Aggregating Buoy Data using dplyr and tidyr

Set the working directory. Make a folder **../G18** somewhere on your computer. You will be using this directory for the rest of the day.
Mac: `setwd("~/Dropbox/G18/")`
PC: `setwd("C:/Users/hilarydugan/Dropbox/G18/")`

Load required libraries

```
library(readr) # read in data
library(dplyr,quietly = T) # clean data
library(tidyr) # clean data
library(rLakeAnalyzer) # lake analyses
library(lubridate) # working with time
library(LakeMetabolizer) # lake analyses
```

Load datasets

```
met = read_csv('data/Sparkling2014domet_3month.csv')
```

```
## Warning: 151893 parsing failures.
## row        col   expected   actual
##   1 sampletime valid date 12:00:00
##   2 sampletime valid date 12:01:00
##   3 sampletime valid date 12:02:00
##   4 sampletime valid date 12:03:00
##   5 sampletime valid date 12:04:00
## ... .......... .......... ........
## .See problems(...) for more details.
```

There is an error that states that the function expects the column 'sampletime' to be a 'date'. Looking at the file, 'sampletime' is a time. Therefore, tell the function `read_csv` the class of sampletime should be a character. **Can use a compact string representation where each character represents one column: c = character, i = integer, n = number, d = double, l = logical, D = date, T = date time, t = time, ? = guess, or _/- to skip the column**

```
met = read_csv('data/Sparkling2014domet_3month.csv',col_types = cols(sampletime = 'c'))
buoy = read_csv('data/Sparkling2014wtemp_3month.csv',col_types = cols(sampletime = 'c'))
```

Aggregate buoy data

Use dplyr to convert data from minute to hourly.

```
buoyHourly <- buoy %>%
  mutate(hour = hour(sampledate)) %>% # create new column for hour of timestamps
  mutate(date = as.Date(sampledate)) %>% # create new column for date of timestamps
  group_by(date,hour,depth) %>% # group the dataset by date and hour (hourly data)
  summarise(wtemp_h = mean(wtemp,na.rm=T),datetime = first(sampledate)) %>% # take
  # the mean water temperature and first sampledate of each group
  ungroup() %>% # ungroup data
  dplyr::select(datetime,depth,wtemp_h)
```

Use tidyr to convert data from long to wide format.

```
buoyHourlyWide <- buoyHourly %>% #select only three columns
  spread(depth,wtemp_h) # convert dataset from 'long' format to 'wide
```

Look at the first three rows of our data

```
head(buoyHourlyWide,3)
```

```
## Source: local data frame [3 x 29]
##
##              datetime        0     0.25      0.5     0.75        1
##                <time>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 2014-05-18 12:00:00 9.437500 8.279333 9.314333 9.285833 9.266333
## 2 2014-05-18 13:00:00 9.499333 8.322833 9.408667 9.375500 9.334833
## 3 2014-05-18 14:00:00 9.922167 8.378333 9.737500 9.652000 9.544833
## Variables not shown: 1.5 <dbl>, 2 <dbl>, 2.5 <dbl>, 3 <dbl>, 3.5 <dbl>, 4
##   <dbl>, 4.5 <dbl>, 5 <dbl>, 5.5 <dbl>, 6 <dbl>, 6.5 <dbl>, 7 <dbl>, 7.5
##   <dbl>, 8 <dbl>, 9 <dbl>, 10 <dbl>, 11 <dbl>, 12 <dbl>, 13 <dbl>, 14
##   <dbl>, 15 <dbl>, 16.5 <dbl>, 18 <dbl>.
```

