# 1 Exploratory Data Analysis using R

Panagiotis Lamprakis[a]

[a]PhD Candidate, AI|LS Laboratory, National Technical University of Athens, panoslamprakis@mail.ntua.gr, 03003142

Tetuan is a city located in the north of Morocco which occupies an area of around 10375 km² and its population is about 550.374 inhabitants, according to the last Census of 2014, and is increasing rapidly, approximately 1.96% annually. Since it is located along the Mediterranean Sea, its weather is mild and rainy in the winter, hot and dry during the summer months. The power consumption data was collected from Supervisory Control and Data Acquisition System (SCADA) of Amendis which is a public service operator and in charge of the distribution of drinking water and electricity since 2002. The distribution network is powered by 3 source stations, namely: Quads, Smir and Boussafou. The three stations power 3 different areas of the city (zones).

## 1 Dataset description

We will use the "Tetuan City power consumption.csv" dataset, located in course's website (click here to directly download it). The dataset contains seven columns and has 52416 rows of data. The R code for this analysis can be found in github (repo is private, please ask for access).

| Column Name | Description |
|---|---|
| DateTime | Time window of ten minutes: 01/01/2017 00:00 to 30/12/2017 23:50 |
| Temperature | Weather Temperature in °C |
| Humidity | Weather Humidity in % |
| Wind Speed | Wind Speed in km/h |
| Zone 1 Power Consumption | in KiloWatts (KW) |
| Zone 2 Power Consumption | in KiloWatts (KW) |
| Zone 3 Power Consumption | in KiloWatts (KW) |

## 2 Reading the file and transforming the data

In the following code snippet, we read the data in a data table structure. In order to achieve better performance, we read the first 100 rows, determine the type of the columns from those 100 rows and then we read the entire dataset. We make sure that the dataset is in data table representation.

```
library(data.table)
# Read dataset sample to determine the data table column classes
dataSample <- read.table(DATASET_PATH, nrows=100, sep=",", header = TRUE)

# Read the data by specifying each column type and make sure it is a data
    table
classes <- sapply(dataSample, class)
data <- read.table(DATASET_PATH, colClasses = classes, sep=",", header =
    TRUE)
setDT(data, keep.rownames=T)
```

For the EDA, we want to know which days of the year are bank holidays. For this reason, we will use data from (holidayapi.com 2022). We manually produce a csv file, which we read and create a new column *isBankHoliday* in the original data table:

```
officialBankHolidays <- read.csv(BANK_HOLIDAYS_PATH_OFC, header = F)
```

We will create new features (*isBankHoliday*) and extract features from current data (*Date*, *DoW*, *Month*, *DoM*, *Hour*, *isWeekend*, *season*).

```
dateTimeSplitList <- strsplit(data$DateTime, " +")

# create new date and time columns.
data[, "DateStr"] = sapply(dateTimeSplitList, "[[", 1)
data[, "Time"] = sapply(dateTimeSplitList, "[[", 2)

# Create a column to see if the date was a bank holiday
data[, "isBankHoliday"] = rep(1, nrow(data))
data$isBankHoliday = ifelse(data$DateStr %in% officialBankHolidays[1]$V1,
    1, 0)  # kept as numeric for the correlation map

# Convert Date column to date type, so to extract the month, the day of
    the week and the day of the month
data$Date <- as.Date(data$DateStr, format = "%m/%d/%Y")


data[, "DoW"] = wday(data$Date, week_start=1) # kept as numeric for the
    correlation map
data[, "Month"] = as.numeric(format(data$Date, "%m")) # kept as numeric
    for the correlation map
data[, "DoM"] = as.numeric(format(data$Date, "%d")) # kept as numeric for
    the correlation map
data[, "Hour"] <- as.numeric(format(as_datetime(data$DateTime, format = "%
    m/%d/%Y %H:%M"),'%H'))
```

```
19
20 # Produce info if the day was during the weekend or it was a workday
21 data[, "isWeekend"] = rep(1, nrow(data))
22 data$isWeekend = ifelse(data$DoW %in% c(6, 7), 1, 0) # 6: Sat, 7: Sun
23
24 # Determine the season of the year for the data sample
25 data[, "season"] = sapply(data$Month, getSeason) # convert to numeric for
       the correlation map
26 data[, "seasonF"] = sapply(data$season, encodeSeason)
```

