Create and Alter Date-Time Objects From Strings

Step 1: Load your data.

```
library(tidyverse)
calls <- read.csv("calls.csv", stringsAsFactors = FALSE)

# Prepare data:

# Initialize vectors to store each of the new variables
address <- c()
town <- c()
dt <- c()

for(i in 1:nrow(calls)) { # loop over emergency calls

  # get the description of the i^th call
  callI <- calls[i, "desc"]

  # split the description text based on ";" --> gives a matrix of substrings
  splitCallDesc <- str_split(callI, ";", simplify = TRUE)

  # store the street address, town, and date/time
  address[i] <- splitCallDesc[1]
  town[i] <- splitCallDesc[2]
  dt[i] <- splitCallDesc[3]

}

# add the new variables to the data frame
calls$address <- address
calls$towns <- town
calls$dt <- dt

# For plot:
theme <-  theme(plot.margin = margin(5, 5, 5, 5, "pt"),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        panel.border = element_rect(colour = "#393f47", fill = NA, size = 2),
        axis.text = element_text(size = 30),
        axis.title.x = element_text(size = 30),
        axis.title.y = element_text(size = 30),
        plot.title = element_text(face = "bold", size = 30))
ourTheme <- list(theme, scale_size_manual(values = c(1.5)))
```

Step 2: Examine your data.

Look at the first few rows of the data. Notice that the dt variable contains date-times.

```
head(calls$dt)
```

```
## [1] " 2020-01-01 @ 00:04:06" " 2020-01-01 @ 00:02:25" " 2020-01-01 @ 00:07:21"
## [4] " 2020-01-01 @ 00:07:53" " 2020-01-01 @ 00:20:15" " 2020-01-01 @ 00:20:36"
```

Step 3: Load the lubridate package.

This package is useful for working with date-times.

```
library(lubridate)
```

Step 4: Convert date-times from strings to POSIXct objects.

```
# convert date-times from strings to POSIXct objects
# need to specify the order in which the components appear -- in this case, year-month-day-hour-minute-
calls$dtBetter <- ymd_hms(calls$dt)
head(calls$dtBetter)
```

```
## [1] "2020-01-01 00:04:06 UTC" "2020-01-01 00:02:25 UTC"
## [3] "2020-01-01 00:07:21 UTC" "2020-01-01 00:07:53 UTC"
## [5] "2020-01-01 00:20:15 UTC" "2020-01-01 00:20:36 UTC"
```

Step 5: Set the time zone to follow New York.

```
# convert date-times from strings to POSIX objects, but with the time zone specified
temp <- ymd_hms(calls$dt, tz = "America/New_York")
head(temp)
```

```
## [1] "2020-01-01 00:04:06 EST" "2020-01-01 00:02:25 EST"
## [3] "2020-01-01 00:07:21 EST" "2020-01-01 00:07:53 EST"
## [5] "2020-01-01 00:20:15 EST" "2020-01-01 00:20:36 EST"
```

Step 6: Extract components from the date-time of the first call.

```
# Date and time of first call
firstCall <- calls$dtBetter[1]
firstCall
```

```
## [1] "2020-01-01 00:04:06 UTC"
```

```
# Pull out components
year(firstCall)
```

```
## [1] 2020
```

```
month(firstCall)
```

```
## [1] 1
```

```
day(firstCall)
```

```
## [1] 1
```

```
hour(firstCall)
```

```
## [1] 0
```

```
minute(firstCall)
```

```
## [1] 4
```

```
second(firstCall)
```

```
## [1] 6
```

