Hypothesis Testing

Marie Khalil

2022-07-17

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>
```

number of active users per day

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

users_per_day
```

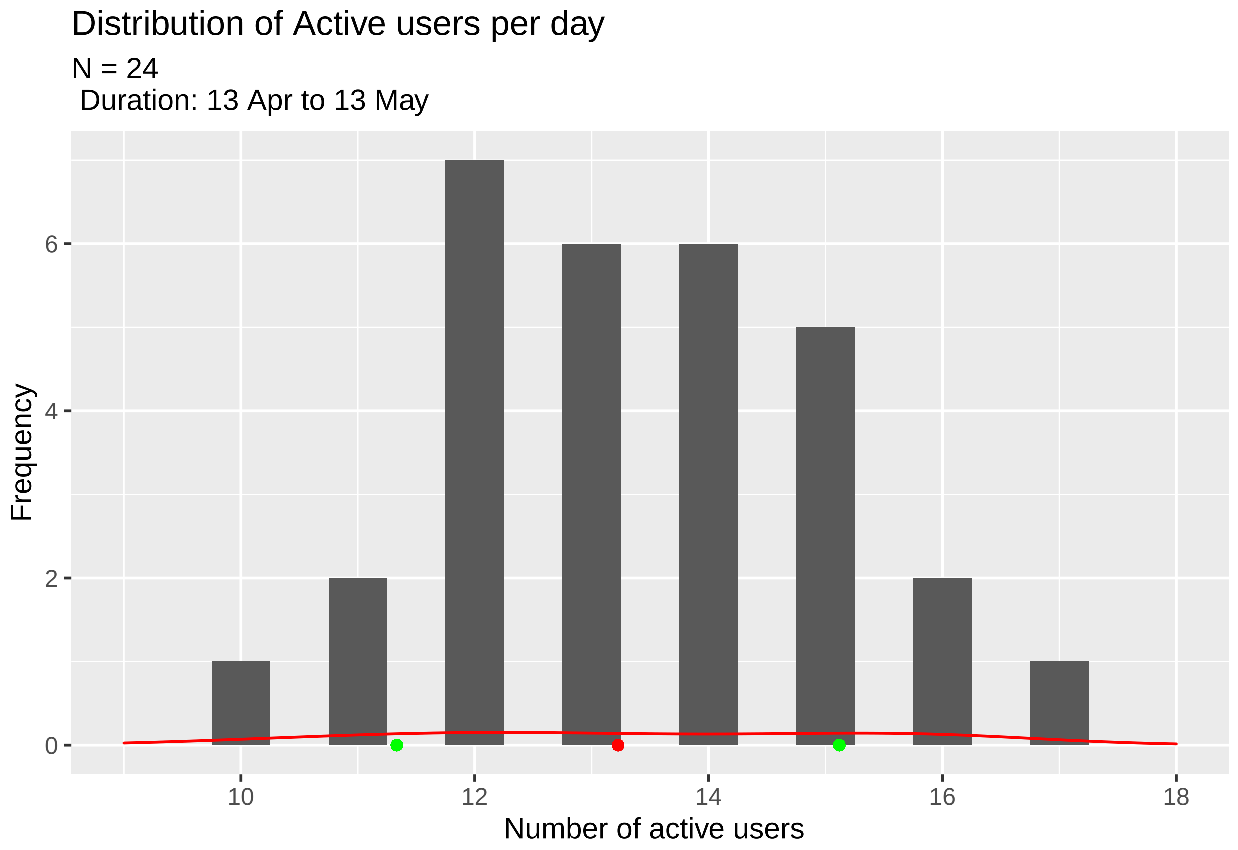
```
## # A tibble: 31 x 2
##    activity_date users_count
##    <date>              <int>
##  1 2016-04-12             13
##  2 2016-04-13             14
##  3 2016-04-14             13
##  4 2016-04-15             17
##  5 2016-04-16             14
##  6 2016-04-17             12
##  7 2016-04-18  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

Checking the distribution of data

```
mean_users_count = mean(users_per_day$users_count)
sd_users_count = sd(users_per_day$users_count)

x__ = rnorm(31, mean = mean_users_count, sd = sd_users_count)

