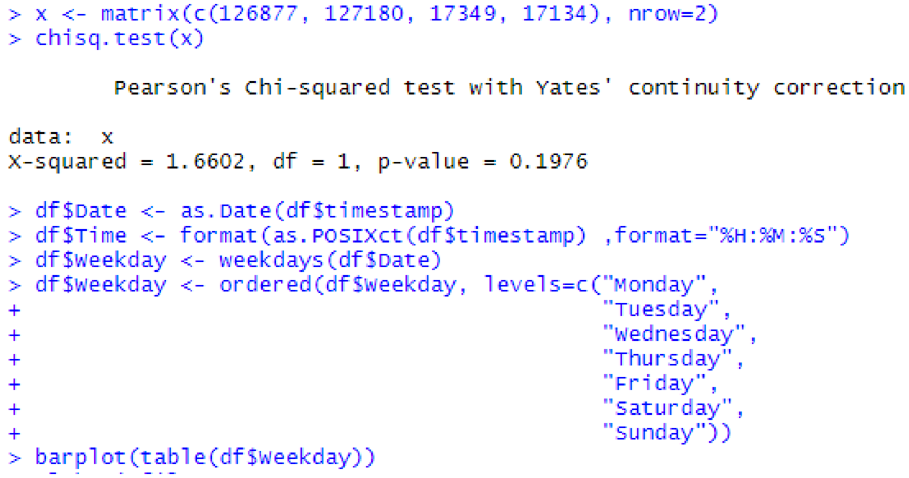
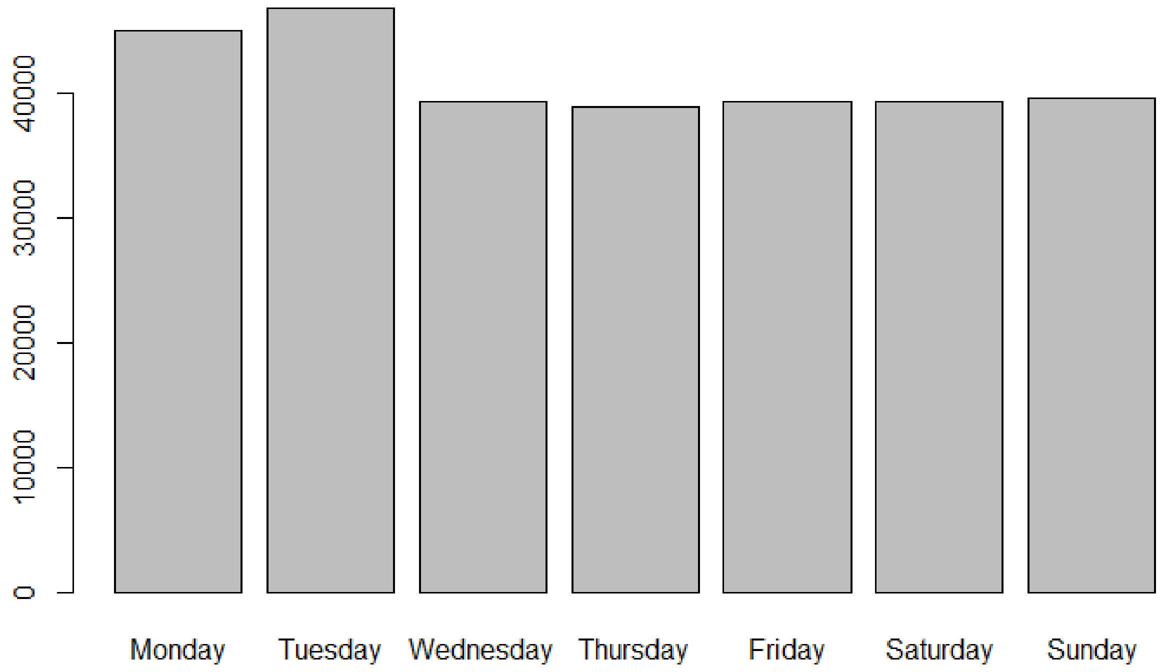
```
> packages <- c("plyr","dplyr","ggplot2")
> lapply(packages, library, character.only = TRUE)

Attaching package: ‘dplyr’

The following objects are masked from ‘package:plyr’:

    arrange, count, desc, failwith, id, mutate, rename, summarise, summarize

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union

[[1]]
[1] "plyr"      "printr"    "stats"     "graphics"  "grDevices" "utils"     "datasets"
[8] "methods"   "base"

[[2]]
 [1] "dplyr"     "plyr"      "printr"    "stats"     "graphics"  "grDevices" "utils"
 [8] "datasets"  "methods"   "base"

[[3]]
 [1] "ggplot2"   "dplyr"     "plyr"      "printr"    "stats"     "graphics"  "grDevices"
 [8] "utils"     "datasets"  "methods"   "base"
> df <- read.csv("D:/DATAMINING/ab_data.csv")
> str(df)
'data.frame':	294478 obs. of  5 variables:
 $ user_id     : int  851104 804228 661590 853541 864975 936923 679687 719014 817355 839785 ...
