Cool Quiet City

Federico Cortese

2023-12-01

Competition

This competition's goal is to use contextual data to predict noise distraction and thermal preference across a diversity of indoor and outdoor spaces.

This competition includes the city-scale collection of 9,808 smartwatch-driven micro-survey responses that were collected alongside 2,659,764 physiological and environmental measurements from 98 people using the open-source Cozie-Apple platform combined with geolocation-driven urban digital twin metrics.

Data

The data is available here https://www.kaggle.com/competitions/cool-quiet-city-competition/data

The dataset includes micro-survey responses from 98 people, who each provided at least 100 micro-survey responses from October 2022 to August 2023.

The data collection process for this competition utilized the open-source Cozie Apple platform. This mobile and smartwatch application for iOS devices allows people to complete a watch-based micro-survey and provide real-time feedback about environmental conditions via their Apple Watch. It leverages the built-in sensors of the smartwatch to collect physiological (e.g., heart rate, activity), environmental (sound level), and location (latitude and longitude) data.

The goal of the competition is to predict the values of the columns q_noise_nearby, q_noise_kind, q_earphones, q_thermal_preferenceas well as possible.

Accuracy

The evaluation metric for this competition is accuracy. The metric is calculated as

$$\frac{TP + TN}{TP + TN + FP + FN}$$

As we have four targets here (q_noise_nearby, q_noise_kind, q_earphones, q_thermal_preference), accuracy for each target column is calculated and the average of the four of them is returned.

Prediction

For each id_unique in the test set, you must predict the target variables: q_noise_nearby, q_noise_kind, q_earphones, q_thermal_preference.

Dataset description

The physiological and environmental data were collected at a higher frequency than micro-survey responses. Therefore, these datasets are sparse, i.e., many values are empty as not every row contains a microsurvey response.

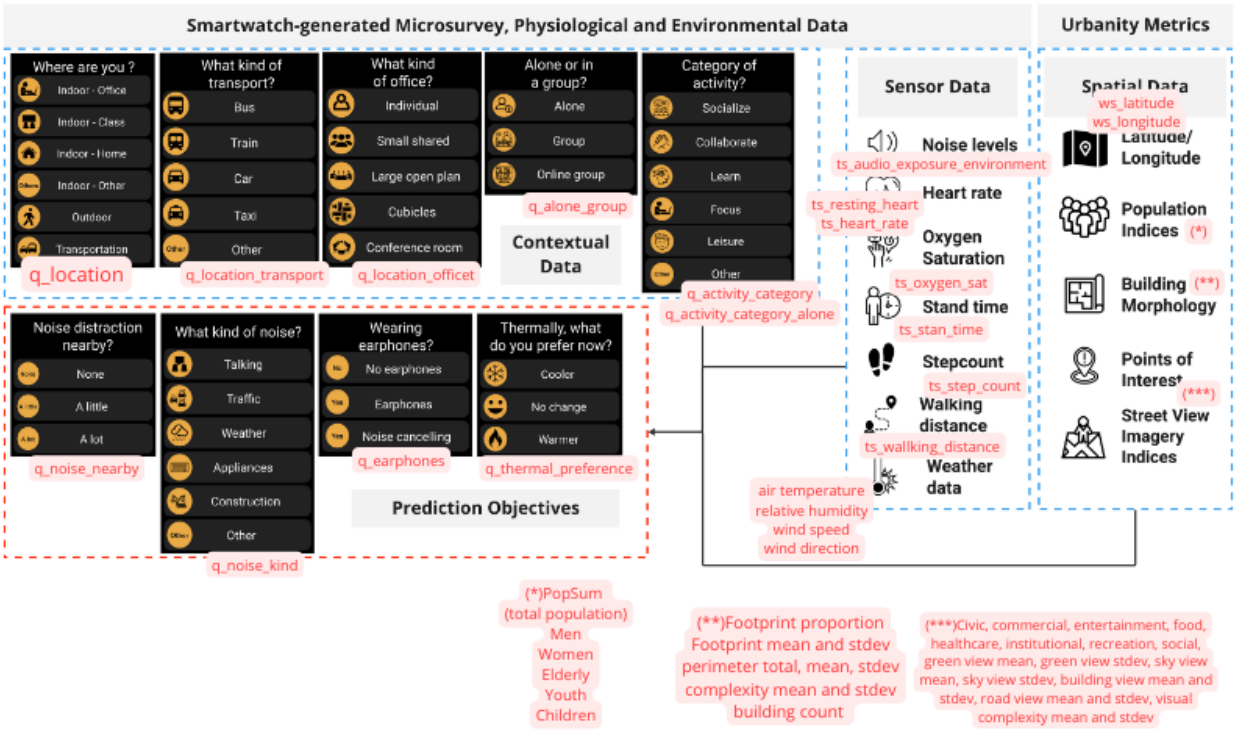


Figure 1: The red small captions refer to the feature names.

