lubridate Practical Solutions

Jumping Rivers

As usual, let's load the packages and data needed for this practical.

```
library("dplyr")
library("lubridate")
library("ggplot2")
data(okcupid, package = "jrTidyverse")
```

When were you born? (you can lie if you want to)

1. Store your birth date as a character variable i.e.

   ```
   bday = "11/04/1967"
   ```

2. Convert it into a date object using `dmy`

   ```
   bday = dmy(bday)
   ```

3. Which day of the week were you born on? Hint: Use `wday()`. Notice R returns the weekday as a number. To clarify this, set the argument `label` equal to `TRUE` inside `wday`.

   ```
   wday(bday, label = TRUE)
   ```

   ```
   ## [1] Tue
   ## Levels: Sun < Mon < Tue < Wed < Thu < Fri < Sat
   ```

4. Using the `year()` function, change the year of your date object to your next birthday. What day is that on?

```
year(bday) = 2018
wday(bday, label = TRUE)
```

```
## [1] Wed
## Levels: Sun < Mon < Tue < Wed < Thu < Fri < Sat
```

5. How many days is it until your next birthday? What about seconds since you were born? Hint: Use `interval` then use the `unit` argument inside `as.period()`

```
today = today()
as.period(interval(today, bday), unit = "year")
```

```
## [1] "-1m -19d 0H 0M 0S"
```

```
as.period(interval(today, bday), unit = "day")
```

```
## [1] "-49d 0H 0M 0S"
```

```
as.period(interval(today, bday), unit = "seconds")
```

```
## [1] "-4233600S"
```

OKCupid

Take our OKcupid data, let's say we want to look at the distribution of the weekday of people's last online time. Effectively asking the question "Which day of the week do people use OKCupid most on?"

1. Using `mutate()` and `ymd_hms()` convert the `last_online` column to a proper date. Hint, remember to set the time zone in the `ymd_hms()` via `tz = "America/Los_Angeles"`.

```
okcupid = okcupid %>%
  mutate(last_online = ymd_hms(last_online, tz = "America/Los_Angeles"))
```

2. Create a new column called `week_day` that contains the day of the week a user accessed OKCupid. Hint: use `mutate()` and `wday()`

```
okcupid = okcupid %>%
    mutate(week_day = wday(last_online, label = TRUE))
```

3. Create a bar chart of the day of the week using `geom_bar()`. Which days are most popular?

```
ggplot(okcupid, aes(x = week_day)) +
  geom_bar() +
  xlab("Week day") +
  ylab("Count")
```

```
# friday and saturday are the two most popular
```

4. How does this compare for men and women?

```
# either use a graph to find out
ggplot(okcupid, aes(x = week_day)) +
  geom_bar() +
  xlab("Week day") +
  ylab("Count") +
  facet_wrap(~sex)

# or a summary data frame

okcupid %>%
  group_by(sex) %>%
  count(week_day)
```

```
## # A tibble: 14 x 3
## # Groups:   sex [2]
##    sex   week_day     n
##    <chr> <ord>    <int>
##  1 f     Sun        343
##  2 f     Mon        276
##  3 f     Tue        310
##  4 f     Wed        365
##  5 f     Thu        447
##  6 f     Fri        610
##  7 f     Sat        762
##  8 m     Sun        957
##  9 m     Mon        816
## 10 m     Tue        860
## 11 m     Wed        912
## 12 m     Thu        999
## 13 m     Fri       1573
```

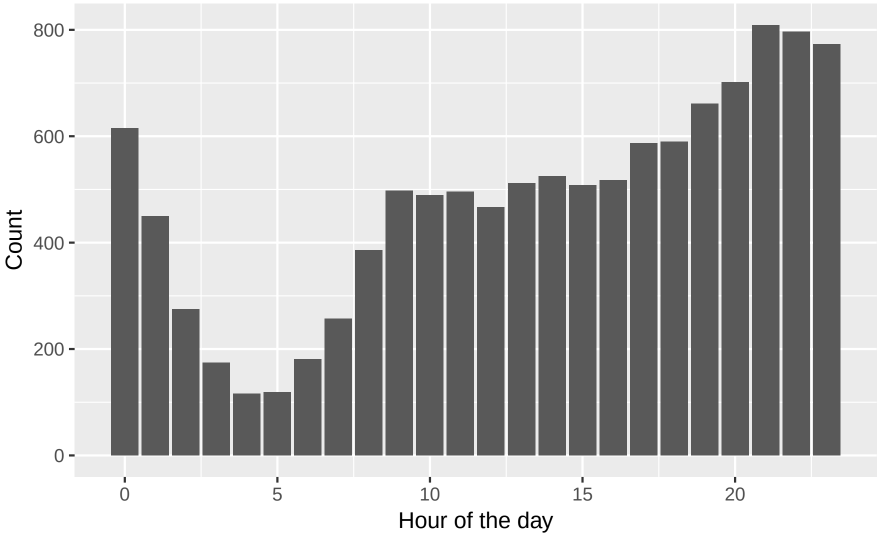

Figure 1: Distribution of access times.

```
## 14 m     Sat        2274
```

5. Create a bar chart showing the distribution for the hour of the day okcupid users were last online? You should end up with something like the figure below

```
okcupid = okcupid  %>%
    mutate(lo_hour = hour(last_online))
```