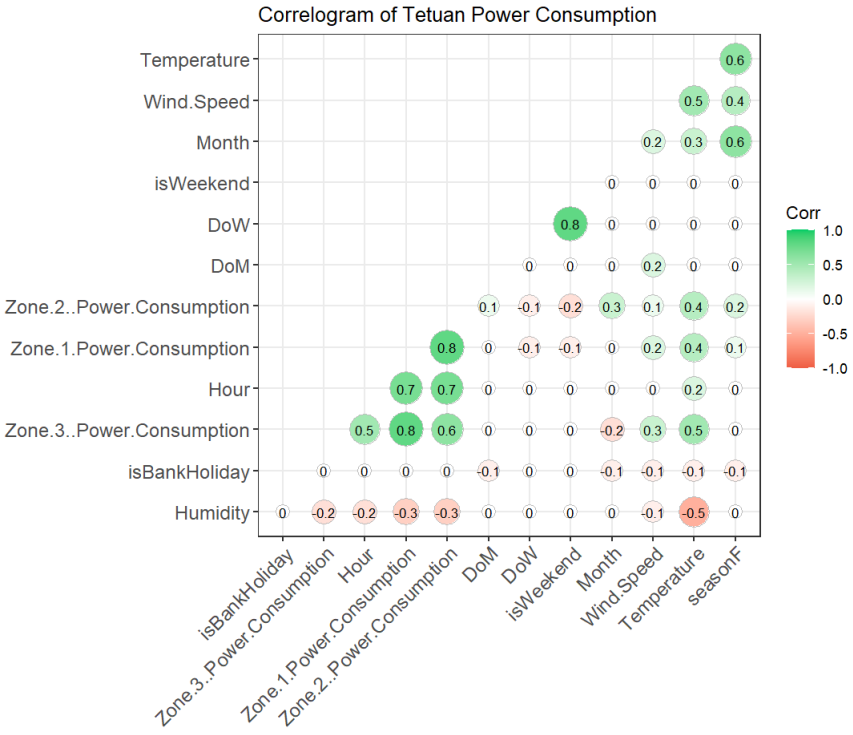
Moreover, in order to produce some other plots we will expand our dataset from its current form. As of now, the dataset contains three columns with the consumption per zone on each column. The transformation we will put in place is to have one column with holding the power consumption information and another one showing to which city zone the consumption refers. The drawback of this approach is that we will duplicate the information we have and that we will end up increasing the memory used by approximately 200%. To achieve this result, we followed the following process:

```
1  dataFiltered = data[, -c("rn", "DateTime", "DateStr", "Time", "season")]
2  dataFiltered$DateF <- as.numeric(factor(data$Date))
3  dataFiltered$TimeF <- as.numeric(factor(data$Time))
4
5  # Expand dataset and have the consumption in one column and another column
       with the zone
6  # This will increase the dataset size by 200% but will help us with the
       plotting
7  dataZone1 <- dataFiltered[, -c("Zone.2..Power.Consumption", "Zone.3..Power
       .Consumption")]
8  setnames(dataZone1, "Zone.1.Power.Consumption", "ZoneConsumption")
9  dataZone1$Zone <- rep(1, nrow(dataZone1))
10
11 dataZone2 <- dataFiltered[, -c("Zone.1.Power.Consumption", "Zone.3..Power.
       Consumption")]
12 setnames(dataZone2, "Zone.2..Power.Consumption", "ZoneConsumption")
13 dataZone2$Zone <- rep(2, nrow(dataZone2))
14
15 dataZone3 <- dataFiltered[, -c("Zone.1.Power.Consumption", "Zone.2..Power.
       Consumption")]
16 setnames(dataZone3, "Zone.3..Power.Consumption", "ZoneConsumption")
17 dataZone3$Zone <- rep(3, nrow(dataZone3))
18
19 dataZoneLevelWithDate <- rbind(dataZone1, dataZone2, dataZone3)
20 dataZoneLevelWithDate$DateF <- as.numeric(factor(dataZoneLevel$Date))
21 dataZoneLevelWithDate$TimeF <- as.numeric(factor(dataZoneLevel$Time))
22
23 dataZoneLevel <- dataZoneLevelWithDate[, -c("Date")]
24 dataZoneLevel[, "ZoneName"] <- factor(sapply(dataZoneLevel$Zone,
       decodeZone))
```