ggplot(data = users_per_day) +
  geom_histogram(aes(x= as.integer(users_count)), binwidth = .5)+
  geom_density(aes(x = x__ ), colour = "red", show.legend = FALSE)+
  geom_point(aes(x = mean_users_count, y = 0 ), colour = "red")+
  geom_point(aes(x = mean_users_count + sd_users_count, y = 0), colour = "green")+
  geom_point(aes(x = mean_users_count - sd_users_count, y = 0), colour = "green")+
  xlim(9,18)+
  xlab(" Number of active users")+
  ylab("Frequency")+
  ggtitle("Distribution of Active users per day",
          subtitle = "N = 24 \n Duration: 13 Apr to 13 May")
```

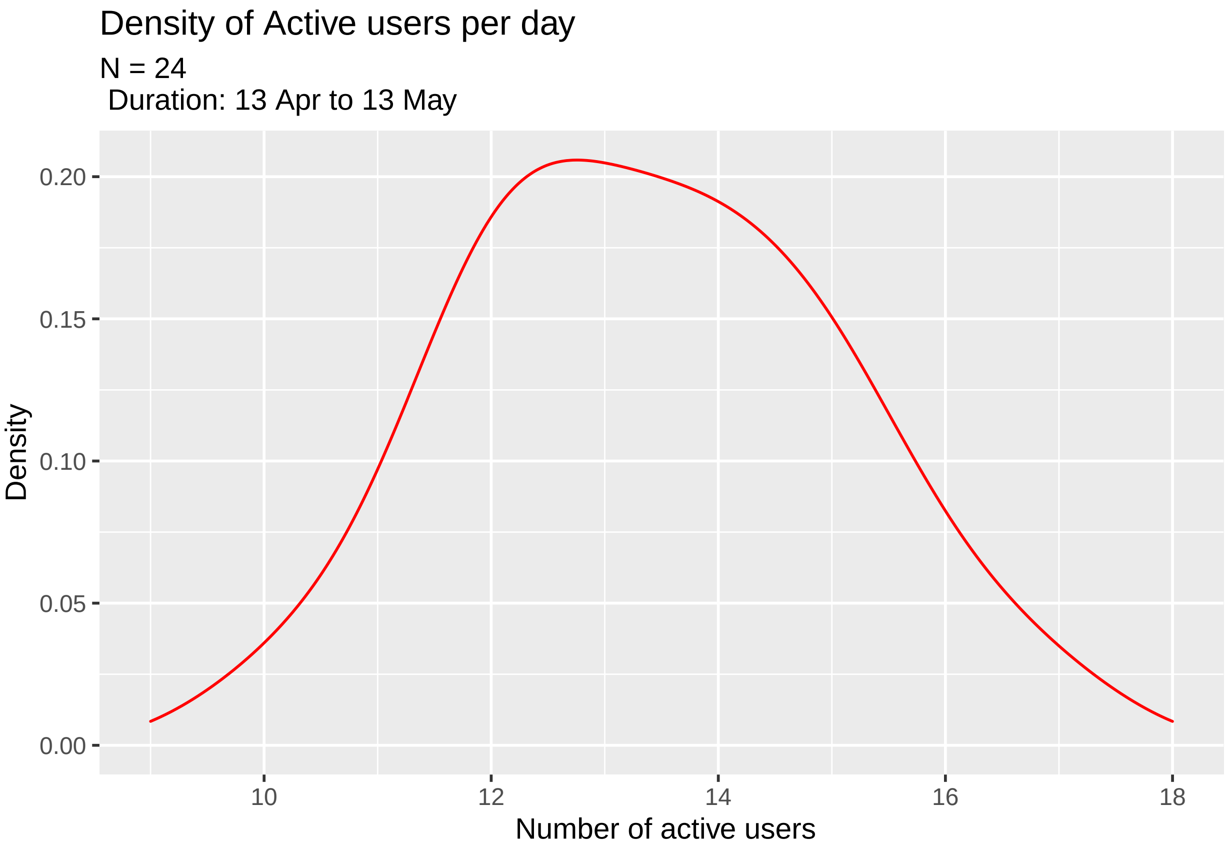
```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



```
ggplot(data = users_per_day) +
  geom_density(aes(x = as.integer(users_count) ), colour = "red", show.legend = FALSE)+
  xlim(9,18)+
  xlab(" Number of active users")+
  ylab("Density")+
  ggtitle("Density of Active users per day",
          subtitle = "N = 24 \n Duration: 13 Apr to 13 May")
```

```
## Warning: Removed 1 rows containing non-finite values (stat_density).
```



according to the density plot, the number of active users per day is normally distributed (bell shaped curve)

we can proceed to check if there is correlation between **weekdays** and **number of Active users**

adding new column weekday and converting the weekday column to be a factor with predefined levels then order the table be weekdays starting from Monday

```
activity_sleep_weight$weekday <- weekdays(activity_sleep_weight$activity_date)
activity_sleep_weight$weekday <- factor(activity_sleep_weight$weekday, levels = c("Monday",
    "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday","Sunday"))

activity_sleep_weight_orderd <- activity_sleep_weight[order(activity_sleep_weight$weekday),]
activity_sleep_weight_orderd
```

```
## # A tibble: 410 x 16
##            id activity_date total_steps total_distance sedentary_minut~ calories
##         <dbl> <date>              <dbl>          <dbl>            <dbl>    <dbl>
##  1 1503960366 2016-04-25          15355           9.80              814     2013
##  2 1503960366 2016-05-02          14727           9.71              798     2004
##  3 1503960366 2016-05-09          12022           7.72              835     1819
##  4 1644430081 2016-05-02           3758           2.73              682     2580
##  5 2026352035 2016-04-25           6017           3.73              821     1576
##  6 2026352035 2016-05-02           7018           4.35              716     1690
##  7 2026352035 2016-05-09          10685           6.62              543     1869
##  8 2347167796 2016-04-18           8247           5.45              678     1944
##  9 2347167796 2016-04-25           9482           6.38              653     2095
## 10 3977333714 2016-04-18          11663           7.80              605     1584
## # ... with 400 more rows, and 10 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>, weekday <fct>
```

```
weekdays_df <- activity_sleep_weight_orderd %>%
  group_by(activity_date,weekday) %>% summarise(user_count= n())
```

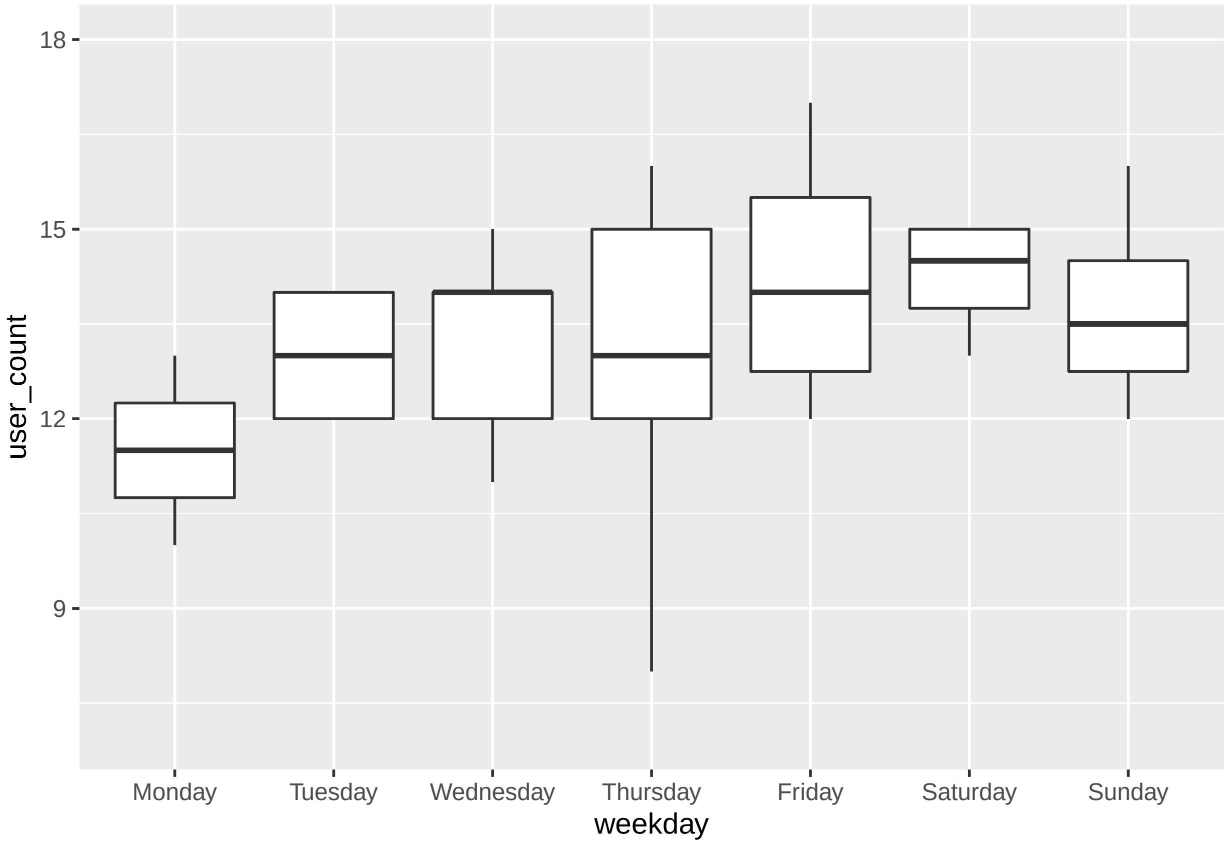
```
## `summarise()` has grouped output by 'activity_date'. You can override using the
## `.groups` argument.
```

```
weekdays_df
```

```
## # A tibble: 31 x 3
## # Groups:   activity_date [31]
##    activity_date weekday   user_count
##    <date>        <fct>          <int>
##  1 2016-04-12    Tuesday           13
##  2 2016-04-13    Wednesday         14
##  3 2016-04-14    Thursday          13
##  4 2016-04-15    Friday            17
##  5 2016-04-16    Saturday          14
##  6 2016-04-17    Sunday            12
##  7 2016-04-18    Monday            10
##  8 2016-04-19    Tuesday           14
##  9 2016-04-20    Wednesday         15
## 10 2016-04-21    Thursday          15
## # ... with 21 more rows
```