The spatial indicators (calculated using the URBANITY PYTHON PACKAGE starting from latitude and longitude) are grouped into Building Morphology and Point of Interest. For more information on the spatial indicators, please refer to Table 1 of Yap, W., Biljecki, F. A Global Feature-Rich Network Dataset of Cities and Dashboard for Comprehensive Urban Analyses. Sci Data 10, 667 (2023).

Urbanity is a network-based Python package to automate the construction of feature rich (contextual and semantic) urban networks at any geographical scale. Through an accessible and simple to use interface, users can request heterogeneous urban information such as street view imagery, building morphology, population (including sub-group), and points of interest for target areas of interest.

```
# Load packages
library(scales)
library(dplyr)
library(tidyr)
library(tidyverse)
library(ggplot2)
library(ggpubr)
library(lubridate)
```

Feature engineering

The features have different last dates

```
load("cleaned_data.Rdata")

# Time ranges
as.Date(range(cozie_train$time))
```

```
## [1] "2022-10-10" "2023-05-31"
```

```
as.Date(range(air_temp$time))
```

```
## [1] "2022-10-09" "2023-07-03"
```

```
as.Date(range(RH$time))
```

```
## [1] "2022-10-09" "2023-07-03"
```

```
as.Date(range(wind_dir$time))
```

```
## [1] "2022-10-09" "2023-07-03"
```

```
as.Date(range(wind_speed$time))
```

```
## [1] "2022-10-09" "2023-07-03"
```

```
as.Date(range(rainfall$time))
```

```
## [1] "2022-10-09" "2023-07-03"
```

Let us have a look at the feature sparsity per the microsurvey variables

```
df1=cozie_train[1:1000,]
df1$time=strptime(df1$time,
                  format="%Y-%m-%d  %H:%M",
                  tz="Asia/Singapore")
df2=air_temp[1:1000,]
df2$time=strptime(df2$time,
                  format="%Y-%m-%d  %H:%M",
                  tz="Asia/Singapore")
#df1=cozie_train[1:1000,]%>%format(time,format='%y-%m-%d %H:%M',tz='UTC')
#df2=air_temp[1:1000,]%>%format(time,format='%y-%m-%d %H:%M',tz="Asia/Singapore")

df_merge=merge(df1,df2,by="time",all=T)

head(df_merge[c("q_thermal_preference","S43")])
```

```
##   q_thermal_preference  S43
## 1                 <NA> 28.0
## 2                 <NA>   NA
## 3                 <NA> 27.9
## 4                 <NA> 27.9
## 5                 <NA> 27.9
## 6                 <NA> 27.9
```

