Weather API Project

Valentino Mascherini

January 29, 2022

```
knitr::opts_chunk$set(collapse = TRUE, message = FALSE, results = FALSE, warning = FALSE)
```

Libraries

```
library(jsonlite)
library(dplyr)
library(tidyverse)
library(lubridate)
library(RPostgres)
library(sf)
```

SQL Database connection

```
con <- dbConnect(RPostgres::Postgres(),
                 "hydenv", host = "localhost",
                 port = 5432,
                 user = "hydenv",
                 password = "hydenv")
```

SQL query overview

```
sql_overview <- dbGetQuery(con,
          {'WITH data_tot AS (
            SELECT * FROM data
            JOIN metadata meta ON data.meta_id=meta.id
            WHERE variable_id=1
          ),
            mean_temp AS (
              SELECT data_tot.id,
              AVG(value) AS t_avg
              FROM data_tot
              GROUP BY data_tot.id
            ),
            day_temp AS (
```

```
              SELECT data_tot.id,
              AVG(value) AS t_day
              FROM data_tot
              WHERE date_part(\'hour\', tstamp) >= 6 AND date_part(\'hour\', tstamp) < 18
              GROUP BY data_tot.id
            ),
            night_temp AS (
              SELECT data_tot.id,
              AVG(value) AS t_night
              FROM data_tot
              WHERE date_part(\'hour\', tstamp) < 6 OR date_part(\'hour\', tstamp) >= 18
              GROUP BY data_tot.id
            ),
            t_var AS (
              SELECT tv.id,
              t_max - t_min AS t_t_var
              FROM (
                SELECT  data_tot.id,
                date_trunc(\'day\', tstamp) AS day,
                MIN(value) AS t_min,
                MAX(value) AS t_max
                FROM data_tot
                GROUP BY data_tot.id, date_trunc(\'day\', tstamp)
              ) tv
            ),
            amount AS (
              SELECT id, count(*) AS "count" FROM data_tot GROUP BY id
            )
            SELECT  mean_temp.t_avg,
            day_temp.t_day,
            night_temp.t_night,
            t_var.t_t_var,
            amount."count",
            meta.id
            FROM metadata meta
            JOIN mean_temp ON meta.id=mean_temp.id
            JOIN day_temp ON meta.id=day_temp.id
            JOIN night_temp ON meta.id=night_temp.id
            JOIN amount ON meta.id=amount.id
            JOIN t_var ON meta.id=t_var.id
            LIMIT 15'} # SQL query code
          )
```

JSON importer

```
url1 <- as.character("https://raw.githubusercontent.com/data-hydenv/data/master/extra/weather/data/2021_

num1 <- as.character("11")

url2 <- as.character("_raw_dump.json")
```

```
url3 <- as.character("https://raw.githubusercontent.com/data-hydenv/data/master/extra/weather/data/2022_

urlt <- paste(url1, num1, url2, sep= "")

# for january

num2 <- as.character("1")
```

```
dtt <- data.frame(dttm =as.Date(character()), th = as.double(double()))
```

```
while (as.integer(num1) < 43) {

  #---- json importer

  js <- fromJSON(urlt, flatten =TRUE)

  jdf <- js$historic$hourly %>% as.data.frame()

  dfor <- jdf %>% select(dt, temp)%>%
    mutate(th = temp) %>%
    mutate(dttm = as.POSIXct(dt, origin ="1970-01-01")) %>%
    select(dttm, th)

  #---- create data frame

  dtt <- union(dtt, dfor)

  dtt_check <- dtt %>% mutate(year = year(dttm), month = month(dttm), day = day(dttm))

  #---- names and url update

  ifelse(num1 < 31, #condition yes = december

         {

           num1 <- ifelse(num1 == 26, as.integer(num1) + 2, as.integer(num1) + 1)

           urlt <- paste(url1, as.character(num1), url2, sep= "")}, # yes

         { # no = january

           num1 <- as.integer(num1) + 1

           urlt <- paste(url3, num2, url2, sep= "")

           num2 <- as.integer(num2) + 1} # no

  )}

dtt <- dtt %>% filter(dttm > "2021-12-12 23:00:00", dttm < "2022-01-10") %>%
  mutate(th = th -273.15)
```

