MGT6203 Group Project

Final Report

04/21/2024

```
library(lubridate)
library(ggplot2)
library(dplyr)
library(plotly)
library(visdat)
library(tidyr)
library(data.table)
library(raster)
library(nnet)
library(purrr)
library(DataExplorer)
library(pscl)
library(tree)
library(rpart)
library(rpart.plot)
library(ISLR)
library(randomForest)
library(kableExtra)
library(broom)
library(rattle)
library(corrplot)
library(Metrics)
```

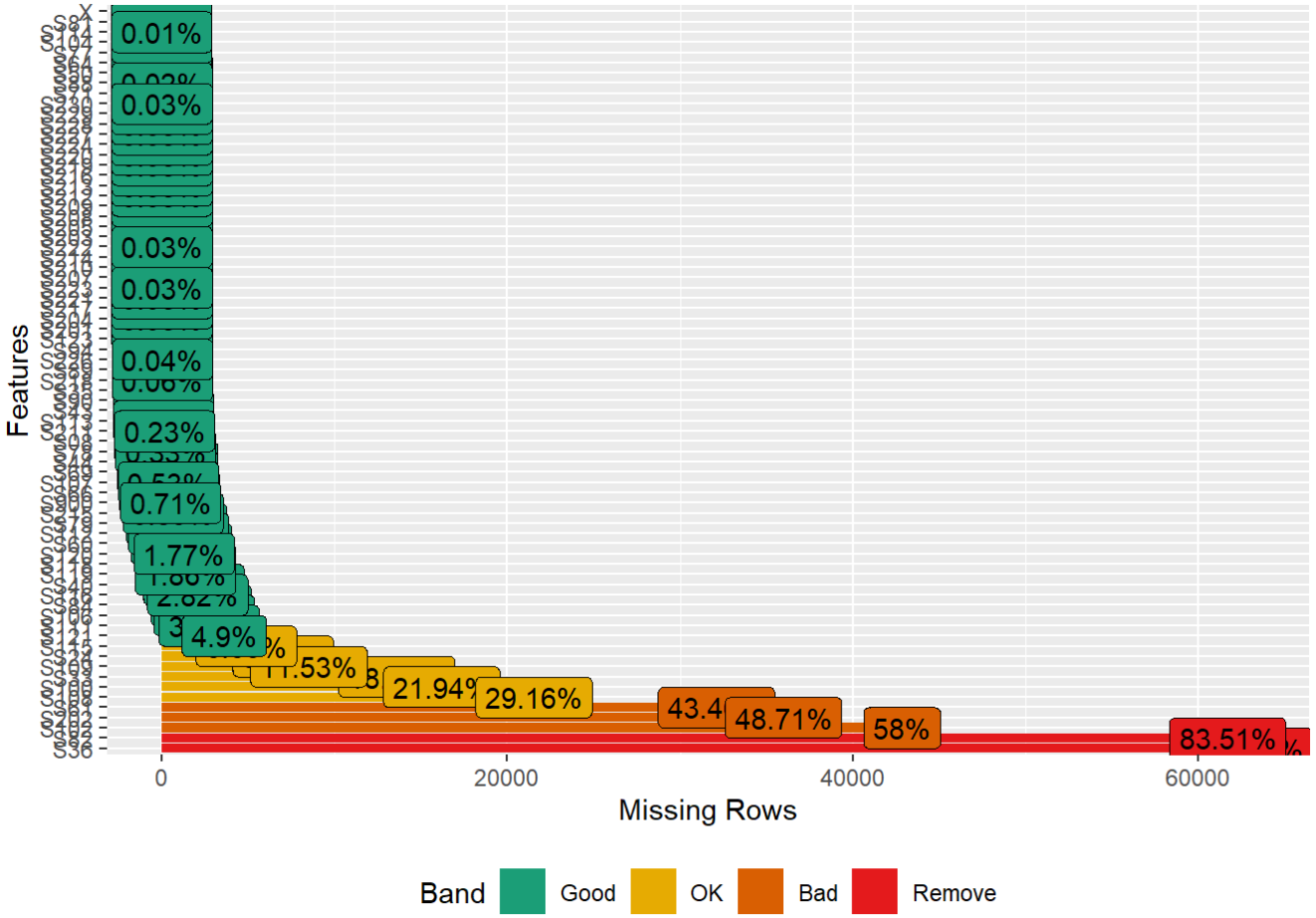
```
# read our data
survey_data_test <- read.csv("cozie_responses_and_physiological_data_test_public.csv",
                        sep = ",", header = TRUE)
```

```
# read our data
survey_data <- read.csv("cozie_responses_and_physiological_data_training.csv",
                        sep = ",", header = TRUE)
```

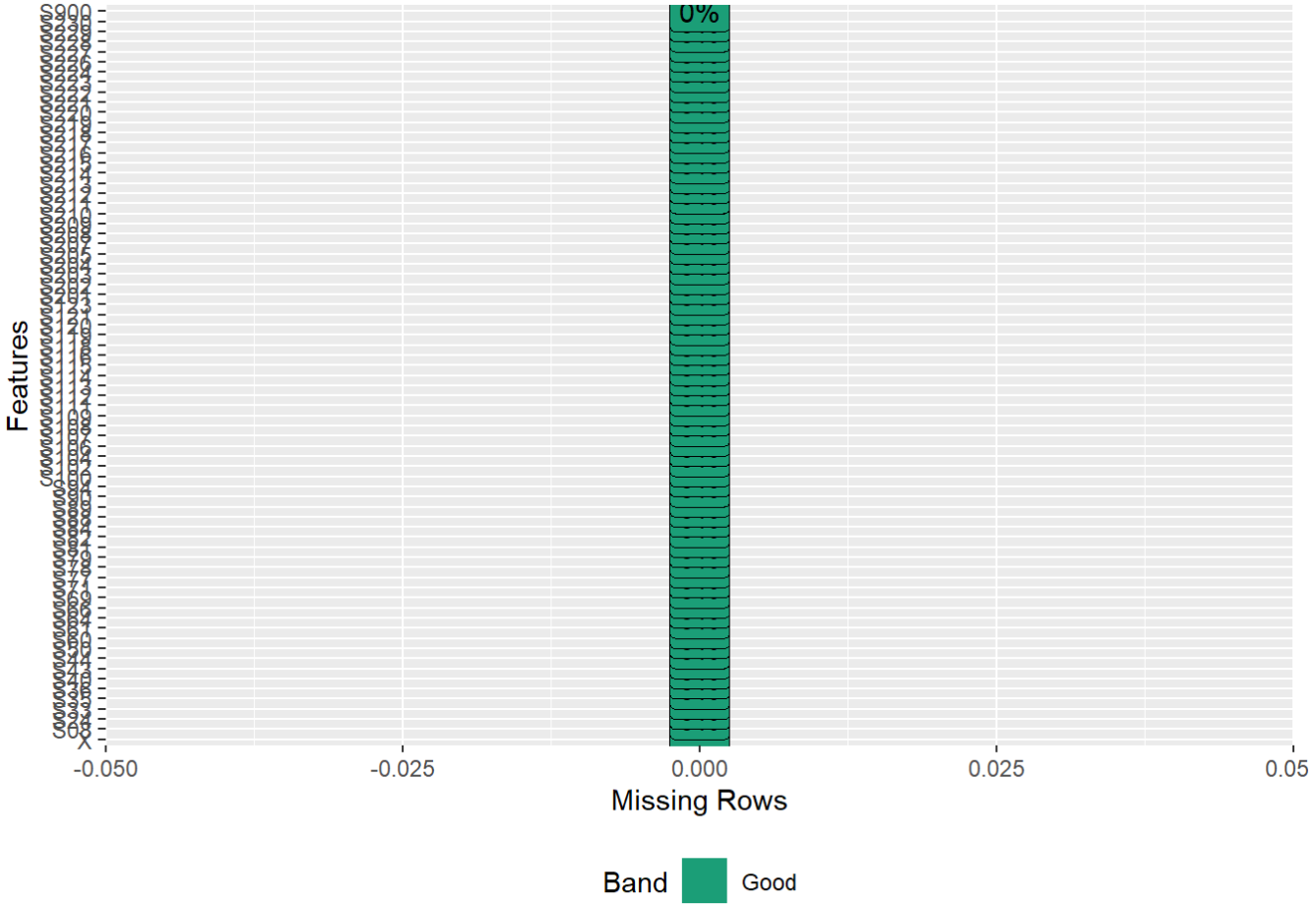
```
weather_rainfall_data <- read.csv("weather_rainfall.csv",
                        sep = ",", header = TRUE)

weather_wind_speed_data <- read.csv("weather_wind-speed.csv",
                        sep = ",", header = TRUE)

weather_wind_direction_data <- read.csv("weather_wind-direction.csv",
                        sep = ",", header = TRUE)

weather_stations_data <- read.csv("weather_stations.csv",
                        sep = ",", header = TRUE)

weather_temperature_data <- read.csv("weather_air-temperature.csv",
                        sep = ",", header = TRUE)

weather_humidity_data <- read.csv("weather_relative-humidity.csv",
                        sep = ",", header = TRUE)
```

```
clean_weather_data <- function(weather_data){
  # remove rows that have only missing values and replace remaining NA values with average
of row
  avg_temps <-
    rowMeans(subset(weather_data[rowSums(is.na(weather_data)) != ncol(weather_data),], sel
ect = c(-X)), na.rm = T)
  weather_data <-
    weather_data[rowSums(is.na(weather_data)) != ncol(weather_data),] %>%
    mutate(across(where(is.numeric),
                  ~ if_else(is.na(.), avg_temps, .)))

  weather_data <- weather_data %>%
    mutate_at(vars(colnames(weather_data)[colnames(weather_data) != "X"]), as.numeric)

  return(weather_data)
}
```

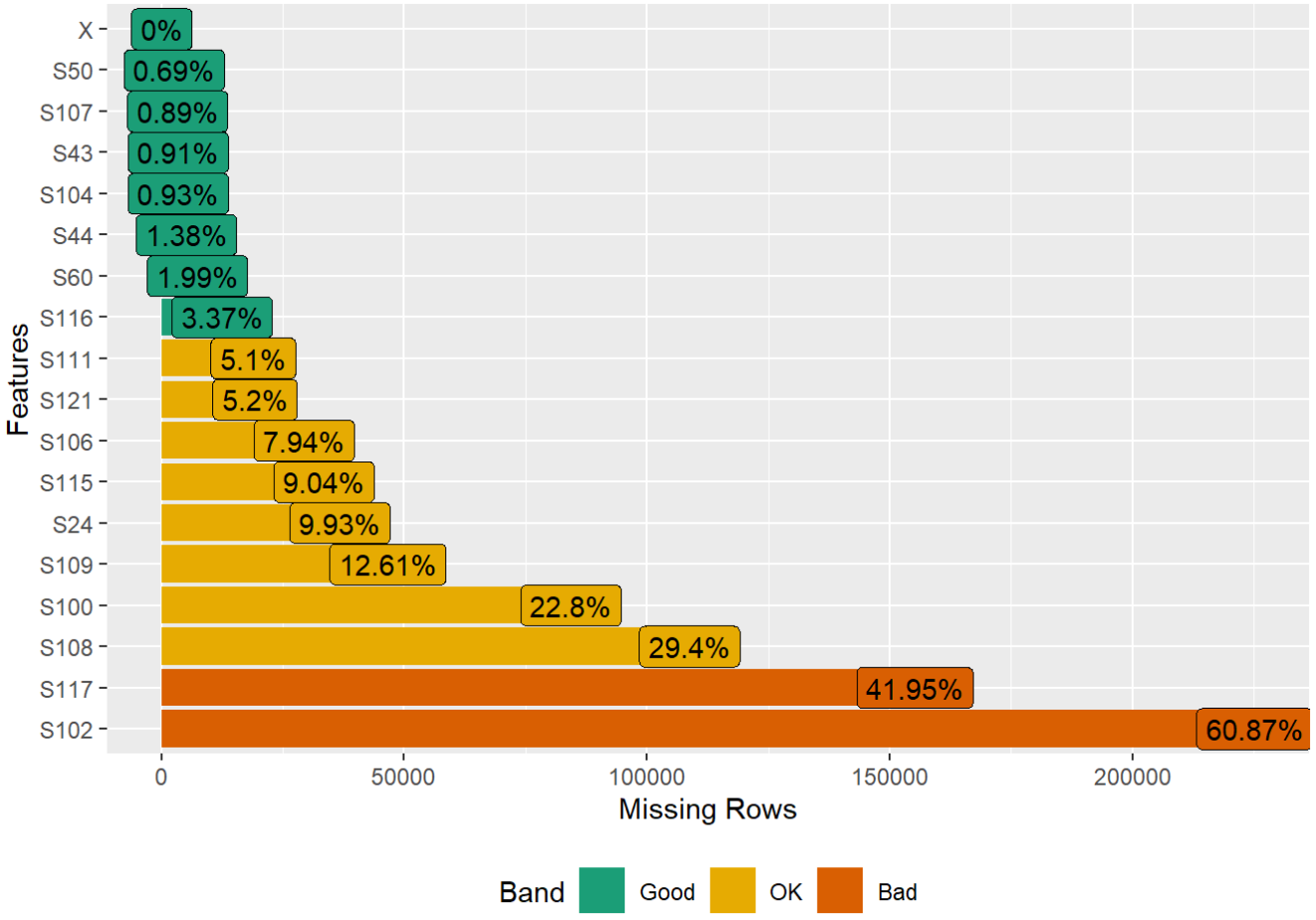
```
weather_rainfall_data %>% plot_missing()
```



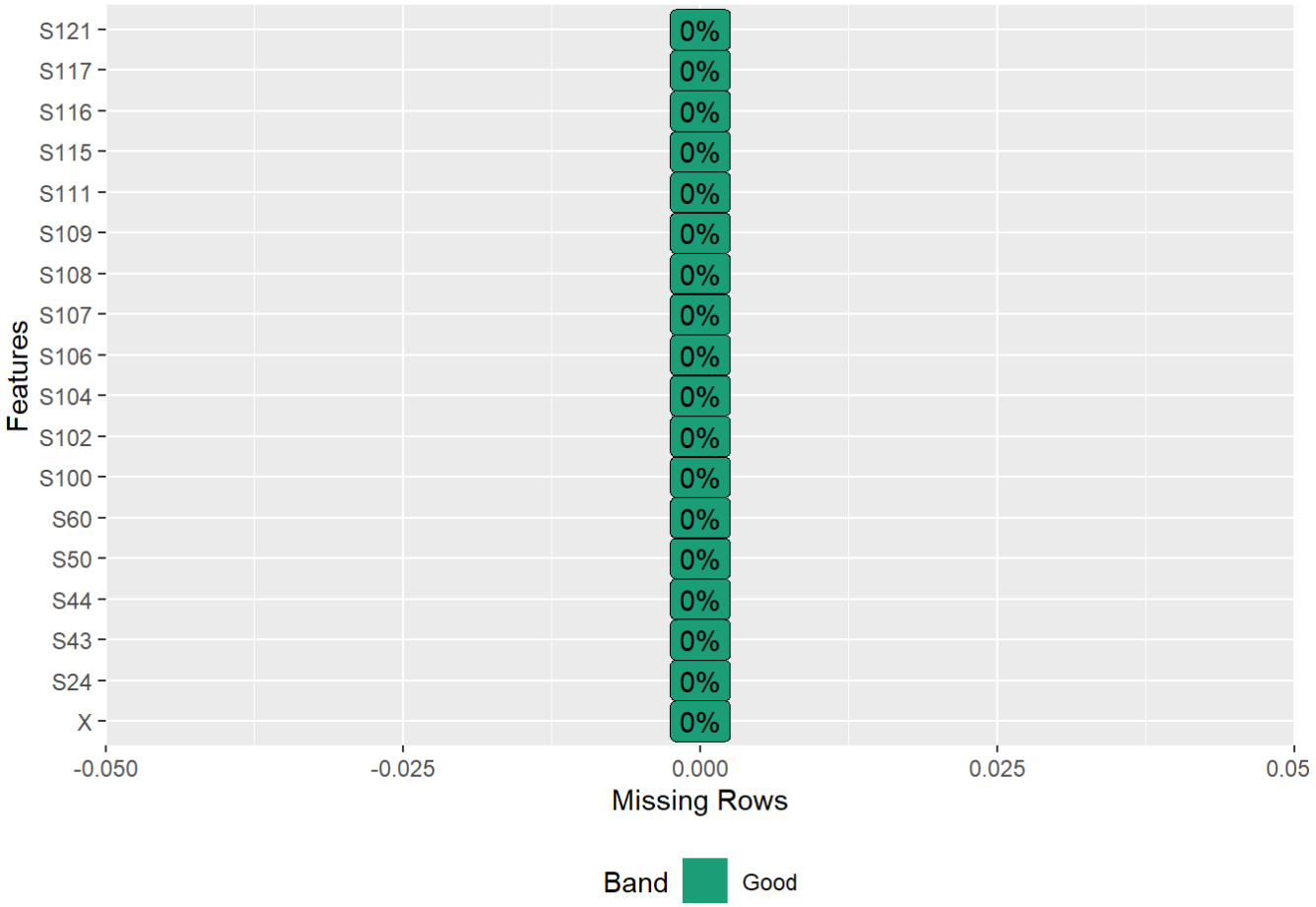

```
weather_rainfall_data_clean <-clean_weather_data(weather_rainfall_data)
weather_rainfall_data_clean %>% plot_missing()
```



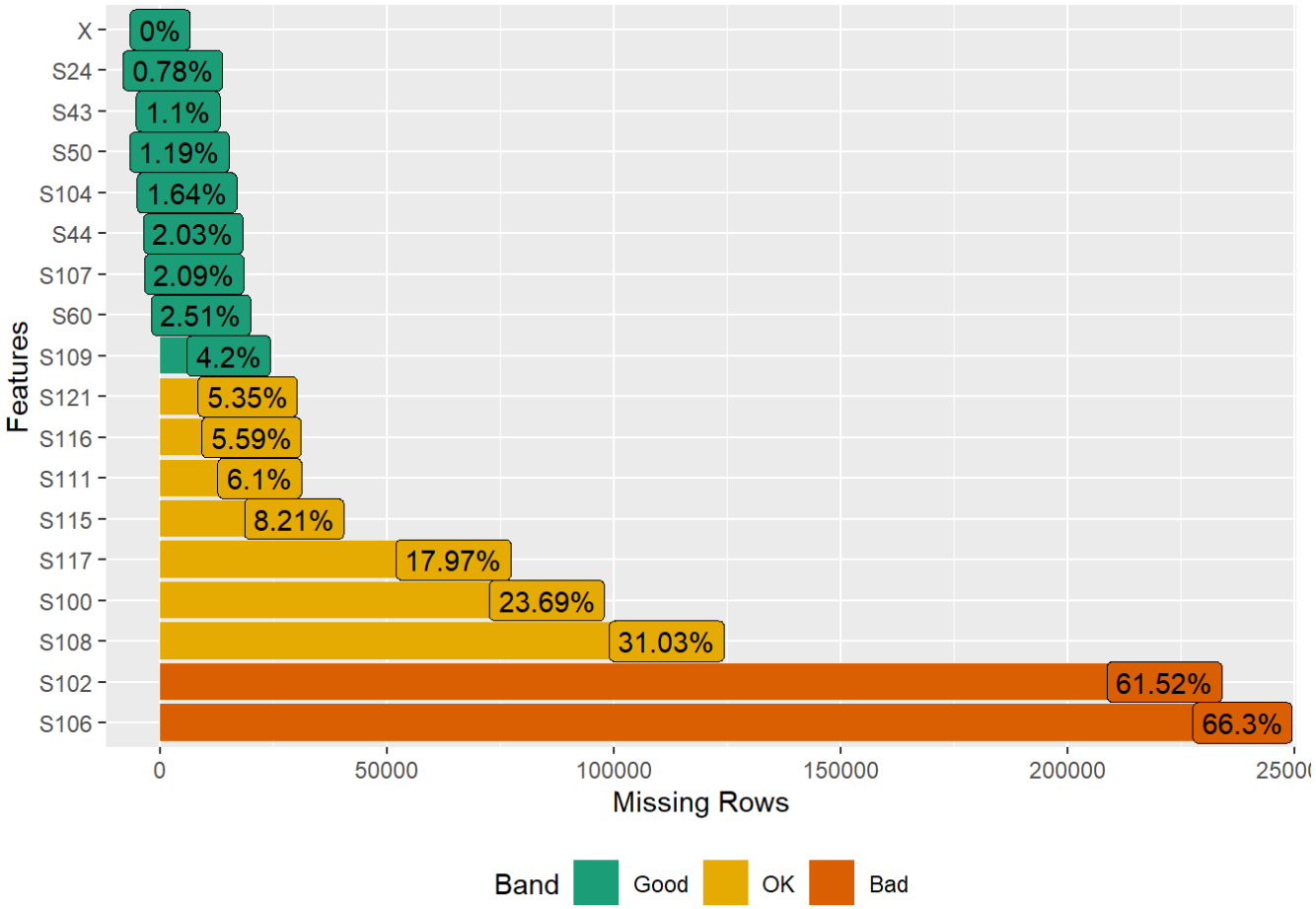

```
weather_temperature_data %>% plot_missing()
```



