Clean and Preprocess Station Data

Yurk

3/2/2021

```
rm(list=ls())
gc()
```

```
##          used (Mb) gc trigger (Mb) limit (Mb) max used (Mb)
## Ncells 423698 22.7     881527 47.1         NA   658077 35.2
## Vcells 806335  6.2    8388608 64.0      32768  1802945 13.8
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.1     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

We first construct a function that will remove repetitive rows and rows without valid time stamps that are present in some data sets. The function also merges data from separate csv files covering different time ranges. The function also sets names and data types for the columns and creates a column with sample times in a time format.

```
stationClean <- function(infileList,rowRangeList){

  #labels and data types for columns
  col_names <- c("Date","Time","Temp","RH","Dewpt","WS","WD","Gust","LoCloudHt","MedCloudHt",
                 "HiCloudHt","Vis","AtmPres","SLP","Altim","Precip","WindChill","HeatIndex",
                 "empty")
  col_types <- cols("D","t","d","d","d","d","d","d","d","d","d","d","d","d","d","d","d","d","d")
  col_units <- c("Date","EST","F","perc","F","mph","deg","ft","ft","ft","mi","hPa","hPa","hPa",
                 "in","F","F")


  data <- NULL
```

```
  for(i in 1:length(infileList)){
    fname <- infileList[[i]]
    skipRows <- rowRangeList[[i]][1]
    nRows <- rowRangeList[[i]][2]
    d <- read_csv(fname,skip=skipRows,col_names=col_names,col_types=col_types,
                  na=c("","m","M","NC"),n_max=nRows) %>% select(-empty)
    data <- bind_rows(data,d)
  }

  #drop any rows without a valid time stamp
  if(any(is.na(data$Time))){
    prob_rows <- which(is.na(data$Time))
    data <- data[-prob_rows,]
  }

  # create time object with sample times
  data <- data %>% mutate(datetime=ymd(Date,tz="EST")+hms(Time))

  return(data)
}
```

Next we construct a function that will calculate drift potential per time, following Fryberger and Dean. Note that this allows the station height to be specified if it not at the standard 10m height. The function assumes that wind speed units are mph.

```
dp_from_ws <- function(ws,ht=10){ #assumes ws units are miles per hour
  ws10 <- ws*log(200)/log(20*ht) #estimate speed at 10m
  ws10kt <- ws10*0.868976 #mph to knots
  q <- ws10kt^2*(ws10kt-12) #threshold 12 knots as in Fryberger and Dean
  q <- q*(q>0) #dp should be zero if ws is below threshold
  return(q)
}
```

Next we create a function to do the cleaning steps, compute drift potentials, and save the output file for a single station.

```
stat_clean_preproc <- function(stat_par){
  data <- stationClean(infileList=stat_par[["infiles"]],
                       rowRangeList=stat_par[["rowRange"]])
  data <- data %>% mutate(dp = dp_from_ws(WS))
  outfile <- stat_par[["outfile"]]
  saveRDS(data,file=outfile)
  return(data)
}
```

Now we specify the necessary information for using the functions to clean and preprocess the data for each station.

```
#beh
beh_par <- list(infiles=list("raw_data/BEH_19730101_20201207.csv"),
                rowRange = list(c(9+1102,320439-1102)),
                outfile="clean_data/BEH_19730101_20201207_dp.rds")
#grb
grb_par <- list(infiles=list("raw_data/GRB_19500101_19591231.csv",
                             "raw_data/GRB_19600101_19891231.csv",
                             "raw_data/GRB_19900101_20201207.csv"),
```