Comparison plot

```
# Importing tables from Postgres

hobos_data <- dbReadTable(con, "data") %>% # quality checked data
  filter(tstamp >= "2021-12-13 00:00:00") %>%
  rename(dttm = tstamp, th = value)

hobos_md <- dbReadTable(con, "metadata") %>%
  filter(id >= 37, id <= 67)

hobos_id <- hobos_md %>% select(id, device_id)

dtl <- as.data.frame(read_csv("https://raw.githubusercontent.com/vm17399/weather_api/main/api_data_weatl

# joining df

h_data <- merge(hobos_data, hobos_id, by.x = c("meta_id"), by.y = c("id")) %>%
  select(-variable_id) %>% select(-meta_id, -quality_flag_id)

dth <- dtl %>%  mutate(device_id = "api")

hobo_api <- union(h_data, dth)

hobo_api <- merge(hobo_api, dth, by = "dttm") %>% select(-device_id.y)

names(hobo_api) <- c("dttm", "th", "device_id", "temp")

# plotting

cols <- c("OpenWeather" = "red", "HOBOs" = "grey")

g0 <- ggplot(hobo_api, aes(dttm)) +
  geom_line(aes(y = th), color = "grey") +
  geom_line(aes(y = temp, color = "OpenWeather"), size = 0.7) +
  labs(x = 'Date', y = 'Temperature', color = 'Legend') +
  scale_color_manual(values = cols) +
  theme_minimal(14) +
  theme(legend.position = c(0.3, 0.85),
        legend.background = element_rect(fill = "white", color = "grey"),
        legend.key.size = unit(3, "line"))
```

Correlation of single HOBOs

```
# Importing tables from Postgres and api

hobos_tot <- dbReadTable(con, "data") %>% # quality checked data tot
  rename(dttm = tstamp, th = value, id = meta_id)

hobos_id_tot <- dbReadTable(con, "metadata") %>%
```

```
  select(id, device_id)

# merging with id metadata frame

hobos_full <- merge(hobos_tot, hobos_id_tot, by = "id")

# split the df in two years

hobos_2021 <- hobos_full %>% filter(dttm < "2021-5-1")

hobos_2122 <- hobos_full %>% filter(dttm > "2021-12-13 00:00:00",
                                    dttm < "2021-12-26 01:00:00" |
                                      dttm > "2021-12-27 00:00:00")

cyc <- data.frame(name = c("hobos_2021","hobos_2122"))

i <- 1

while (i < nrow(cyc) + 1) {

  # ---- HOBO and API model comparison ----

  nam <- cyc[[i, 1]]

  curr <- eval(parse(text = nam))

  # get the names

  h_names <- data.frame(device_id = (curr$device_id), id = curr$id) %>% distinct_all()

  # empty data frame and counter

  rel <- data.frame(device_id = NA, pear = NA, cov = NA, meta_id = NA)

  dth <- dtt

  count <- 1

  while (count < (nrow(h_names) + 1)) {

    filtr <- h_names[[count, 1]]

    # filter by device

    dev <- curr %>% filter(device_id == filtr)

    if(i == 1) {

      dth <- dtt %>% slice(1:nrow(dev))
    } else {

      dth <- dtt %>% slice(1:nrow(dev))
    }
```

```
    cova <- cov(x = dev$th, y = dth$th)

    pears <- cor.test(x = dev$th, y = dth$th, model ="pearson")

    rel <- union(rel, data.frame(pear = pears$estimate,
                                 device_id =  h_names[[count, 1]],
                                 cov = cova,
                                 meta_id =  h_names[[count, 2]]))

    count <- count + 1

  }

  rel <- as.data.frame(rel) %>%  drop_na()

  assign(paste0("corr_", cyc[[i, 1]]), rel)

  i <- i + 1

}
```

