```r
######## DESCRIPTIVES ###########

# Make a data.frame containing summary statistics of interest
summ_stats <- fBasics::basicStats(hour[, -c("dteday")])
summ_stats <- as.data.frame(t(summ_stats))

# Rename some of the columns for convenience
summ_stats <- summ_stats[c("Mean", "Stdev", "Minimum", "1. Quartile",
                           "Median",  "3. Quartile", "Maximum")]
colnames(summ_stats)[colnames(summ_stats) %in% c('1. Quartile',
                                                 '3. Quartile')] <- c('Lower quartile',
                                                                      'Upper quartile')

row.names(summ_stats) <- c("Fall", "Spring", "Summer", "Winter",
                           "Year", "Month", "Hour",
                           "Holiday", "Day of Week", "Workday",
                           "Clear Weather", "Misty Weather",
                           "Rainy Weather", "Thunderstorm", "Temperature",
                           "Felt Temperature", "Humidity", "Windspeed", "Casual Users",
                           "Registered Users", "All Users")
# save
stargazer(summ_stats,
          type = "latex",
          summary=FALSE, rownames=TRUE,
          digits = 2) -> sumstats

tabular_positions <- grep("tabular", sumstats)
sumstats <- sumstats[tabular_positions[1]:tabular_positions[2]]
write(sumstats,  file="output/tables/summary_stats.tex")

######### BY DAY

# Make a data.frame containing summary statistics of interest
summ_stats <- fBasics::basicStats(sumstats_day[, -c("dteday", "yr", "casual",
                                                    "registered", "mnth")])
summ_stats <- as.data.frame(t(summ_stats))

# Rename some of the columns for convenience
summ_stats <- summ_stats[c("Mean", "Stdev", "Minimum",  "Median",  "Maximum")]
colnames(summ_stats)[colnames(summ_stats) %in% c('Minimum',
                                                 'Maximum')] <- c('Min.',
                                                                  'Max.')

row.names(summ_stats) <- c("Fall", "Spring", "Summer", "Winter",
                           "Holiday", "Day of Week", "Workday", "Clear Weather",
                           "Misty Weather",
                           "Rainy Weather",  "Temperature",
                           "Felt Temperature", "Humidity", "Windspeed",
                           "All Users")



# save
```

```r
stargazer(summ_stats,
          type = "latex",
          summary=FALSE, rownames=TRUE,
          digits = 2) -> sumstats

tabular_positions <- grep("tabular", sumstats)
sumstats <- sumstats[tabular_positions[1]:tabular_positions[2]]
sumstats[1] <- "\\begin{tabular}{@{\\extracolsep{5pt}} lccccc} "
write(sumstats,  file="output/tables/summary_stats_day.tex")




hour_plot <- hour
colnames(hour_plot) <- c("Date", "Fall", "Spring", "Summer", "Winter",
                         "Year", "Month", "Hour",
                         "Holiday", "Day of Week", "Workday", "Clear Weather",
                         "Misty Weather",
                         "Rainy Weather", "Thunderstorm", "Temperature",
                         "Felt Temperature", "Humidity", "Windspeed",
                         "Casual Users",
                         "Registered Users", "All Users")

# correlation plot
pairwise_pvalues <- psych::corr.test(hour_plot, hour_plot)$p
png(filename="output/plots/corrplot.png")
pairwise_pvalues <- psych::corr.test(hour_plot, hour_plot)$p
corrplot(cor(hour_plot),
         type="upper",
         tl.col="black",
         order="hclust",
         tl.cex=0.6,
         addgrid.col = "black",
         p.mat=pairwise_pvalues,
         sig.level=0.05,
         number.font=10,
         insig="blank")
dev.off()

# plot daily trends
day %>%
  ggplot(aes(dteday, cnt)) + geom_point(size = 0.5) +
  xlab("Day") + ylab("Total Number of Users")  +
  theme_linedraw() +
  theme(axis.ticks = element_blank())

ggsave("output/plots/users_by_day.png", width = 8, height = 4)

# average number of users by hour
users_by_hour <- hour[, .(avg_users_by_hours = mean(cnt)), by = c("hr")]

users_by_hour %>%
  ggplot(aes(hr, avg_users_by_hours)) + geom_line(size = 0.5) +
```

```
  xlab("Hour") + ylab("Average Number of Users")
ggsave("output/plots/average_number_users_by_hour.png")

# average number of users by weekday
users_by_weekday <- day[, .(avg_users_by_weekday = mean(cnt)), by = c("weekday")]

users_by_weekday %>%
  ggplot(aes(weekday, avg_users_by_weekday)) + geom_line(size = 0.5) +
  xlab("Day") + ylab("Average Number of by Weekday")
ggsave("output/plots/average_number_users_by_weekday.png")
```