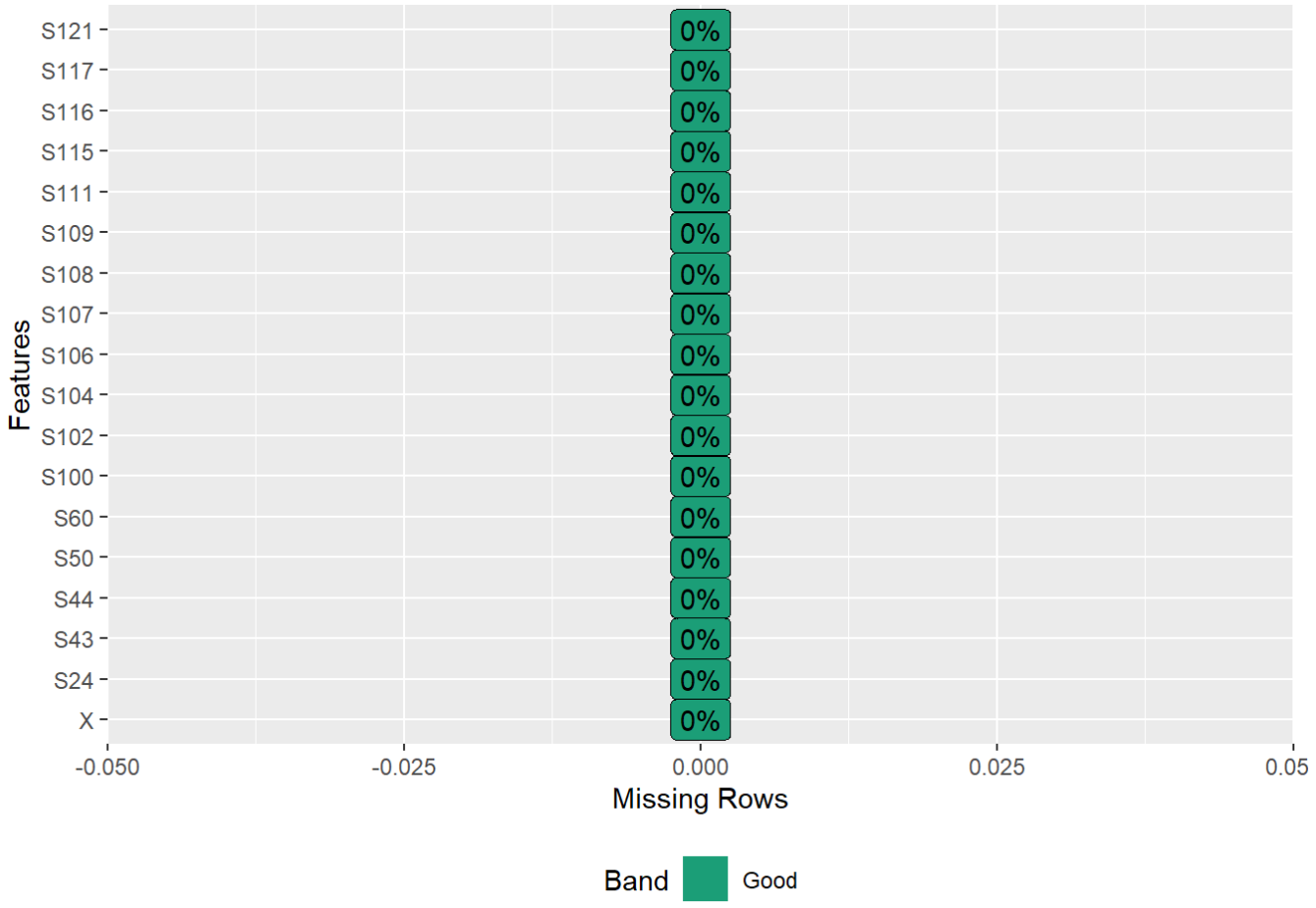
```
weather_temperature_data_clean <-clean_weather_data(weather_temperature_data)
weather_temperature_data_clean %>% plot_missing()
```



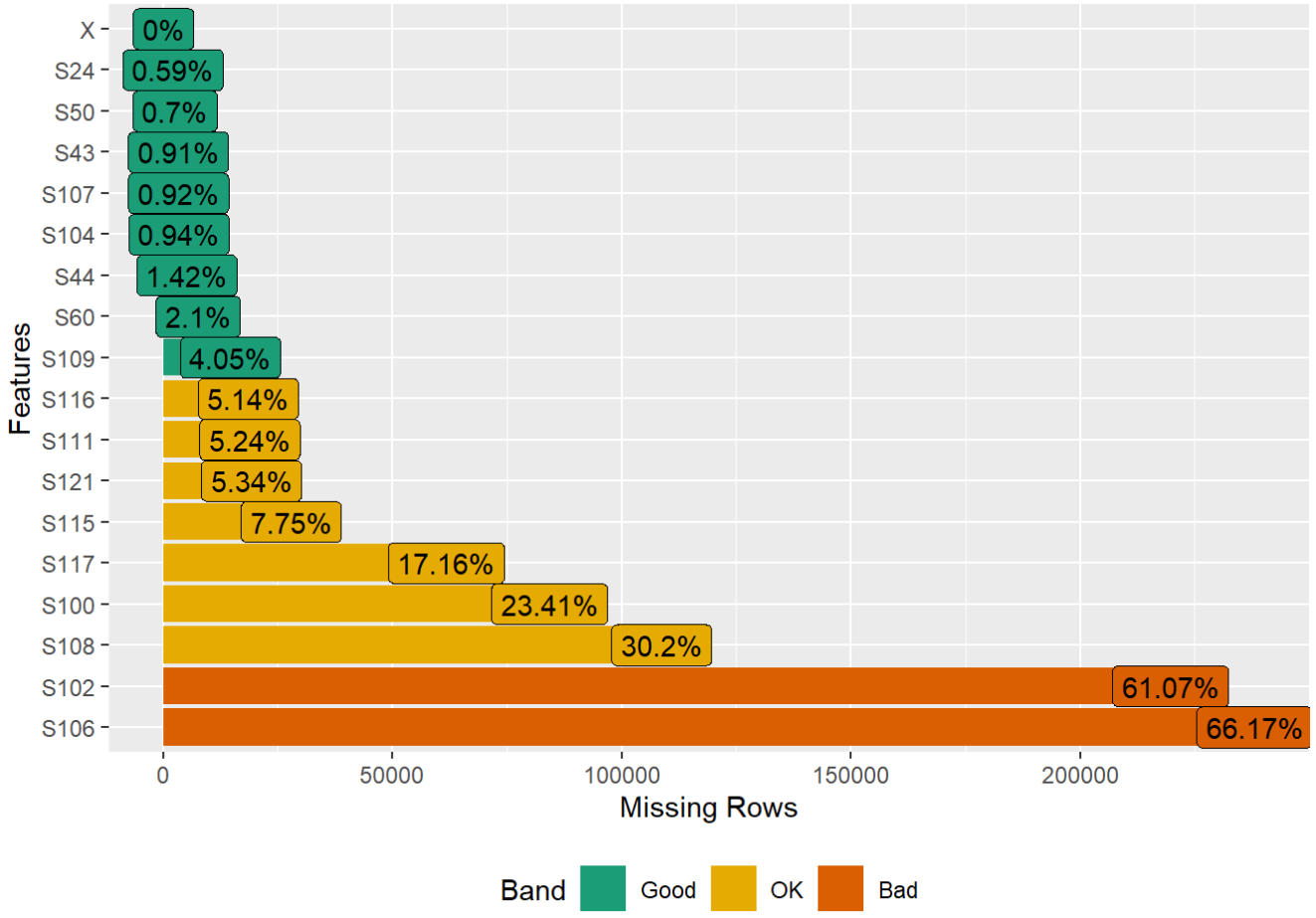

```
weather_wind_direction_data %>% plot_missing()
```



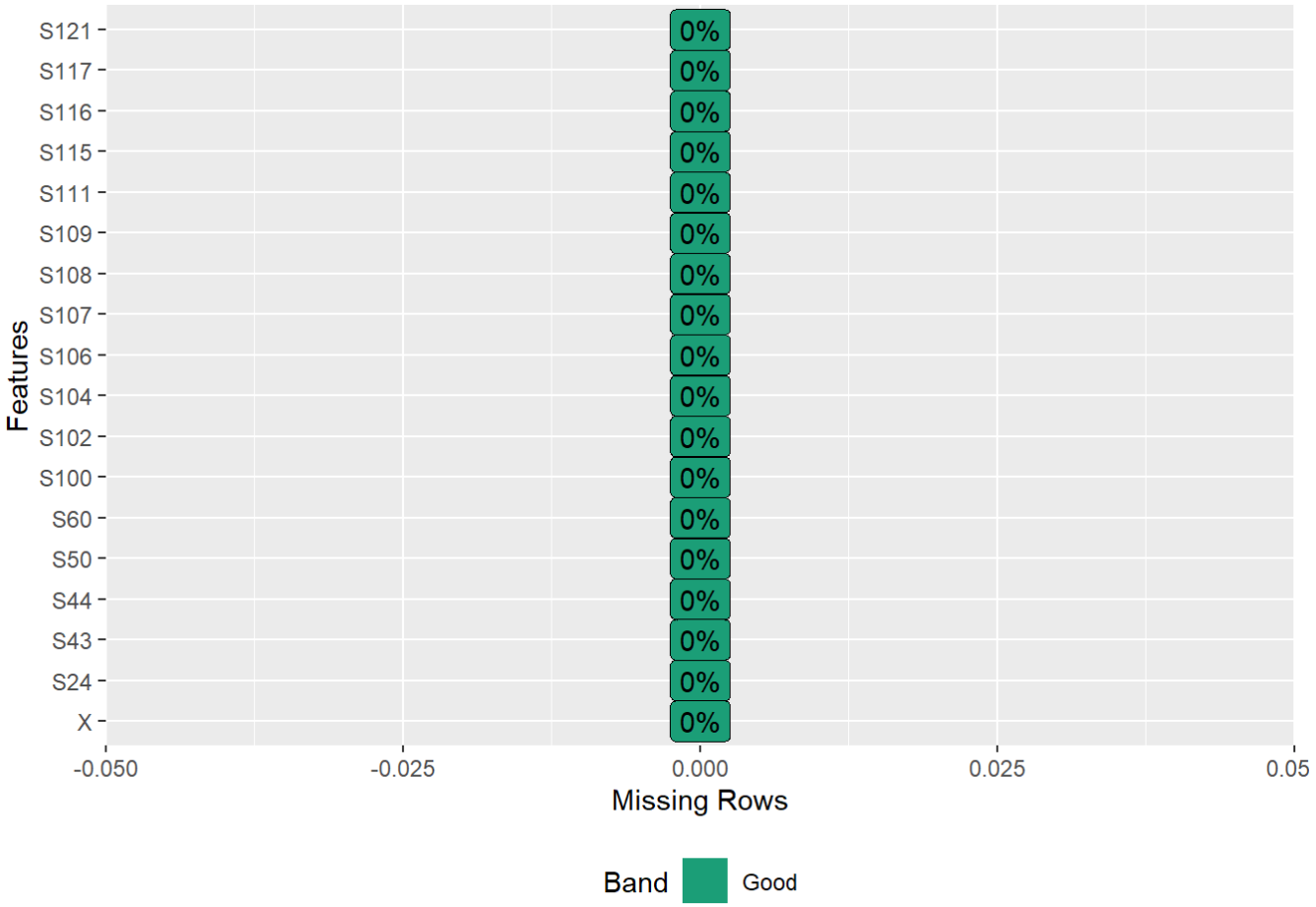

```
weather_wind_direction_data_clean <-clean_weather_data(weather_wind_direction_data)
weather_wind_direction_data_clean %>% plot_missing()
```



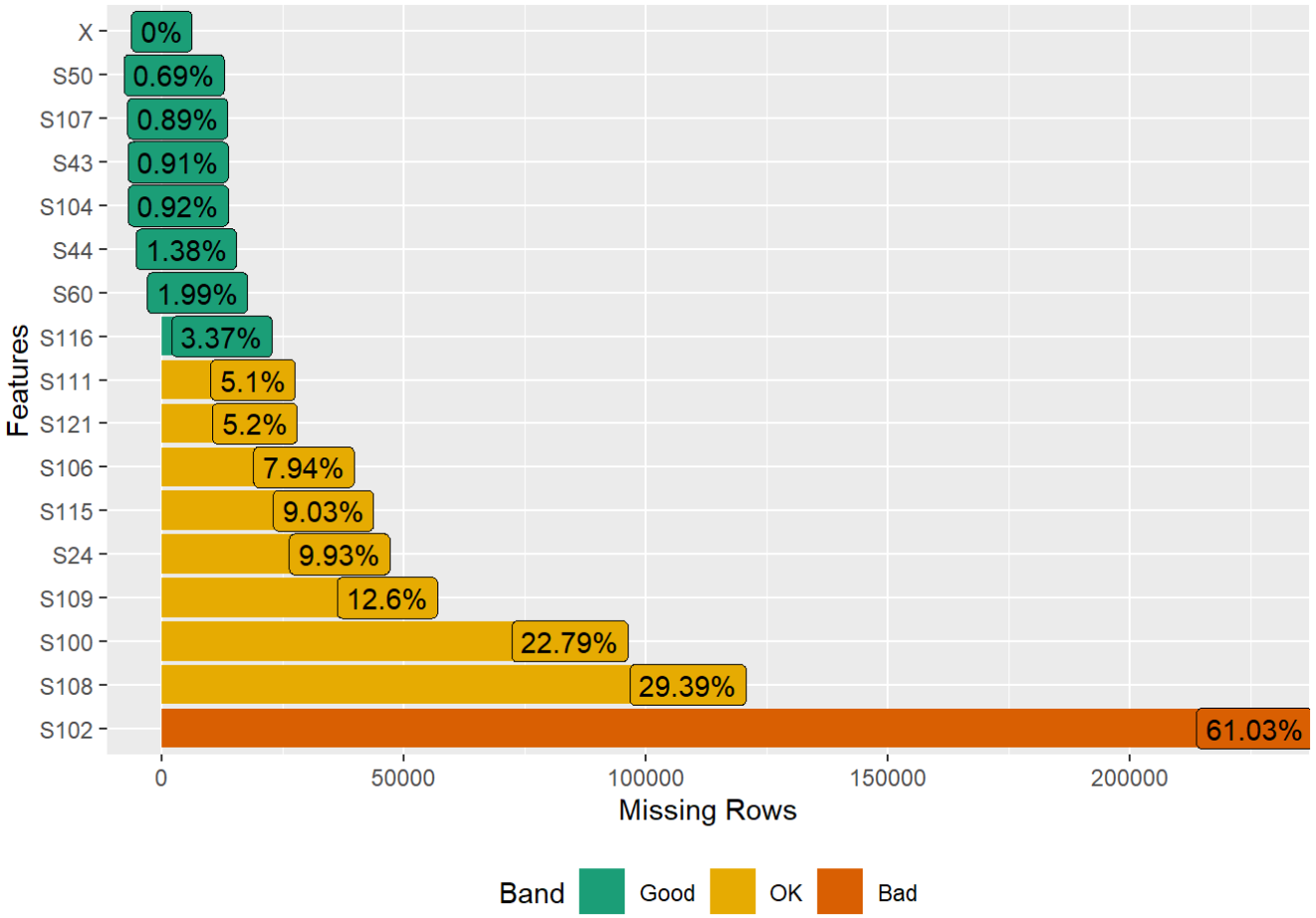

```
weather_wind_speed_data %>% plot_missing()
```



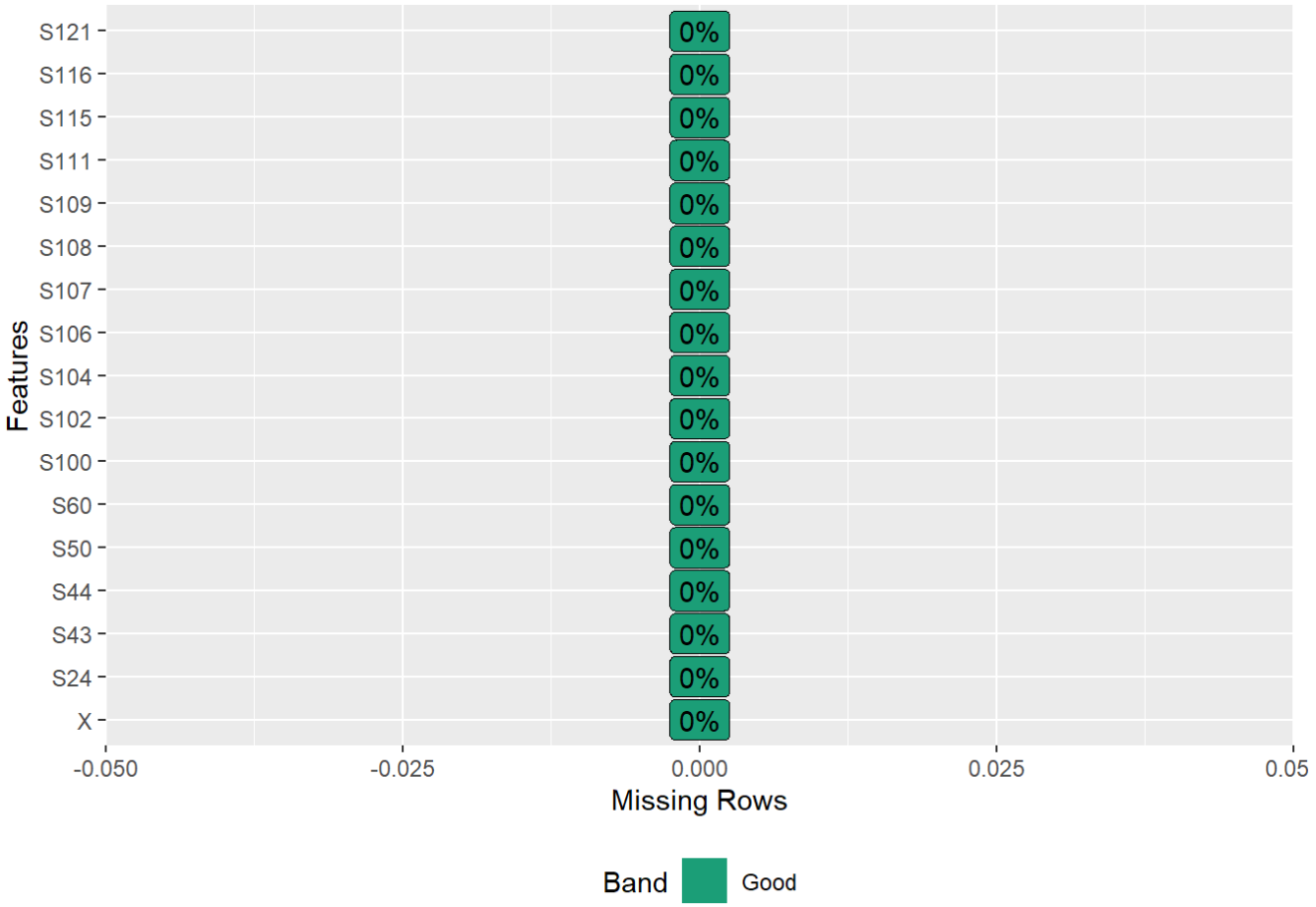
```
weather_wind_speed_data_clean <-clean_weather_data(weather_wind_speed_data)
weather_wind_speed_data_clean %>% plot_missing()
```




```
weather_humidity_data %>% plot_missing()
```




```
weather_humidity_data_clean <-clean_weather_data(weather_humidity_data)
weather_humidity_data_clean %>% plot_missing()
```

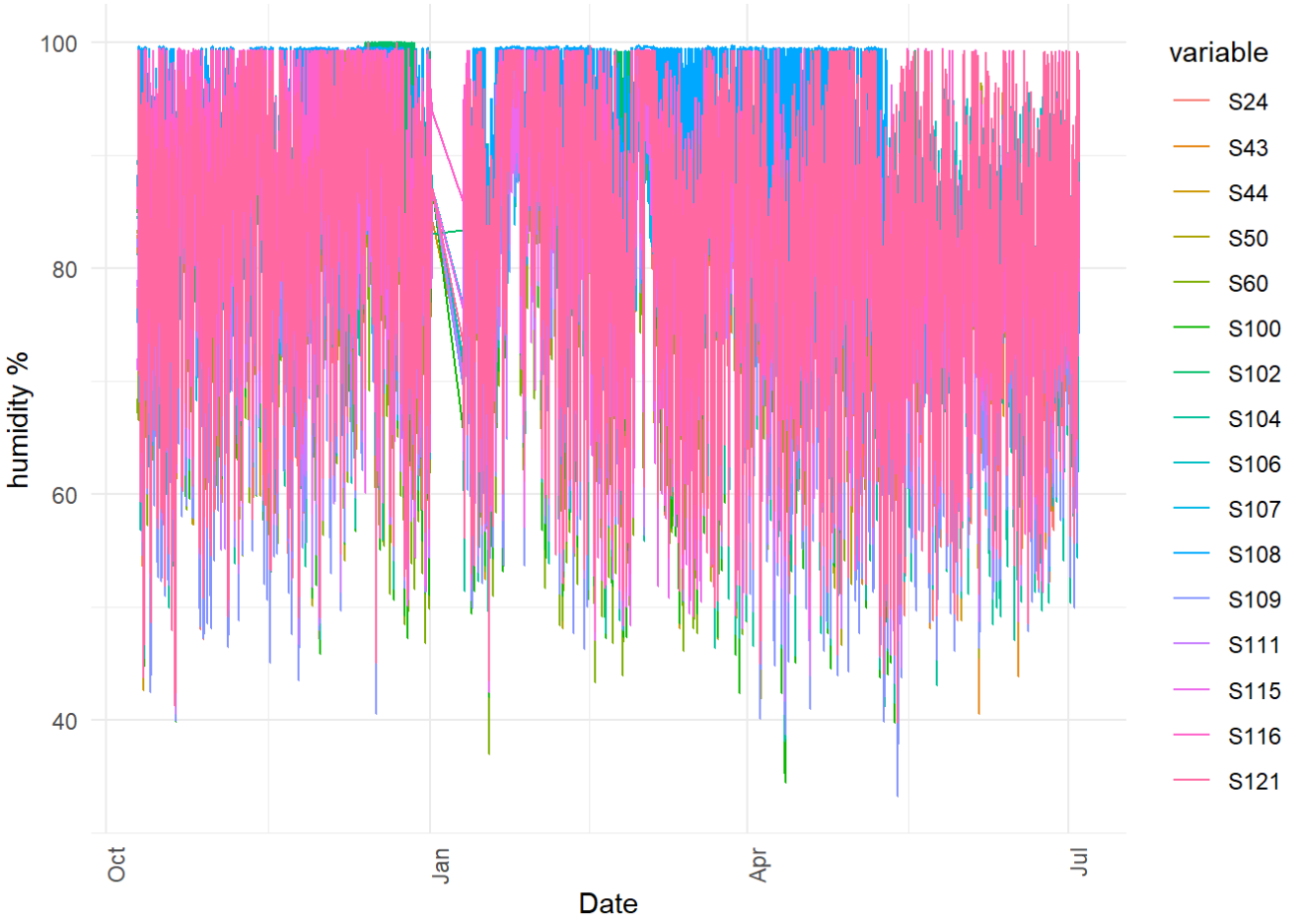

S121
S116
S115
S111
S109
S108
S107
S106
S104
S102
S100
S60
S50
S44
S43
S24
X
Features
0%
-0.050
-0.025
0.000
0.025
0.05
Missing Rows
Band
Good

```
weather_humidity_data_clean <- weather_humidity_data_clean %>% # convert time to time obje
ct
  #mutate(date_time = ymd_hms(weather_humidity_data_clean$X))
  mutate(date_time = ymd_hms(unlist(map(strsplit(weather_humidity_data_clean$X, split='+',
fixed=TRUE), 1))))

weather_temperature_data_clean <- weather_temperature_data_clean %>% # convert time to tim
e object
  #mutate(date_time = ymd_hms(weather_temperature_data_clean$X))
  mutate(date_time = ymd_hms(unlist(map(strsplit(weather_temperature_data_clean$X, split
='+', fixed=TRUE), 1))))

weather_wind_direction_data_clean <- weather_wind_direction_data_clean %>% # convert time
to time object
  #mutate(date_time = ymd_hms(weather_wind_direction_data_clean$X))
  mutate(date_time = ymd_hms(unlist(map(strsplit(weather_wind_direction_data_clean$X, spli
t='+', fixed=TRUE), 1))))

weather_wind_speed_data_clean <- weather_wind_speed_data_clean %>% # convert time to time
object
  #mutate(date_time = ymd_hms(weather_wind_speed_data_clean$X))
  mutate(date_time = ymd_hms(unlist(map(strsplit(weather_wind_speed_data_clean$X, split
='+', fixed=TRUE), 1))))

weather_rainfall_data_clean <- weather_rainfall_data_clean %>% # convert time to time obje
ct
  #mutate(date_time = ymd_hms(weather_rainfall_data_clean$X))
  mutate(date_time = ymd_hms(unlist(map(strsplit(weather_rainfall_data_clean$X, split='+',
fixed=TRUE), 1))))
```

```
inspect_weather_data <- function(weather_data, value_name){
  weather_start_date <- min(weather_data$date_time)
  weather_end_date <- max(weather_data$date_time)
  weather_duration <- max(weather_data$date_time) - min(weather_data$date_time)
  weather_frequency <- nrow(weather_data) / as.numeric(weather_duration)
  print(paste('First day of measurement', weather_start_date))
  print(paste('Last day of measurement', weather_end_date))
  print(paste('Duration of measurement in days', weather_duration))
  print(paste('Frequency of measurement per day', weather_frequency))

  weather_plot <-
    ggplot(data = as.data.frame(reshape2::melt(subset(weather_data, select = -c(X)), id="d
ate_time")), aes(x = date_time, y = value, col = variable)) +
    geom_line() +
    theme_minimal() +
    theme() +
    theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
    xlab("Date") +
    ylab(value_name)

  return(weather_plot)
}
```