boxplot

```
ggplot(data=weekdays_df)+
  geom_boxplot(aes(x=weekday,y = user_count),
               outlier.colour="red",
               outlier.shape=8,
               outlier.size=4)+
  ylim(7,18)
```



according to the boxplot

- Tuesday and Thursday have the same median but number of users is more variable on Thursday. (sometimes it is high and sometimes it is very low)
- Wednesday and Friday have the same median but number of users is more variable on Friday.
- Monday has the lowest median and it is less variable than other workdays (the least number of Active users is on Monday) and there is notable difference between the number of users on Monday and any other day
- Saturday has the highest median but the number of users is the least variable (number of users on Sat differ greatly (higher) than Tuesday and Wednesday)
- Sunday has lower median than Saturday but the number of users is more variable

in conclusion: the number of users start increasing from Tuesday to Friday then on Saturday it the is highest and mostly stable (not variable) also on Sunday it start decreasing until it reaches the lowest number of users by Monday

** as the data is normally distributed we can test the following hypothesis

H: some days have greater number of users than others (number of users is associated with weekdays)

H0: There is no association between number of Active users and weekdays.

---

due to the small sample size we need to combine some days in groups

labeling Tuesday-Friday as Tue_Fri, Saturday& Sunday as Sat_Sun AND Monday will be on it‘s own

```
weekdays_df_2 <- weekdays_df
weekdays_df_2$day_type <- "W"
weekdays_df_2
```

```
## # A tibble: 31 x 4
## # Groups:   activity_date [31]
##    activity_date weekday   user_count day_type
##    <date>        <fct>          <int> <chr>
##  1 2016-04-12    Tuesday           13 W
##  2 2016-04-13    Wednesday         14 W
##  3 2016-04-14    Thursday          13 W
##  4 2016-04-15    Friday            17 W
##  5 2016-04-16    Saturday          14 W
##  6 2016-04-17    Sunday            12 W
##  7 2016-04-18    Monday            10 W
##  8 2016-04-19    Tuesday           14 W
##  9 2016-04-20    Wednesday         15 W
## 10 2016-04-21    Thursday          15 W
## # ... with 21 more rows
```