```
                rowRange = list(c(9+2376,92946-2376),c(9,213627),c(9,270376)),
                outfile="clean_data/GRB_19490901_20201207_dp.rds")
#kbiv
kbiv_par <- list(infiles=list("raw_data/KBIV_19961231_20201203.csv"),
                 rowRange = list(c(9,209477)),
                 outfile="clean_data/KBIV_19961231_20201203_dp.rds")
#kmkg
kmkg_par <- list(infiles=list("raw_data/KMKG_19480101_19591231.csv",
                              "raw_data/KMKG_19600101_19891231.csv",
                              "raw_data/KMKG_19900101_20201204.csv"),
                rowRange = list(c(9,86135),c(9,215644),c(9,271048)),
                outfile="clean_data/KMKG_19480101_20201204_dp.rds")
#mdw
mdw_par <- list(infiles=list("raw_data/MDW_19480101_19591231.csv",
                             "raw_data/MDW_19600101_19891231.csv",
                             "raw_data/MDW_19900101_20201207.csv"),
                rowRange = list(c(9,105186),c(9,259730),c(9,269147)),
                outfile="clean_data/MDW_19480101_20201207_dp.rds")
#mke
mke_par <- list(infiles=list("raw_data/MKE_19480101_19591231.csv",
                             "raw_data/MKE_19600101_19891231.csv",
                             "raw_data/MKE_19900101_20201207.csv"),
                rowRange = list(c(9+2376,107574-2376),c(9,233776),c(9,270397)),
                outfile="clean_data/MKE_19471231_20201207_dp.rds")
#tvc
tvc_par <- list(infiles=list("raw_data/TVC_19490101_19591231.csv",
                             "raw_data/TVC_19600101_19891231.csv",
                             "raw_data/TVC_19900101_20201207.csv"),
                rowRange = list(c(9+2376,99519-2376),c(9,213786),c(9,270396)),
                outfile="clean_data/TVC_19481201_20201207_dp.rds")

station_par_list <- list(beh=beh_par,grb=grb_par,kbiv=kbiv_par,kmkg=kmkg_par,
                         mdw=mdw_par,mke=mke_par,tvc=tvc_par)
```

Finally we run the cleaning/preprocessing function on the list of stations.

```
stat_data_list<-lapply(station_par_list,stat_clean_preproc)
```

```
## Warning: 4 parsing failures.
##    row  col   expected actual                                    file
## 114652 Time time like    00:7 'raw_data/BEH_19730101_20201207.csv'
## 114676 Time time like    00:1 'raw_data/BEH_19730101_20201207.csv'
## 115876 Time time like    00:6 'raw_data/BEH_19730101_20201207.csv'
## 174988 Time time like    00:4 'raw_data/BEH_19730101_20201207.csv'

## Warning: 2 parsing failures.
##   row  col   expected actual                                    file
##  6198 Time time like    00:6 'raw_data/KBIV_19961231_20201203.csv'
## 24270 Time time like    00:5 'raw_data/KBIV_19961231_20201203.csv'

## Warning: 1 parsing failure.
##    row  col   expected actual                                    file
## 222005 Time time like    00:6 'raw_data/MDW_19600101_19891231.csv'
```

We can refer to stations in the list by name.

```
stat_data_list[["mdw"]]
```

```
## # A tibble: 634,062 x 20
##    Date       Time    Temp    RH Dewpt    WS    WD  Gust LoCloudHt MedCloudHt
##    <date>     <time> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>     <dbl>       <dbl>
##  1 1948-01-01 00:00     32    95    31    17    68    NA        NA          NA
##  2 1948-01-01 01:00     32   100    32    20    45    NA        60          NA
##  3 1948-01-01 02:00     31    92    29    23    68    NA        80          NA
##  4 1948-01-01 03:00     30    95    29    24    45    NA        NA          NA
##  5 1948-01-01 04:00     30    85    26    23    68    NA        90          NA
##  6 1948-01-01 05:00     30    85    26    21    68    NA        80          NA
##  7 1948-01-01 06:00     30    88    27    21    45    NA        NA          NA
##  8 1948-01-01 07:00     30    95    29    22    68    NA        80          NA
##  9 1948-01-01 08:00     31    92    29    14    45    NA        70          NA
## 10 1948-01-01 09:00     31   100    31    14    68    NA        NA          NA
## # ... with 634,052 more rows, and 10 more variables: HiCloudHt <dbl>,
## #   Vis <dbl>, AtmPres <dbl>, SLP <dbl>, Altim <dbl>, Precip <dbl>,
## #   WindChill <dbl>, HeatIndex <dbl>, datetime <dttm>, dp <dbl>
```