# 3 Data Analysis

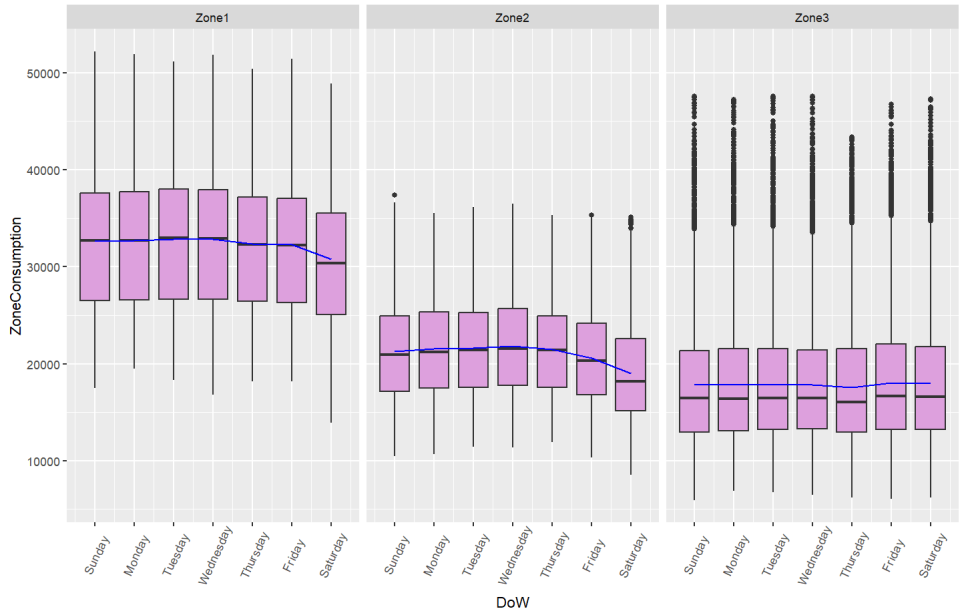
## 3.1 Correlation Analysis

To begin with the analysis, we will first examine the correlation between the dataset features. From the produced correlation map, it is easy to observe that features as *DoM*, *isBankHoliday* are not related to the power consumption, since their correlation score is zero. These are features that we produced/extracted in the previous step. Features such as *DoW*, *isWeekend*, *season* have weak correlation with the power consumption ( correlation coefficient's absolute value ≤0.2), while features such as *Wind.Speed*, *Month*, *Humidity* have moderate correlation with the power consumption. Finally, *Temperature* and *Hour* seem to be related to the power consumption (correlation score ≥0.4). It's also easy to observe that the power consumption in one zone is highly related to the power consumption in the other two zones (correlation score ≥0.6).



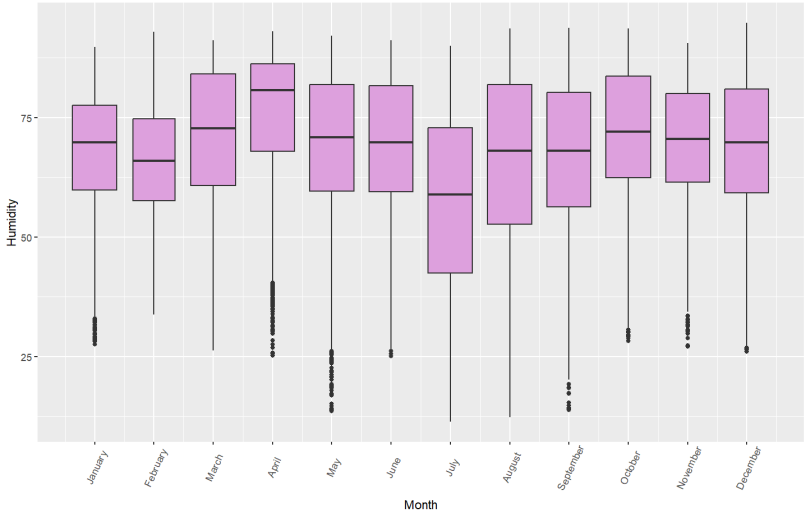
```
dataOnlyNumeric <- data[, -c("rn", "DateStr", "DateTime", "Date", "
Time", "season")]

corr <- round(cor(dataOnlyNumeric), 1)
ggcorrplot(corr, hc.order = TRUE,
           type = "lower",
           lab = TRUE,
           lab_size = 3,
           method="circle",
           colors = c("tomato2", "white", "springgreen3"),
           title="Correlogram of Tetuan Power Consumption",
           ggtheme=theme_bw)
```