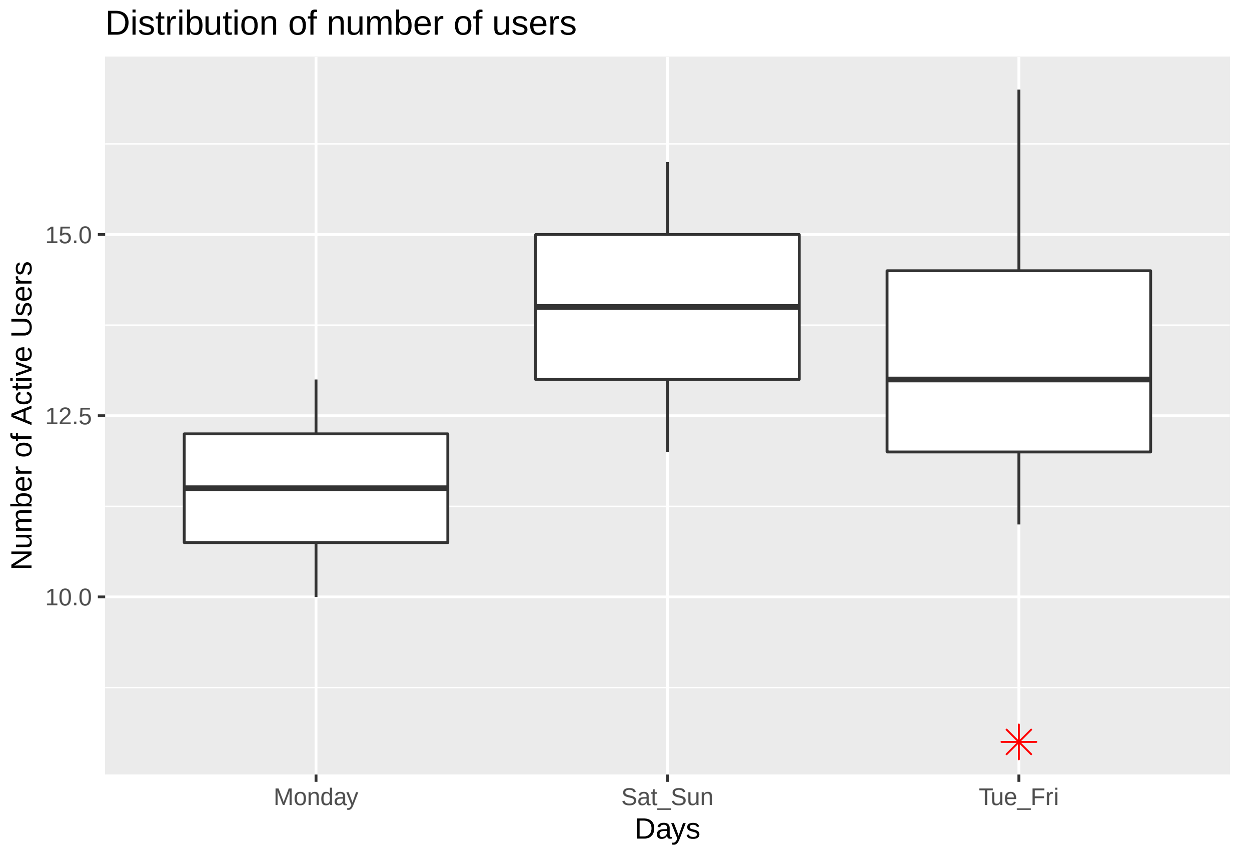
```
Tue_Fri <- c(
    "Tuesday", "Wednesday", "Thursday", "Friday")
Sat_Sun <- c("Saturday","Sunday")


for (i in 1:31){
  #print(i)
  if (weekdays_df_2$weekday[i] %in% Tue_Fri){
    print(i)
    weekdays_df_2[i,]["day_type"] <- "Tue_Fri"
  }
  if (weekdays_df_2$weekday[i] %in% Sat_Sun){
    weekdays_df_2[i,]["day_type"] <- "Sat_Sun"
  }
  if (weekdays_df_2$weekday[i] == "Monday"){
    weekdays_df_2[i,]["day_type"] <- "Monday"
  }
  i = i + 1
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 15
## [1] 16
## [1] 17
## [1] 18
## [1] 22
## [1] 23
## [1] 24
## [1] 25
## [1] 29
## [1] 30
## [1] 31
```

```
weekdays_df_2
```

```
## # A tibble: 31 x 4
## # Groups:   activity_date [31]
##    activity_date weekday   user_count day_type
##    <date>        <fct>          <int> <chr>
##  1 2016-04-12    Tuesday           13 Tue_Fri
##  2 2016-04-13    Wednesday         14 Tue_Fri
##  3 2016-04-14    Thursday          13 Tue_Fri
##  4 2016-04-15    Friday            17 Tue_Fri
##  5 2016-04-16    Saturday          14 Sat_Sun
##  6 2016-04-17    Sunday            12 Sat_Sun
##  7 2016-04-18    Monday            10 Monday
##  8 2016-04-19    Tuesday           14 Tue_Fri
##  9 2016-04-20    Wednesday         15 Tue_Fri
## 10 2016-04-21    Thursday          15 Tue_Fri
## # ... with 21 more rows
```

```
#png("Distribution of number of users.png",
#    width = 800, height = 500)
ggplot(data=weekdays_df_2)+
  geom_boxplot(aes(x=day_type,y = user_count),
               outlier.colour="red",
               outlier.shape=8,
               outlier.size=4) +
  xlab("Days")+
  ylab("Number of Active Users")+
  ggtitle("Distribution of number of users")
```



```
  #ylim(7,18)
#dev.off()
```

Sat-Sun group has the highest median (hence greater number of users) Tue-Fri group has lower median than Sat-Sun Group however it is more variable and has an outlier

Monday has the lowest median (there is huge difference between number of users on Monday and any other day(the lowest))

In conclusion: more users (consistent) use the device on weekends (Sat-Sun) than workdays (Tue-Fri). the least number of users is on Monday (consistent)

We should apply Mann-Whitney U test To test the hypothesis

###Step 1 Data preparation

```
weekdays_test_df <- weekdays_df_2 %>%
  select(user_count,day_type)
```

```
## Adding missing grouping variables: `activity_date`
```

```
weekdays_test_df <- weekdays_test_df %>%
  subset(select= -activity_date)
```

```
weekdays_test_df
```

```
## # A tibble: 31 x 2
##    user_count day_type
##         <int> <chr>
##  1          13 Tue_Fri
##  2          14 Tue_Fri
##  3          13 Tue_Fri
##  4          17 Tue_Fri
##  5          14 Sat_Sun
##  6          12 Sat_Sun
##  7          10 Monday
##  8          14 Tue_Fri
##  9          15 Tue_Fri
## 10          15 Tue_Fri
## # ... with 21 more rows
```

```
weekdays_test_df %>% group_by(day_type) %>%
  summarise(sum=n())
```

```
## # A tibble: 3 x 2
##   day_type   sum
##   <chr>    <int>
## 1 Monday       4
## 2 Sat_Sun      8
## 3 Tue_Fri     19
```