Mapping

```
# read .csv and maps, metadata

h21 <- read.csv("https://raw.githubusercontent.com/vm17399/weather_api/main/cor_hobos_2021.csv")

h22<- read.csv("https://raw.githubusercontent.com/vm17399/weather_api/main/cor_hobos_2122.csv")

hwt <- union(h21, h22) %>% rename(id = meta_id)

districts <- dbReadTable(con, "osm_nodes")

hmd <- dbReadTable(con, "metadata") %>% select(-description)

# we have two pq_geometries being in hmd and in districts, we must convert them

distr <- st_as_sfc(districts$geom)

distr1 <- sf::st_as_sf(distr) %>% st_set_crs("WGS84")

distr1 <- distr1 %>% mutate(id = districts$id)

hmd <- hmd %>%  mutate(coord = st_as_sf(sf::st_as_sfc(hmd$location)))

hmd$coord %>% st_set_crs("WGS84")

hmdt <- merge(hmd, hwt, by = "id")%>%
  mutate(term = ifelse(term_id == 11, "WT21", "WT22"))
```

```r
hmd22 <- hmdt %>% filter(term == "WT22")

g1 <- ggplot() +
  geom_sf(data = distr1, colour = "white", fill = "grey70") +
  geom_sf(data = hmdt$coord, aes(fill = hmdt$pear, shape = hmdt$term), size = 3, alpha= 0.8) +
  theme_minimal(14) +
  scale_shape_manual(values = c(24, 21), name = "Term") +
  scale_fill_viridis_b(option = "plasma",
                       name = "Pearson correlation",
                       n.breaks = 5) +
  ggtitle("HOBOs in Freiburg") +
  theme(legend.key.size = unit(0.3, "line"),
        legend.position = "bottom",
        legend.background = element_rect(fill = "white", colour = "grey"))
```

```r
dist_coord <- districts %>% mutate(coord = st_as_sf(sf::st_as_sfc(geom))) %>%
  mutate(did = c(1:28))

dist_coord$coord %>% st_set_crs("WGS84")

dist_coord$coord %>% st_within(hmdt$coord[[1]])

points <- hmdt$coord %>% st_within(dist_coord$coord)

with <- st_within(hmdt$coord, dist_coord$coord)

point <- hmdt %>%
  mutate(did = with)

hobo_dist <- point %>% select(-location, -device_id.y, -sensor_id, -term_id)

hobo_dist$did <- hobo_dist$did %>% as.integer()

hobo_districts <- inner_join(hobo_dist, dist_coord, by = "did")

hobo_but <- hobo_districts %>%
  select(id.x, device_id.x, pear, name) %>%
  rename(id = id.x, device_id = device_id.x)

distribution <- hobo_but %>% select(name) %>% group_by(name) %>% count()

pearper <- hobo_but %>% group_by(name) %>%
  summarise(p_avg = mean(pear),
            name) %>% distinct_all()

pearper22 <- hobo_but %>% filter(id > 36) %>%
  group_by(name) %>%
  summarise(p_avg = mean(pear),
            name) %>% distinct_all()
```

```r
#by WT 21
```

```r
# get data frames

hobos_2021 <- read_csv("https://raw.githubusercontent.com/vm17399/weather_api/main/hobos_2021.csv")

pointer21 <- point %>% filter(term_id == 11) %>% select(id, did) %>%
  mutate(did = as.integer(did)) %>%
  filter(is.na(did) == FALSE)

didnt <- dist_coord %>% select(name, did)

hobos_did <- full_join(hobos_2021, pointer21, by = "id")

hobos_did_avg21 <- hobos_did %>% group_by(did, dttm) %>%
  summarise(tavg = mean(th), did, dttm) %>%
  distinct_all() %>% filter(is.na(tavg) == FALSE)

didder <- data.frame(did = as.integer(pointer21$did)) %>%
  distinct_all() %>%
  filter(did != "integer(0)")

real <- data.frame(pear = NA, did = NA)

dtg <- dtt

j <- 1

while (j < nrow(didder) + 1) {

  hda <- hobos_did_avg21 %>% filter(did == didder[[j,1]])

  dtg <- dtt %>% slice(1:nrow(hda))

  pearson <- cor.test(x = hda$tavg, y = dtg$th, model ="pearson")

  real <- union(real, data.frame(pear = pearson$estimate,
                                 did =  didder[[j, 1]]))
  j <- j + 1
}

peareal21 <- merge(real, didnt, by = "did")

# average for WT 22

# get data frames

pointer <- read_csv("https://raw.githubusercontent.com/vm17399/weather_api/main/pointer22.csv")

pointer22 <- point %>% filter(term_id == 13) %>% select(id, did) %>% mutate(did = as.integer(did)) %>%
  filter(is.na(did) == FALSE)

hobos_2022 <- read_csv("https://raw.githubusercontent.com/vm17399/weather_api/main/hobos_2022.csv")

hobos_did <- full_join(hobos_2022, pointer, by = "id")
```

```
hobos_did_avg22 <- hobos_did %>% group_by(did, dttm) %>%
  summarise(tavg = mean(th), did, dttm) %>%
  distinct_all() %>% filter(is.na(tavg) == FALSE)

didnt <- dist_coord %>% select(name, did)

didder <- data.frame(did = as.integer(pointer$did)) %>%
  distinct_all() %>%
  filter(did != "integer(0)")

real <- data.frame(pear = NA, did = NA)

dtg <- dtt

j <- 1

while (j < nrow(didder) + 1) {

  hda <- hobos_did_avg22 %>% filter(did == didder[[j,1]]) %>% slice(c(1:648))

  pearson <- cor.test(x = hda$tavg, y = dtg$th, model ="pearson")

  real <- union(real, data.frame(pear = pearson$estimate,
                                 did =  didder[[j, 1]]))
  j <- j + 1
}

peareal <- merge(real, didnt, by = "did")
```

```
# WT 22

colorz2 <- merge(peareal, districts, by = "name") %>% slice(1:11)

colorz2 <- left_join(colorz2, distr1, by = "id")

g4 <- ggplot(colorz2$x) +
  geom_sf(data = distr1, colour = "white", fill = "grey70") +
  geom_sf(aes(fill = colorz2$pear)) +
  geom_sf(data = hmd22$coord, size = 3, alpha= 0.8) +
  scale_shape_manual(values = c(21)) +
  scale_fill_viridis_c(option = "plasma",
                       name = "P. average",
                       n.breaks = 3) +
  ggtitle("Pearson correlation per district 2022") +
  theme(legend.key.size = unit(0.55, "line"),
        legend.position = c(0.1,0.18),
        legend.background = element_rect(fill = "white", colour = "grey"))

# WT 21 + WT 22

peartot <- union(peareal, peareal21)
```

```r
colorz3 <- merge(peartot, districts, by = "name")

colorz3 <- left_join(colorz3, distr1, by = "id")

g5 <- ggplot(colorz3$x) +
  geom_sf(data = distr1, colour = "white", fill = "grey70") +
  geom_sf(aes(fill = colorz3$pear)) +
  geom_sf(data = hmdt$coord, aes(shape = hmdt$term), fill = "black",size = 3, alpha= 0.8) +
  scale_shape_manual(values = c(24, 21), name = "Term") +
  scale_fill_viridis_c(name = "P. average",
                       n.breaks = 5) +
  ggtitle("Pearson correlation per district for 2021 and 2022") +
  theme(legend.key.size = unit(0.55, "line"),
        legend.position = c(0.1,0.18),
        legend.background = element_rect(fill = "white", colour = "grey"))
```