We establish our argument that *Dow* is not related to the power consumption, with the following plot. We see that the average power consumption per day is almost constant in each city zone. The code used to produce the plot follows it.



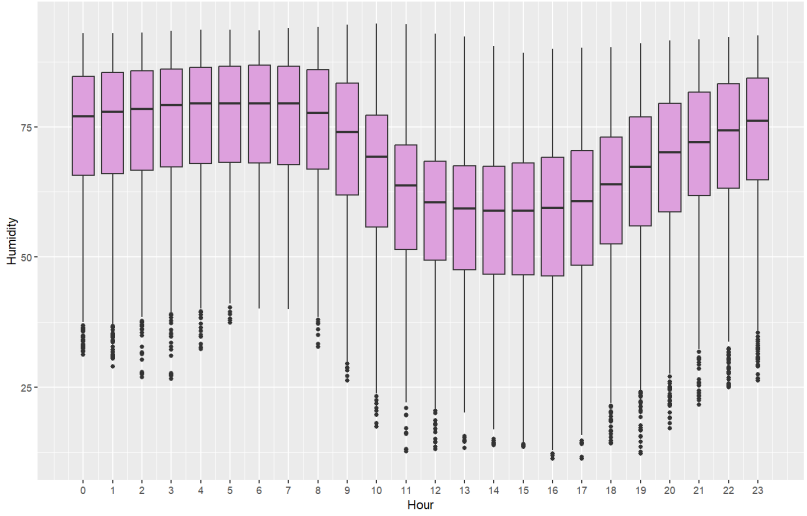

```
# Facet per zone per DoW consumption
facetBoxPlotBreaksDoW <- seq(1, 7)
dataZoneLevel[, "DoWName"] <- sapply(dataZoneLevel$DoW, decodeDoW)
gg <- ggplot(dataZoneLevel, aes (x = DoW , y = ZoneConsumption )) +
      geom_boxplot(varwidth=T, fill="plum", aes(group=DoWName)) +
      stat_summary(aes(group=ZoneName),fun=mean,geom="line", color="blue
") +
      facet_grid(~ZoneName) +
      scale_x_continuous(labels = unique(dataZoneLevel$DoWName), breaks
= facetBoxPlotBreaksDoW) +
      theme(axis.text.x = element_text (angle=65, vjust =0.6))
plot(gg)
```

Another interesting insight we get from the correlation map, is that humidity is not related to month. From the plot, we easily see that the average humidity level is almost constant across months, with the exception of April and July.



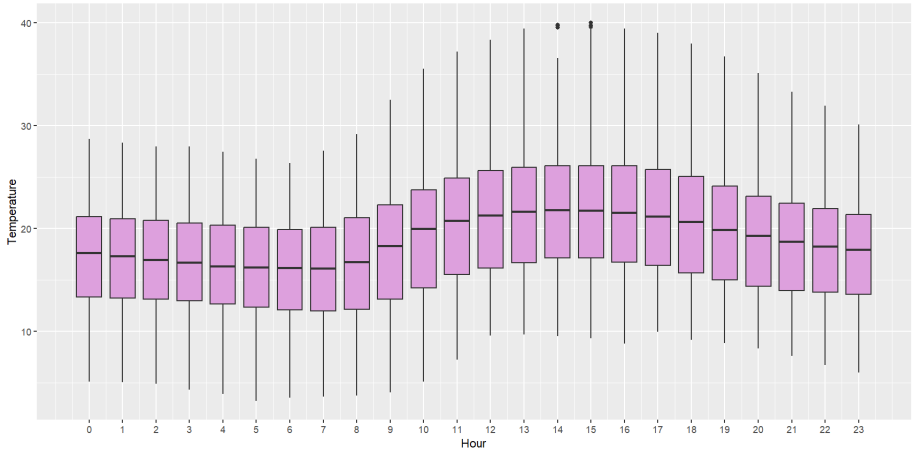

```
gg <- ggplot(dataZoneLevel, aes(x = Month , y = Humidity )) +
      geom_boxplot(varwidth=T, fill="plum", aes(group=MonthName)) +
      scale_x_continuous(labels = unique(dataZoneLevel$MonthName),
breaks = facetBoxPlotBreaksMonth) +
      theme(axis.text.x = element_text (angle=65, vjust =0.6))
plot(gg)
```

