Water Analytics: Predicting water level direction in the Petrignano aquifer

Chiara Cavalagli, Stefanija Galevska, Santiago Víquez

July, 2022

1. Introduction

Water is one of the main ingredients for life on earth. Due to environmental changes every year more and more waterbodies such as lakes, rivers and aquifers are drying up. Waterbodies in Italy are not the exception. In the time of writing many regions in Italy have declared state of emergency since during the current summer many waterbodies that were used to supply water to towns are now almost empty.

The fact that Italy is currently facing a shortage of water motivated us to explore if we can predict the future of one of it's waterbodies. Specifically the Petrignano aquifer located in Umbria a city in the center of the country. In this report we display our analysis and the code we used to achieve it, we talk about anomalies in the data, explore each feature that we had access, and predicted the direction of the water levels using a baseline and a logistic regression model with lag variables.

2. Data

In this analysis we will be working with a dataset from the Acea Group an Italian multiutility operator. Acea Group manages and develops water and electricity networks and environmental services. For this analysis we will focus on one of their data sets of the Petrignano aquifer.

The Petrignano aquifer is located in Umbria, a region of central Italy. This aquifer is fed by three other underground aquifers and the Chiascio river.

In **Table 1** we can see the information that we have available about the Petrignano aquifer. Depth_to_Groundwater_P24 and Depth_to_Groundwater_P25 are our main measures of interests. We will try to forecast the trend direction of both of them by using their past values and other information such as rainfall, temperature, volume, etc.

Field	Format	Description
Date	Daily Date	Uniquely identifies a day (Primary Key)
Rainfall_Bastia_Umbra	Real Number	Quantity of rain falling, expressed in millimeters (mm)
Depth_to_Groundwater_P24	Real Number	Groundwater level, expressed in (m) from the ground floor, detected by the piezometer P24
Depth_to_Groundwater_P25	Real Number	Groundwater level, expressed in (m) from the ground floor, detected by the piezometer P25
Temperature_Bastia_Umbra	Real Number	Temperature, expressed in °C
Temperature_Petrignano	Real Number	Temperature, expressed in °C

Field	Format	Description
Volume_C10_Petrignano	Real Number	Volume of water, expressed in cubic meters (mc)
Hydrometry_Fiume_Chiascio_Petrignano	Real Number	Groundwater level, expressed in meters (m)

Table 1: Feature description

This experiment is particularly interesting because of the importance of predicting the trend direction of the depth to groundwater. This is something every water supply company would like to explore. Since during different periods of the year waterbodies start to drain, the water companies need to forecast the water level so they can successfully handle daily consumption.

2.1 Data Cleaning

Let's begin by taking a look at basic statistics of the Petrignano dataset.

```
df <- read.csv("Aquifer_Petrignano.csv")
colnames(df)[1]="Date"
summary(df)
```

```
##      Date           Rainfall_Bastia_Umbra Depth_to_Groundwater_P24
##  Length:5223        Min.   : 0.000        Min.   :-34.47
##  Class :character   1st Qu.: 0.000        1st Qu.:-28.25
##  Mode  :character   Median : 0.000        Median :-25.99
##                     Mean   : 1.557        Mean   :-26.26
##                     3rd Qu.: 0.100        3rd Qu.:-23.82
##                     Max.   :67.300        Max.   :-19.66
##                     NA's   :1024          NA's   :55
##  Depth_to_Groundwater_P25 Temperature_Bastia_Umbra Temperature_Petrignano
##  Min.   :-33.71           Min.   :-3.70            Min.   :-4.20
##  1st Qu.:-27.62           1st Qu.: 8.80            1st Qu.: 7.70
##  Median :-25.54           Median :14.70            Median :13.50
##  Mean   :-25.69           Mean   :15.03            Mean   :13.74
##  3rd Qu.:-23.43           3rd Qu.:21.40            3rd Qu.:20.00
##  Max.   :-19.10           Max.   :33.00            Max.   :31.10
##  NA's   :39               NA's   :1024             NA's   :1024
##  Volume_C10_Petrignano Hydrometry_Fiume_Chiascio_Petrignano
##  Min.   :-45545        Min.   :0.000
##  1st Qu.:-31679        1st Qu.:2.100
##  Median :-28689        Median :2.400
##  Mean   :-29043        Mean   :2.373
##  3rd Qu.:-26218        3rd Qu.:2.700
##  Max.   :     0        Max.   :4.100
##  NA's   :198           NA's   :1024
```