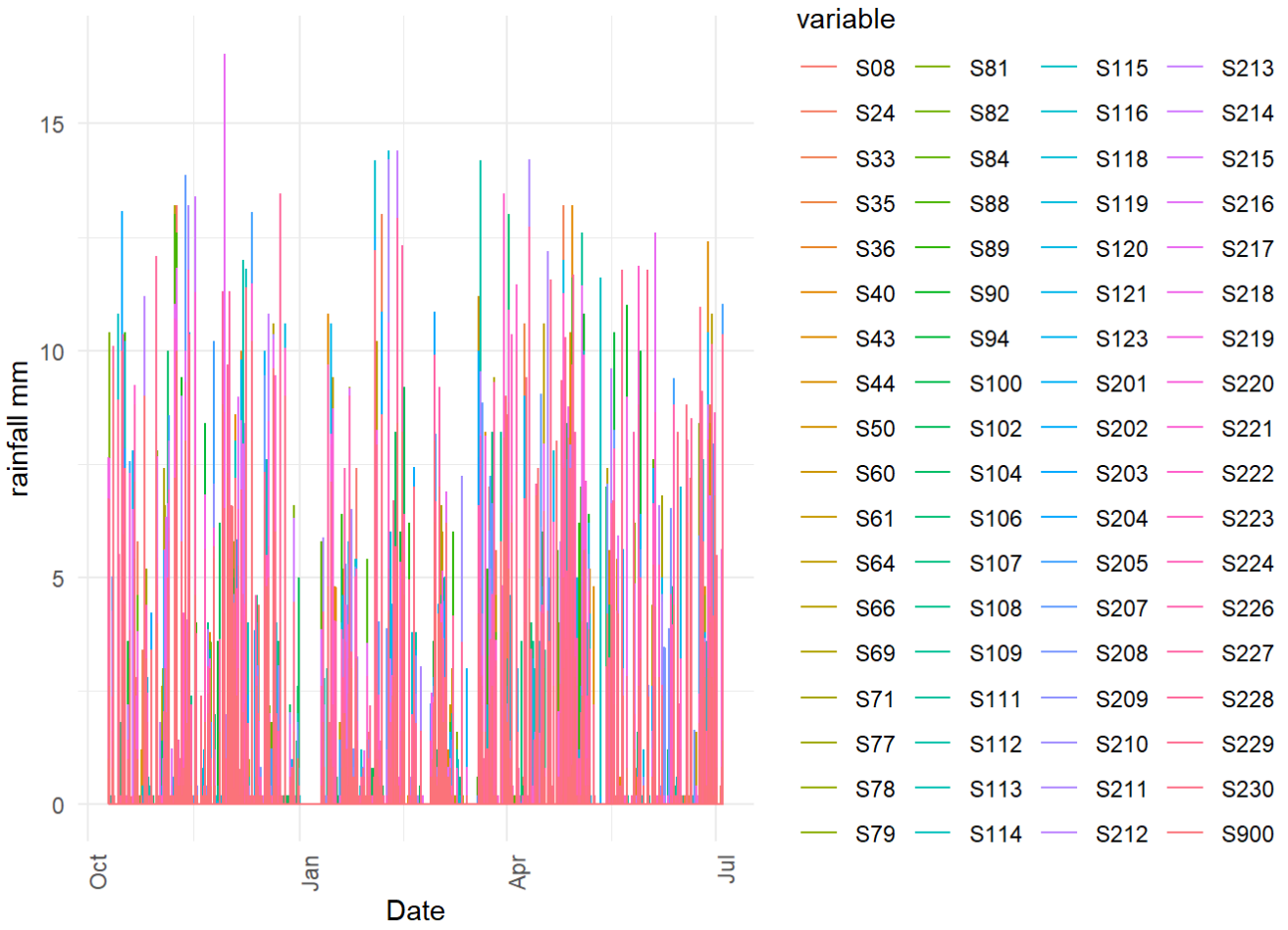
```
inspect_weather_data(weather_humidity_data_clean, "humidity %")
```

```
## [1] "First day of measurement 2022-10-10 00:01:00"
## [1] "Last day of measurement 2023-07-03 23:59:00"
## [1] "Duration of measurement in days 266.998611111111"
## [1] "Frequency of measurement per day 1384.83866437091"
```



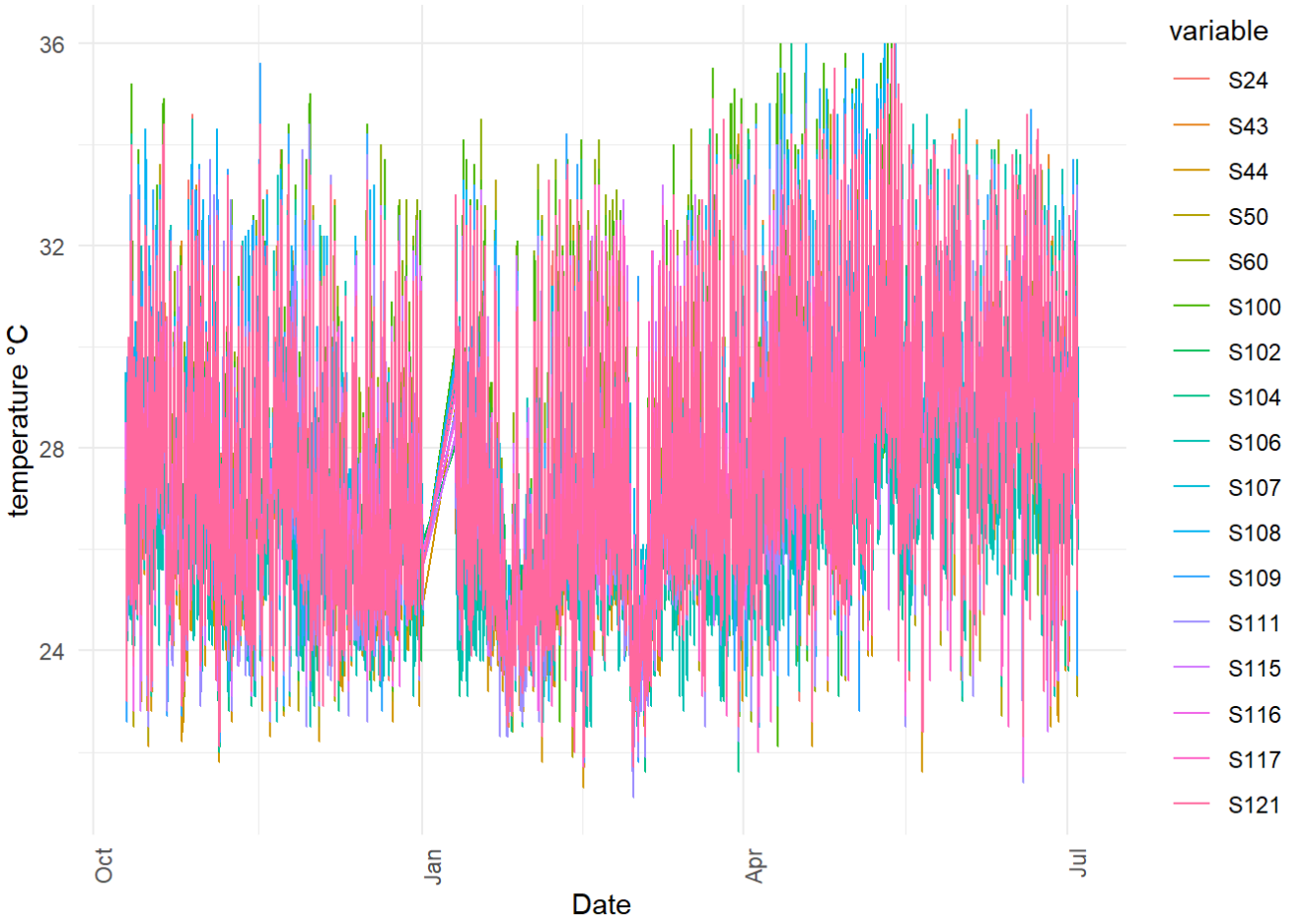

```
inspect_weather_data(weather_rainfall_data_clean, "rainfall mm")
```

```
## [1] "First day of measurement 2022-10-10 00:05:00"
## [1] "Last day of measurement 2023-07-03 23:55:00"
## [1] "Duration of measurement in days 266.993055555556"
## [1] "Frequency of measurement per day 276.598954404765"
```



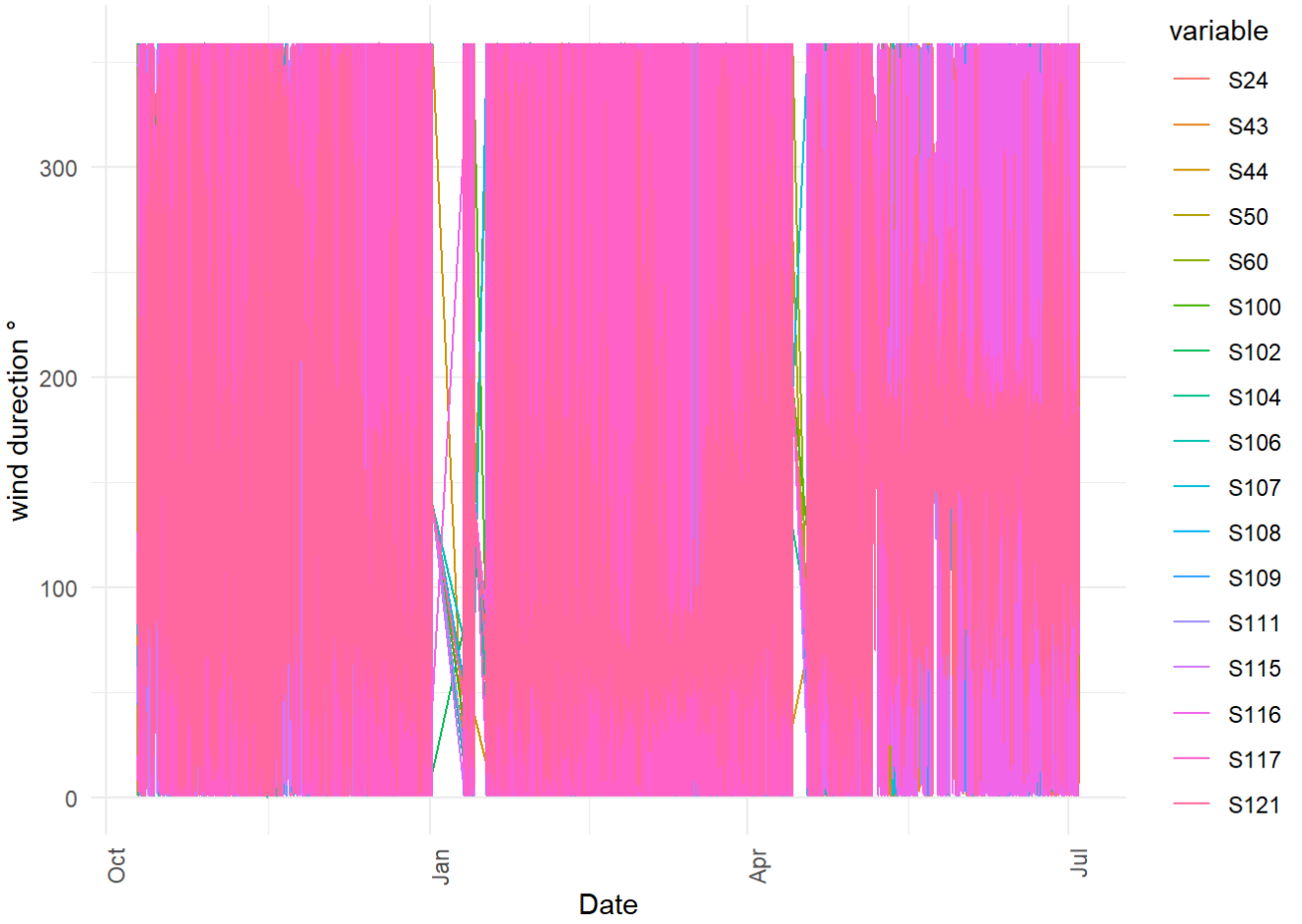

```
inspect_weather_data(weather_temperature_data_clean, "temperature °C")
```

```
## [1] "First day of measurement 2022-10-10 00:01:00"
## [1] "Last day of measurement 2023-07-03 23:59:00"
## [1] "Duration of measurement in days 266.998611111111"
## [1] "Frequency of measurement per day 1385.01094991131"
```



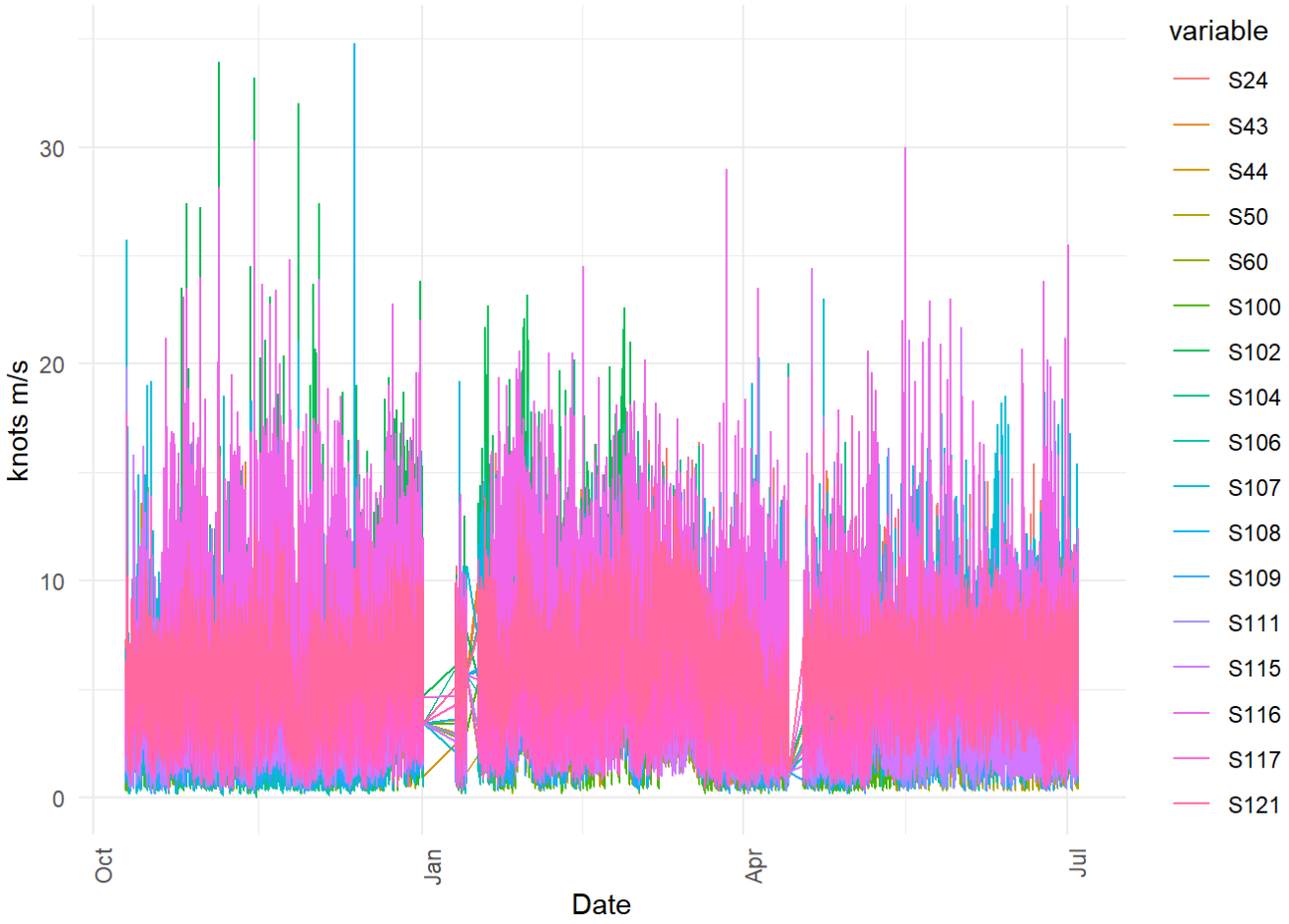

```
inspect_weather_data(weather_wind_direction_data_clean, "wind durection °")
```

```
## [1] "First day of measurement 2022-10-10 00:01:00"
## [1] "Last day of measurement 2023-07-03 23:59:00"
## [1] "Duration of measurement in days 266.998611111111"
## [1] "Frequency of measurement per day 1347.32910595665"
```




```
inspect_weather_data(weather_wind_speed_data_clean, "knots m/s")
```

```
## [1] "First day of measurement 2022-10-10 00:01:00"
## [1] "Last day of measurement 2023-07-03 23:59:00"
## [1] "Duration of measurement in days 266.998611111111"
## [1] "Frequency of measurement per day 1347.33285129448"
```



How many weather stations are there?

```
str(weather_stations_data)
```

```
## 'data.frame':    74 obs. of  5 variables:
##  $ X        : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ id       : chr  "S07" "S08" "S100" "S102" ...
##  $ name     : chr  "Lornie Road" "Upper Thomson Road" "Woodlands Road" "Semakau Landfil
l" ...
##  $ latitude : num  1.34 1.37 1.42 1.19 1.44 ...
##  $ longitude: num  104 104 104 104 104 ...
```

What are the variable names?

```
names(survey_data)
```

```
##  [1] "time"                          "q_alone_group"
##  [3] "q_earphones"                   "id_participant"
##  [5] "ws_latitude"                   "q_location"
##  [7] "q_location_office"             "q_location_transport"
##  [9] "ws_longitude"                  "q_noise_kind"
## [11] "q_noise_nearby"                "q_thermal_preference"
## [13] "ws_timestamp_location"         "ws_timestamp_start"
## [15] "ts_oxygen_saturation"          "ts_resting_heart_rate"
## [17] "ts_stand_time"                 "ts_step_count"
## [19] "ts_walking_distance"           "ws_survey_count"
## [21] "Footprint.Proportion"          "Footprint.Mean"
## [23] "Footprint.Stdev"               "Perimeter.Total"
## [25] "Perimeter.Mean"                "Perimeter.Stdev"
## [27] "Complexity.Mean"               "Complexity.Stdev"
## [29] "Building.Count"                "PopSum"
## [31] "Men"                           "Women"
## [33] "Elderly"                       "Youth"
## [35] "Children"                      "Civic"
## [37] "Commercial"                    "Entertainment"
## [39] "Food"                          "Healthcare"
## [41] "Institutional"                 "Recreational"
## [43] "Social"                        "Green.View.Mean"
## [45] "Green.View.Stdev"              "Sky.View.Mean"
## [47] "Sky.View.Stdev"                "Building.View.Mean"
## [49] "Building.View.Stdev"           "Road.View.Mean"
## [51] "Road.View.Stdev"               "Visual.Complexity.Mean"
## [53] "Visual.Complexity.Stdev"       "dT"
## [55] "q_activity_category_alone"     "q_activity_category_group"
## [57] "ts_heart_rate"                 "ts_audio_exposure_environment"
## [59] "id_unique"
```

```
survey_data <- survey_data %>%
  # convert time to time object
  #mutate(date_time = ymd_hms(survey_data$time)) %>%
  mutate(date_time = ymd_hms(unlist(map(strsplit(survey_data$time, split='.', fixed=TRUE),
1)))) %>%
  # replace empty strings with NA
  mutate(across(where(is.character), ~na_if(., "")))
head(survey_data)
```

	time <chr>	**q_alone_group** <chr>	**q_earphones** <chr>	**id_participant** <chr>	**w**
1	2022-10-10 09:32:04.588000+0800	*NA*	*NA*	xesh001	
2	2022-10-10 09:32:04.588000+0800	*NA*	*NA*	xesh001	
3	2022-10-10 09:33:08.713000+0800	*NA*	*NA*	xesh001	
4	2022-10-10 09:35:11.600000+0800	*NA*	*NA*	xesh001	
5	2022-10-10 09:38:59.100000+0800	*NA*	*NA*	xesh001	
6	2022-10-10 09:43:32.100000+0800	*NA*	*NA*	xesh001	