However, humidity is related to the hour of the day. Humidity levels decrease during sunlight and are increased during night hours.



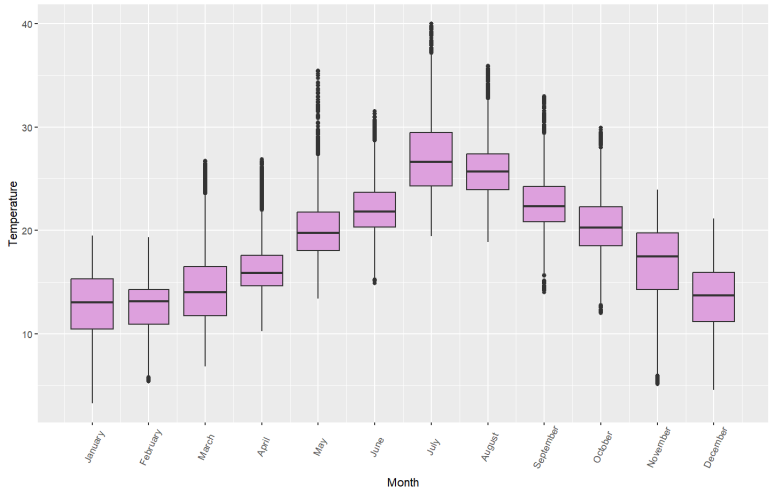
```
gg <- ggplot(dataZoneLevel, aes (x = Hour , y = Humidity )) +
      geom_boxplot(varwidth=T, fill="plum", aes(group=Hour)) +
      scale_x_continuous(labels = facetBoxPlotLabelsHour, breaks =
  facetBoxPlotBreaksHour)
  plot(gg)
```

The below box plot visualizes that Temperature is related to hour (correlation coefficient = 0.2). We observed that temperature increases during afternoon hours.




```
gg <- ggplot(dataZoneLevel, aes (x = Hour , y = Temperature )) +
      geom_boxplot(varwidth=T, fill="plum", aes(group=Hour)) +
      scale_x_continuous(labels = facetBoxPlotLabelsHour, breaks =
  facetBoxPlotBreaksHour)
plot(gg)
```

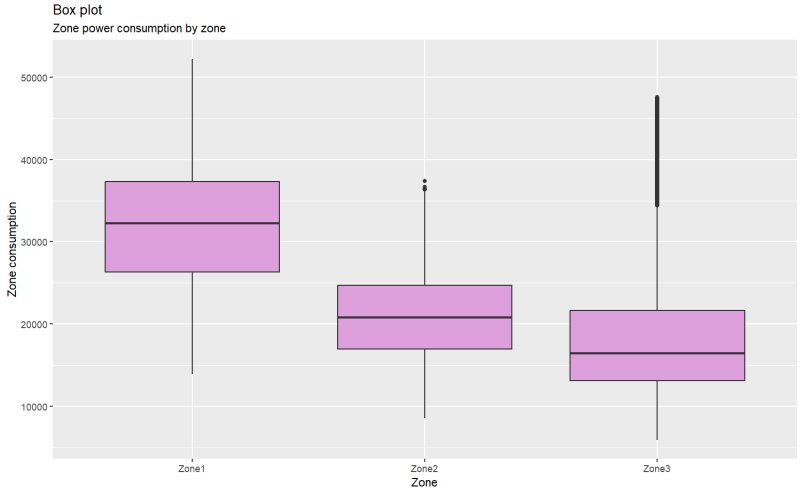
More interesting is the box plot for the Temperature compared to Month. May to October are the most hot months of the year for Tetuan, with July to be the hottest among all.




```
gg <- ggplot(dataZoneLevel, aes (x = Month , y = Temperature )) +
      geom_boxplot(varwidth=T, fill="plum", aes(group=MonthName)) +
      scale_x_continuous(labels = unique(dataZoneLevel$MonthName),
breaks = facetBoxPlotBreaksMonth) +
      theme(axis.text.x = element_text (angle=65, vjust =0.6))
plot(gg)
```

## 3.2 Power Consumption Analysis

From now and on, our feature of interest will be the power consumption. In order to study the distribution of power consumption we will plot its Box plot.