We see that all columns, including the targets: Depth_to_Groundwater_P24, Depth_to_Groundwater_P25 contain NA's values. Let's take a deeper look to see if we can find for which dates these values are NA.

```
library(dplyr)
df$Date <- as.Date(df$Date, format="%d/%m/%Y")
df$Month <- format(as.Date(df$Date, format="%m"), "%m")
df$Year <- format(as.Date(df$Date, format="%d/%m/%Y"),"%Y")

year.nan <- df %>%
```

```
  group_by(Year) %>%
  summarise_all(funs(sum(is.na(.))))

print(year.nan %>%
  select(Year, Rainfall_Bastia_Umbra:Temperature_Bastia_Umbra))
```

```
## # A tibble: 15 x 5
##    Year  Rainfall_Bastia_Umb~ Depth_to_Ground~ Depth_to_Ground~ Temperature_Bas~
##    <chr>                <int>            <int>            <int>            <int>
##  1 2006                   293                3                0              293
##  2 2007                   365                2                2              365
##  3 2008                   366               11               10              366
##  4 2009                     0                0                0                0
##  5 2010                     0                0                0                0
##  6 2011                     0                0                0                0
##  7 2012                     0                7                7                0
##  8 2013                     0               21                0                0
##  9 2014                     0                0                9                0
## 10 2015                     0                0                0                0
## 11 2016                     0                0                0                0
## 12 2017                     0                0                0                0
## 13 2018                     0                0                0                0
## 14 2019                     0                7                7                0
## 15 2020                     0                4                4                0
```

```
print(year.nan %>%
  select(Year, Temperature_Petrignano:Hydrometry_Fiume_Chiascio_Petrignano))
```

```
## # A tibble: 15 x 4
##    Year  Temperature_Petrignano Volume_C10_Petrignano Hydrometry_Fiume_Chiascio~
##    <chr>                  <int>                 <int>                      <int>
##  1 2006                     293                   197                        293
##  2 2007                     365                     0                        365
##  3 2008                     366                     0                        366
##  4 2009                       0                     0                          0
##  5 2010                       0                     0                          0
##  6 2011                       0                     0                          0
##  7 2012                       0                     0                          0
##  8 2013                       0                     0                          0
##  9 2014                       0                     0                          0
## 10 2015                       0                     0                          0
## 11 2016                       0                     0                          0
## 12 2017                       0                     0                          0
## 13 2018                       0                     0                          0
## 14 2019                       0                     0                          0
## 15 2020                       0                     1                          0
```

From the printed outputs above we can see that years 2006, 2007 and 2008 are the ones that contain most of the NA's values. Because of the lack of information for these years, we are going to restrict our analysis on data from 2009 to 2020.

```
df <- filter(df, Year > 2008)
```

Let's explore now the NA's in the columns `Depth_to_Groundwater_P24` and `Depth_to_Groundwater_P25`. These two columns are of special interest for the analysis since from them we are going to build the target variables `Direction_P24` and `Direction_P25`. This directions will let us know if the depth level of the

aquifer is growing or not.

```
library(ggplot2)
library(gridExtra)

p1 <- ggplot(df, aes(x=Date, y=Depth_to_Groundwater_P24)) +
  geom_line() +
  xlab("")

p2 <- ggplot(df, aes(x=Date, y=Depth_to_Groundwater_P25)) +
  geom_line() +
  xlab("")

grid.arrange(p1, p2, nrow = 1)
```

Figure 1: Trend of Depth_to_Groundwater_{P24, P25}

In **Figure 1** we see small gaps of records on both `Depth_to_Groundwater` trends. It is easy to spot that the course of the trends is somewhat obvious and applying interpolation would be a good idea to fill in the gaps. We follow with the application and observe the results.

```
library(imputeTS)
inp1 <- na_interpolation(x=df$Depth_to_Groundwater_P24)
inp2 <- na_interpolation(x=df$Depth_to_Groundwater_P25)
inp3 <- na_interpolation(x=df$Volume_C10_Petrignano)

p1 <- ggplot_na_imputations(df$Depth_to_Groundwater_P24, inp1)
p2 <- ggplot_na_imputations(df$Depth_to_Groundwater_P25, inp2)

df$Depth_to_Groundwater_P24 = inp1
```

```
df$Depth_to_Groundwater_P25 = inp2
df$Volume_C10_Petrignano = inp3

grid.arrange(p1, p2, nrow = 1)
```