Step 2

select equal samples

Sample size = 4

and for Tue_Fri group only sample size = 8 and sample size =4

```
Sat_Sun_subset <- weekdays_test_df %>%
  filter(day_type == "Sat_Sun")

Tue_Fri_subset <- weekdays_test_df %>%
  filter(day_type == "Tue_Fri")

Monday_subset <- weekdays_test_df %>%
  filter(day_type == "Monday")
```

```
Sat_Sun_subset
```

```
## # A tibble: 8 x 2
##   user_count day_type
##        <int> <chr>
## 1         14 Sat_Sun
## 2         12 Sat_Sun
## 3         15 Sat_Sun
## 4         13 Sat_Sun
## 5         15 Sat_Sun
## 6         16 Sat_Sun
## 7         13 Sat_Sun
## 8         14 Sat_Sun
```

```
Tue_Fri_subset
```

```
## # A tibble: 19 x 2
##    user_count day_type
##         <int> <chr>
##  1         13 Tue_Fri
##  2         14 Tue_Fri
##  3         13 Tue_Fri
##  4         17 Tue_Fri
##  5         14 Tue_Fri
##  6         15 Tue_Fri
##  7         15 Tue_Fri
##  8         13 Tue_Fri
##  9         14 Tue_Fri
## 10         14 Tue_Fri
## 11         16 Tue_Fri
## 12         15 Tue_Fri
## 13         12 Tue_Fri
## 14         12 Tue_Fri
## 15         12 Tue_Fri
## 16         12 Tue_Fri
## 17         12 Tue_Fri
## 18         11 Tue_Fri
## 19          8 Tue_Fri
```

```
Monday_subset
```

```
## # A tibble: 4 x 2
##   user_count day_type
##        <int> <chr>
## 1         10 Monday
## 2         12 Monday
## 3         13 Monday
## 4         11 Monday
```

```
Sat_Sun_sample <- sample_n(Sat_Sun_subset,4)
```

```
Tue_Fri_sample <- sample_n(Tue_Fri_subset,4)
Tue_Fri_sample_8 <- sample_n(Tue_Fri_subset,8)
```

```
Sat_Sun_sample
```

```
## # A tibble: 4 x 2
##   user_count day_type
##        <int> <chr>
## 1         14 Sat_Sun
## 2         16 Sat_Sun
## 3         15 Sat_Sun
## 4         13 Sat_Sun
```

```
Tue_Fri_sample
```

```
## # A tibble: 4 x 2
##   user_count day_type
##        <int> <chr>
## 1         15 Tue_Fri
## 2         11 Tue_Fri
## 3         12 Tue_Fri
## 4         14 Tue_Fri
```

```
Monday_subset
```

```
## # A tibble: 4 x 2
##   user_count day_type
##        <int> <chr>
## 1         10 Monday
## 2         12 Monday
## 3         13 Monday
## 4         11 Monday
```

```
Tue_Fri_sample_8
```

```
## # A tibble: 8 x 2
##   user_count day_type
##        <int> <chr>
## 1          8 Tue_Fri
## 2         15 Tue_Fri
## 3         13 Tue_Fri
## 4         11 Tue_Fri
## 5         12 Tue_Fri
## 6         14 Tue_Fri
## 7         12 Tue_Fri
## 8         14 Tue_Fri
```

```
Sat_Sun_subset
```

```
## # A tibble: 8 x 2
##   user_count day_type
##        <int> <chr>
## 1         14 Sat_Sun
## 2         12 Sat_Sun
## 3         15 Sat_Sun
## 4         13 Sat_Sun
## 5         15 Sat_Sun
## 6         16 Sat_Sun
## 7         13 Sat_Sun
## 8         14 Sat_Sun
```

We have 3 groups (Tue-Fri, Sat-Sun, Monday) so there will be multiple tests for each pair.

Constructing pairs

adding (Sat-Sun) & (Tue-Fri) in one df Sat__Tue__test__df (sample size=8) adding (Sat-Sun) & (Monday) in one df Sat__Monday__test__df (sample size=4) adding (Tue-Fri) & (Monday) in one df Tue__Monday__test__df (sample size=4)

```
Sat_Tue_test_df <-rbind(Sat_Sun_subset, Tue_Fri_sample_8)

Tue_Monday_test_df <-rbind(Tue_Fri_sample, Monday_subset)
```

```
Sat_Monday_test_df <-rbind(Sat_Sun_sample, Monday_subset)
```

```
Sat_Tue_test_df
```

```
## # A tibble: 16 x 2
##    user_count day_type
##         <int> <chr>
##  1         14 Sat_Sun
##  2         12 Sat_Sun
##  3         15 Sat_Sun
##  4         13 Sat_Sun
##  5          15 Sat_Sun
##  6          16 Sat_Sun
##  7          13 Sat_Sun
##  8          14 Sat_Sun
##  9           8 Tue_Fri
## 10          15 Tue_Fri
## 11          13 Tue_Fri
## 12          11 Tue_Fri
## 13          12 Tue_Fri
## 14          14 Tue_Fri
## 15          12 Tue_Fri
## 16          14 Tue_Fri
```

```
Sat_Monday_test_df
```

```
## # A tibble: 8 x 2
##   user_count day_type
##        <int> <chr>
## 1         14 Sat_Sun
## 2         16 Sat_Sun
## 3         15 Sat_Sun
## 4         13 Sat_Sun
## 5         10 Monday
## 6         12 Monday
## 7         13 Monday
## 8         11 Monday
```

```
Tue_Monday_test_df
```

```
## # A tibble: 8 x 2
##   user_count day_type
##        <int> <chr>
## 1         15 Tue_Fri
## 2         11 Tue_Fri
## 3         12 Tue_Fri
## 4         14 Tue_Fri
## 5         10 Monday
## 6         12 Monday
## 7         13 Monday
## 8         11 Monday
```

Saving the dataframes as csv to be used in the analysis

```
Sat_Tue_test_df
```

```
## # A tibble: 16 x 2
##    user_count day_type
##         <int> <chr>
##  1         14 Sat_Sun
##  2         12 Sat_Sun
##  3         15 Sat_Sun
##  4         13 Sat_Sun
##  5         15 Sat_Sun
##  6         16 Sat_Sun
##  7         13 Sat_Sun
##  8         14 Sat_Sun
##  9          8 Tue_Fri
## 10         15 Tue_Fri
## 11         13 Tue_Fri
## 12         11 Tue_Fri
## 13         12 Tue_Fri
## 14         14 Tue_Fri
## 15         12 Tue_Fri
## 16         14 Tue_Fri
```

```
Sat_Monday_test_df
```

```
## # A tibble: 8 x 2
##   user_count day_type
##        <int> <chr>
## 1         14 Sat_Sun
## 2         16 Sat_Sun
## 3         15 Sat_Sun
## 4         13 Sat_Sun
## 5         10 Monday
## 6         12 Monday
## 7         13 Monday
## 8         11 Monday
```

```
Tue_Monday_test_df
```

```
## # A tibble: 8 x 2
##   user_count day_type
##        <int> <chr>
## 1         15 Tue_Fri
## 2         11 Tue_Fri
## 3         12 Tue_Fri
## 4         14 Tue_Fri
## 5         10 Monday
## 6         12 Monday
## 7         13 Monday
## 8         11 Monday
```

```
#write.csv(Sat_Tue_test_df, "./Data/cleaned_data/Sat_Tue_test_df_28_05_2022_v01.csv",row.names = FALSE)

#write.csv(Sat_Monday_test_df, "./Data/cleaned_data/Sat_Monday_test_df_28_05_2022_v01.csv",row.names = 

#write.csv(Tue_Monday_test_df, "./Data/cleaned_data/Tue_Monday_test_df_28_05_2022_v01.csv",row.names = 
```

read dataframes

```
Sat_Tue_test_df_csv <- read.csv("Sat_Tue_test_df_28_05_2022_v01.csv")
Sat_Monday_test_df_csv <- read.csv("Sat_Monday_test_df_28_05_2022_v01.csv")
Tue_Monday_test_df_csv <- read.csv("Tue_Monday_test_df_28_05_2022_v01.csv")
```