Convert the colnames of the data to include *'wtr_'*

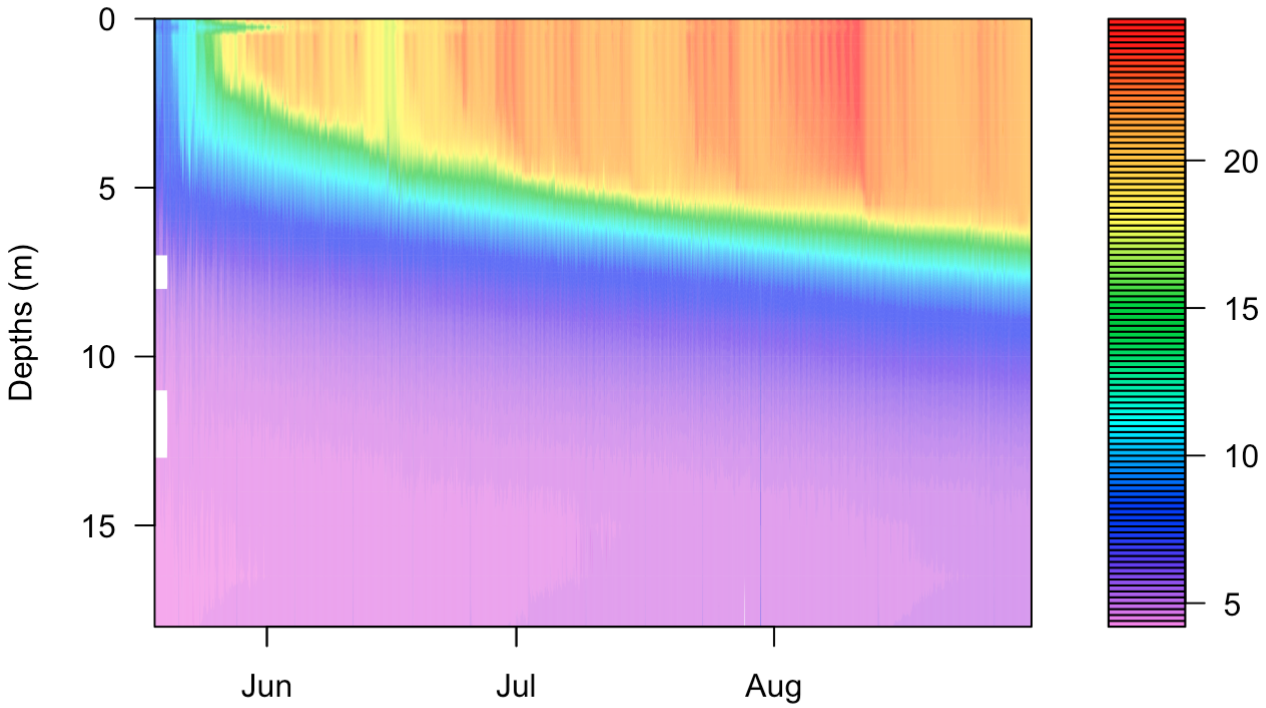
```
colnames(buoyHourlyWide)[-1] = paste('wtr_',colnames(buoyHourlyWide)[-1],sep='')
names(buoyHourlyWide)
```

```
##  [1] "datetime" "wtr_0"    "wtr_0.25" "wtr_0.5"  "wtr_0.75" "wtr_1"
##  [7] "wtr_1.5"  "wtr_2"    "wtr_2.5"  "wtr_3"    "wtr_3.5"  "wtr_4"
## [13] "wtr_4.5"  "wtr_5"    "wtr_5.5"  "wtr_6"    "wtr_6.5"  "wtr_7"
## [19] "wtr_7.5"  "wtr_8"    "wtr_9"    "wtr_10"   "wtr_11"   "wtr_12"
## [25] "wtr_13"   "wtr_14"   "wtr_15"   "wtr_16.5" "wtr_18"
```

Plotting buoy data

Plot the water temperature profile of our data

```
wtr.heat.map(buoyHourlyWide)
```



Aggregate met data

Use the same process for the meteorological data

```
metHourly <- met %>%
  dplyr::select(sampledate,air_temp,rel_hum,wind_speed_2m,par,opt_wtemp,opt_do_raw) %>%
  mutate(hour = hour(sampledate)) %>%
  mutate(date = as.Date(sampledate)) %>%
  group_by(date,hour) %>%
  summarise_each(funs(mean(., na.rm = TRUE))) %>%
  ungroup() %>%
  dplyr::select(-date,-hour)
```

Look at the first three rows of our data

```
head(metHourly,3)
```

```
## Source: local data frame [3 x 7]
##
##            sampledate air_temp  rel_hum wind_speed_2m      par opt_wtemp
##                <time>    <dbl>    <dbl>         <dbl>    <dbl>     <dbl>
## 1 2014-05-18 12:29:30 18.90333 25.23333      2.811667 986.7500  10.56383
## 2 2014-05-18 13:29:30 17.85667 28.56667      3.353333 582.7233  10.58000
## 3 2014-05-18 14:29:30 18.67167 27.31500      2.473333 811.6083  10.89800
## Variables not shown: opt_do_raw <dbl>.
```

Because we took the mean of the sampledate column. Let's round that column to the nearest hour

```
metHourly$sampledate = as.POSIXct(round(metHourly$sampledate,'hours'))
head(metHourly,3)
```

```
## Source: local data frame [3 x 7]
##
##            sampledate air_temp  rel_hum wind_speed_2m      par opt_wtemp
##                <time>    <dbl>    <dbl>         <dbl>    <dbl>     <dbl>
## 1 2014-05-18 12:00:00 18.90333 25.23333      2.811667 986.7500  10.56383
## 2 2014-05-18 13:00:00 17.85667 28.56667      3.353333 582.7233  10.58000
## 3 2014-05-18 14:00:00 18.67167 27.31500      2.473333 811.6083  10.89800
## Variables not shown: opt_do_raw <dbl>.
```

Output new data

Write data to .csv

```
write.csv(buoyHourly,'data/Sparkling2014wtemp_hourly_long.csv',row.names = F)
write.csv(buoyHourlyWide,'data/Sparkling2014wtemp_hourly_wide.csv',row.names = F)
write.csv(metHourly,'data/Sparkling2014domet_hourly.csv',row.names = F)
```