We compare it with environmental variables

```
#merge()
```

```
# How many available obs?
length(which(cozie_train$q_thermal_preference!=""))
```

```
## [1] 4900
```

```
# Create  no-sparse version
cozie_train_nosparse=cozie_train[which(cozie_train$q_thermal_preference!=""),]
dim(cozie_train_nosparse)
```

```
## [1] 4900   59
```

```
head(cozie_train_nosparse)
```

```
##                    time q_alone_group    q_earphones id_participant ws_latitude
## 28  2022-10-10 11:34:28         Group   No earphones        xesh001    1.297257
## 85  2022-10-10 11:52:50         Group   No earphones        xesh002    1.297251
## 223 2022-10-10 13:08:45         Group Not applicable        xesh003    1.297280
## 230 2022-10-10 13:13:19         Alone   No earphones        xesh001    1.297577
## 376 2022-10-10 13:52:13         Group   No earphones        xesh002    1.297598
## 428 2022-10-10 14:50:29         Group   No earphones        xesh002    1.297592
##          q_location q_location_office q_location_transport ws_longitude
## 28  Indoor - Office      Small shared       Not applicable     103.7711
## 85  Indoor - Office      Small shared       Not applicable     103.7710
## 223 Indoor - Office      Small shared       Not applicable     103.7710
## 230  Indoor - Class    Not applicable       Not applicable     103.7713
## 376  Indoor - Class    Not applicable       Not applicable     103.7713
## 428  Indoor - Class    Not applicable       Not applicable     103.7713
##       q_noise_kind q_noise_nearby q_thermal_preference ws_timestamp_location
## 28           Other       A little               Cooler   2022-10-10 03:33:55
## 85         Talking       A little            No change   2022-10-10 03:52:40
## 223 Not applicable           None               Cooler   2022-10-10 05:08:28
## 230        Talking       A little               Cooler   2022-10-10 05:13:04
## 376        Talking       A little               Cooler   2022-10-10 05:52:07
## 428     Appliances       A little               Cooler   2022-10-10 06:50:17
##      ws_timestamp_start ts_oxygen_saturation ts_resting_heart_rate
## 28  2022-10-10 03:33:51                   NA                    NA
## 85  2022-10-10 03:43:11                   NA                    NA
## 223 2022-10-10 05:07:58                   NA                    NA
## 230 2022-10-10 05:12:57                   NA                    NA
```

```
## 376 2022-10-10 05:52:03                  NA                   NA
## 428 2022-10-10 06:50:14                  NA                   NA
##     ts_stand_time ts_step_count ts_walking_distance ws_survey_count
## 28             NA            NA                  NA               1
## 85             NA            NA                  NA               1
## 223            NA            NA                  NA               1
## 230            NA            NA                  NA               2
## 376            NA            NA                  NA               3
## 428            NA            NA                  NA               4
##     Footprint.Proportion Footprint.Mean Footprint.Stdev Perimeter.Total
## 28                 0.277       1411.898        1496.111        4977.404
## 85                 0.263       1456.581        1502.083        4922.148
## 223                0.264       1368.192        1490.957        5012.360
## 230                0.323       1470.377        1562.229        6062.666
## 376                0.323       1437.235        1553.514        6125.874
## 428                0.322       1437.235        1553.514        6125.874
##     Perimeter.Mean Perimeter.Stdev Complexity.Mean Complexity.Stdev
## 28         165.913         125.857          27.413           12.893
## 85         