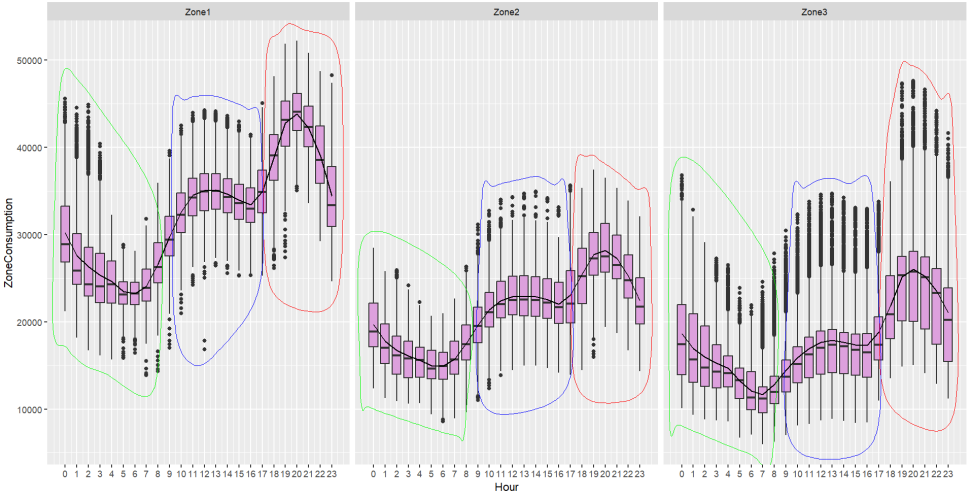
```
g <- ggplot(dataZoneLevel, aes(Zone, ZoneConsumption)) +
     geom_boxplot(varwidth=T, fill="plum") +
     labs(title="Box plot",
          subtitle="Zone power consumption by zone",
          x="Zone",
          y="Zone consumption")
plot(g)
ld <- layer_data(g) # retrieve box plot data
```

From the box plot itself, it is easy to see that the highest average power consumption is in Zone 1 while Zone 3 has the lowest consumption. This leads us to think that Zone 1 may be a rich neighborhood in the city, while Zone 3 is the least rich. Table 1 describes the box plot in precision.

Table 1: Box plot data

| | Mean | 25% | 75% | Whisker(High) | Whisker(Low) |
|---|---|---|---|---|---|
| Zone 1 | 32265.92 | 26310.67 | 37309.02 | 52204.40 | 13895.696 |
| Zone 2 | 20823.17 | 16980.77 | 24713.72 | 36201.48 | 8560.081 |
| Zone 3 | 16415.12 | 13129.33 | 21624.10 | 34361.17 | 5935.174 |

We plot the power consumption per zone per hour. We split semantically the plot into three subsections. We assume that people sleep in time window 00:00 - 8:00 (*green* circle), they work in time window 9:00 - 17:00 (*blue* circle) and they rest at home in time window 18:00 - 23:00 (*red* circle).