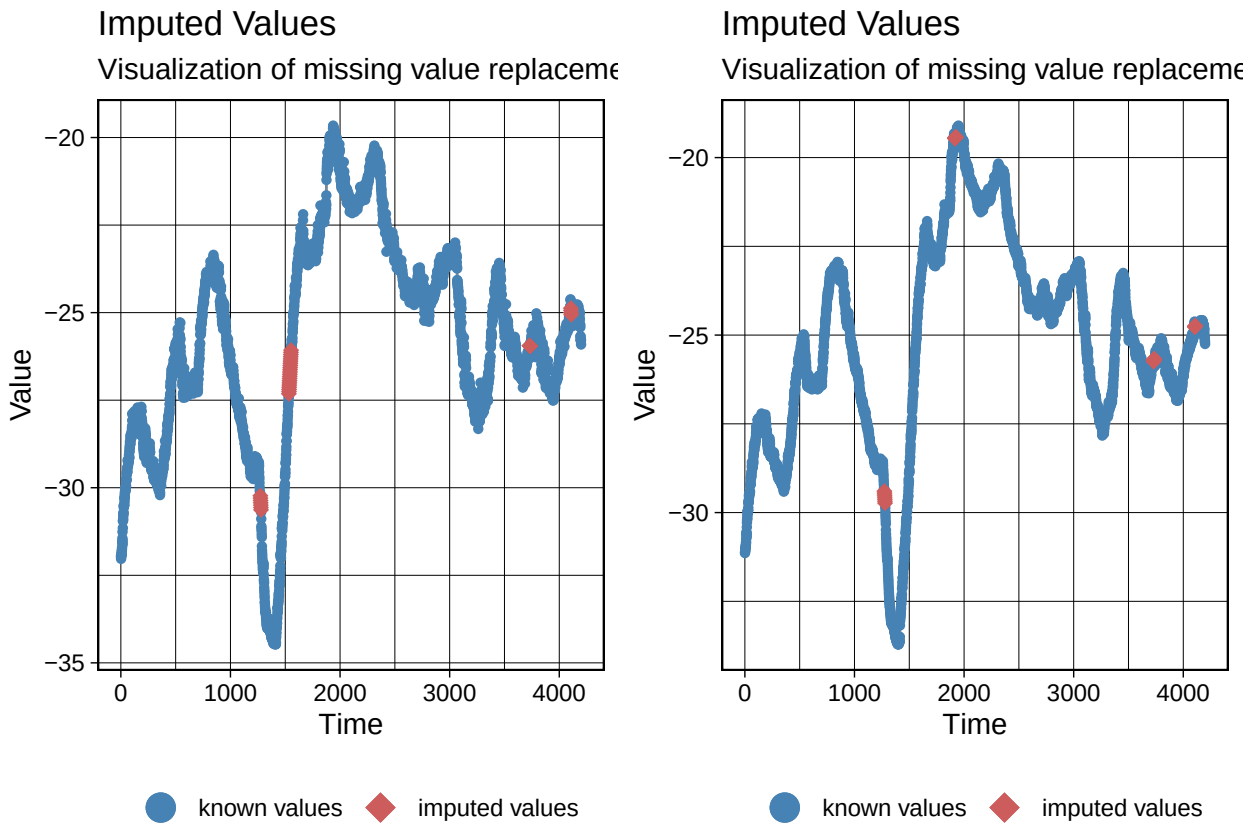

Figure 2:
Left: Imputed values for Depth_to_Groundwater_P24. Right: Imputed values for Depth_to_Groundwater_P25.