169.729         126.306          27.778           12.962
## 223        161.689         125.957          26.939           12.948
## 230        168.407         129.718          27.466           12.990
## 376        165.564         129.068          27.156           12.946
## 428        165.564         129.068          27.156           12.946
##     Building.Count PopSum Men Women Elderly Youth Children Civic Commercial
## 28              30   1184 562   621     256   119       55     0          2
## 85              29   1157 549   607     250   117       53     0          2
## 223             31   1157 549   607     250   117       53     0          2
## 230             36   1388 659   728     301   140       64     0          2
## 376             37   1361 647   714     295   137       63     0          2
## 428             37   1374 653   721     298   139       64     0          2
##     Entertainment Food Healthcare Institutional Recreational Social
## 28              0    7          0             1            4      0
## 85              0    6          0             1            4      0
## 223             0    7          0             1            4      0
## 230             0    7          0             0            3      0
## 376             0    7          0             0            3      0
## 428             0    7          0             0            4      0
##     Green.View.Mean Green.View.Stdev Sky.View.Mean Sky.View.Stdev
## 28        0.3412654        0.1773330    0.10444444     0.08794980
## 85        0.3477455        0.1800441    0.10499394     0.08989828
## 223       0.3446845        0.1806241    0.10669048     0.08955084
## 230       0.3283605        0.1874471    0.09272789     0.08146623
## 376       0.3284362        0.1861950    0.09324161     0.08106151
## 428       0.3293400        0.1858989    0.09441333     0.08205369
##     Building.View.Mean Building.View.Stdev Road.View.Mean Road.View.Stdev
## 28           0.1513148           0.1856637      0.1534321       0.1300839
## 85           0.1457212           0.1847960      0.1526242       0.1288227
## 223          0.1464107           0.1840601      0.1515655       0.1286354
## 230          0.1760272           0.1930748      0.1438571       0.1289262
## 376          0.1740940           0.1925150      0.1435302       0.1280900
## 428          0.1729600           0.1923699      0.1431200       0.1277583
##     Visual.Complexity.Mean Visual.Complexity.Stdev        dT
## 28                1.567822              0.2665034        NA
## 85                1.561133              0.2730608        NA
```

```
## 223               1.563568               0.2758377        NA
## 230               1.560096               0.2819969  98.85017
## 376               1.563620               0.2817242 119.38598
## 428               1.563154               0.2808353  58.26572
##     q_activity_category_alone q_activity_category_group ts_heart_rate
## 28             Not applicable                     Learn            NA
## 85             Not applicable                 Socialize            NA
## 223            Not applicable                     Learn            NA
## 230                     Focus            Not applicable            NA
## 376            Not applicable                     Learn            NA
## 428            Not applicable                     Learn            NA
##     ts_audio_exposure_environment id_unique
## 28                             NA        28
## 85                             NA        85
## 223                            NA       223
## 230                            NA       230
## 376                            NA       376
## 428                            NA       428
```