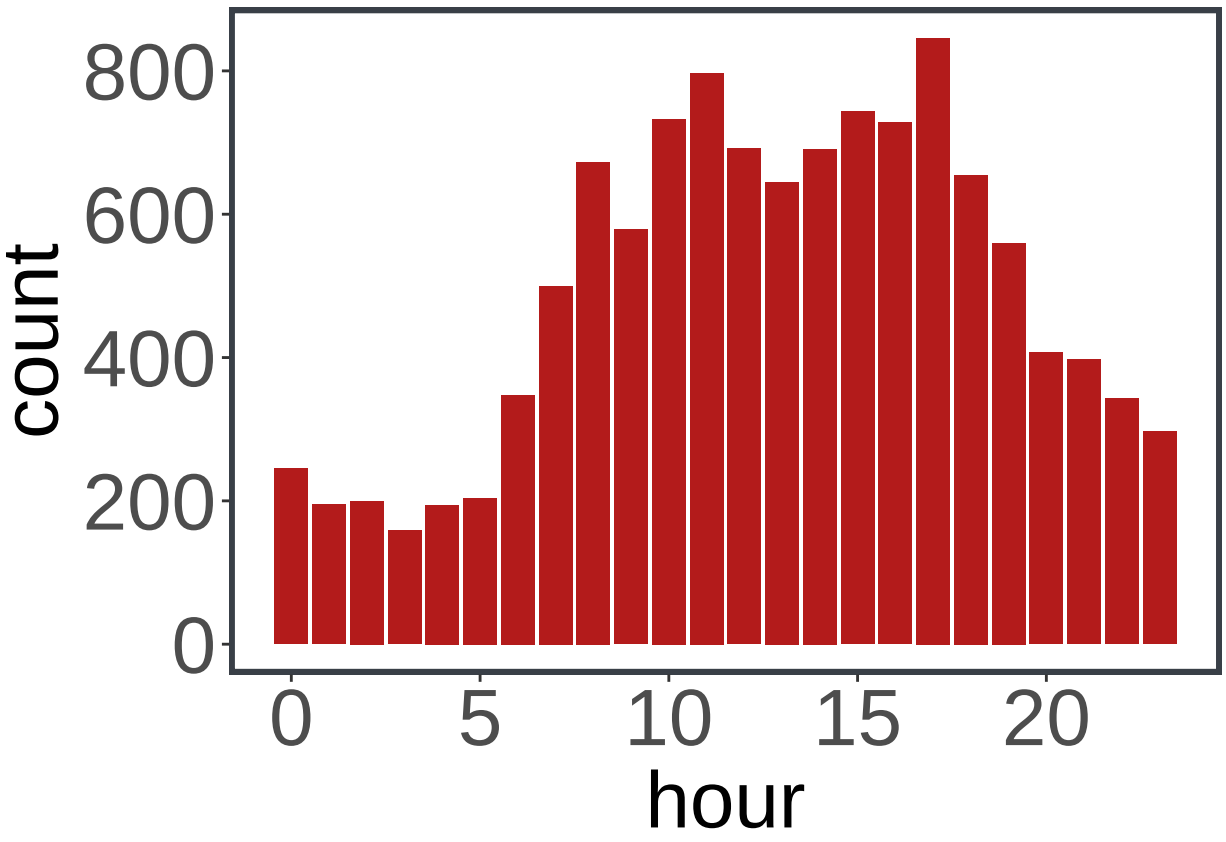
Step 7: Make a new variable giving the hour of each 911 call.

```
# create a new variable called "hour" that gives the hour (0-23) of the emergency call
calls$hour <- hour(calls$dtBetter)
head(calls)
```

```
##                                                                    desc   zip
## 1      BRADFIELD RD & SUSQUEHANNA RD; ABINGTON; 2020-01-01 @ 00:04:06 19001
## 2 E CITY AVE & PRESIDENTIAL BLVD; LOWER MERION; 2020-01-01 @ 00:02:25 19004
## 3            MAPLE AVE AND W 6TH ST;  LANSDALE; 2020-01-01 @ 00:07:21 19446
## 4                        DEKALB ST; BRIDGEPORT; 2020-01-01 @ 00:07:53    NA
## 5                         BEECH ST;  POTTSTOWN; 2020-01-01 @ 00:20:15 19464
## 6           DEKALB ST AND W 5TH ST; BRIDGEPORT; 2020-01-01 @ 00:20:36 19405
##                       title                         address         towns
## 1          Fire: FIRE ALARM  BRADFIELD RD & SUSQUEHANNA RD      ABINGTON
## 2 Traffic: VEHICLE ACCIDENT E CITY AVE & PRESIDENTIAL BLVD  LOWER MERION
## 3          EMS: LACERATIONS         MAPLE AVE AND W 6TH ST      LANSDALE
## 4    Fire: WOODS/FIELD FIRE                      DEKALB ST    BRIDGEPORT
## 5             EMS: STABBING                       BEECH ST     POTTSTOWN
## 6 Traffic: VEHICLE ACCIDENT         DEKALB ST AND W 5TH ST    BRIDGEPORT
##                         dt            dtBetter hour
## 1  2020-01-01 @ 00:04:06 2020-01-01 00:04:06    0
## 2  2020-01-01 @ 00:02:25 2020-01-01 00:02:25    0
## 3  2020-01-01 @ 00:07:21 2020-01-01 00:07:21    0
## 4  2020-01-01 @ 00:07:53 2020-01-01 00:07:53    0
## 5  2020-01-01 @ 00:20:15 2020-01-01 00:20:15    0
## 6  2020-01-01 @ 00:20:36 2020-01-01 00:20:36    0
```

Step 8: Create a barplot that illustrates how call volume changes over the course of the day.

```
# make a barplot of the number of calls in each hour
ggplot(calls, aes(x = hour)) + geom_bar(fill = "#b31b1b") + ourTheme
```


800
600
400
200
0
count
0
5
10
15
20
hour