changing day_type to a factor and labeling the groups “Sat_Sun” = 1 “Tue_Fri” = 2 “Monday” = 3

```
#attach(Sat_Tue_test_df_csv)
Sat_Tue_test_df_csv$day_type <- factor(Sat_Tue_test_df_csv$day_type, c("Sat_Sun","Tue_Fri"), labels = c
```

```
Sat_Monday_test_df_csv$day_type <- factor(Sat_Monday_test_df_csv$day_type, c("Sat_Sun","Monday"), label
```

```
Tue_Monday_test_df_csv$day_type <- factor(Tue_Monday_test_df_csv$day_type, c("Tue_Fri","Monday"), label
str(Sat_Tue_test_df_csv)
```

```
## 'data.frame':    16 obs. of  2 variables:
##  $ user_count: int  14 12 15 13 15 16 13 14 12 14 ...
##  $ day_type  : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 2 2 ...
```

```
str(Sat_Monday_test_df_csv)
```

```
## 'data.frame':    8 obs. of  2 variables:
##  $ user_count: int  13 16 14 12 10 12 13 11
##  $ day_type  : Factor w/ 2 levels "1","3": 1 1 1 1 2 2 2 2
```

```
str(Tue_Monday_test_df_csv)
```

```
## 'data.frame':    8 obs. of  2 variables:
##  $ user_count: int  8 13 11 13 10 12 13 11
##  $ day_type  : Factor w/ 2 levels "2","3": 1 1 1 1 2 2 2 2
```

```
Sat_Tue_test_df_csv
```

```
##    user_count day_type
## 1          14        1
## 2          12        1
## 3          15        1
## 4          13        1
## 5          15        1
## 6          16        1
## 7          13        1
## 8          14        1
## 9          12        2
## 10         14        2
## 11         11        2
## 12         14        2
## 13         13        2
## 14         12        2
## 15         12        2
## 16         13        2
```

```
Sat_Monday_test_df_csv
```

```
##   user_count day_type
## 1         13        1
## 2         16        1
## 3         14        1
## 4         12        1
## 5         10        3
## 6         12        3
## 7         13        3
## 8         11        3
```

```
Tue_Monday_test_df_csv
```

```
##   user_count day_type
## 1          8        2
## 2         13        2
## 3         11        2
## 4         13        2
## 5         10        3
## 6          12         3
## 7          13         3
## 8          11         3
```

Pair 1 (Sat_sun & Tue_Fri)

H: There is difference between number of users on Saturday & Sunday compared to the number of users on Tuesday to Friday

H0: There is no difference between number of users on Saturday & Sunday compared to the number of users on Tuesday to Friday

showing summary statistics

```
Sat_Tue_test_df_csv %>% group_by(day_type)%>% summarise(median_data= median(user_count), iqr=IQR(user_c
```

```
## # A tibble: 2 x 3
##   day_type median_data   iqr
##   <fct>          <dbl> <dbl>
## 1 1               14    2
## 2 2               12.5  1.25
```

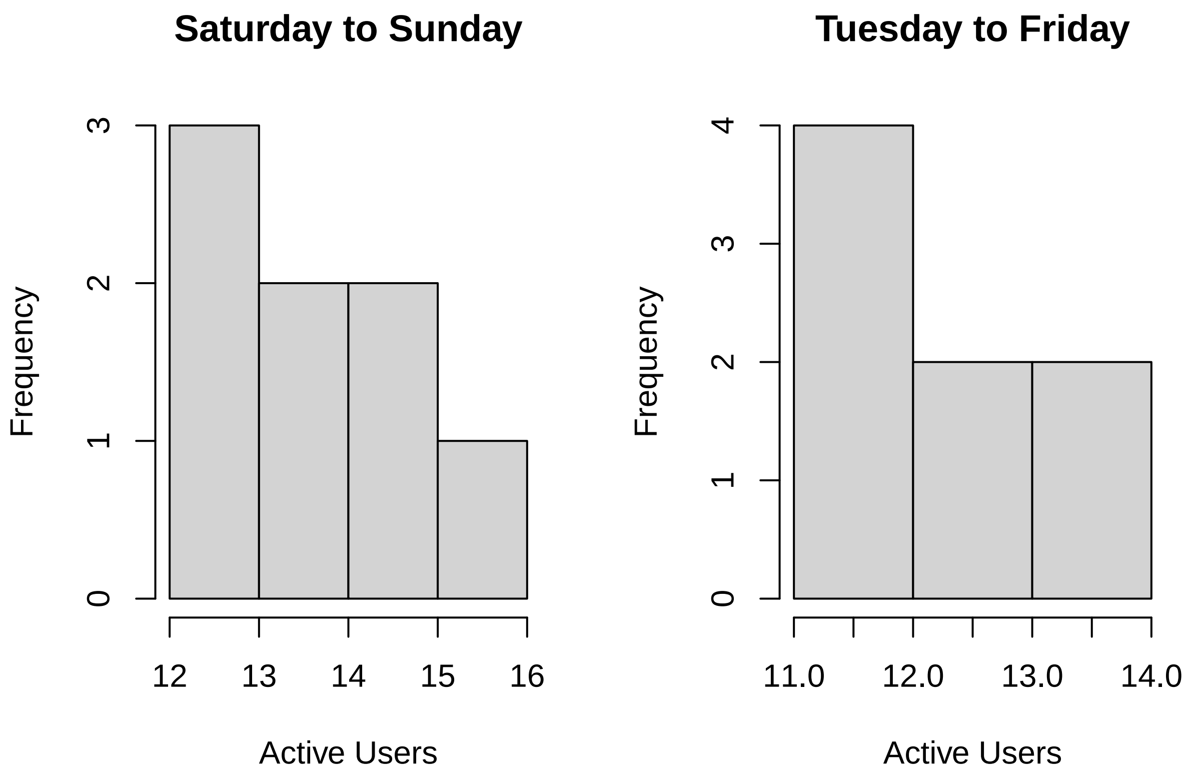
group 2 (Tue_Fri) median and iqr for this sample is less than group 1 Sat_Sun

```
Sat_Tue_test_df_csv
```

```
##    user_count day_type
## 1          14        1
## 2          12        1
## 3          15        1
## 4          13        1
## 5          15        1
## 6          16        1
## 7          13        1
## 8          14        1
## 9          12        2
## 10         14        2
## 11         11        2
## 12         14        2
## 13         13        2
## 14         12        2
## 15         12        2
## 16         13        2
```

```
x_Sat_Sun <- filter(Sat_Tue_test_df_csv,day_type == "1")
x_Tue_Fri <- filter(Sat_Tue_test_df_csv,day_type == "2")
```

```
par(mfrow = c(1,2))
hist(x_Sat_Sun$user_count , main = "Saturday to Sunday", xlab = "Active Users")
hist(x_Tue_Fri$user_count, main = "Tuesday to Friday", xlab = "Active Users")
```