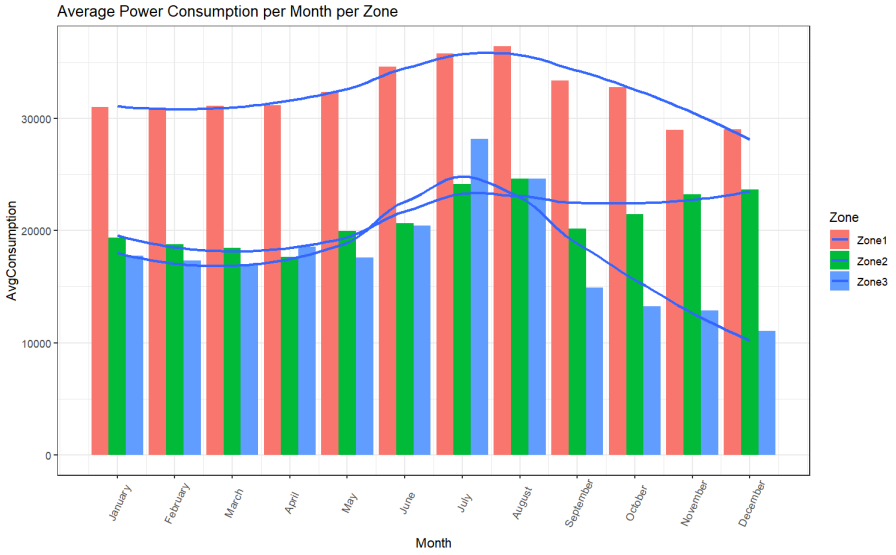
```
# Facet per zone per Hour consumption
facetBoxPlotLabelsHour <- seq(0, 23)
facetBoxPlotBreaksHour <- seq(0, 23)
gg <- ggplot(dataZoneLevel, aes (x = Hour , y = ZoneConsumption )) +
      geom_boxplot(varwidth=T, fill="plum", aes(group=Hour)) +
      stat_summary(aes(group=ZoneName),fun=mean,geom="line") +
      facet_grid(~ZoneName) +
      scale_x_continuous(labels = facetBoxPlotLabelsHour, breaks =
facetBoxPlotBreaksHour) +
      geom_encircle(data = dataZoneLevel[Hour >= 18 & Hour <= 23 ],
aes(x = Hour , y = ZoneConsumption), colour="red") +
      geom_encircle(data = dataZoneLevel[Hour > 9 & Hour < 17 ], aes(x
 = Hour , y = ZoneConsumption), colour="blue") +
      geom_encircle(data = dataZoneLevel[Hour >= 0 & Hour < 8 ], aes(x
 = Hour , y = ZoneConsumption), colour="green")
plot(gg)
```

In the 3.1 section, we saw that in time window 18:00-10:00 (*green* and *red* circles), temperature is lower and humidity is higher. In the ZoneConsumption vs Hour plot, we see that in all zones, the power consumption is higher in the time period 18:00-23:00. This observation makes sense, since people leave their jobs (decentralize heating/cooling from corporate environments to home, shops, malls, etc) in the evening and they need to do household chores (cooking, laundry, etc). Also, electricity could be used for dehumidifier appliances at home. Power consumption drops in time window defined by *green* circle, since people sleep at that hours.


```
# Facet per zone per Month consumption
facetBoxPlotBreaksMonth <- seq(1, 12)
dataZoneLevel[, "MonthName"] <- sapply(dataZoneLevel$Month,
decodeMonth)
```

```
gg <- ggplot(dataZoneLevel, aes (x = Month , y = ZoneConsumption )) +
      geom_boxplot(varwidth=T, fill="plum", aes(group=Month)) +
      stat_summary(aes(group=ZoneName),fun=mean,geom="line", color="
blue") +
      facet_grid(~ZoneName) +
      scale_x_continuous(labels = unique(dataZoneLevel$MonthName),
breaks = facetBoxPlotBreaksMonth) +
      theme(axis.text.x = element_text (angle=65, vjust =0.6))
plot(gg)
```