6 rows | 1-7 of 61 columns

```
str(survey_data)
```

```
## 'data.frame':    1149136 obs. of  60 variables:
##  $ time                          : chr  "2022-10-10 09:32:04.588000+0800" "2022-10-10 0
9:32:04.588000+0800" "2022-10-10 09:33:08.713000+0800" "2022-10-10 09:35:11.600000+0800"
...
##  $ q_alone_group                 : chr  NA NA NA NA ...
##  $ q_earphones                   : chr  NA NA NA NA ...
##  $ id_participant                : chr  "xesh001" "xesh001" "xesh001" "xesh001" ...
##  $ ws_latitude                   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ q_location                    : chr  NA NA NA NA ...
##  $ q_location_office             : chr  NA NA NA NA ...
##  $ q_location_transport          : chr  NA NA NA NA ...
##  $ ws_longitude                  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ q_noise_kind                  : chr  NA NA NA NA ...
##  $ q_noise_nearby                : chr  NA NA NA NA ...
##  $ q_thermal_preference          : chr  NA NA NA NA ...
##  $ ws_timestamp_location         : chr  NA NA NA NA ...
##  $ ws_timestamp_start            : chr  NA NA NA NA ...
##  $ ts_oxygen_saturation          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ ts_resting_heart_rate         : num  57 NA NA NA NA NA NA NA NA NA ...
##  $ ts_stand_time                 : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ ts_step_count                 : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ ts_walking_distance           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ ws_survey_count               : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Footprint.Proportion          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Footprint.Mean                : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Footprint.Stdev               : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Perimeter.Total               : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Perimeter.Mean                : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Perimeter.Stdev               : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Complexity.Mean               : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Complexity.Stdev              : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Building.Count                : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ PopSum                        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Men                           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Women                         : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Elderly                       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Youth                         : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Children                      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Civic                         : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Commercial                    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Entertainment                 : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Food                          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Healthcare                    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Institutional                 : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Recreational                  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Social                        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Green.View.Mean               : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Green.View.Stdev              : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Sky.View.Mean                 : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Sky.View.Stdev                : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Building.View.Mean            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Building.View.Stdev           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Road.View.Mean                : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Road.View.Stdev               : num  NA NA NA NA NA NA NA NA NA NA ...
```

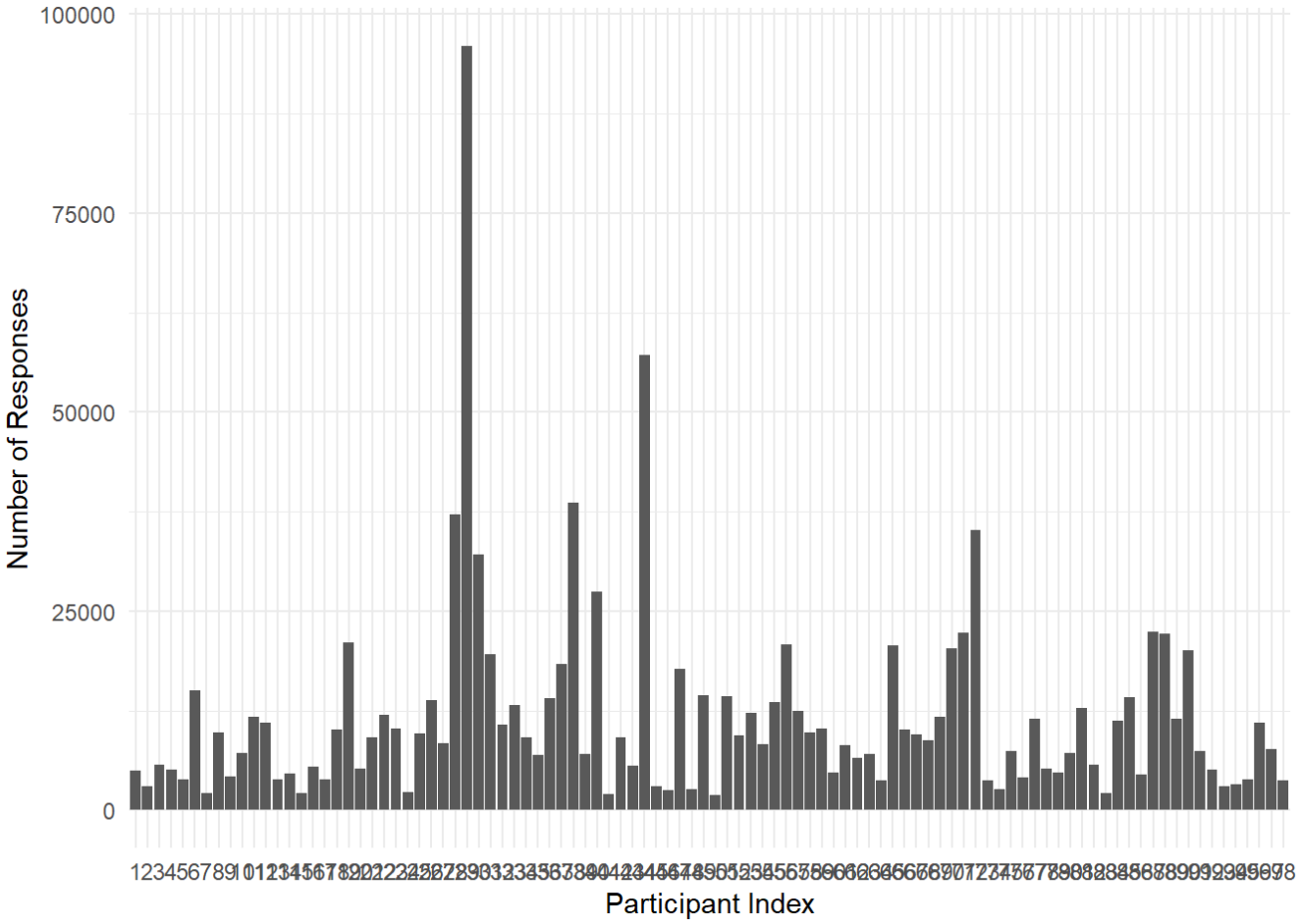
```
##  $ Visual.Complexity.Mean       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Visual.Complexity.Stdev      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ dT                           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ q_activity_category_alone    : chr  NA NA NA NA ...
##  $ q_activity_category_group    : chr  NA NA NA NA ...
##  $ ts_heart_rate                : num  NA 83 59 61 63 67 74 76 78 90 ...
##  $ ts_audio_exposure_environment: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ id_unique                    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ date_time                    : POSIXct, format: "2022-10-10 09:32:04" "2022-10-10 09
:32:04" ...
```

How many users are in the study? How many data points for each user?

```
nrow(table(survey_data$id_participant))
```

```
## [1] 98
```

```
ggplot(survey_data[!is.na(survey_data$id_participant),],aes(id_participant)) +
  geom_bar(stat = "count", position = "dodge") +
  xlab("Participant Index") +
  ylab("Number of Responses") +
  scale_x_discrete(labels=seq(1,nrow(table(survey_data$id_participant)))) +
  theme_minimal()
```



```
#
# ggsave("Number_of_Responses_per_partisipant.png",
#        width = 18, height = 6, dpi = 200, units = "in", device='png')
```

```
i1 <- which(!is.na(survey_data$q_thermal_preference))

logs <- data.frame(log_row_index = rownames(survey_data[i1,]),
                   date_time = survey_data[i1,]$date_time)
```

```
grouped_logs <- merge(survey_data, logs, by = "date_time", all.x = TRUE) %>%
  arrange(date_time) %>% # sort by time
  fill(log_row_index, .direction = "down") %>% # fill group index
  fill(log_row_index, .direction = "up") %>% # fill group index
  group_by(id_participant) %>% # group by participant
  arrange(id_participant, date_time) %>% #sort by user id then by time
  group_by(log_row_index) # group by log group

nrow(grouped_logs)
```

```
## [1] 1149184
```

```
avg_heart_rate_past_10 <- grouped_logs %>% # group by log interval
  summarize(average_heart_rate = mean(tail(na.omit(ts_heart_rate)), n = 10))

nrow(avg_heart_rate_past_10)
```

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

```
total_dist_past_10 <- grouped_logs %>% # group by log interval
  summarize(dist_walked = sum(tail(na.omit(ts_walking_distance)), n = 10))
  # take mean of last 10 non NaN walked distance measures
nrow(total_dist_past_10)
```

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

```
activity_data <- left_join(avg_heart_rate_past_10,
                           total_dist_past_10,
                           by = "log_row_index",
                           keep = FALSE)
head(activity_data)
```

log_row_index <chr>	average_heart_rate <dbl>	dist_walked <dbl>
100059	70.0	45.27379
1000732	NaN	10.00000
1000734	92.5	10.00000

log_row_index <chr>	average_heart_rate <dbl>	dist_walked <dbl>
1000737	NaN	10.00000
1000738	NaN	10.00000
1000739	NaN	10.00000

6 rows

```
activity_data_full <- survey_data[activity_data$log_row_index,]
activity_data_full$log_row_index <- activity_data$log_row_index
activity_data_full <- left_join(activity_data, activity_data_full, by = "log_row_index")
head(activity_data_full)
```

log_row_index <chr>	average_heart_rate <dbl>	dist_walked <dbl>	time <chr>
100059	70.0	45.27379	2022-11-09 12:38:16.448000+0800
1000732	NaN	10.00000	2023-05-05 09:00:22.264000+0800
1000734	92.5	10.00000	2023-05-05 09:00:24.703000+0800
1000737	NaN	10.00000	2023-05-05 09:00:28.513000+0800
1000738	NaN	10.00000	2023-05-05 09:00:35.877000+0800
1000739	NaN	10.00000	2023-05-05 09:00:37.219000+0800

6 rows | 1-5 of 63 columns

```
humidity_stations_ids <- colnames(weather_humidity_data_clean)[colnames(weather_humidity_d
ata_clean) %in% weather_stations_data$id]
humidity_stations <- weather_stations_data[weather_stations_data$id %in% humidity_stations
_ids, ]

d <- pointDistance(activity_data_full[,c("ws_longitude", "ws_latitude")],
                   humidity_stations[,c("longitude", "latitude")],
                   lonlat=TRUE, allpairs=T)
i <- apply(d, 1, which.min)

activity_data_full$humidity_ID = humidity_stations$id[i]


rainfall_stations_ids <- colnames(weather_rainfall_data_clean)[colnames(weather_rainfall_d
ata_clean) %in% weather_stations_data$id]
rainfall_stations <- weather_stations_data[weather_stations_data$id %in% rainfall_stations
_ids, ]

d <- pointDistance(activity_data_full[,c("ws_longitude", "ws_latitude")],
                   rainfall_stations[,c("longitude", "latitude")],
                   lonlat=TRUE, allpairs=T)
i <- apply(d, 1, which.min)

activity_data_full$rainfall_ID = rainfall_stations$id[i]


temperature_stations_ids <- colnames(weather_temperature_data_clean)[colnames(weather_temp
erature_data_clean) %in% weather_stations_data$id]
temperature_stations <- weather_stations_data[weather_stations_data$id %in% temperature_st
ations_ids, ]

d <- pointDistance(activity_data_full[,c("ws_longitude", "ws_latitude")],
                   temperature_stations[,c("longitude", "latitude")],
                   lonlat=TRUE, allpairs=T)
i <- apply(d, 1, which.min)

activity_data_full$temperature_ID = temperature_stations$id[i]


wind_speed_stations_ids <- colnames(weather_wind_speed_data_clean)[colnames(weather_wind_s
peed_data_clean) %in% weather_stations_data$id]
wind_speed_stations <- weather_stations_data[weather_stations_data$id %in% wind_speed_stat
ions_ids, ]

d <- pointDistance(activity_data_full[,c("ws_longitude", "ws_latitude")],
                   wind_speed_stations[,c("longitude", "latitude")],
                   lonlat=TRUE, allpairs=T)
i <- apply(d, 1, which.min)

activity_data_full$wind_speed_ID = wind_speed_stations$id[i]


wind_direction_stations_ids <- colnames(weather_wind_direction_data_clean)[colnames(weathe
r_wind_direction_data_clean) %in% weather_stations_data$id]
```

```
wind_direction_stations <- weather_stations_data[weather_stations_data$id %in% wind_direct
ion_stations_ids, ]

d <- pointDistance(activity_data_full[,c("ws_longitude", "ws_latitude")],
                   wind_direction_stations[,c("longitude", "latitude")],
                   lonlat=TRUE, allpairs=T)
i <- apply(d, 1, which.min)

activity_data_full$wind_direction_ID = wind_direction_stations$id[i]

head(activity_data_full)
```

log_row_index <chr>	average_heart_rate <dbl>	dist_walked <dbl>	time <chr>
100059	70.0	45.27379	2022-11-09 12:38:16.448000+0800
1000732	NaN	10.00000	2023-05-05 09:00:22.264000+0800
1000734	92.5	10.00000	2023-05-05 09:00:24.703000+0800
1000737	NaN	10.00000	2023-05-05 09:00:28.513000+0800
1000738	NaN	10.00000	2023-05-05 09:00:35.877000+0800
1000739	NaN	10.00000	2023-05-05 09:00:37.219000+0800

6 rows | 1-5 of 68 columns

```
str(weather_humidity_data_clean[,c(humidity_stations_ids,'date_time')])
```

```
## 'data.frame':    369750 obs. of  17 variables:
##  $ S24      : num  78.2 77.7 76.6 76.3 75.9 75.4 75.6 75.1 75.7 77.1 ...
##  $ S43      : num  82.2 81.2 82.5 82.6 82.8 ...
##  $ S44      : num  81.4 81.4 81.6 82.1 82.6 82.5 82.5 82.4 82.4 82.3 ...
##  $ S50      : num  83.3 83.4 83.4 83.3 83.2 83 83 83.2 83.3 83.7 ...
##  $ S60      : num  67.1 67.3 67.8 68.4 67.9 67.7 67.6 67.3 66.7 66.5 ...
##  $ S100     : num  84.8 84.9 85 85.1 85.1 85.2 85.4 85.4 85.4 85.5 ...
##  $ S102     : num  81.3 81.2 81.3 81.2 81.2 ...
##  $ S104     : num  84.5 84.5 84.5 84.4 84.3 84.6 84.6 84.5 84.4 84.2 ...
##  $ S106     : num  89.5 89.4 89.3 89.3 89.2 89.1 89.2 89.3 89.2 89.2 ...
##  $ S107     : num  75.1 75.8 76.4 75.8 76.3 75.5 75.6 76.4 76.6 78 ...
##  $ S108     : num  87.2 87 87.2 87.6 87.9 88.2 88.5 88.7 88.9 89.1 ...
##  $ S109     : num  82.5 82.6 82.5 82.3 82.1 81.9 81.6 81.5 81.5 81.5 ...
##  $ S111     : num  78.9 78.8 78.3 78.3 78.5 78.3 78.3 77.8 77.9 77.9 ...
##  $ S115     : num  71 70.9 71.5 72 71.6 71.7 72 72.1 72.4 72.5 ...
##  $ S116     : num  87.1 87.2 86.5 85.5 85.2 85.4 85.8 87 87.8 88.4 ...
##  $ S121     : num  86.2 86 86 85.7 85.8 85.6 85.4 85.6 86 86.1 ...
##  $ date_time: POSIXct, format: "2022-10-10 00:01:00" "2022-10-10 00:02:00" ...
```

```
melted_humidity_data <- reshape2::melt(weather_humidity_data_clean[,c(humidity_stations_ids,'date_time')], id='date_time')
colnames(melted_humidity_data)[colnames(melted_humidity_data) == 'variable'] <- 'humidity_ID'
colnames(melted_humidity_data)[colnames(melted_humidity_data) == 'value'] <- 'humidity'
setDT(melted_humidity_data)

melted_rainfall_data <- reshape2::melt(weather_rainfall_data_clean[,c(rainfall_stations_ids,'date_time')], id='date_time')
colnames(melted_rainfall_data)[colnames(melted_rainfall_data) == 'variable'] <- 'rainfall_ID'
colnames(melted_rainfall_data)[colnames(melted_rainfall_data) == 'value'] <- 'rainfall'
setDT(melted_rainfall_data)

melted_temperature_data <- reshape2::melt(weather_temperature_data_clean[,c(temperature_stations_ids,'date_time')], id='date_time')
colnames(melted_temperature_data)[colnames(melted_temperature_data) == 'variable'] <- 'temperature_ID'
colnames(melted_temperature_data)[colnames(melted_temperature_data) == 'value'] <- 'temperature'
setDT(melted_temperature_data)

melted_wind_speed_data <- reshape2::melt(weather_wind_speed_data_clean[,c(wind_speed_stations_ids,'date_time')], id='date_time')
colnames(melted_wind_speed_data)[colnames(melted_wind_speed_data) == 'variable'] <- 'wind_speed_ID'
colnames(melted_wind_speed_data)[colnames(melted_wind_speed_data) == 'value'] <- 'wind_speed'
setDT(melted_wind_speed_data)

melted_wind_direction_data <- reshape2::melt(weather_wind_direction_data_clean[,c(wind_direction_stations_ids,'date_time')], id='date_time')
colnames(melted_wind_direction_data)[colnames(melted_wind_direction_data) == 'variable'] <- 'wind_direction_ID'
colnames(melted_wind_direction_data)[colnames(melted_wind_direction_data) == 'value'] <- 'wind_direction'
setDT(melted_wind_direction_data)
```

```
setDT(activity_data_full)

activity_data_full <- activity_data_full[, c("humidityTime", "humidity") :=
    melted_humidity_data[activity_data_full, on = c("humidity_ID", "date_time"), roll = In
f, .(x.date_time, x.humidity)]][]

activity_data_full <- activity_data_full[, c("rainfallTime", "rainfall") :=
    melted_rainfall_data[activity_data_full, on = c("rainfall_ID", "date_time"), roll = In
f, .(x.date_time, x.rainfall)]][]

activity_data_full <- activity_data_full[, c("temperatureTime", "temperature") :=
    melted_temperature_data[activity_data_full, on = c("temperature_ID", "date_time"), rol
l = Inf, .(x.date_time, x.temperature)]][]

activity_data_full <- activity_data_full[, c("wind_speedTime", "wind_speed") :=
    melted_wind_speed_data[activity_data_full, on = c("wind_speed_ID", "date_time"), roll
= Inf, .(x.date_time, x.wind_speed)]][]

activity_data_full <- activity_data_full[, c("wind_directionTime", "wind_direction") :=
    melted_wind_direction_data[activity_data_full, on = c("wind_direction_ID", "date_tim
e"), roll = Inf, .(x.date_time, x.wind_direction)]][]

head(activity_data_full, 100)
```

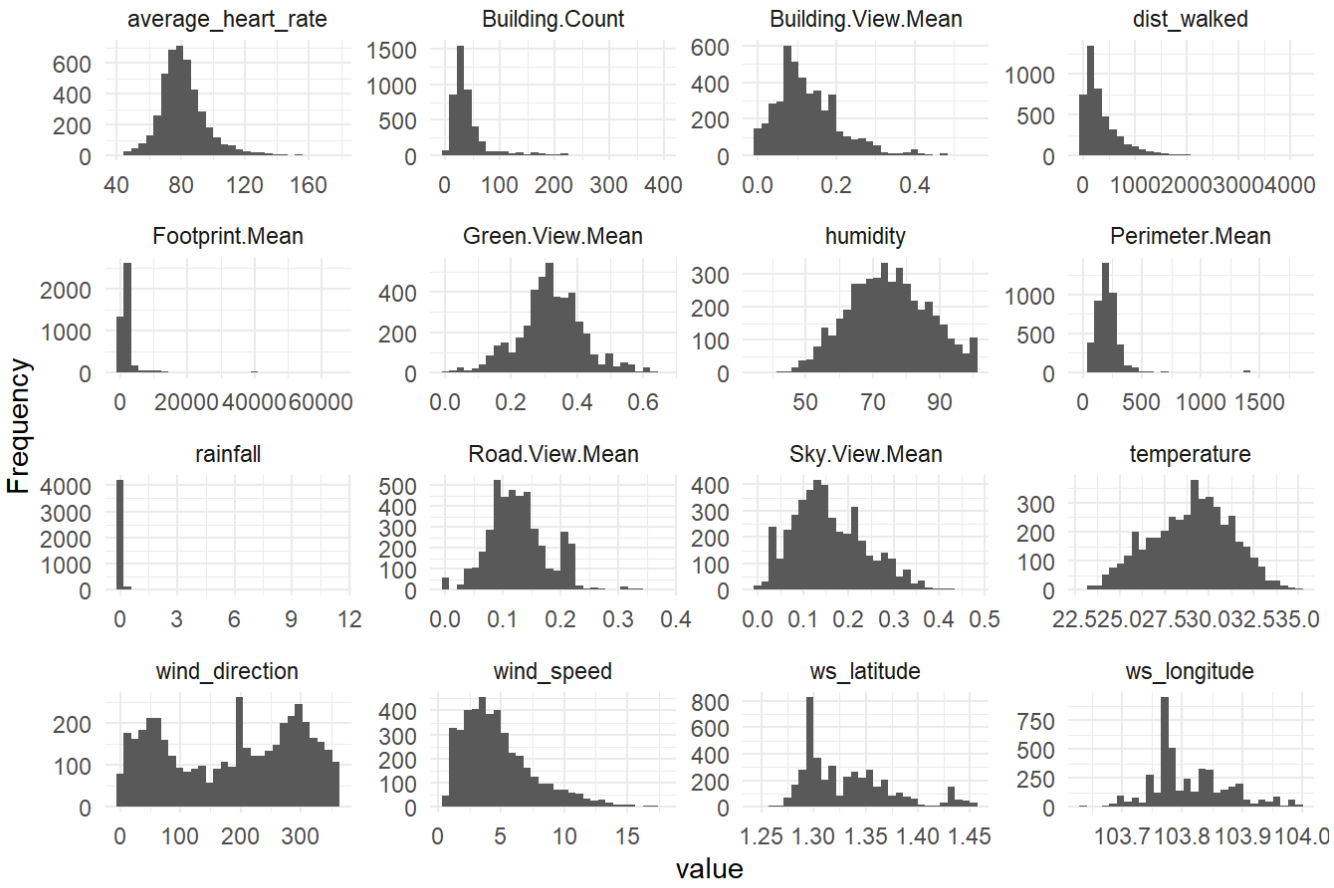
log_row_index <chr>	average_heart_rate <dbl>	dist_walked <dbl>	time <chr>
100059	70.00000	45.27379	2022-11-09 12:38:16.448000+0800
1000732	NaN	10.00000	2023-05-05 09:00:22.264000+0800
1000734	92.50000	10.00000	2023-05-05 09:00:24.703000+0800
1000737	NaN	10.00000	2023-05-05 09:00:28.513000+0800
1000738	NaN	10.00000	2023-05-05 09:00:35.877000+0800
1000739	NaN	10.00000	2023-05-05 09:00:37.219000+0800
1000741	110.00000	39.72900	2023-05-05 09:00:38.665000+0800
1000746	87.66667	184.40908	2023-05-05 09:00:48.304000+0800
1000809	99.83333	232.78524	2023-05-05 09:05:34.242000+0800
1001113	81.33333	173.50558	2023-05-05 09:41:53.011000+0800

