Solution to exercise 4

Data manipulation

Manipulate the `temperature` dataset

i) Load the dataset `temperature.csv` from the `01_Data` folder and assign it to an object with a meaningful name

```
# Set the working directory
setwd("~/R_Basic_Introduction/01_Data") # replace with your path to the folder "01_Data"
```

```
# Load data
temperature <- read.csv(file = "temperature.csv")

# Get an overview of dataset
head(temperature) # show the first six rows
```

```
##     site       temp day month
## 1 Zurich -2.6652164   6     1
## 2 Zurich -1.1469265   7     1
## 3 Zurich  1.9932443   8     1
## 4 Zurich  0.9122417   9     1
## 5 Zurich -4.1277218  10     1
## 6 Zurich -3.5909123  11     1
```

```
tail(temperature) # show the last six rows
```

```
##     site      temp day month
## 175 Bern  2.194550   9     4
## 176 Bern  4.683131  10     4
## 177 Bern  7.688624  11     4
## 178 Bern  4.467412  12     4
## 179 Bern  6.198005  13     4
## 180 Bern 13.362449  14     4
```

```
str(temperature) # show the structure of the dataset
```

```
## 'data.frame':    180 obs. of  4 variables:
##  $ site : Factor w/ 2 levels "Bern","Zurich": 2 2 2 2 2 2 2 2 2 2 ...
##  $ temp : num  -2.665 -1.147 1.993 0.912 -4.128 ...
##  $ day  : int  6 7 8 9 10 11 21 22 23 24 ...
##  $ month: int  1 1 1 1 1 1 1 1 1 1 ...
```

ii) Filter the dataset for the site `Bern`

```
# Filter for site Bern
temp_be <- subset(temperature, site == "Bern")
unique(temp_be$site) # check the result
```

```
## [1] Bern
## Levels: Bern Zurich
```

iii) Add a column with the variable `year` (the year is `2013`)

```
# Add year column
temp_be$year <- 2013
head(temp_be) # check the result
```

```
##    site       temp day month year
## 91 Bern  0.6474913   6     1 2013
## 92 Bern  1.3407568   7     1 2013
## 93 Bern  5.1327719   8     1 2013
## 94 Bern  3.8262902   9     1 2013
## 95 Bern -0.7606448  10     1 2013
## 96 Bern -1.6092732  11     1 2013
```

iv) Create a new `date` column

- Create a new column that is a combination of the variables `year`, `month` and `day` (in the form of `"2013-01-25"`)
- Convert the `class` of the column from `"character"` to `"Date"`
- Calculate the number of days between the first and last measurement

```
# Create a date vector
date_vec <- paste(temp_be$year, temp_be$month, temp_be$day, sep = "-")
head(date_vec)
```

```
## [1] "2013-1-6"  "2013-1-7"  "2013-1-8"  "2013-1-9"  "2013-1-10" "2013-1-11"
```

```
# Add the date vector as a column to the dataset
temp_be$date <- date_vec
head(temp_be)
```

```
##    site       temp day month year      date
## 91 Bern  0.6474913   6     1 2013  2013-1-6
## 92 Bern  1.3407568   7     1 2013  2013-1-7
## 93 Bern  5.1327719   8     1 2013  2013-1-8
## 94 Bern  3.8262902   9     1 2013  2013-1-9
## 95 Bern -0.7606448  10     1 2013 2013-1-10
## 96 Bern -1.6092732  11     1 2013 2013-1-11
```

```r
# Change the class of the 'date' column to "Date"
class(temp_be$date)
```

```
## [1] "character"
```

```r
temp_be$date <- as.Date(temp_be$date)
class(temp_be$date)
```

```
## [1] "Date"
```

```r
# Calculate the number of days between first and last measurement
time_diff <- max(temp_be$date) - min(temp_be$date)
time_diff
```

```
## Time difference of 98 days
```

v) Calculate the average temperature for periods without frost (i.e. the temperature is above `0 °C`) for the site `Zurich`

```r
# Filter the dataset for the site Zurich
temp_zh <- subset(temperature, site == "Zurich")

# Filter the dataset for periods without frost
temp_zh_nofrost <- subset(temp_zh, temp > 0)

# Example of a combination of the two filter arguments above
temp_zh_nofrost <- subset(temperature, site == "Zurich" & temp > 0)

# Calculate the average temperature
mean(temp_zh_nofrost$temp)
```

```
## [1] 2.561974
```

vi) Load the internal dataset `airquality` and change the column names to lower case

```
# Load airquality dataset
data(airquality)

# Assign the dataset to a new object
airquality_1 <- airquality

# Option 1: manual change
colnames(airquality_1) <- c("ozone", "solar.r", "wind", "temp", "month",
                            "day")
head(airquality_1) # check the result
```

```
##   ozone solar.r wind temp month day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

To find a function for a specific task it is usually helpful to Google for answers. For this specific exercise I would use the following query:
r convert column names to lower case

```
# Option 2: use function 'tolower'
?tolower
colnames(airquality_1) <- tolower(colnames(airquality))
head(airquality_1) # check the result
```

```
##   ozone solar.r wind temp month day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```