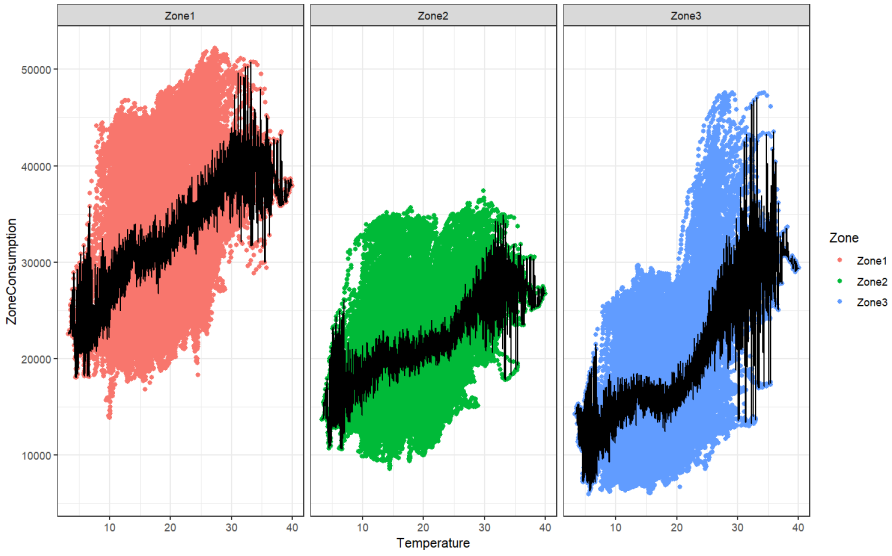
We also saw that May to October are the hottest months of the year for Tetuan, while humidity is not related to the month (humidity levels are almost constant across months). From the below plot, we see that the power consumption during summertime is higher for all three city zones (positive correlation ratio).




```
# Average consumption PER MONTH PER ZONE
avgConPerMonPerZoneData = dataZoneLevel[, c("ZoneName", "Month", "
ZoneConsumption")]
avgConPerMonPerZoneData[, "MonthName"] = sapply(
avgConPerMonPerZoneData$Month, decodeMonth)

avgConPerMonPerZoneData = avgConPerMonPerZoneData[, .(mean(
ZoneConsumption)) , by = .(Month, ZoneName)]
setnames(avgConPerMonPerZoneData, "V1", "AvgConsumption")

# It is aggregated per month per zone, so first 12 rows
# contain all discrete months in the proper order
months <- sapply(avgConPerMonPerZoneData$Month, decodeMonth)

gg <- ggplot(avgConPerMonPerZoneData, aes (x = Month, y =
AvgConsumption , fill = ZoneName )) +
      geom_bar(position = "dodge", stat = "identity") +
```

```
            geom_smooth(method="loess", se=F) +
            scale_x_continuous(breaks=seq(1, 12), labels=months[1:12]) +
            theme(axis.text.x = element_text (angle=65, vjust =0.6)) +
            labs(title="Average Power Consumption per Month per Zone")
    plot(gg)
```

In principle, we see that the power consumption increases as temperature increases in all zones.




```
    # Temperature Vs Consumption Facet per Zone
    gg <- ggplot(dataZoneLevel, aes (x = Temperature , y = ZoneConsumption
     )) +
          geom_point(aes(color=ZoneName)) +
          stat_summary(aes(group=ZoneName),fun=mean,geom="line") +
          facet_grid(~Zone)
    plot(gg)

```

# 4 Appendix

Auxiliary functions declared in scope of the analysis:

```
# Declare auxiliary functions
getSeason <- function(month) {
  if (month %in% c(12, 1, 2)) {
    "winter"
  } else if (month %in% c(3, 4, 5)) {
    "spring"
  } else if (month %in% c(6, 7, 8)) {
    "summer"
  } else if (month %in% c(9, 10, 11)) {
    "autumn"
  } else {
    "unknown"
  }
}

encodeSeason <- function(season) {
  if (season == "winter") {
    1
  } else if (season == "spring") {
    2
  } else if (season == "summer") {
    3
  } else if (season == "autumn") {
    4
  } else {
    5
  }
}

decodeMonth <- function(month) {
  if (month == 1) {
    "January"
  } else if (month == 2) {
    "February"
  } else if (month == 3) {
    "March"
  } else if (month == 4) {
    "April"
  } else if (month == 5) {
    "May"
  } else if (month == 6) {
    "June"
  } else if (month == 7) {
    "July"
  } else if (month == 8) {
    "August"
  } else if (month == 9) {
    "September"
  } else if (month == 10) {
    "October"
  } else if (month == 11) {
```

```
    "November"
  } else {
    "December"
  }
}

decodeZone <- function(zoneId) {
  if (zoneId == 1) {
    "Zone1"
  } else if (zoneId == 2) {
    "Zone2"
  } else {
    "Zone3"
  }
}

encodeZone <- function(zoneName) {
  if (zoneName == "Zone1") {
    1
  } else if (zoneName == "Zone2") {
    2
  } else {
    3
  }
}
```

## References

holidayapi.com. 2022. https://holidayapi.com/countries/ma-01/2017 (27 December, 2022).