 $ timestamp   : Factor w/ 294478 levels "2017-01-02 13:42:05.378582",..: 258966 131019 122512 8287
1 247589 108211 221819 194300 29089 176584 ...
 $ group       : Factor w/ 2 levels "control","treatment": 1 1 2 2 1 1 2 1 2 2 ...
 $ landing_page: Factor w/ 2 levels "new_page","old_page": 2 2 1 1 2 2 1 2 1 1 ...
 $ converted   : int  0 0 0 0 1 0 1 0 1 1 ...
> colSums(is.na(df))
     user_id    timestamp        group landing_page    converted 
           0            0            0            0            0 
> unique_id <- unique(df$user_id)
> length(unique_id)
[1] 290584
> df <- df[!duplicated(df$user_id), ]
> summary(df$group)
  control treatment 
   145232    145352 
> summary(df$landing_page)
new_page old_page 
  145320   145264 
> freqgrouplanding_page <- ddply(df, .(df$group, df$landing_page), nrow)
> names(freqgrouplanding_page) <- c("group", "landing_page", "Freq")
> freqgrouplanding_page
      group landing_page   Freq
1   control     new_page   1006
2   control     old_page 144226
3 treatment     new_page 144314
4 treatment     old_page   1038
> dfclean1 <- dplyr::filter(df, group == "control" & landing_page == "old_page")
> dfclean2 <- dplyr::filter(df, group == "treatment" & landing_page == "new_page")
> df <- rbind(dfclean1, dfclean2)
> dfgrouplanding_page <- ddply(df, .(df$group, df$landing_page), nrow)
> names(dfgrouplanding_page) <- c("group", "landing_page", "Freq")
> dfgrouplanding_page
      group landing_page   Freq
1   control     old_page 144226
2 treatment     new_page 144314
> p <- ggplot(dfgrouplanding_page, aes(x = landing_page, y = Freq))+
+   geom_col(aes(fill = group), width = 0.7)
> p
> groupconvertfreq <- ddply(df, .(df$group, df$converted), nrow)
> names(groupconvertfreq) <- c("group", "converted", "Freq")
> groupconvertfreq
      group converted   Freq
1   control         0 126877
2   control         1  17349
3 treatment         0 127180
4 treatment         1  17134
```



```
> x <- matrix(c(126877, 127180, 17349, 17134), nrow=2)
> chisq.test(x)

        Pearson's Chi-squared test with Yates' continuity correction

data:  x
X-squared = 1.6602, df = 1, p-value = 0.1976

> df$Date <- as.Date(df$timestamp)
> df$Time <- format(as.POSIXct(df$timestamp) ,format="%H:%M:%S")
> df$Weekday <- weekdays(df$Date)
> df$Weekday <- ordered(df$Weekday, levels=c("Monday",
+                                             "Tuesday",
+                                             "Wednesday",
+                                             "Thursday",
+                                             "Friday",
+                                             "Saturday",
+                                             "Sunday"))
> barplot(table(df$Weekday))
```


Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday