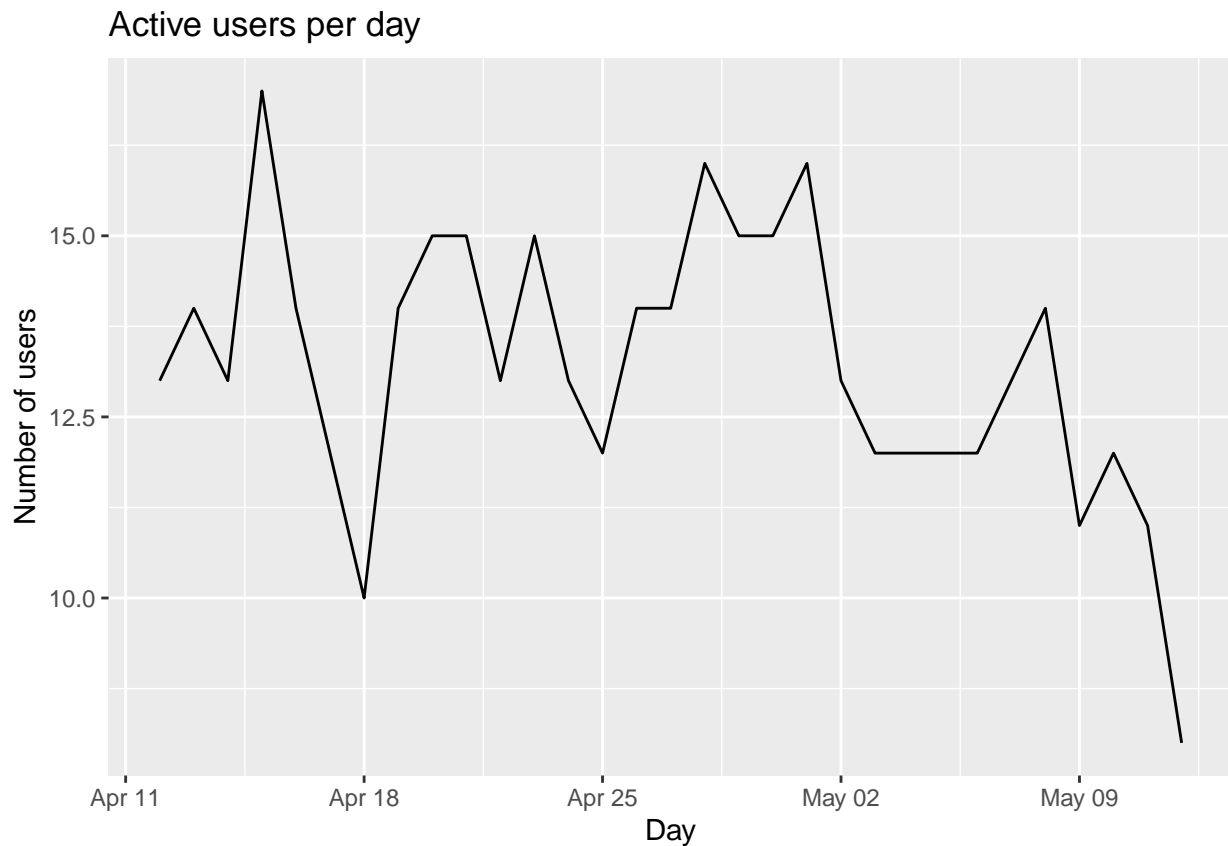
```
users_per_day <- activity_sleep_weight %>%
  group_by(activity_date) %>%
  summarise(users_count = n())
```

users_per_day

```
## # A tibble: 31 x 2
##   activity_date users_count
##   <date>         <int>
## 1 2016-04-12         13
## 2 2016-04-13         14
## 3 2016-04-14         13
## 4 2016-04-15         17
## 5 2016-04-16         14
## 6 2016-04-17         12
## 7 2016-04-18         10
## 8 2016-04-19         14
## 9 2016-04-20         15
## 10 2016-04-21        15
## # ... with 21 more rows
```

creating line chart to show how many users use the device per day

```
#png("Active_users_perday.png", width = 800, height = 500)
ggplot(data = users_per_day, aes(x= activity_date, y = users_count)) +
  geom_line()+
  ggtitle("Active users per day")+
  xlab("Day")+
  ylab("Number of users")
```



```
#dev.off()
```

Note: We need more data to confirm patterns

counting how many users log their weight

```
users_log_weight <- activity_sleep_weight %>% filter(weight_kg >=1)
```

```
weight_logs <- users_log_weight %>%
  group_by(id) %>%
  summarise(weight_logs_count = n())
```

```
weight_logs
```

```
## # A tibble: 5 x 2
##       id weight_logs_count
##   <dbl>         <int>
## 1 1503960366             2
## 2 1927972279             1
## 3 4558609924             1
## 4 5577150313             1
```


5 6962181067

30

creating a summary table for weight

```
weight_logs_count <- nrow(weight_logs)
no_weight_logs_count <- nrow(table_sumarry)-nrow(weight_logs)
total_users <- nrow(table_sumarry)
weight_logs_perc <- round((weight_logs_count/total_users)*100,0)
no_weight_logs_perc <- round(no_weight_logs_count/total_users*100,0)
total_percent <- weight_logs_perc+ no_weight_logs_perc

users_summary_weight <- data.frame(users_behavior = c("weight logged","no weight logged","total"),
                                   users_count = c(weight_logs_count,no_weight_logs_count,total_users),
                                   users_perc = c(weight_logs_perc,no_weight_logs_perc,total_percent))

users_summary_weight
```

```
##      users_behavior users_count users_perc
## 1    weight logged         5         21
## 2 no weight logged        19         79
## 3          total         24        100
```

the new table to be used in the pie chart

```
users_summary_weight_no_tatal <- users_summary_weight %>% filter(users_perc < 100)
#View(users_summary_weight_no_tatal)
users_summary_weight_no_tatal$users_perc_lbl <- str_c(users_summary_weight_no_tatal$users_perc,"%")

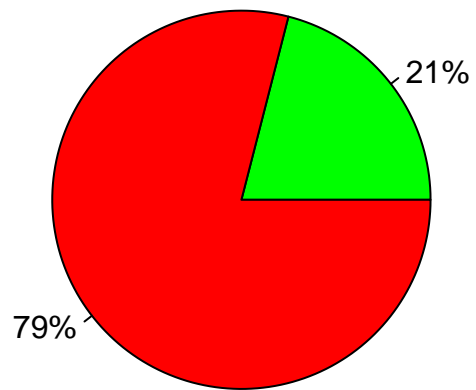
users_summary_weight_no_tatal
```

```
##      users_behavior users_count users_perc users_perc_lbl
## 1    weight logged         5         21         21%
## 2 no weight logged        19         79         79%
```

creating a pie chart to show the percentage of users who logged and did not log their weight

```
#png("Active_Users_and_Weight_logs.png", width = 400, height = 400)
pie(users_summary_weight_no_tatal$users_perc,
    labels = users_summary_weight_no_tatal$users_perc_lbl,
    col = c("green","red"),
    main = "Active Users and Weight logs")
```

Active Users and Weight logs



`#dev.off()`