both histograms are positively skewed
so it is better to use medians to summaries the differences between number of users on (Sat_sun / Tue_Fri)
if the histograms looks different we should use the mean

carrying out Mann-Whitney U test

```
wilcox.test(Sat_Tue_test_df_csv$user_count~Sat_Tue_test_df_csv$day_type)
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Sat_Tue_test_df_csv$user_count by Sat_Tue_test_df_csv$day_type
## W = 50.5, p-value = 0.05299
## alternative hypothesis: true location shift is not equal to 0
```

Accepting NULL Hypothesis (H0) for Pair 1 (Sat_sun&Tue_Fri)

H0: There is no difference between number of users on Saturday & Sunday compared to the number of users on Tuesday to Friday

p-value = 0.05299 Reporting Mann-Whitney U test

A Mann-Whitney U test showed that there is no significant difference (W = 50.5, p-value = 0.05299) between number of users on Sat_Sun compared to the number of users on Tue_Fri (there is no huge difference in the medians of the two groups)

the median number of users for Sat_Sun group was 14 and Tue_Fri group was 12.5

Pair 2 (Sat_sun & Monday)

H: There is difference between number of users on Saturday & Sunday compared to the number of users on Monday

H0: There is no difference between number of users on Saturday & Sunday compared to the number of users on Monday

showing summary statistics

```
Sat_Monday_test_df_csv %>% group_by(day_type)%>% summarise(median_data= median(user_count), iqr=IQR(use
```

```
## # A tibble: 2 x 3
##   day_type median_data   iqr
##   <fct>          <dbl> <dbl>
## 1 1               13.5  1.75
## 2 3               11.5  1.5
```

group 3 (Monday) median and iqr for this sample is less than group 1 Sat_Sun

```
Sat_Monday_test_df_csv
```

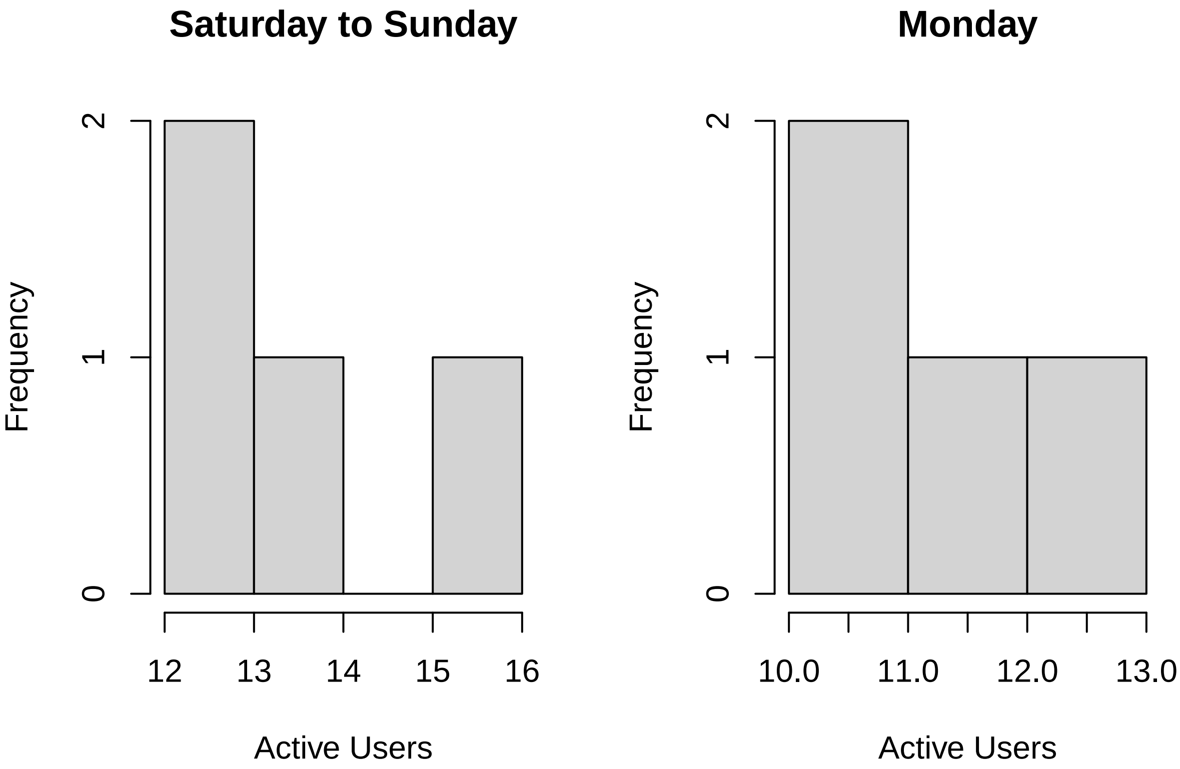
```
##   user_count day_type
## 1         13        1
## 2         16        1
## 3         14        1
## 4         12        1
## 5         10        3
## 6         12        3
## 7         13        3
## 8         11        3
```

```
x_Sat_Sun_2 <- filter(Sat_Monday_test_df_csv,day_type == "1")
x_Monday <- filter(Sat_Monday_test_df_csv,day_type == "3")
```

```
x_Sat_Sun_2
```

```
##   user_count day_type
## 1         13        1
## 2         16        1
## 3         14        1
## 4         12        1
```

```
par(mfrow = c(1,2))
hist(x_Sat_Sun_2$user_count, main = "Saturday to Sunday", xlab = "Active Users")
hist(x_Monday$user_count, main = "Monday", xlab = "Active Users")
```



both histograms are positively skewed
so it is better to use medians to summaries the differences between number of users on (Sat_sun / Monday)

if the histograms looks different we should use the mean

carrying out Mann-Whitney U test

```
wilcox.test(Sat_Monday_test_df_csv$user_count~Sat_Monday_test_df_csv$day_type)
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Sat_Monday_test_df_csv$user_count by Sat_Monday_test_df_csv$day_type
## W = 14, p-value = 0.1081
## alternative hypothesis: true location shift is not equal to 0
```