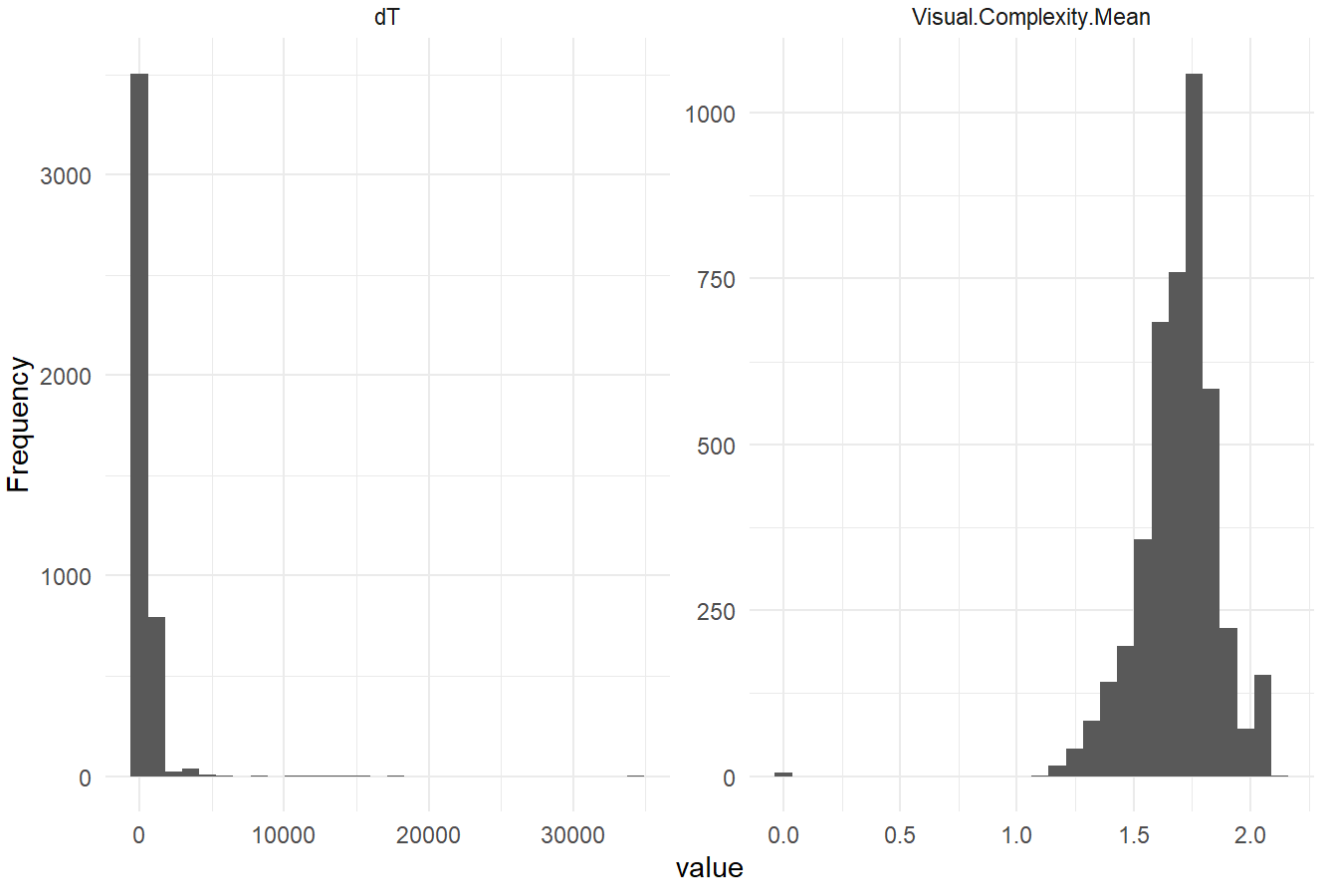
```
selected_data <- activity_data_full[,c(
  'id_participant',
  'ws_longitude',
  'ws_latitude',
  'dist_walked',
  'average_heart_rate',
  'q_location',
  'Green.View.Mean',
  'Footprint.Mean',
  'Perimeter.Mean',
  'Building.Count',
  'Sky.View.Mean',
  'Building.View.Mean',
  'Road.View.Mean',
  'humidity',
  'rainfall',
  'temperature',
  'wind_speed',
  'wind_direction',
  'q_thermal_preference',
  'date_time',
  'dT',
  'Visual.Complexity.Mean'
)]

selected_data <- drop_na(selected_data)


selected_data$q_location <- as.factor(selected_data$q_location)
selected_data$q_thermal_preference <-
  as.factor(selected_data$q_thermal_preference)
```

```
summary(selected_data)
```

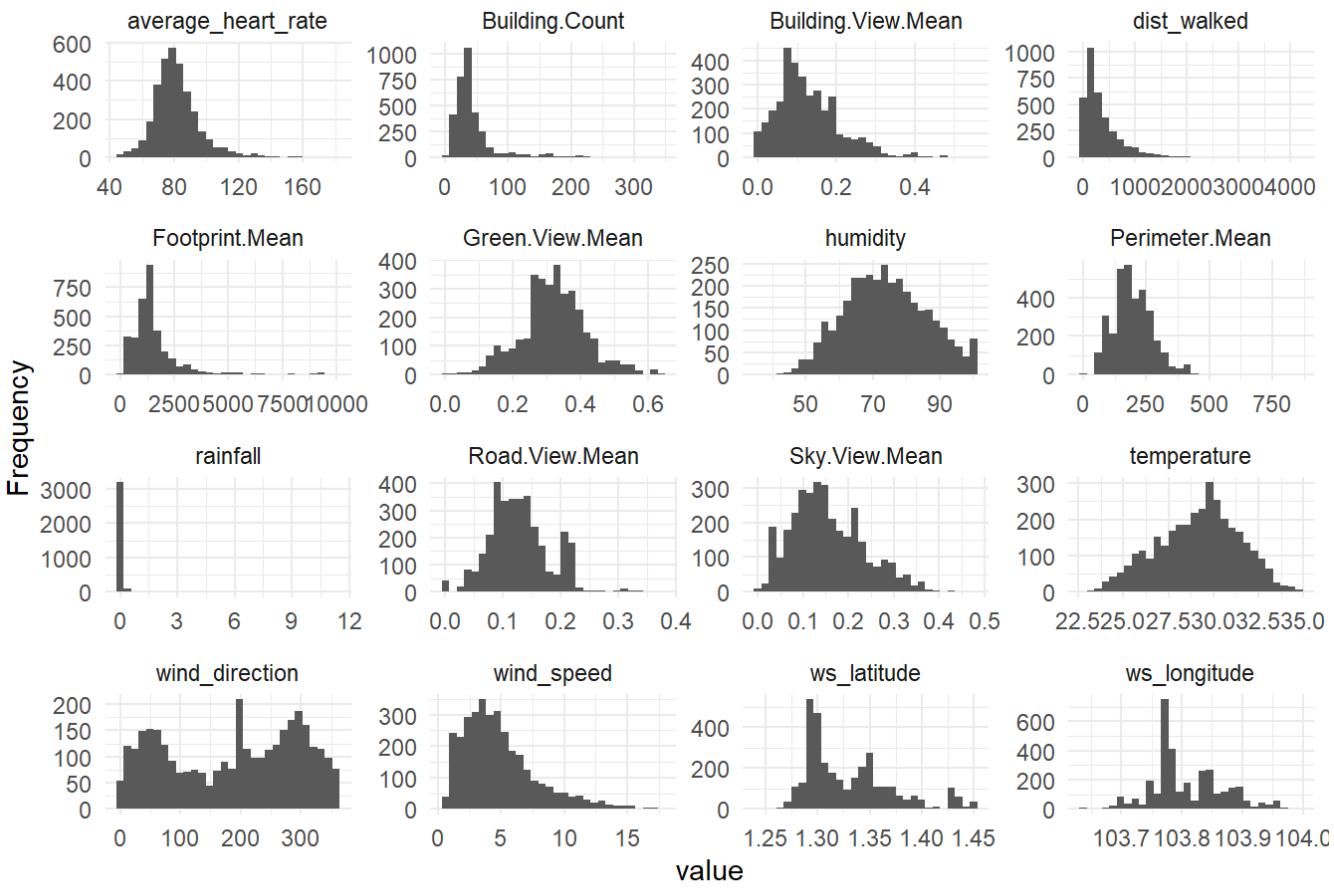
```
##  id_participant      ws_longitude    ws_latitude     dist_walked
##  Length:4379        Min.   :103.6   Min.   :1.246   Min.   :  10.0
##  Class :character   1st Qu.:103.8   1st Qu.:1.297   1st Qu.: 107.7
##  Mode  :character   Median :103.8   Median :1.319   Median : 227.9
##                     Mean   :103.8   Mean   :1.332   Mean   : 342.6
##                     3rd Qu.:103.8   3rd Qu.:1.355   3rd Qu.: 472.1
##                     Max.   :104.0   Max.   :1.454   Max.   :4173.2
##  average_heart_rate           q_location   Green.View.Mean  Footprint.Mean
##  Min.   : 43.50     Indoor - Class : 267   Min.   :0.0000   Min.   :    0
##  1st Qu.: 72.17     Indoor - Home  :2039   1st Qu.:0.2670   1st Qu.: 1045
##  Median : 79.67     Indoor - Office: 704   Median :0.3225   Median : 1357
##  Mean   : 81.37     Indoor - Other : 581   Mean   :0.3204   Mean   : 2367
##  3rd Qu.: 88.00     Outdoor        : 517   3rd Qu.:0.3836   3rd Qu.: 1834
##  Max.   :178.50     Transportation : 271   Max.   :0.6560   Max.   :64330
##  Perimeter.Mean   Building.Count   Sky.View.Mean    Building.View.Mean
##  Min.   :   0.0   Min.   :  0.00   Min.   :0.0000   Min.   :0.00000
##  1st Qu.: 150.9   1st Qu.: 22.50   1st Qu.:0.1000   1st Qu.:0.07229
##  Median : 191.7   Median : 32.00   Median :0.1453   Median :0.10687
##  Mean   : 219.7   Mean   : 41.91   Mean   :0.1577   Mean   :0.12802
##  3rd Qu.: 251.7   3rd Qu.: 44.00   3rd Qu.:0.2113   3rd Qu.:0.17388
##  Max.   :1834.8   Max.   :395.00   Max.   :0.4772   Max.   :0.55050
##  Road.View.Mean       humidity        rainfall         temperature
##  Min.   :0.00000   Min.   :34.90   Min.   : 0.00000   Min.   :22.9
##  1st Qu.:0.09087   1st Qu.:66.10   1st Qu.: 0.00000   1st Qu.:27.4
##  Median :0.12242   Median :74.40   Median : 0.00000   Median :29.2
##  Mean   :0.12753   Mean   :74.76   Mean   : 0.04111   Mean   :29.0
##  3rd Qu.:0.15535   3rd Qu.:83.30   3rd Qu.: 0.00000   3rd Qu.:30.6
##  Max.   :0.37500   Max.   :99.50   Max.   :11.31200   Max.   :35.0
##    wind_speed     wind_direction q_thermal_preference
##  Min.   : 0.300   Min.   :  1    Cooler   :1736
##  1st Qu.: 2.600   1st Qu.: 73    No change:2377
##  Median : 4.200   Median :196    Warmer   : 266
##  Mean   : 4.795   Mean   :185
##  3rd Qu.: 6.200   3rd Qu.:286
##  Max.   :17.400   Max.   :359
##    date_time                          dT           Visual.Complexity.Mean
##  Min.   :2022-10-10 13:13:19.00   Min.   :   55.00   Min.   :0.000
##  1st Qu.:2022-11-16 15:07:59.00   1st Qu.:   79.59   1st Qu.:1.601
##  Median :2023-03-29 17:45:24.00   Median :  121.10   Median :1.712
##  Mean   :2023-02-08 08:38:30.94   Mean   :  368.76   Mean   :1.694
##  3rd Qu.:2023-04-24 18:08:26.50   3rd Qu.:  275.25   3rd Qu.:1.793
##  Max.   :2023-05-30 17:00:31.00   Max.   :34368.42   Max.   :2.126
```

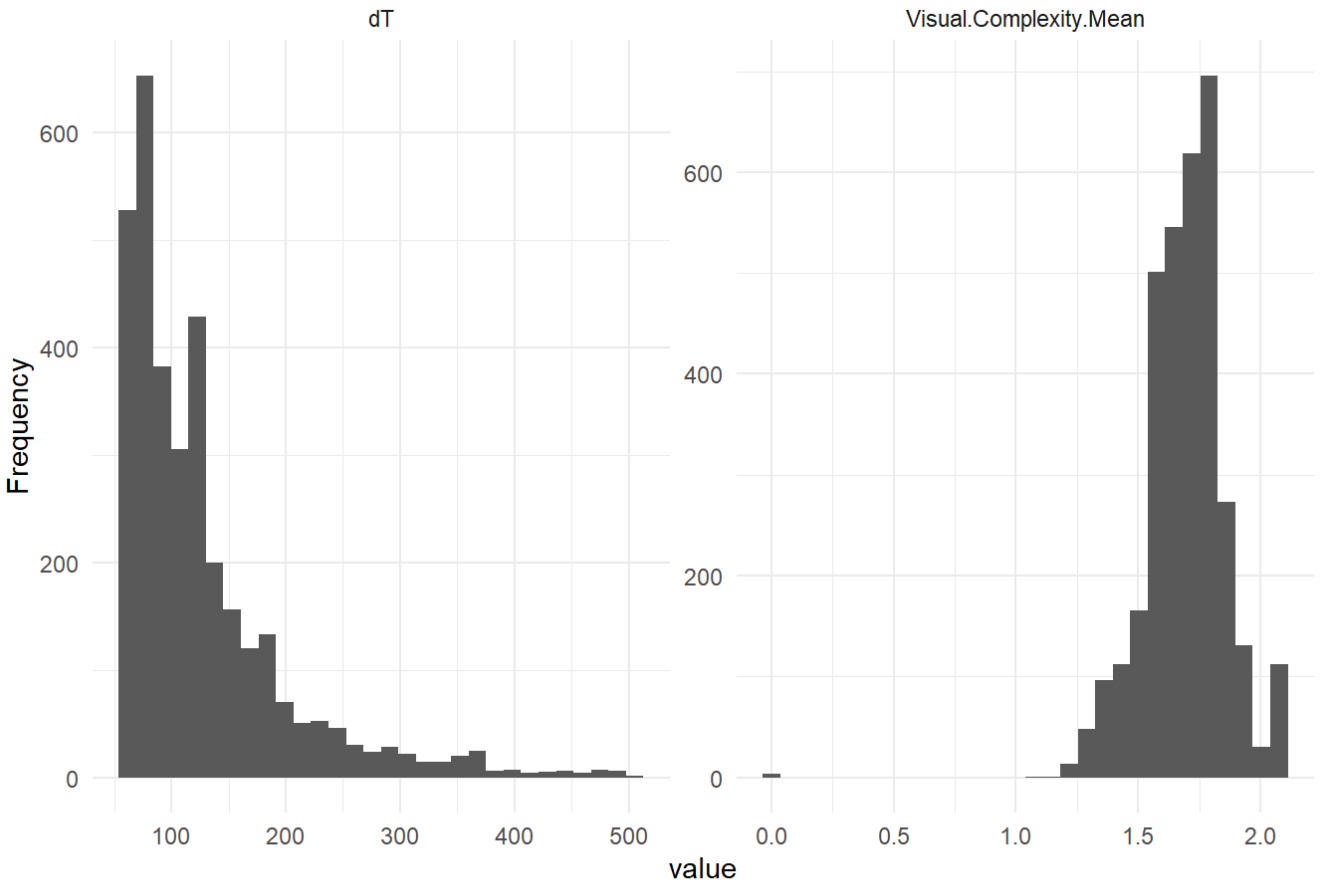
```
selected_data %>% plot_histogram(ggtheme = theme_minimal())
```


average_heart_rate
Building.Count
Building.View.Mean
dist_walked
Footprint.Mean
Green.View.Mean
humidity
Perimeter.Mean
rainfall
Road.View.Mean
Sky.View.Mean
temperature
wind_direction
wind_speed
ws_latitude
ws_longitude
Frequency
value

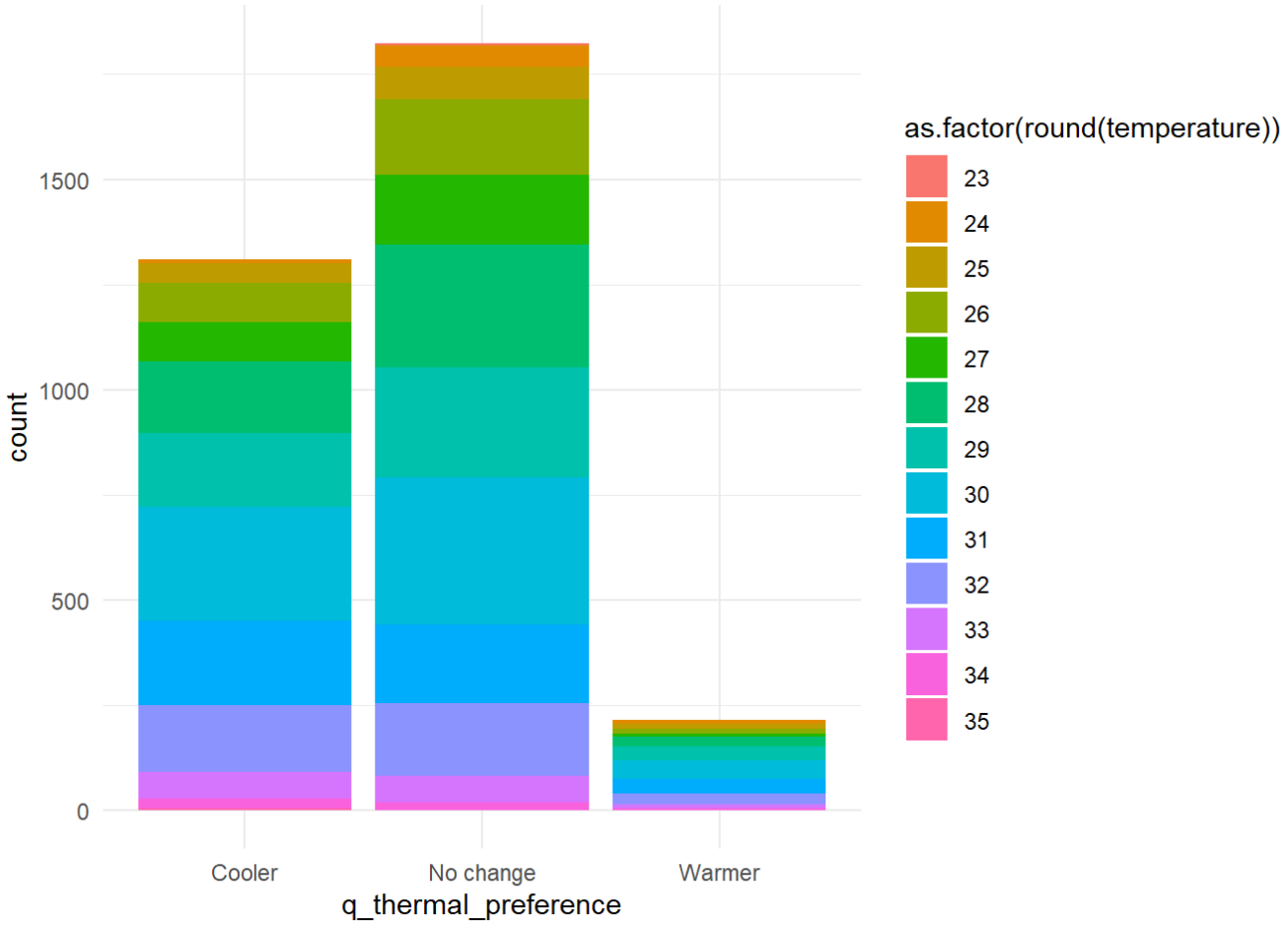

dT
Visual.Complexity.Mean
Frequency
value

```
selected_data_no_outliers <- selected_data[(
  selected_data$dT < 500 & # remove measurements that are infrequent
  selected_data$Footprint.Mean < 10000
   ),]


#selected_data_no_outliers <- selected_data_no_outliers[selected_data_no_outliers$q_locati
on == "Outdoor", ]
selected_data_no_outliers %>% plot_histogram(ggtheme = theme_minimal())
```





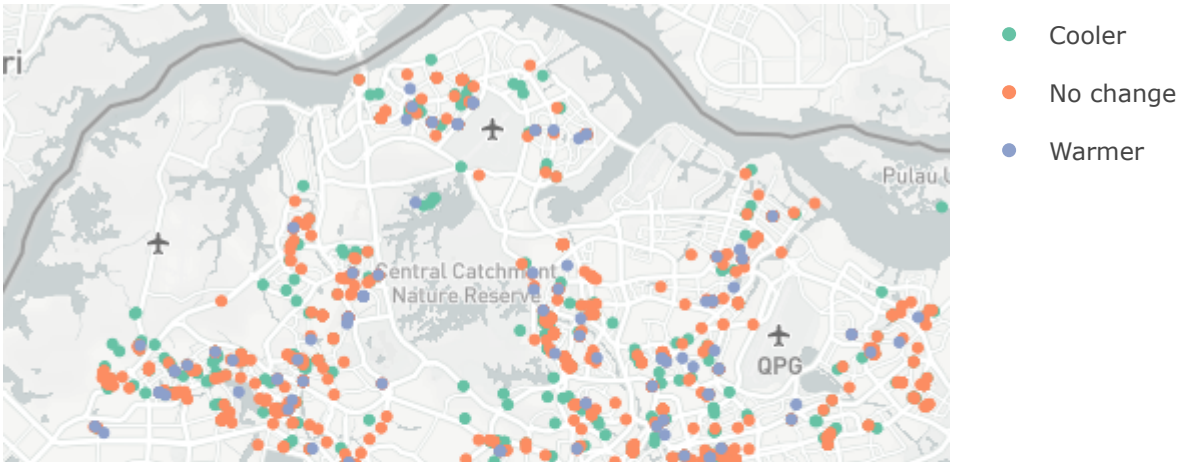

```
ggplot(data=selected_data_no_outliers,aes(q_thermal_preference)) +
  geom_bar(aes(fill=as.factor(round(temperature)))) + theme_minimal() +
  scale_color_gradient2(low = "blue", mid = "white", high = "red", space = "Lab" )
```