Modelling

```r
# importing

hobos_2021 <- read_csv("https://raw.githubusercontent.com/vm17399/weather_api/main/hobos_2021.csv")

hobos_2022 <- read_csv("https://raw.githubusercontent.com/vm17399/weather_api/main/hobos_2022.csv")

pointer <- read_csv("https://raw.githubusercontent.com/vm17399/weather_api/main/pointer22.csv")

# unified df

hobos_full <- union(hobos_2021, hobos_2022)

hobos_didt <- full_join(hobos_full, pointer, by = "id")

model_data <- hobos_didt %>% group_by(did, dttm) %>%
  summarise(tavg = mean(th), did, dttm) %>%
  distinct_all() %>% filter(is.na(tavg) == FALSE)

pointer_n <- merge(pointer, dist_coord, by ="did") %>% select(did, name, id.x)

pn22 <- pointer_n %>% filter(id.x > 36) %>% distinct(did, name)

pn21 <- pointer_n %>% filter(id.x < 37)

c <- 1

# this code eliminates district in WT21 that are present in WT22, for accuracy

while (c < nrow(pn22)) {

  pn21 <- pn21 %>% filter(did != pn22[[c,1]])

  c <- c + 1
```

```r
}

pn21 <- pn21 %>% distinct(did, name)

pntot <- union(pn21, pn22)

# models per district

k <- 1

while (k < nrow(pntot) + 1) {

  md <- model_data %>% filter(did == pntot[[k, 1]])

  if (nrow(md) > 800) {md <- md %>% filter(dttm > "2021-01-01")} else {md <- md}

  dtk <- dtt %>% slice(1:nrow(md))

  model <- lm(md$tavg ~ dtk$th)

  assign(paste0("lm_", pntot[[k,2]]), model)

  k <- k + 1

}

# the models for the available districts are now objects named "lm_districtname", for example:

#summary(data_Wiehre)
```