Distribution of the microsurvey target variables

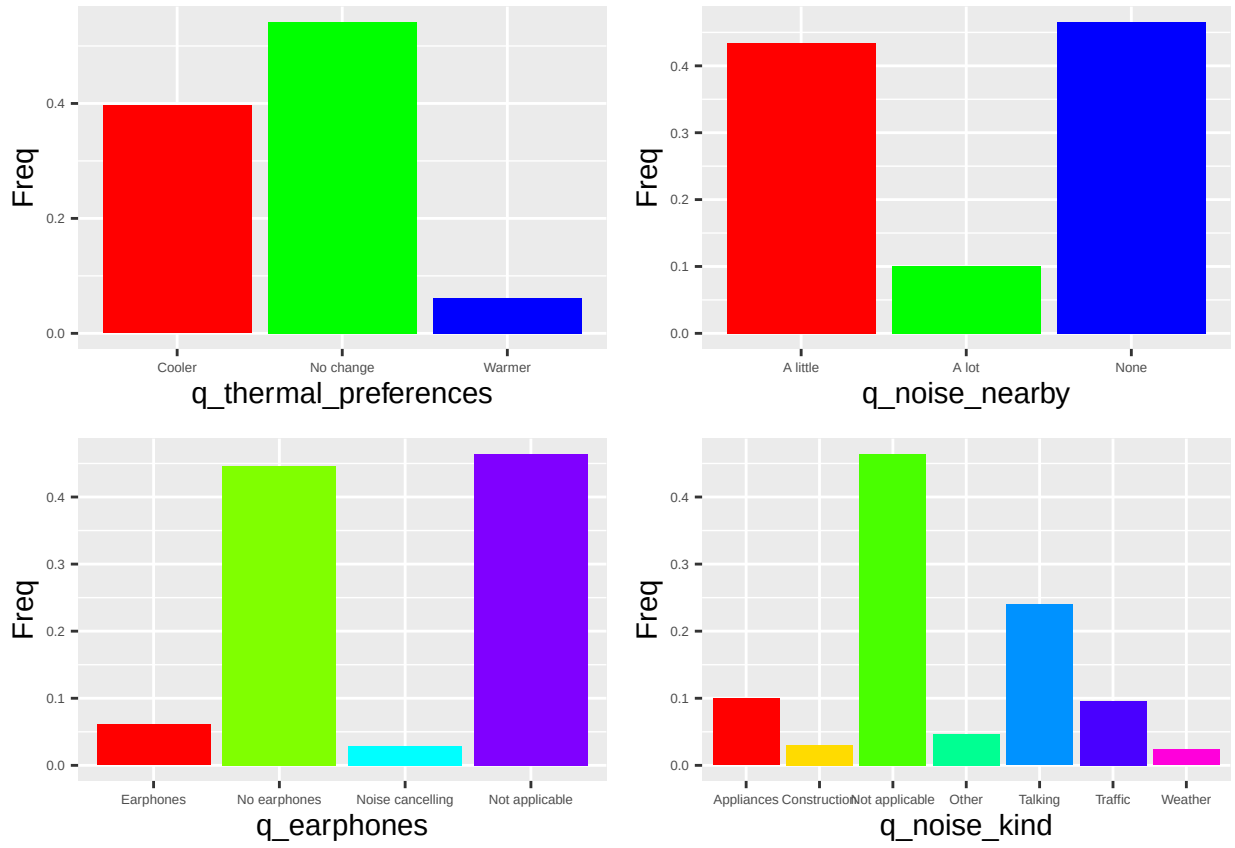
```
attach(cozie_train_nosparse)
P1=ggplot(cozie_train_nosparse, aes(x=q_thermal_preference )) +
   #ggplot(data=tips, aes(x=day)) +
      geom_bar(aes(y = (..count..)/sum(..count..)),
               fill=rainbow(length(unique(q_thermal_preference))))+
  labs(y="Freq",x="q_thermal_preferences")+
  theme(axis.text=element_text(size=5))

P2=ggplot(cozie_train_nosparse, aes(x=q_noise_nearby )) +
   #ggplot(data=tips, aes(x=day)) +
      geom_bar(aes(y = (..count..)/sum(..count..)),
               fill=rainbow(length(unique(q_noise_nearby))))+
  labs(y="Freq",x="q_noise_nearby")+
  theme(axis.text=element_text(size=5))

P3=ggplot(cozie_train_nosparse, aes(x=q_earphones )) +
   #ggplot(data=tips, aes(x=day)) +
      geom_bar(aes(y = (..count..)/sum(..count..)),
               fill=rainbow(length(unique(q_earphones))))+
  labs(y="Freq",x="q_earphones")+
  theme(axis.text=element_text(size=5))

P4=ggplot(cozie_train_nosparse, aes(x=q_noise_kind )) +
   #ggplot(data=tips, aes(x=day)) +
      geom_bar(aes(y = (..count..)/sum(..count..)),
               fill=rainbow(length(unique(q_noise_kind))))+
  labs(y="Freq",x="q_noise_kind")+
  theme(axis.text=element_text(size=5))

detach(cozie_train_nosparse)
ggarrange(P1,P2,P3,P4,nrow=2,ncol=2)
```