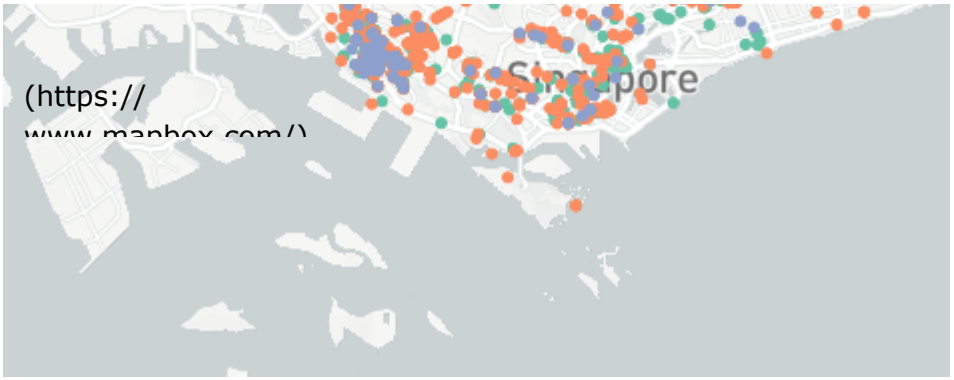



```
mapboxToken <- paste(readLines("mapbox_token.txt"), collapse="")
```

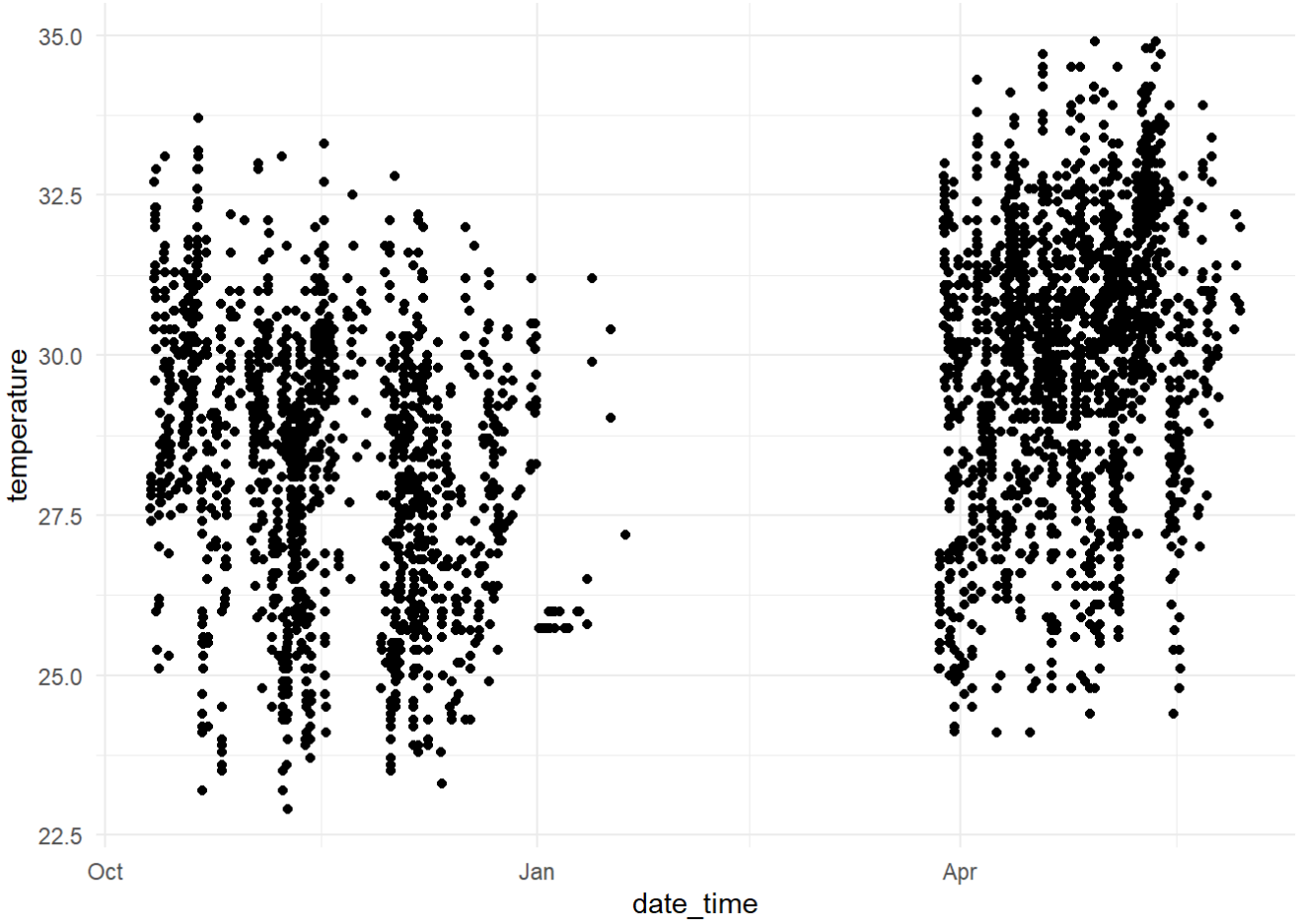
```
## Warning in readLines("mapbox_token.txt"): incomplete final line found on
## 'mapbox_token.txt'
```

```r
# creating a sample data.frame with your lat/lon points
lon <- selected_data_no_outliers$ws_longitude
lat <- selected_data_no_outliers$ws_latitude
thermal_preference <- selected_data_no_outliers$q_thermal_preference

df <- as.data.frame(cbind(lon,lat))

df <- df %>%
  arrange(thermal_preference) %>%
  mutate(thermal_preference = as.factor(thermal_preference),
         color = recode(thermal_preference,'Cooler' = "#fe4a49",
                        "No change" = "#fed766", "Warmer" = "#009fb7"))

fig <- df

fig <- fig %>%
  plot_ly(
    lat = ~lat,
    lon = ~lon,
    type = 'scattermapbox',
    mode = "markers",
    color  = ~thermal_preference,
    legendgroup  = ~thermal_preference,
    marker = list(size=7))


fig <- fig %>%
  layout(
    mapbox = list(
      style = 'light',
      zoom =10,
      center = list(lon = mean(df$lon), lat = mean(df$lat))))

fig <- fig %>%

  config(mapboxAccessToken = mapboxToken)

pb <- plotly_build(fig)

pb
```

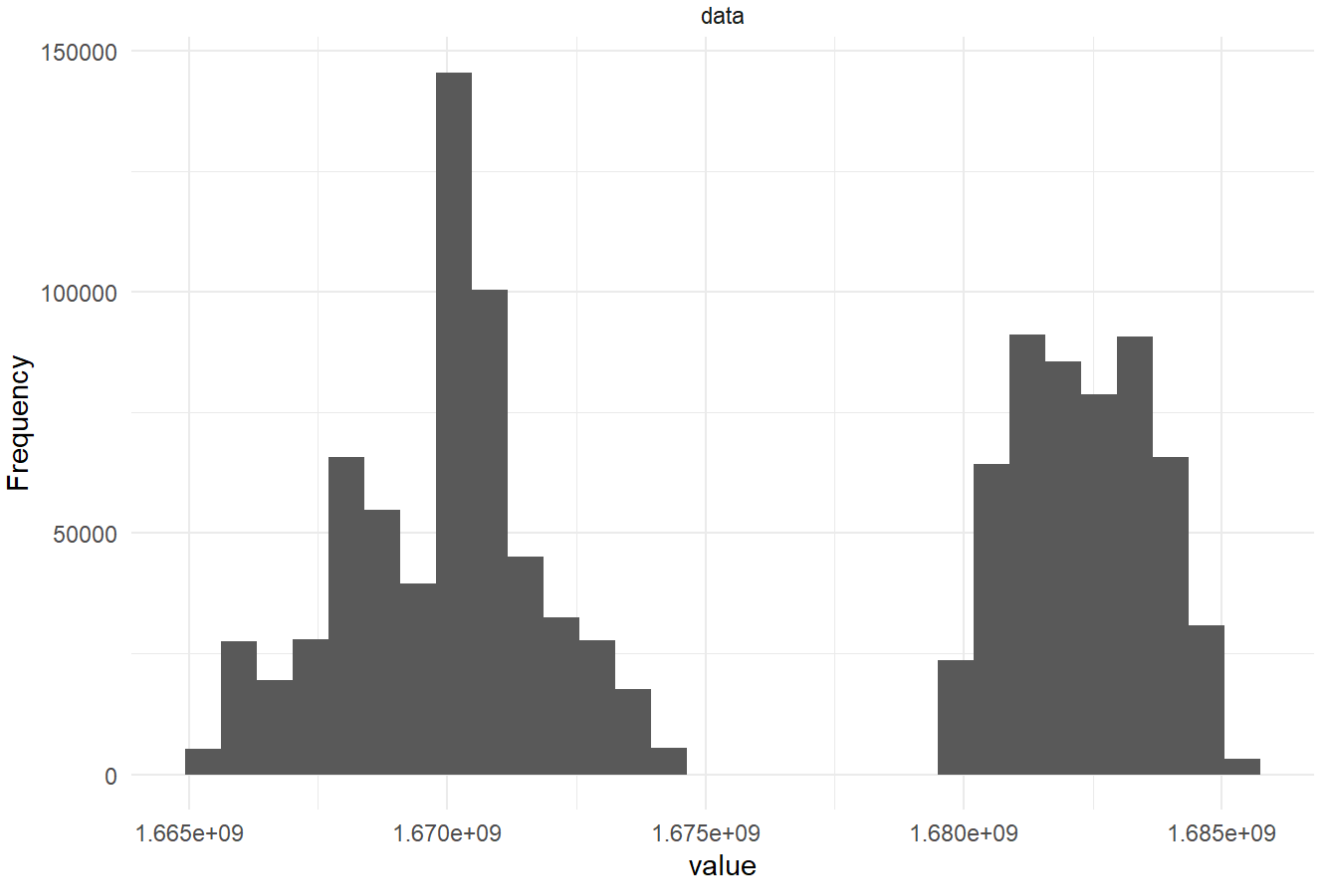




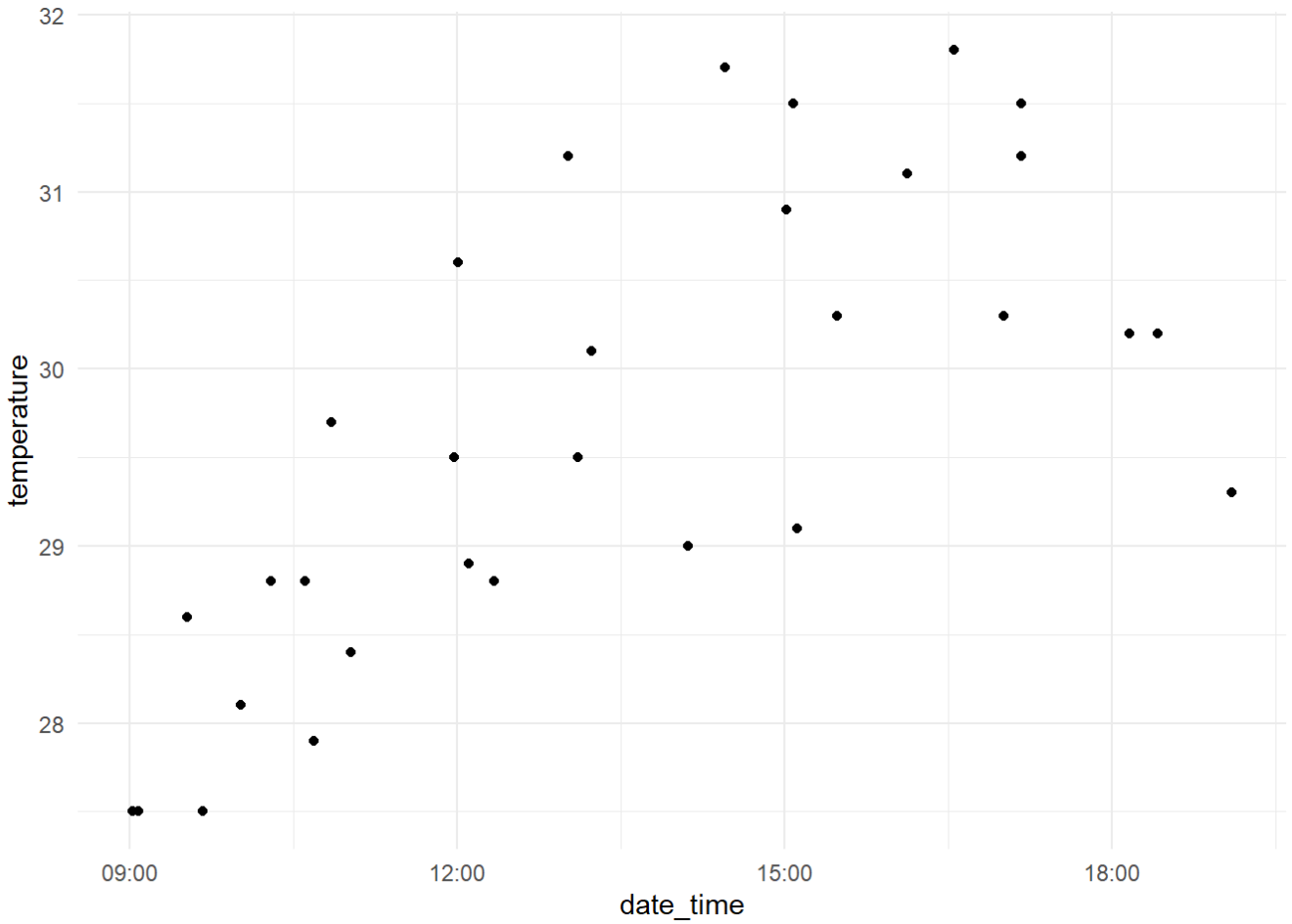

```
ggplot(selected_data_no_outliers, aes(date_time, temperature)) + geom_point() + theme_mini
mal()
```




```
as.numeric(survey_data$date_time) %>%
  plot_histogram(ggtheme = theme_minimal())
```

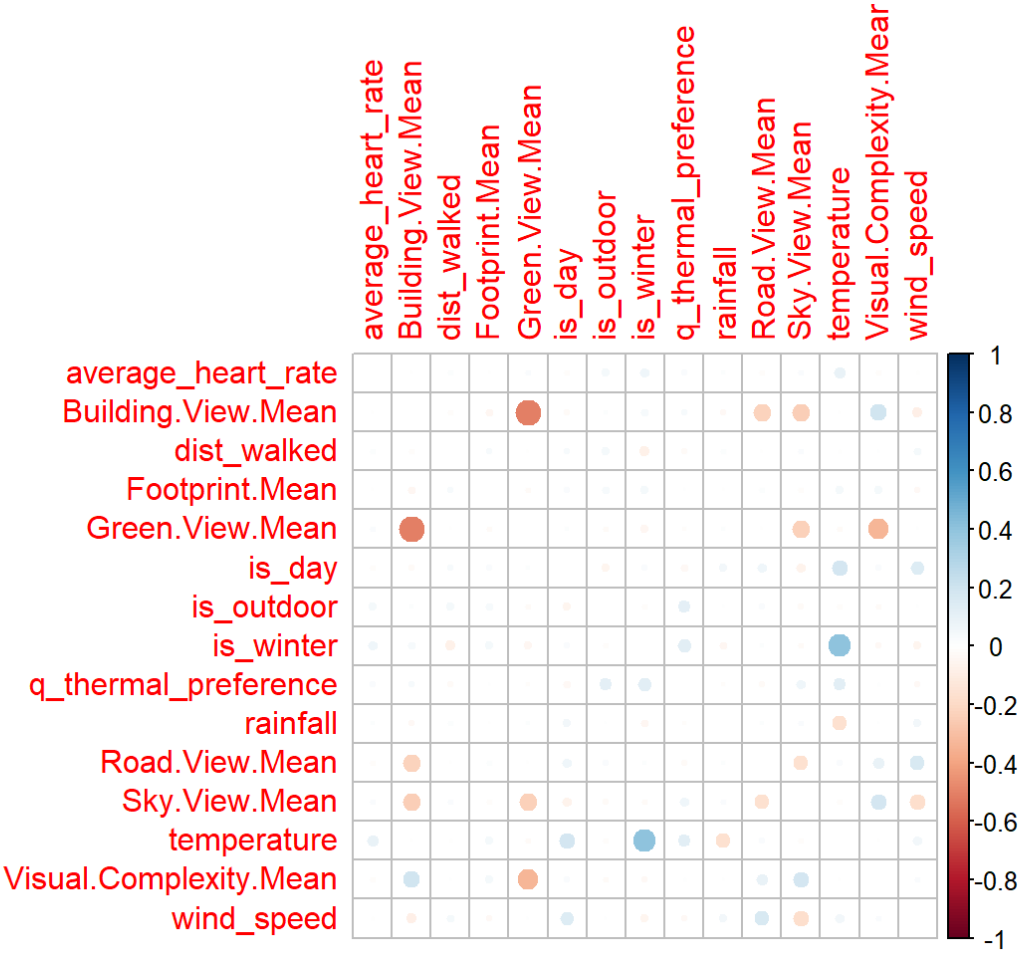



```
# Temperature increases during the day
ggplot(selected_data[ selected_data$date_time > ymd("2022/10/18") &
                      selected_data$date_time < ymd("2022/10/19") #&
                     ,], aes(date_time, temperature)) + geom_point() + theme_minimal()
```



==========Model Testing Starts Here==========

```
selected_data_log <-
  subset(
    selected_data_no_outliers,
    select = -c(
      id_participant,
      ws_longitude,
      ws_latitude,
      Perimeter.Mean,
      humidity,
      wind_direction,
      Building.Count,
      dT
    )
  )


selected_data_log$q_thermal_preference <-
  as.factor(selected_data_log$q_thermal_preference == "Cooler")

selected_data_log$is_outdoor <-
  as.factor(selected_data_log$q_location == "Outdoor")

#
selected_data_log$is_winter <- as.factor(selected_data_log$date_time > ym("2023/04"))
selected_data_log$is_day <- as.factor((hour(selected_data_log$date_time) > 12 &
                                         hour(selected_data_log$date_time) < 18) == T)

selected_data_log <-
  subset(selected_data_log, select = -c(q_location, date_time))

set.seed(1)

# Divide the data into 80% training and 20% testing
train <-
  sample(1:nrow(selected_data_log),
         size = round(nrow(selected_data_log) * 0.8),
         replace = FALSE)

selected_data_log_train <- selected_data_log[train, ]
selected_data_log_test <- selected_data_log[-train, ]

selected_data_log %>%
  mutate(is_outdoor = as.numeric(is_outdoor)) %>%
  mutate(q_thermal_preference = as.numeric(q_thermal_preference)) %>%
  mutate(is_winter = as.numeric(is_winter)) %>%
  mutate(is_day = as.numeric(is_day)) %>%
  cor(use = "pairwise.complete.obs") %>%
  corrplot(order = 'alphabet', diag = F)
```




```
str(selected_data_log_train)
```

```
## Classes 'data.table' and 'data.frame':   2678 obs. of  15 variables:
##  $ dist_walked           : num  405.4 28.8 237 89.1 980.4 ...
##  $ average_heart_rate    : num  75 75.3 69 79.7 90 ...
##  $ Green.View.Mean       : num  0.265 0.405 0.367 0.259 0.276 ...
##  $ Footprint.Mean        : num  1255 1435 1273 1802 1079 ...
##  $ Sky.View.Mean         : num  0.106 0.132 0.152 0.191 0.288 ...
##  $ Building.View.Mean    : num  0.2684 0.0782 0.032 0.1297 0.0589 ...
##  $ Road.View.Mean        : num  0.1155 0.2062 0.1972 0.2013 0.0868 ...
##  $ rainfall              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ temperature           : num  25.5 29.7 26.4 30.7 29.3 27.9 32.5 30.9 26.4 26 ...
##  $ wind_speed            : num  1.6 5.7 2.5 4.4 7.2 2 4.2 4.7 4.4 2.2 ...
##  $ q_thermal_preference  : Factor w/ 2 levels "FALSE","TRUE": 1 2 2 1 1 2 1 1 2 1 ...
##  $ Visual.Complexity.Mean: num  1.78 1.78 1.38 1.86 1.81 ...
##  $ is_outdoor            : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 2 2 1 1 ...
##  $ is_winter             : Factor w/ 2 levels "FALSE","TRUE": 1 1 2 1 1 2 2 2 1 1 ...
##  $ is_day                : Factor w/ 2 levels "FALSE","TRUE": 1 2 1 2 1 1 2 2 1 1 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
model <- glm(q_thermal_preference ~ . , family = binomial(link = "logit"), data = selected
_data_log_train)
summary(model)
```

```
##
## Call:
## glm(formula = q_thermal_preference ~ ., family = binomial(link = "logit"),
##     data = selected_data_log_train)
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -4.941e+00  8.657e-01  -5.708 1.15e-08 ***
## dist_walked            -2.028e-04  1.211e-04  -1.674  0.09403 .
## average_heart_rate      2.274e-04  2.790e-03   0.081  0.93505
## Green.View.Mean         1.670e+00  5.934e-01   2.814  0.00489 **
## Footprint.Mean         -4.790e-06  3.066e-05  -0.156  0.87585
## Sky.View.Mean           2.894e+00  6.642e-01   4.357 1.32e-05 ***
## Building.View.Mean      2.149e+00  7.171e-01   2.997  0.00273 **
## Road.View.Mean          1.055e+00  8.797e-01   1.199  0.23052
## rainfall               -2.109e-02  1.612e-01  -0.131  0.89593
## temperature             1.037e-01  2.086e-02   4.971 6.65e-07 ***
## wind_speed              1.133e-02  1.454e-02   0.780  0.43557
## Visual.Complexity.Mean -6.921e-02  2.788e-01  -0.248  0.80392
## is_outdoorTRUE          7.059e-01  1.216e-01   5.807 6.37e-09 ***
## is_winterTRUE           3.599e-01  8.951e-02   4.021 5.80e-05 ***
## is_dayTRUE             -1.947e-01  8.453e-02  -2.303  0.02125 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3578.7  on 2677  degrees of freedom
## Residual deviance: 3452.2  on 2663  degrees of freedom
## AIC: 3482.2
##
## Number of Fisher Scoring iterations: 4
```

```
model2 <-
  glm(
    q_thermal_preference ~ is_winter + is_outdoor + is_day + temperature + Sky.View.Mean,
    family = binomial(link = "logit"),
    data = selected_data_log_train
  )
summary(model2)
```

```
##
## Call:
## glm(formula = q_thermal_preference ~ is_winter + is_outdoor +
##     is_day + temperature + Sky.View.Mean, family = binomial(link = "logit"),
##     data = selected_data_log_train)
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -3.88353    0.57885  -6.709 1.96e-11 ***
## is_winterTRUE   0.35545    0.08829   4.026 5.67e-05 ***
## is_outdoorTRUE  0.68305    0.12022   5.682 1.33e-08 ***
## is_dayTRUE     -0.19882    0.08324  -2.389  0.01691 *
## temperature     0.10306    0.02025   5.090 3.58e-07 ***
## Sky.View.Mean   1.59007    0.51023   3.116  0.00183 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3578.7  on 2677  degrees of freedom
## Residual deviance: 3466.4  on 2672  degrees of freedom
## AIC: 3478.4
##
## Number of Fisher Scoring iterations: 4
```

```
# Step 3: Predict probabilities
probabilities <- predict(model2, selected_data_log_test, type = "response")

# Step 4 and 5: Use different cutoffs and calculate accuracy
cutoffs <- c(0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)

accuracies <- sapply(cutoffs, function(cutoff) {
  # Convert probabilities to binary predictions
  predictions <- ifelse(probabilities > cutoff, 2, 1)

  # Calculate accuracy
  accuracy(as.numeric(selected_data_log_test$q_thermal_preference), predictions)
  #mean(predictions == as.numeric(selected_data_log_test$q_thermal_preference))
})

# Print the accuracies for each cutoff
names(accuracies) <- cutoffs
accuracies
```