Accepting NULL Hypothesis (H0) for Pair 2 (Sat_sun&Monday)

H0: There is no difference between number of users on Saturday & Sunday compared to the number of users on Monday

p-value = 0.1081 Reporting Mann-Whitney U test

A Mann-Whitney U test showed that there is no significant difference (W = 14, p-value = 0.1081) between number of users on Sat_Sun compared to the number of users on Monday (there is no huge difference in the medians of the two groups)

Pair 3 (Tue_Fri & Monday)

H: There is difference between number of users on Tuesday to Friday compared to the number of users on Monday

H0: There is no difference between number of users on Tuesday to Friday compared to the number of users on Monday

showing summary statistics

```
Tue_Monday_test_df_csv %>% group_by(day_type)%>% summarise(median_data= median(user_count), iqr=IQR(use
```

```
## # A tibble: 2 x 3
##   day_type median_data   iqr
##   <fct>          <dbl> <dbl>
## 1 2               12    2.75
## 2 3               11.5  1.5
```

group 3 (Monday) median and iqr for this sample is less than group 2 Tue_Fri

```
Sat_Monday_test_df_csv
```

```
##   user_count day_type
## 1         13        1
## 2         16        1
## 3         14        1
## 4         12        1
## 5         10        3
## 6         12        3
## 7         13        3
## 8         11        3
```

```
x_Sat_Sun_3 <- filter(Sat_Monday_test_df_csv,day_type == "1")
x_Monday <- filter(Sat_Monday_test_df_csv,day_type == "3")
```

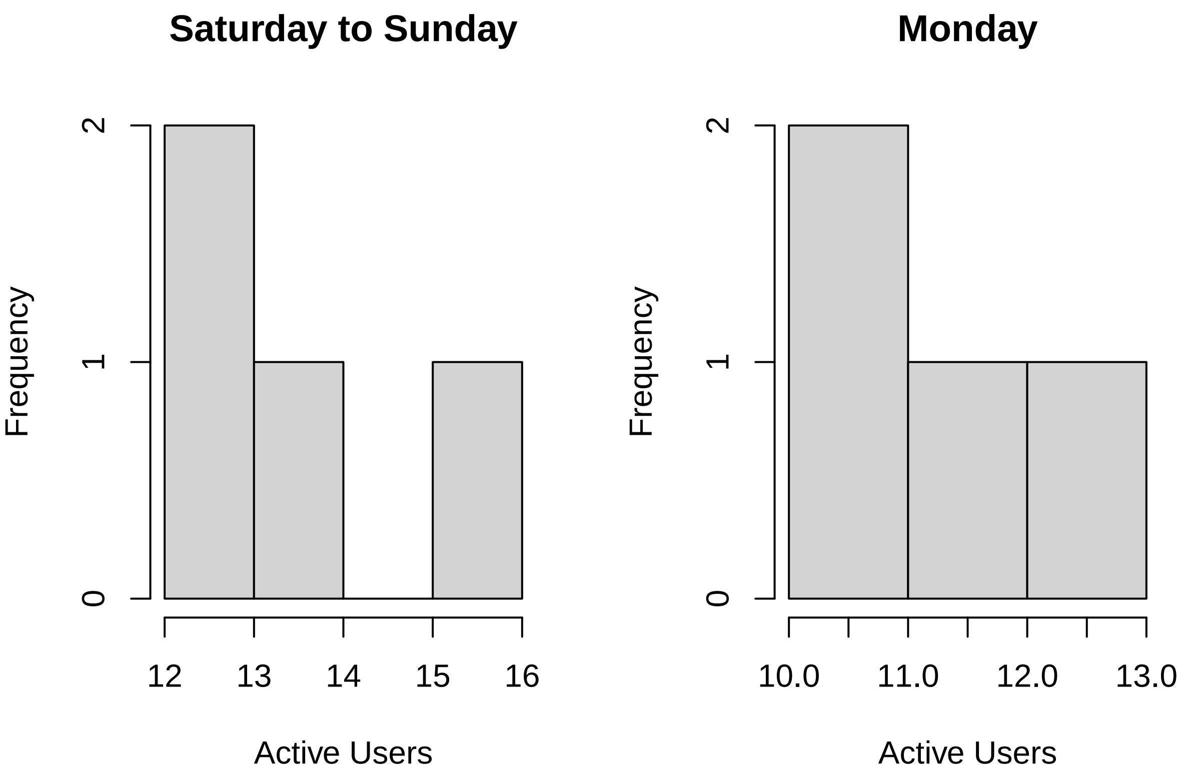
```
x_Sat_Sun_3
```

```
##   user_count day_type
## 1         13        1
## 2         16        1
## 3         14        1
## 4         12        1
```

```
x_Sat_Sun_2
```

```
##   user_count day_type
## 1         13        1
## 2         16        1
## 3         14        1
## 4         12        1
```

```
par(mfrow = c(1,2))
hist(x_Sat_Sun_2$user_count, main = "Saturday to Sunday", xlab = "Active Users")
hist(x_Monday$user_count, main = "Monday", xlab = "Active Users")
```



both histograms are positively skewed
so it is better to use medians to summaries the differences between number of users on (Tue_Fri / Monday)

if the histograms looks different we should use the mean

carrying out Mann-Whitney U test

```
wilcox.test(Tue_Monday_test_df_csv$user_count~Tue_Monday_test_df_csv$day_type)
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Tue_Monday_test_df_csv$user_count by Tue_Monday_test_df_csv$day_type
## W = 8.5, p-value = 1
## alternative hypothesis: true location shift is not equal to 0
```

Accepting NULL Hypothesis (H0) for Pair 3 (Tue_Fri&Monday)

H0: There is no difference between number of users on Tuesday to Friday compared to the number of users on Monday

p-value = 1 Reporting Mann-Whitney U test

A Mann-Whitney U test showed that there is no significant difference (W = 8.5, p-value = 1) between number of users on Tue_Fri compared to the number of users on Monday (there is no huge difference in the medians of the two groups)

in conclusion: We accept the NULL Hypothesis H0: There is no association between number of Active users and weekdays.

NOTE: Due to the small sample size 24 (N<30)the results might not be much accurate. We need more data to apply this result on the population.