Distribution of the environmental variables for a given site

The environmental features have different levels of sparsity for a given site.

```
plot_onesite=function(x1=air_temp,x2=rainfall,x3=RH,x4=wind_dir,x5=wind_speed,
                      site="S24"){

  P_x1=ggplot(data=x1,aes(x=time,y=x1[[site]]))+
    geom_line(linewidth=.65,color="#9999CC")+
    labs(y="Air temperature",x="Time", title=site)+
    theme_bw()+
    scale_x_datetime(labels = date_format("%y-%m-%d"))+
    theme(axis.text=element_text(size=6),
        axis.title=element_text(size=6,face="bold"))

  P_x2=ggplot(data=x2,aes(x=time,y=x2[[site]]))+
    geom_line(linewidth=.65,color="#9999CC")+
    labs(y="Rainfall",x="Time")+
    theme_bw()+
    scale_x_datetime(labels = date_format("%y-%m-%d"))+
    theme(axis.text=element_text(size=6),
        axis.title=element_text(size=6,face="bold"))

  P_x3=ggplot(data=x3,aes(x=time,y=x3[[site]]))+
    geom_line(linewidth=.65,color="#9999CC")+
```

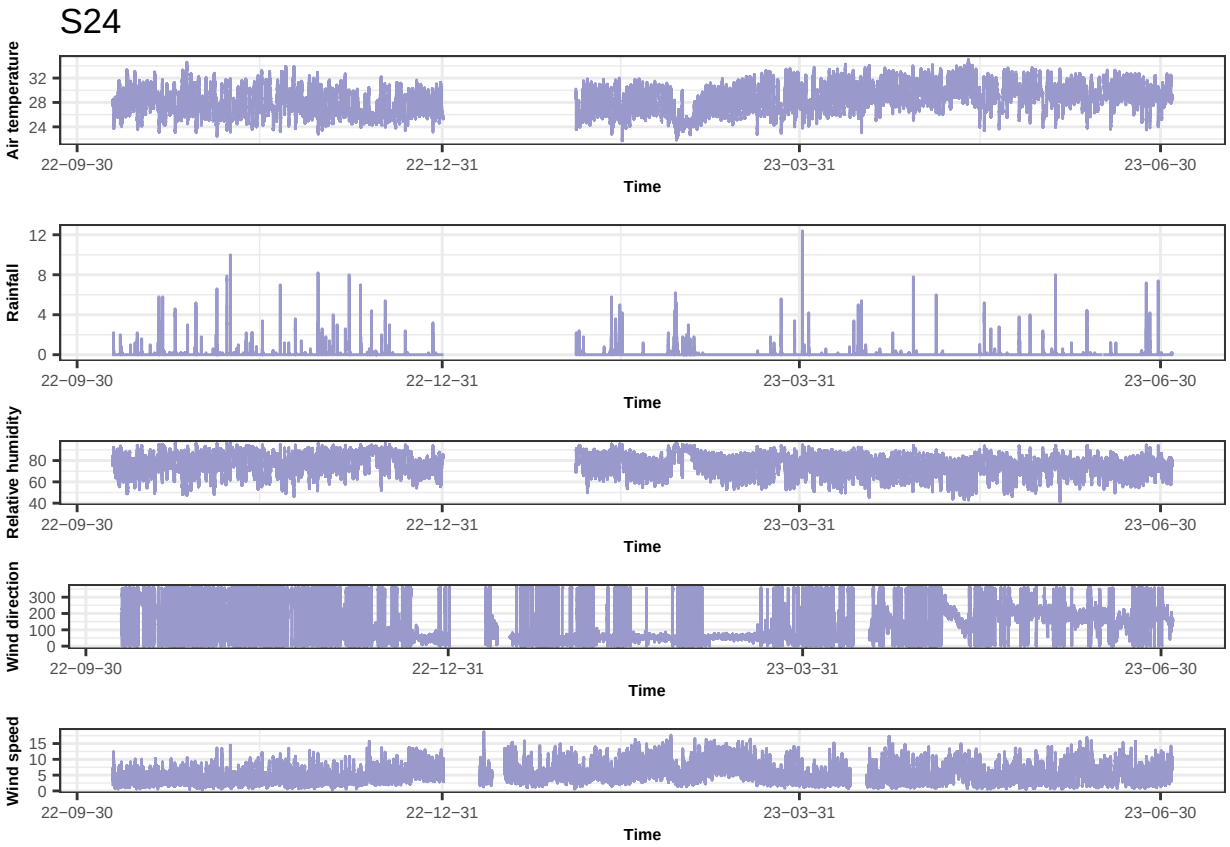
```
    labs(y="Relative humidity",x="Time")+
    theme_bw()+
    scale_x_datetime(labels = date_format("%y-%m-%d"))+
    theme(axis.text=element_text(size=6),
        axis.title=element_text(size=6,face="bold"))

  P_x4=ggplot(data=x4,aes(x=time,y=x4[[site]]))+
    geom_line(linewidth=.65,color="#9999CC")+
    labs(y="Wind direction",x="Time")+
    theme_bw()+
    scale_x_datetime(labels = date_format("%y-%m-%d"))+
    theme(axis.text=element_text(size=6),
        axis.title=element_text(size=6,face="bold"))

  P_x5=ggplot(data=x5,aes(x=time,y=x5[[site]]))+
    geom_line(linewidth=.65,color="#9999CC")+
    labs(y="Wind speed",x="Time")+
    theme_bw()+
    scale_x_datetime(labels = date_format("%y-%m-%d"))+
    theme(axis.text=element_text(size=6),
        axis.title=element_text(size=6,face="bold"))

  all_P=ggarrange(P_x1, P_x2,P_x3,P_x4,P_x5,
                  nrow = 5,heights = c(1.5,1.5,1,1,1))
  return(all_P)
}


plot_onesite(site="S24")
```