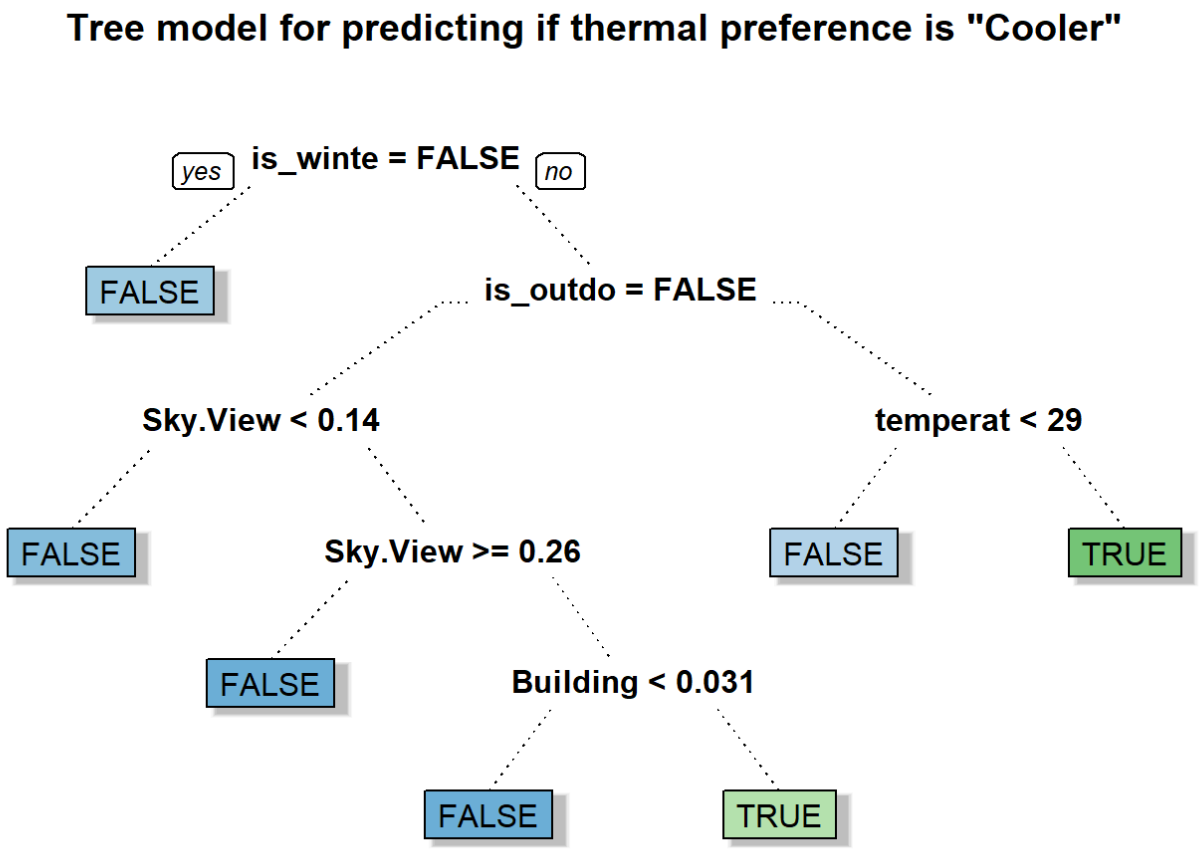
```
##       0.2       0.3       0.4       0.5       0.6       0.7       0.8       0.9
## 0.4059701 0.4701493 0.6029851 0.6238806 0.6208955 0.6014925 0.6000000 0.6000000
```

```
pR2(model2)['McFadden']
```

```
## fitting null model for pseudo-r2
```

```
##   McFadden
## 0.03139527
```

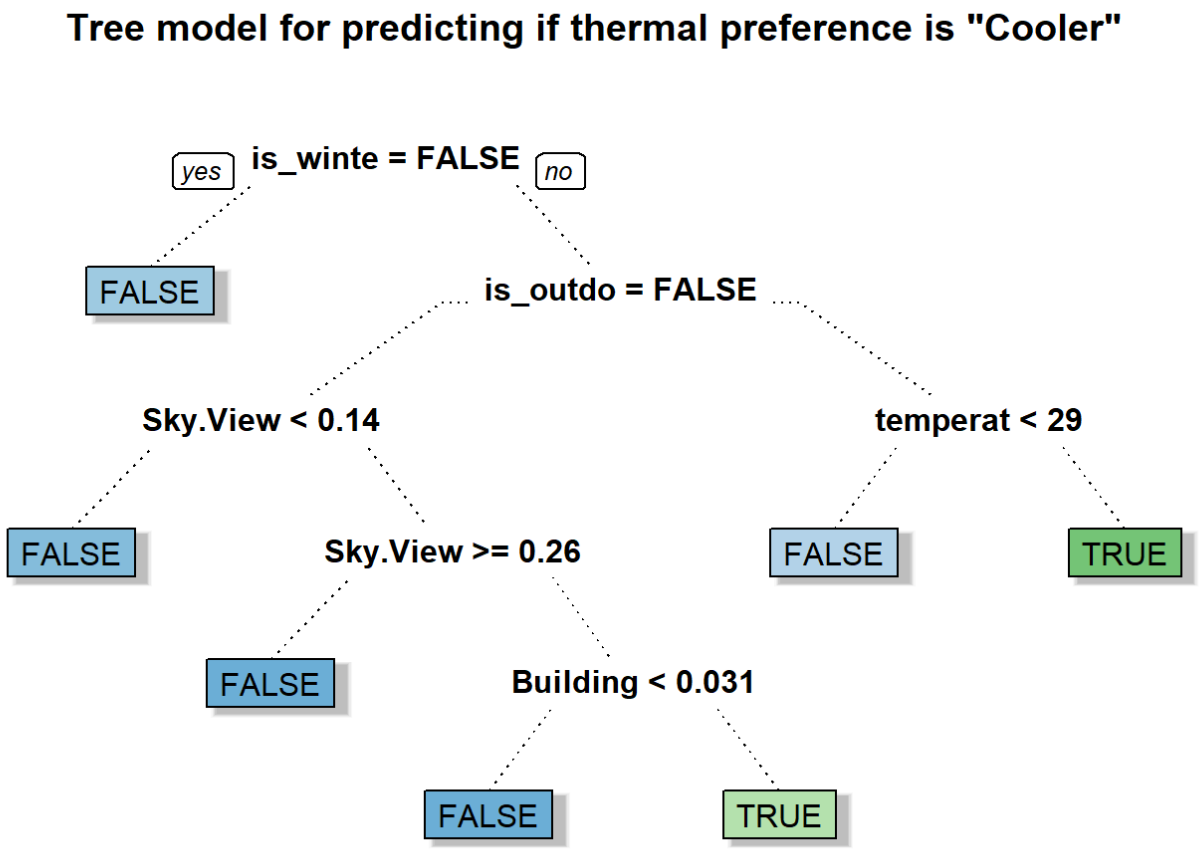
```
fit.tree = rpart(q_thermal_preference ~ ., data=selected_data_log_train, method="class", c
p=0.008)
prp(fit.tree,
    main = "Tree model for predicting if thermal preference is \"Cooler\"",
    box.palette = "auto",
    fallen.leaves = F,
    shadow.col = "gray",
    branch.lty = 3,
    branch = .5,
    faclen = 0,
    round = 0)
```




```
printcp(fit.tree)
```

```
##
## Classification tree:
## rpart(formula = q_thermal_preference ~ ., data = selected_data_log_train,
##     method = "class", cp = 0.008)
##
## Variables actually used in tree construction:
## [1] Building.View.Mean is_outdoor         is_winter          Sky.View.Mean
## [5] temperature
##
## Root node error: 1041/2678 = 0.38872
##
## n= 2678
##
##         CP nsplit rel error  xerror     xstd
## 1 0.023055      0   1.00000 1.00000 0.024232
## 2 0.010567      5   0.86647 0.94524 0.023966
## 3 0.008000      6   0.85591 0.93084 0.023888
```

```
bestcp <- fit.tree$cptable[which.min(fit.tree$cptable[,"xerror"]),"CP"]
pruned.tree <- prune(fit.tree, cp = bestcp)

prp(pruned.tree,
    main = "Tree model for predicting if thermal preference is \"Cooler\"",
    box.palette = "auto",
    fallen.leaves = F,
    shadow.col = "gray",
    branch.lty = 3,
    branch = .5,
    faclen = 0,
    round = 0)
```




```
predicted <- as.numeric(predict(pruned.tree, selected_data_log_test, type = "class"))
sum(predicted == as.numeric(selected_data_log_test$q_thermal_preference)) / nrow(selected_
data_log_test)
```

```
## [1] 0.6641791
```

```
Metrics::accuracy(as.numeric(selected_data_log_test$q_thermal_preference), predicted)
```

```
## [1] 0.6641791
```

```
SS_tot <- sum((as.numeric(selected_data_log_train$q_thermal_preference) - mean(as.numeric
(selected_data_log_train$q_thermal_preference))) ^ 2)
SS_res_tree <- sum((as.numeric(selected_data_log_train$q_thermal_preference) - as.numeric
(predict(pruned.tree, selected_data_log_train, type = "class"))) ^ 2)

R_sq_lm <- 1 - SS_res_tree / SS_tot
R_sq_lm
```

```
## [1] -0.4001961
```

```
set.seed(50)

model_forest <-
  randomForest(
    q_thermal_preference ~ . ,
    data = selected_data_log_train,
    importance = TRUE,
    ntree = 150
  )

predicted <- as.numeric(predict(model_forest, selected_data_log_test))
sum(predicted == as.numeric(selected_data_log_test$q_thermal_preference)) / nrow(selected_
data_log_test)
```

```
## [1] 0.7059701
```

```
Metrics::accuracy(as.numeric(selected_data_log_test$q_thermal_preference), predicted)
```

```
## [1] 0.7059701
```

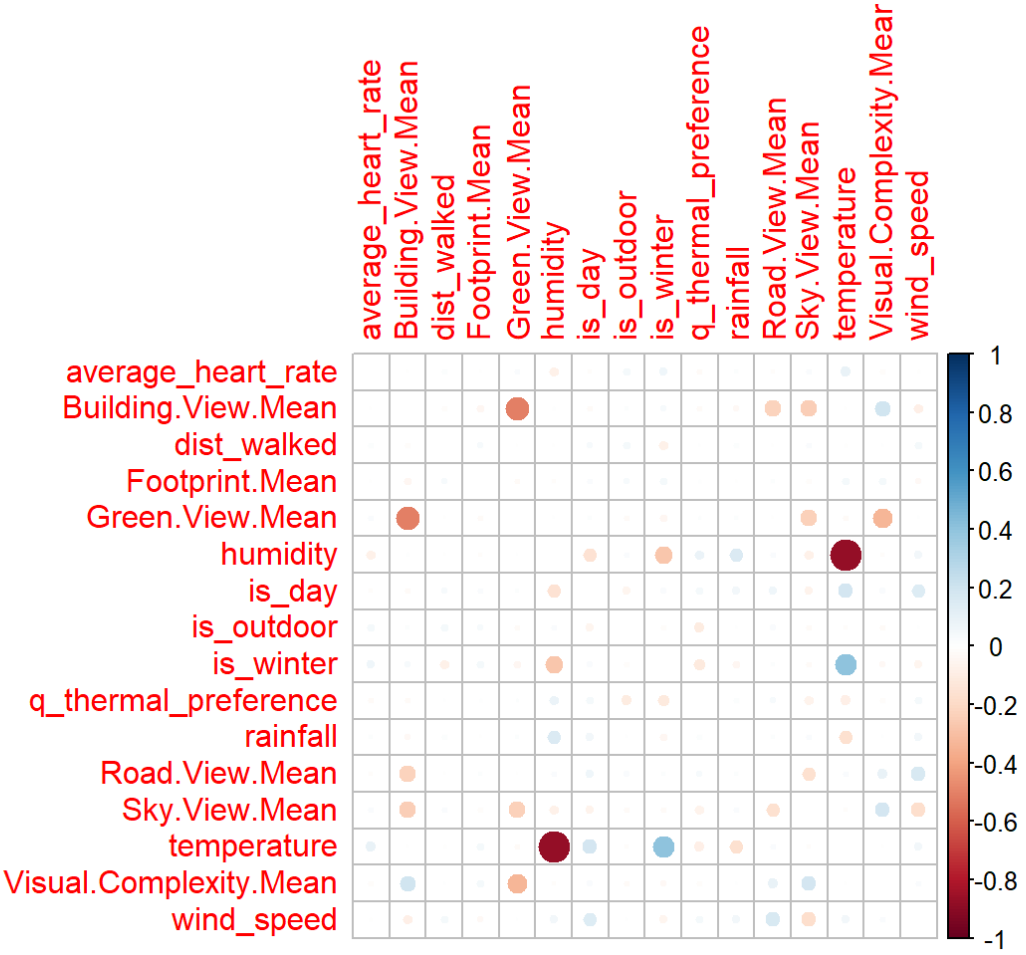
```
randomForest::importance(model_forest)
```

```
##                              FALSE        TRUE MeanDecreaseAccuracy
## dist_walked              0.8563182  1.37172663            1.4746404
## average_heart_rate      -0.8833980  0.05711924           -0.6839402
## Green.View.Mean         12.0010150  9.75821062           16.8150583
## Footprint.Mean          13.6651554  9.88235906           17.5411607
## Sky.View.Mean           13.7253629 12.71720495           19.0889554
## Building.View.Mean      12.2702192 10.35629188           18.4378167
## Road.View.Mean          13.4612004 13.76639433           18.6040803
## rainfall                 3.7616951  0.46017916            3.8177337
## temperature              9.3007950  5.27828073           10.9330602
## wind_speed               5.9826921  0.98871531            5.3589595
## Visual.Complexity.Mean 13.9860610 12.11331136           16.6666900
## is_outdoor               9.1658534 13.02425758           14.0035736
## is_winter               11.5291223 10.40958961           13.1356087
## is_day                   0.1084446  1.07094103            0.7524310
##                         MeanDecreaseGini
## dist_walked                    124.84328
## average_heart_rate             116.39439
## Green.View.Mean                103.87575
## Footprint.Mean                 129.76265
## Sky.View.Mean                  120.27174
## Building.View.Mean             108.88783
## Road.View.Mean                 110.00430
## rainfall                        11.21246
## temperature                    127.13448
## wind_speed                     113.41158
## Visual.Complexity.Mean         120.65905
## is_outdoor                      25.65583
## is_winter                       21.58707
## is_day                          20.35907
```

```
SS_tot <- sum((as.numeric(selected_data_log_train$q_thermal_preference) - mean(as.numeric
(selected_data_log_train$q_thermal_preference))) ^ 2)
SS_res_tree <- sum((as.numeric(selected_data_log_train$q_thermal_preference) - as.numeric
(predict(model_forest, selected_data_log_train))) ^ 2)

R_sq_lm <- 1 - SS_res_tree / SS_tot
R_sq_lm
```

```
## [1] 0.9984285
```

```r
selected_data_multinom <-
  subset(
    selected_data_no_outliers,
    select = -c(
      id_participant,
      ws_longitude,
      ws_latitude,
      Perimeter.Mean,
      wind_direction,
      Building.Count,
      dT
    )
  )


selected_data_multinom$q_thermal_preference <-
  as.factor(selected_data_multinom$q_thermal_preference)

selected_data_multinom$is_outdoor <-
  as.factor(selected_data_multinom$q_location == "Outdoor")

selected_data_multinom$is_winter <- as.factor(selected_data_multinom$date_time > ym("2023/
04"))
selected_data_multinom$is_day <-  as.factor((hour(selected_data_multinom$date_time) > 12 &
                                               hour(selected_data_multinom$date_time) < 18)
== T)

selected_data_multinom <-
  subset(selected_data_multinom, select = -c(q_location, date_time))

set.seed(2)

# Divide the data into 80% training and 20% testing
train <-
  sample(1:nrow(selected_data_multinom),
         size = round(nrow(selected_data_multinom) * 0.8),
         replace = FALSE)

selected_data_multinom_train <- selected_data_multinom[train, ]
selected_data_multinom_test <- selected_data_multinom[-train, ]

selected_data_multinom %>%
  mutate(is_outdoor = as.numeric(is_outdoor)) %>%
  mutate(q_thermal_preference = as.numeric(q_thermal_preference)) %>%
  mutate(is_winter = as.numeric(is_winter)) %>%
  mutate(is_day = as.numeric(is_day)) %>%
  cor(use = "pairwise.complete.obs") %>%
  corrplot(order = 'alphabet', diag = F)
```




```
model_multinom <- multinom(q_thermal_preference ~ ., data = selected_data_multinom_train)
```

```
## # weights:  51 (32 variable)
## initial  value 2942.083709
## iter  10 value 2621.325508
## iter  20 value 2333.878059
## iter  30 value 2262.114329
## iter  40 value 2261.403953
## final  value 2261.403665
## converged
```

```
summary(model_multinom)
```

```
## Call:
## multinom(formula = q_thermal_preference ~ ., data = selected_data_multinom_train)
##
## Coefficients:
##           (Intercept)   dist_walked average_heart_rate Green.View.Mean
## No change   3.7752521  0.0002199776       -0.001969002       -1.199221
## Warmer      0.4628009 -0.0006249347        0.002440529       -1.283316
##           Footprint.Mean Sky.View.Mean Building.View.Mean Road.View.Mean
## No change   2.388761e-06     -2.831785         -2.2248894      -1.000906
## Warmer      1.008029e-04     -1.112459          0.8609189       2.105141
##               humidity    rainfall temperature   wind_speed
## No change  0.004302523 0.002536116 -0.09158742 -0.003618452
## Warmer    -0.001101559 0.122157617 -0.02482563  0.048528087
##           Visual.Complexity.Mean is_outdoorTRUE is_winterTRUE is_dayTRUE
## No change              0.2782135     -0.6701643    -0.3502048  0.1534436
## Warmer                -0.9386555     -0.4311649    -0.4710044  0.3879659
##
## Std. Errors:
##            (Intercept)  dist_walked average_heart_rate Green.View.Mean
## No change 0.0014990947 0.0001231608        0.002853019    0.0012844538
## Warmer    0.0005027521 0.0002912159        0.005606995    0.0003877144
##           Footprint.Mean Sky.View.Mean Building.View.Mean Road.View.Mean
## No change   3.176422e-05  0.0023649486       0.0008636186   0.0006949420
## Warmer      5.492044e-05  0.0004844795       0.0003588285   0.0001798301
##              humidity   rainfall temperature wind_speed Visual.Complexity.Mean
## No change 0.002684677 0.10679755 0.009805298 0.01436564            0.004708234
## Warmer    0.005160460 0.04063834 0.017780836 0.02636042            0.001055362
##           is_outdoorTRUE is_winterTRUE is_dayTRUE
## No change     0.11756837    0.08220748 0.07980995
## Warmer        0.01780133    0.01578459 0.01499350
##
## Residual Deviance: 4522.807
## AIC: 4586.807
```

```
tidy(model_multinom, conf.int = TRUE) %>%
  kable() %>%
  kable_styling("basic", full_width = FALSE)
```

y.level	term	estimate	std.error	statistic	p.value	conf.low
No change	(Intercept)	3.7752521	0.0014991	2518.3546648	0.0000000	3.7723139
No change	dist_walked	0.0002200	0.0001232	1.7861005	0.0740830	-0.0000214
No change	average_heart_rate	-0.0019690	0.0028530	-0.6901468	0.4901019	-0.0075608
No change	Green.View.Mean	-1.1992206	0.0012845	-933.6424328	0.0000000	-1.2017381

y.level	term	estimate	std.error	statistic	p.value	conf.low
No change	Footprint.Mean	0.0000024	0.0000318	0.0752029	0.9400533	-0.0000599
No change	Sky.View.Mean	-2.8317855	0.0023649	-1197.3983070	0.0000000	-2.8364207
No change	Building.View.Mean	-2.2248894	0.0008636	-2576.2409104	0.0000000	-2.2265821
No change	Road.View.Mean	-1.0009062	0.0006949	-1440.2730335	0.0000000	-1.0022682
No change	humidity	0.0043025	0.0026847	1.6026222	0.1090181	-0.0009593
No change	rainfall	0.0025361	0.1067976	0.0237469	0.9810545	-0.2067832
No change	temperature	-0.0915874	0.0098053	-9.3406054	0.0000000	-0.1108055
No change	wind_speed	-0.0036185	0.0143656	-0.2518825	0.8011319	-0.0317746
No change	Visual.Complexity.Mean	0.2782135	0.0047082	59.0908400	0.0000000	0.2689856
No change	is_outdoorTRUE	-0.6701643	0.1175684	-5.7002093	0.0000000	-0.9005941
No change	is_winterTRUE	-0.3502048	0.0822075	-4.2600111	0.0000204	-0.5113285
No change	is_dayTRUE	0.1534436	0.0798100	1.9226122	0.0545288	-0.0029810
Warmer	(Intercept)	0.4628009	0.0005028	920.5351224	0.0000000	0.4618156
Warmer	dist_walked	-0.0006249	0.0002912	-2.1459497	0.0318770	-0.0011957
Warmer	average_heart_rate	0.0024405	0.0056070	0.4352651	0.6633700	-0.0085490
Warmer	Green.View.Mean	-1.2833160	0.0003877	-3309.9517812	0.0000000	-1.2840759
Warmer	Footprint.Mean	0.0001008	0.0000549	1.8354347	0.0664413	-0.0000068
Warmer	Sky.View.Mean	-1.1124589	0.0004845	-2296.1938892	0.0000000	-1.1134085
Warmer	Building.View.Mean	0.8609189	0.0003588	2399.2487940	0.0000000	0.8602156
Warmer	Road.View.Mean	2.1051415	0.0001798	11706.2797585	0.0000000	2.1047890

