Data Cleaning and (preliminary) EDA

Optimizing HVAC Operation for Occupant Comfort and Energy Savings

Caleb Neale

Spring 2021

Load libraries

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.3
## v tidyr   1.1.2     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
```

```
library(fpp3)
```

```
## -- Attaching packages ---------------------------------------- fpp3 0.4.0 --

## v tsibble     1.0.0     v feasts      0.1.7
## v tsibbledata 0.2.0     v fable       0.3.0

## -- Conflicts ------------------------------------------------- fpp3_conflicts --
## x lubridate::date()    masks base::date()
## x dplyr::filter()      masks stats::filter()
## x tsibble::intersect() masks base::intersect()
## x tsibble::interval()  masks lubridate::interval()
## x dplyr::lag()         masks stats::lag()
## x tsibble::setdiff()   masks base::setdiff()
## x tsibble::union()     masks base::union()
```

Import Data and convert to tibble

```
read_and_clean <- function(csv_path){
  df <- read.csv(csv_path, sep=";", row.names = NULL)
  colnames(df) <- c("series", 'time','value')
  df$value <- as.numeric(df$value)
  df<- df[-1,]
  df <- as_tibble(df)
  return(df)
}

co2 <- read_and_clean('co2.csv')
```

```
## Warning in read_and_clean("co2.csv"): NAs introduced by coercion
```

```
occupied_status <- read_and_clean('occupied_status.csv')
```

```
## Warning in read_and_clean("occupied_status.csv"): NAs introduced by coercion
```

```
supply_air_flow <- read_and_clean('supply_air_flow.csv')
```

```
## Warning in read_and_clean("supply_air_flow.csv"): NAs introduced by coercion
```

```
supply_fan <- read_and_clean('supply_fan.csv')
```

```
## Warning in read_and_clean("supply_fan.csv"): NAs introduced by coercion
```

```
temperature <- read_and_clean('temperature.csv')
```

```
## Warning in read_and_clean("temperature.csv"): NAs introduced by coercion
```

Convert time data to datetime format

```
convert_to_datetime <- function(df){
  df$time <- gsub("-04:00$", "-0400", df$time)
  df$time <- gsub("-05:00$", "-0500", df$time)
  df$time <- strptime(df$time, format ="%Y-%m-%dT%H:%M:%S%z")
  return(df)
}

co2 <- convert_to_datetime(co2)
occupied_status <- convert_to_datetime(occupied_status)
supply_air_flow <- convert_to_datetime(supply_air_flow)
supply_fan <- convert_to_datetime(supply_fan)
temperature <- convert_to_datetime(temperature)
```

Convert to tsibble objects

```
co2 <- as_tsibble(co2, key= series, index = time)
occupied_status <- as_tsibble(occupied_status, key= series, index = time)
supply_air_flow <- as_tsibble(supply_air_flow, key= series, index = time)
supply_fan <- as_tsibble(supply_fan, key= series, index = time)
temperature <- as_tsibble(temperature, key= series, index = time)
```

Merge Tables on the time column

```
# tbd, need strategies for doing this or if it is even a good idea
```

EDA

```
print(co2)
```

```
## # A tsibble: 8,022 x 3 [3h] <?>
## # Key:       series [6]
##    series                                          time                value
##    <chr>                                           <dttm>              <dbl>
##  1 co2_ppm.mean {location_specific: 203 Olsson} 2020-08-31 23:00:00  434.
##  2 co2_ppm.mean {location_specific: 203 Olsson} 2020-09-01 02:00:00  431.
##  3 co2_ppm.mean {location_specific: 203 Olsson} 2020-09-01 05:00:00  433.
##  4 co2_ppm.mean {location_specific: 203 Olsson} 2020-09-01 08:00:00  442.
##  5 co2_ppm.mean {location_specific: 203 Olsson} 2020-09-01 11:00:00  439.
##  6 co2_ppm.mean {location_specific: 203 Olsson} 2020-09-01 14:00:00  435.
##  7 co2_ppm.mean {location_specific: 203 Olsson} 2020-09-01 17:00:00  430.
##  8 co2_ppm.mean {location_specific: 203 Olsson} 2020-09-01 20:00:00  436.
##  9 co2_ppm.mean {location_specific: 203 Olsson} 2020-09-01 23:00:00  444.
## 10 co2_ppm.mean {location_specific: 203 Olsson} 2020-09-02 02:00:00  446.
## # ... with 8,012 more rows
```