```
summary(stat_data_list[["mdw"]])
```

```
##       Date                 Time               Temp              RH
##  Min.   :1948-01-01   Length:634062     Min.   :-24.00   Min.   :  4.00
##  1st Qu.:1966-01-31   Class1:hms        1st Qu.: 35.00   1st Qu.: 54.00
##  Median :1984-04-24   Class2:difftime   Median : 52.00   Median : 67.00
##  Mean   :1984-06-06   Mode  :numeric    Mean   : 51.26   Mean   : 66.41
##  3rd Qu.:2002-11-06                     3rd Qu.: 69.00   3rd Qu.: 80.00
##  Max.   :2020-12-07                     Max.   :106.00   Max.   :214.00
##                                         NA's   :9063     NA's   :11009
##      Dewpt              WS               WD             Gust
##  Min.   :-41.00   Min.   : 0.00   Min.   :  0     Min.   : 0.0
##  1st Qu.: 26.00   1st Qu.: 7.00   1st Qu.:100     1st Qu.:21.0
##  Median : 40.00   Median : 9.00   Median :200     Median :24.0
##  Mean   : 39.63   Mean   :10.17   Mean   :189     Mean   :22.8
##  3rd Qu.: 56.00   3rd Qu.:14.00   3rd Qu.:270     3rd Qu.:29.0
##  Max.   : 83.00   Max.   :67.00   Max.   :360     Max.   :72.0
##  NA's   :10898    NA's   :7181    NA's   :16223   NA's   :583890
##    LoCloudHt        MedCloudHt       HiCloudHt           Vis
##  Min.   :    0    Min.   :   10    Min.   :   40    Min.   : 0.000
##  1st Qu.: 2100    1st Qu.: 4527    1st Qu.: 7870    1st Qu.: 6.000
##  Median : 5910    Median : 9840    Median :15000    Median :10.000
##  Mean   : 7223    Mean   :12437    Mean   :15701    Mean   : 8.047
##  3rd Qu.: 8000    3rd Qu.:20000    3rd Qu.:24610    3rd Qu.:10.000
##  Max.   :72180    Max.   :68900    Max.   :35000    Max.   :10.000
##  NA's   :195841   NA's   :494987   NA's   :592944   NA's   :6624
##     AtmPres            SLP             Altim            Precip
##  Min.   : 957.9   Min.   : 955.7   Min.   : 977.7   Min.   :0
##  1st Qu.: 989.8   1st Qu.:1012.2   1st Qu.:1012.5   1st Qu.:0
##  Median : 994.1   Median :1016.6   Median :1016.9   Median :0
##  Mean   : 994.0   Mean   :1016.8   Mean   :1016.8   Mean   :0
##  3rd Qu.: 998.3   3rd Qu.:1021.3   3rd Qu.:1021.3   3rd Qu.:0
##  Max.   :1023.1   Max.   :1064.5   Max.   :1048.1   Max.   :6
##  NA's   :208475   NA's   :163054   NA's   :228547   NA's   :40755
##    WindChill        HeatIndex         datetime
```

```
##  Min.   :-51.0    Min.   : 80.0    Min.   :1948-01-01 00:00:00
##  1st Qu.: 16.0    1st Qu.: 83.0    1st Qu.:1966-01-31 01:15:00
##  Median : 25.0    Median : 86.0    Median :1984-04-24 19:30:00
##  Mean   : 23.1    Mean   : 87.4    Mean   :1984-06-06 15:45:40
##  3rd Qu.: 34.0    3rd Qu.: 90.0    3rd Qu.:2002-11-06 16:38:00
##  Max.   : 47.0    Max.   :124.0    Max.   :2020-12-07 07:53:00
##  NA's   :371639   NA's   :592153
##        dp
##  Min.   :     0.00
##  1st Qu.:     0.00
##  Median :     0.00
##  Mean   :   245.51
##  3rd Qu.:    24.52
##  Max.   :156678.06
##  NA's   :7181
```