y.level	term	estimate	std.error	statistic	p.value	conf.low
Warmer	humidity	-0.0011016	0.0051605	-0.2134615	0.8309670	-0.0112159
Warmer	rainfall	0.1221576	0.0406383	3.0059696	0.0026474	0.0425079
Warmer	temperature	-0.0248256	0.0177808	-1.3962017	0.1626538	-0.0596754
Warmer	wind_speed	0.0485281	0.0263604	1.8409453	0.0656296	-0.0031374
Warmer	Visual.Complexity.Mean	-0.9386555	0.0010554	-889.4156712	0.0000000	-0.9407240
Warmer	is_outdoorTRUE	-0.4311649	0.0178013	-24.2209411	0.0000000	-0.4660549
Warmer	is_winterTRUE	-0.4710044	0.0157846	-29.8395155	0.0000000	-0.5019416
Warmer	is_dayTRUE	0.3879659	0.0149935	25.8756094	0.0000000	0.3585792

```
model_multinom2 <- multinom(q_thermal_preference ~ is_outdoor + is_winter + temperature +
humidity + Green.View.Mean + Sky.View.Mean + Building.View.Mean + Road.View.Mean, data = s
elected_data_multinom_train)
```

```
## # weights:  30 (18 variable)
## initial  value 2942.083709
## iter  10 value 2339.496857
## iter  20 value 2276.639460
## final  value 2276.637693
## converged
```

```
summary(model_multinom)
```

```
## Call:
## multinom(formula = q_thermal_preference ~ ., data = selected_data_multinom_train)
##
## Coefficients:
##           (Intercept)   dist_walked average_heart_rate Green.View.Mean
## No change   3.7752521  0.0002199776       -0.001969002       -1.199221
## Warmer      0.4628009 -0.0006249347        0.002440529       -1.283316
##           Footprint.Mean Sky.View.Mean Building.View.Mean Road.View.Mean
## No change   2.388761e-06     -2.831785         -2.2248894      -1.000906
## Warmer      1.008029e-04     -1.112459          0.8609189       2.105141
##               humidity    rainfall temperature   wind_speed
## No change  0.004302523 0.002536116 -0.09158742 -0.003618452
## Warmer    -0.001101559 0.122157617 -0.02482563  0.048528087
##           Visual.Complexity.Mean is_outdoorTRUE is_winterTRUE is_dayTRUE
## No change              0.2782135     -0.6701643    -0.3502048  0.1534436
## Warmer                -0.9386555     -0.4311649    -0.4710044  0.3879659
##
## Std. Errors:
##            (Intercept)  dist_walked average_heart_rate Green.View.Mean
## No change 0.0014990947 0.0001231608        0.002853019    0.0012844538
## Warmer    0.0005027521 0.0002912159        0.005606995    0.0003877144
##           Footprint.Mean Sky.View.Mean Building.View.Mean Road.View.Mean
## No change   3.176422e-05  0.0023649486       0.0008636186   0.0006949420
## Warmer      5.492044e-05  0.0004844795       0.0003588285   0.0001798301
##              humidity   rainfall temperature wind_speed Visual.Complexity.Mean
## No change 0.002684677 0.10679755 0.009805298 0.01436564            0.004708234
## Warmer    0.005160460 0.04063834 0.017780836 0.02636042            0.001055362
##           is_outdoorTRUE is_winterTRUE is_dayTRUE
## No change     0.11756837    0.08220748 0.07980995
## Warmer        0.01780133    0.01578459 0.01499350
##
## Residual Deviance: 4522.807
## AIC: 4586.807
```

```
tidy(model_multinom2, conf.int = TRUE) %>%
  kable() %>%
  kable_styling("basic", full_width = FALSE)
```

y.level	term	estimate	std.error	statistic	p.value	conf.low	conf.h
No change	(Intercept)	4.1357486	1.7413921	2.3749669	0.0175505	0.7226828	7.5488
No change	is_outdoorTRUE	-0.6759504	0.1230572	-5.4929750	0.0000000	-0.9171381	-0.4347
No change	is_winterTRUE	-0.3716983	0.0916106	-4.0573721	0.0000496	-0.5512517	-0.1921
No change	temperature	-0.0870918	0.0406158	-2.1442827	0.0320102	-0.1666974	-0.0074

y.level	term	estimate	std.error	statistic	p.value	conf.low	conf.h
No change	humidity	0.0038375	0.0072002	0.5329792	0.5940480	-0.0102745	0.0179
No change	Green.View.Mean	-1.3402841	0.5842772	-2.2939181	0.0217952	-2.4854464	-0.1951
No change	Sky.View.Mean	-2.7604770	0.6583995	-4.1927079	0.0000276	-4.0509164	-1.4700
No change	Building.View.Mean	-2.1534922	0.7238279	-2.9751438	0.0029285	-3.5721689	-0.7348
No change	Road.View.Mean	-0.7749980	0.9030069	-0.8582415	0.3907591	-2.5448589	0.9948
Warmer	(Intercept)	-2.1556697	3.3693787	-0.6397825	0.5223140	-8.7595307	4.4481
Warmer	is_outdoorTRUE	-0.4533492	0.2486565	-1.8231946	0.0682739	-0.9407070	0.0340
Warmer	is_winterTRUE	-0.4751693	0.1821329	-2.6089147	0.0090830	-0.8321433	-0.1181
Warmer	temperature	0.0289362	0.0789534	0.3664977	0.7139937	-0.1258096	0.1836
Warmer	humidity	0.0057392	0.0138387	0.4147213	0.6783459	-0.0213841	0.0328
Warmer	Green.View.Mean	-1.3286374	1.1668583	-1.1386451	0.2548512	-3.6156377	0.9583
Warmer	Sky.View.Mean	-2.3908784	1.3161961	-1.8165062	0.0692928	-4.9705754	0.1888
Warmer	Building.View.Mean	-0.1337018	1.3852025	-0.0965215	0.9231064	-2.8486488	2.5812
Warmer	Road.View.Mean	1.7403543	1.7774074	0.9791533	0.3275042	-1.7433001	5.2240

```
predicted <-
  predict(model_multinom2, selected_data_multinom_test, type="class")
sum(predicted == selected_data_multinom_test$q_thermal_preference) / nrow(selected_data_mu
ltinom_test)
```

```
## [1] 0.561194
```

```
pR2(model_multinom2)['McFadden']
```

```
## fitting null model for pseudo-r2
## # weights:  6 (2 variable)
## initial  value 2942.083709
## final  value 2338.881449
## converged
```

```
##   McFadden
## 0.02661262
```

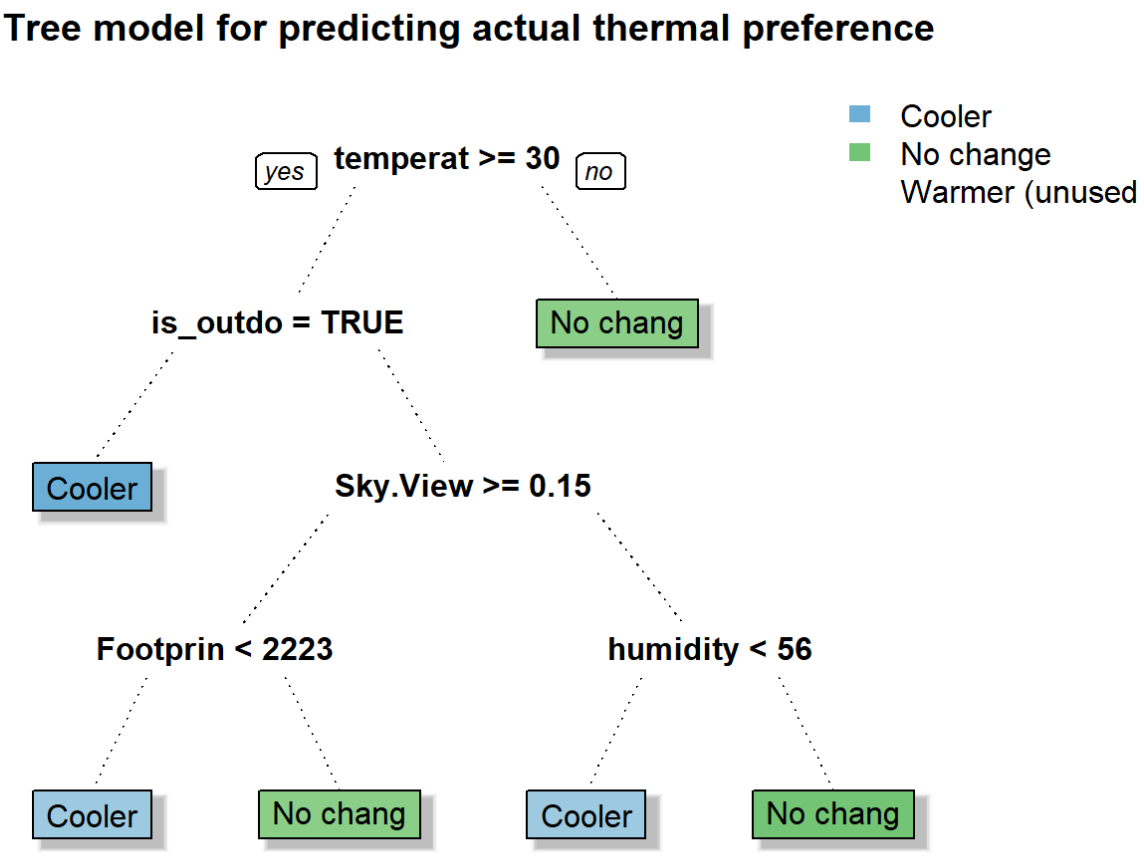
```
Metrics::accuracy(selected_data_multinom_test$q_thermal_preference, predict(model_multinom
2, selected_data_multinom_test, type="class"))
```

```
## [1] 0.561194
```

```
table(predict(model_multinom2, selected_data_multinom_test, type = "class"))
```

```
##
##    Cooler No change    Warmer
##       157       513         0
```

```
fit.tree_multinom = rpart(q_thermal_preference ~ ., data=selected_data_multinom_train, met
hod="class", cp=0.008)
prp(fit.tree_multinom,
    main = "Tree model for predicting actual thermal preference",
    box.palette = "auto",
    fallen.leaves = F,
    shadow.col = "gray",
    branch.lty = 3,
    branch = .5,
    faclen = 0,
    round = 0)
```




```
fit.tree_multinom$variable.importance
```

```
##            temperature               humidity          Sky.View.Mean
##            26.08085381            24.23874296            14.79659833
##             is_outdoor        Footprint.Mean Visual.Complexity.Mean
##            13.51492219            11.84085355             4.19756349
##     Building.View.Mean        Green.View.Mean              is_winter
##             3.52949344             3.24815609             3.11076425
##         Road.View.Mean              wind_speed     average_heart_rate
##             0.64676087             0.20953368             0.09875442
```

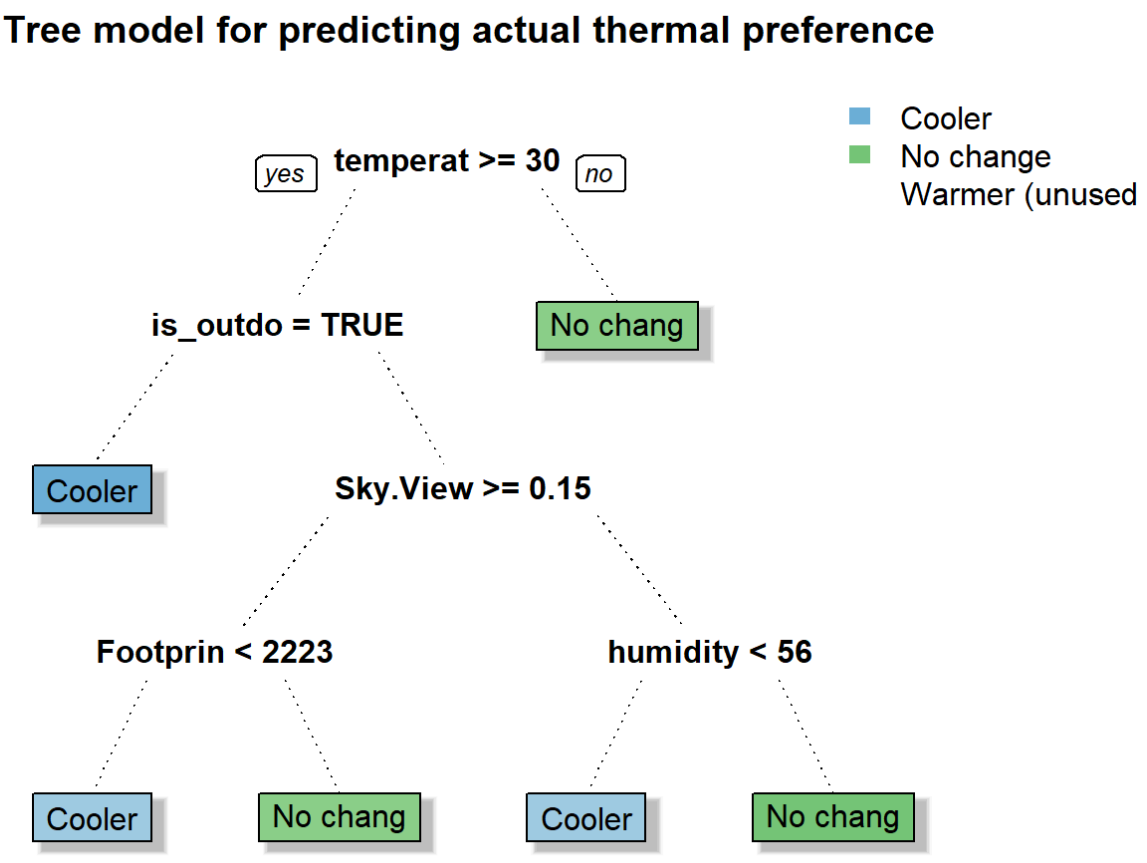
```
bestcp_multinom <- fit.tree_multinom$cptable[which.min(fit.tree_multinom$cptable[,"xerro
r"]),"CP"]
bestcp_multinom
```

```
## [1] 0.008
```

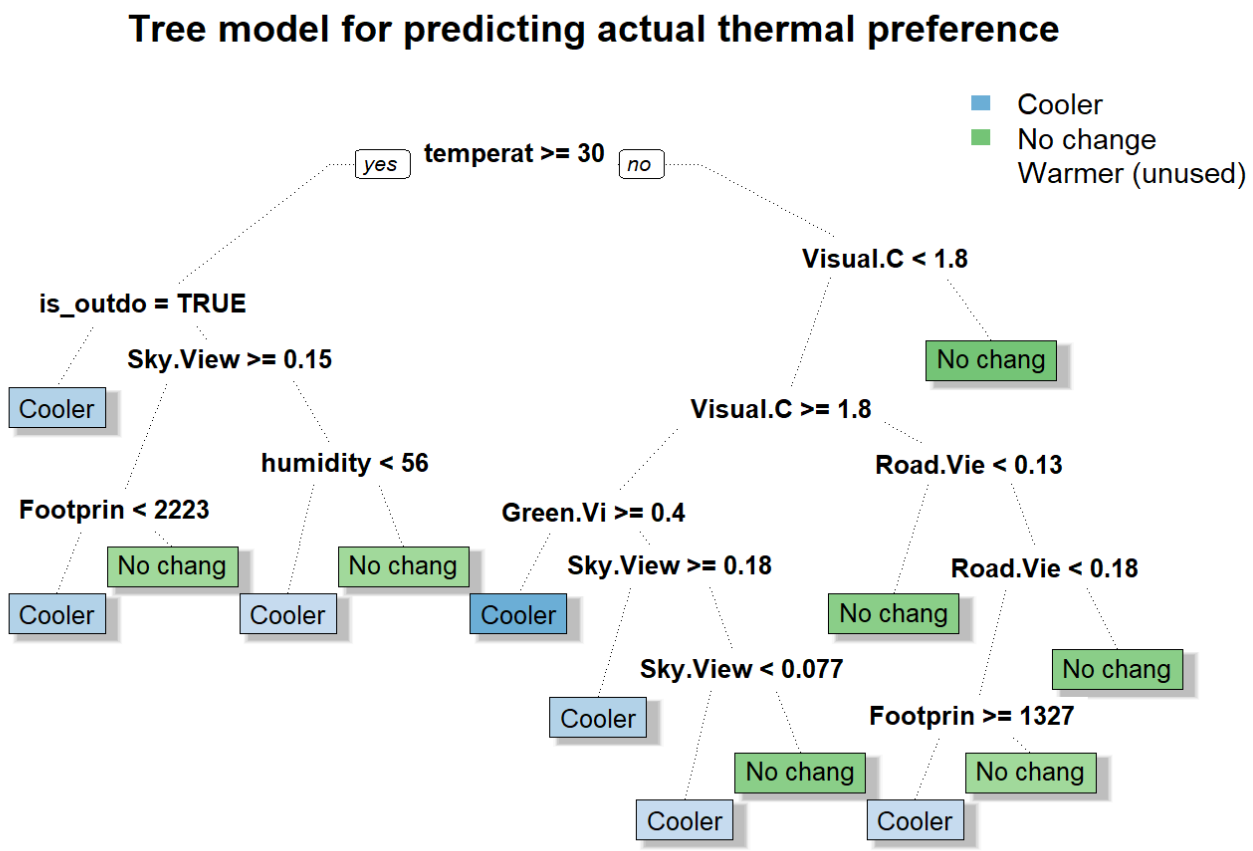
```
final_tree_model <- prune(fit.tree_multinom, cp = bestcp_multinom)
final_tree_model$variable.importance
```

```
##            temperature               humidity          Sky.View.Mean
##             26.08085381            24.23874296             14.79659833
##             is_outdoor         Footprint.Mean Visual.Complexity.Mean
##             13.51492219            11.84085355              4.19756349
##     Building.View.Mean        Green.View.Mean               is_winter
##              3.52949344             3.24815609              3.11076425
##         Road.View.Mean              wind_speed     average_heart_rate
##              0.64676087             0.20953368              0.09875442
```

```
prp(final_tree_model,
    main = "Tree model for predicting actual thermal preference",
    box.palette = "auto",
    fallen.leaves = F,
    shadow.col = "gray",
    branch.lty = 3,
    branch = .5,
    faclen = 0,
    round = 0)
```



```
final_tree_model <-
  rpart(
    q_thermal_preference ~
      Visual.Complexity.Mean +
      Footprint.Mean +
      Sky.View.Mean +
      Green.View.Mean +
      Road.View.Mean +
      Sky.View.Mean +
      temperature +
      humidity +
      is_outdoor,
    data = selected_data_multinom_train,
    method = "class",
    cp = 0.008
  )
prp(final_tree_model,
    main = "Tree model for predicting actual thermal preference",
    box.palette = "auto",
    fallen.leaves = F,
    shadow.col = "gray",
    branch.lty = 3,
    branch = .5,
    faclen = 0,
    round = 0)
```



```
predicted <- as.numeric(predict(final_tree_model, selected_data_multinom_test, type = "cla
ss"))
sum(predicted == as.numeric(selected_data_multinom_test$q_thermal_preference)) / nrow(sele
cted_data_multinom_test)
```

```
## [1] 0.5641791
```

```
table(predict(final_tree_model, selected_data_multinom_test, type = "class"))
```

```
##
##    Cooler No change    Warmer
##       221       449         0
```

```
set.seed(50)

model_forest_multinom <-
  randomForest(
    q_thermal_preference ~ . -average_heart_rate -dist_walked -rainfall -wind_speed,
    data = selected_data_multinom_train,
    importance = TRUE,
    ntree = 200
  )

predicted <- as.numeric(predict(model_forest_multinom, selected_data_multinom_test))
sum(predicted == as.numeric(selected_data_multinom_test$q_thermal_preference)) / nrow(sele
cted_data_multinom_test)
```

```
## [1] 0.6537313
```

```
randomForest::importance(model_forest_multinom)
```

```
##                            Cooler No change    Warmer MeanDecreaseAccuracy
## Green.View.Mean        14.4061414 17.415267 11.719994            23.938412
## Footprint.Mean         17.6342879 15.588526 12.198786            24.695391
## Sky.View.Mean          20.8624277 19.429347 12.203289            31.269214
## Building.View.Mean     18.0361932 18.736024  9.164805            27.435648
## Road.View.Mean         22.3109888 21.242090 12.353233            31.763268
## humidity                8.5017913 13.330698  3.244791            17.787586
## temperature            11.1749083 12.814978  1.724296            17.150709
## Visual.Complexity.Mean 19.8946811 19.867053 13.797653            27.315548
## is_outdoor             12.2169349  8.714548  2.318363            14.959958
## is_winter              13.4873383 14.814294  6.457539            16.964393
## is_day                 -0.1618654  1.811180  1.614403             1.500995
##                        MeanDecreaseGini
## Green.View.Mean               140.96915
## Footprint.Mean                181.56286
## Sky.View.Mean                 153.64321
## Building.View.Mean            149.70710
## Road.View.Mean                149.59443
## humidity                      203.45012
## temperature                   186.22944
## Visual.Complexity.Mean        153.70236
## is_outdoor                     28.69651
## is_winter                      26.44489
## is_day                         32.63674
```

```
table(predict(model_forest_multinom, selected_data_multinom_test, type = "class"))
```

```
##
##    Cooler No change    Warmer
##       233       426        11
```

```
table(selected_data_multinom_test$q_thermal_preference)
```

```
##
##    Cooler No change    Warmer
##       282       346        42
```

```
# if we predict "No Change" we will be correct 51% of the time
sum(selected_data_multinom_test$q_thermal_preference == "No change") / nrow(selected_data_
multinom_test)
```

```
## [1] 0.5164179
```

==========Model Testing Ends Here==========