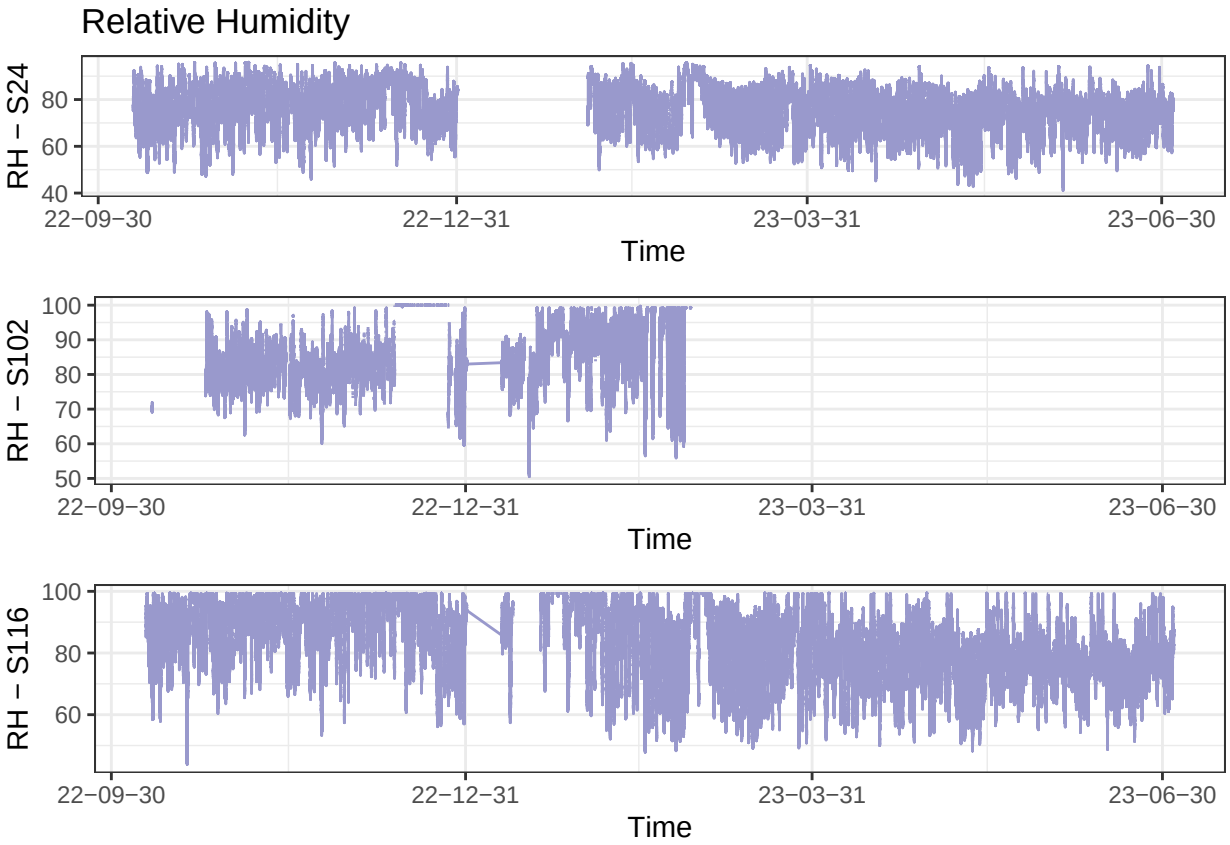


Distribution of one environmental variable for different sites

For a given feature, the level of sparsity differ for different sites.

```
P_s24_RH=ggplot(data=RH,aes(x=time,y=S24))+
  geom_line(linewidth=.65,color="#9999CC")+
  labs(y="RH - S24",x="Time",title="Relative Humidity")+
  theme_bw()+
  scale_x_datetime(labels = date_format("%y-%m-%d"))
P_s102_RH=ggplot(data=RH,aes(x=time,y=S102))+
  geom_line(linewidth=.65,color="#9999CC")+
  labs(y="RH - S102",x="Time")+
  theme_bw()+
  scale_x_datetime(labels = date_format("%y-%m-%d"))
P_s116_RH=ggplot(data=RH,aes(x=time,y=S116))+
  geom_line(linewidth=.65,color="#9999CC")+
  labs(y="RH - S116",x="Time")+
  theme_bw()+
  scale_x_datetime(labels = date_format("%y-%m-%d"))

ggarrange(P_s24_RH, P_s102_RH,P_s116_RH,
          nrow = 3)
```



Workflow

For our purposes, we need to construct a dataset containing a first column (some initial columns) with observation for the target feature(s) and, subsequently, all features we believe relevant for prediction (we call these explanatory features).

Regarding the cozie dataset, the target features are sparse, so we need to drop all NAs. Then, we need to parse the remaining observations with the "temporally closest" value of all explanatory features, or summary statistics of them computed based on most "recent" values.

Considering the varying levels of sparsity across datasets, we follow these steps:

1) Conduct spatio-temporal interpolation of environmental variables to populate empty cells for each time-step, for the entire spatial grid.
2) Synthesize this data by computing averages to align with the micro-survey datasets' dimensions. (Hint: we might average environmental observations (through EMAs) between two micro-survey observations).
3) Temporally match these new features with target variables, result: a complete dataset.
4) Train a machine learning algorithm and subsequently conduct accuracy tests.