In the code cell above we perform an interpolation to fill the NA values for columns `Depth_to_Groundwater_{P24, P25}` and `Volume_C10_Petrignano`. We can see in **Figure 2** the known and imputed values. The imputed values clearly fit in the trend of already known values, it is our opinion that using this method is a safe approach. In theory we should not have any NA's values left in our data, however let's confirm that assuption by running the cell below.

```
colSums(is.na(df))
```

```
##                              Date                Rainfall_Bastia_Umbra
##                                 0                                    0
##             Depth_to_Groundwater_P24             Depth_to_Groundwater_P25
##                                 0                                    0
##             Temperature_Bastia_Umbra               Temperature_Petrignano
##                                 0                                    0
##                Volume_C10_Petrignano Hydrometry_Fiume_Chiascio_Petrignano
##                                 0                                    0
##                                Month                                 Year
##                                 0                                    0
```

2.2 Train, Validation and Test sets

Before further analysis of our data or creating any new features, let's split our imputed data set in three parts. These three parts will serve as our train, validation, and test sets accordingly. We do this to avoid data leakage and to build a robust model validation strategy.

```
train <- filter(df, Year < 2017) # 2922 samples
val <- filter(df, Year >= 2017 & Year <= 2018) # 730 samples
test <- filter(df, Year > 2018) # 547 samples
```

2.3 Feature Engineering

2.3.1 Lag Features

Lag features are values at previous time steps. Our assumption is that what happened in the past can influence what it going to happen in the future.

We are going to create a `lag_1` and `lag_2` for both `Depth_to_Groundwater_{P24, P25}` targets. The assumption of these new features is that the value of 1, 2 time units behind of `Depth_to_Groundwater_{P24, P25}` contain crucial information that can help us estimate it's future value.

```
create_lag <- function(data, col_name, step) {
  name <- paste("lag_", step, "_", col_name, sep="")
  data[name] <- lag(data[col_name], n=step)

  return(data)
}

train <- create_lag(train, "Depth_to_Groundwater_P24", 1)
train <- create_lag(train, "Depth_to_Groundwater_P25", 1)

train <- create_lag(train, "Depth_to_Groundwater_P24", 2)
train <- create_lag(train, "Depth_to_Groundwater_P25", 2)
```

2.3.2 Creating The Direction Column

Part of the main focus of this project is to predict the future directions in which our depth to groundwater values will continue. Meaning, taking into account the previous measurements of the depths and also other features, plainly said we aim to predict if the depth level of the aquifer will grow or not.

```
create_direction <- function(data, col_name) {
  # create direction column for Depth_to_Groundwater_P24
  direction_col <- paste("Direction_", strsplit(col_name, "_")[[1]][[4]], sep="")
  lag_direction_col <- paste("lag_1_", col_name, sep="")
  data[direction_col] <- as.numeric(abs(data[col_name])-abs(data[lag_direction_col]) < 0)
  data[direction_col][is.na(data[direction_col])] <- 0

  return(data)
}

train <- create_direction(train, "Depth_to_Groundwater_P24")
train <- create_direction(train, "Depth_to_Groundwater_P25")
```

2.3.3 Date Time Features

One of the assumptions that we would like to test is to see if particular seasons such as winter and summer have strong effects on the water levels. Because of this we will create a `season` column that states the current

season.

```
create_season <- function(data) {
  data$season <-"winter"
  data$season[as.numeric(data$Month)>2&as.numeric(data$Month)<5]<-"spring"
  data$season[as.numeric(data$Month)>4&as.numeric(data$Month)<9]<-"summer"
  data$season[as.numeric(data$Month)>8&as.numeric(data$Month)<12]<-"autumn"
  data$season<-factor(data$season,levels=c("summer","spring","winter","autumn"))

  return(data)
}

train <- create_season(train)
```

3. Exploratory Data Analysis

3.1 Exploring Depth columns

3.1.1 How does Depth_to_Groundwater_P24 and Depth_to_Groundwater_P25 differ from each other?

As we learned from the data set description, we have two columns with values from two different piezometers (device used for measuring liquid pressure). In the following block we plot the data about these columns and observe the similarities.

```
p1 <- ggplot(train, aes(x=Depth_to_Groundwater_P24)) +
  geom_histogram(aes(y = ..density..),
                 colour = 1, fill = "white") +
  geom_density(lwd = 1, colour = 5,
               fill = 5, alpha = 0.25)


p2 <- ggplot(train, aes(x=Depth_to_Groundwater_P25)) +
  geom_histogram(aes(y = ..density..),
                 colour = 1, fill = "white") +
  geom_density(lwd = 1, colour = 2,
               fill = 2, alpha = 0.25)

grid.arrange(p1, p2, nrow = 1)
```

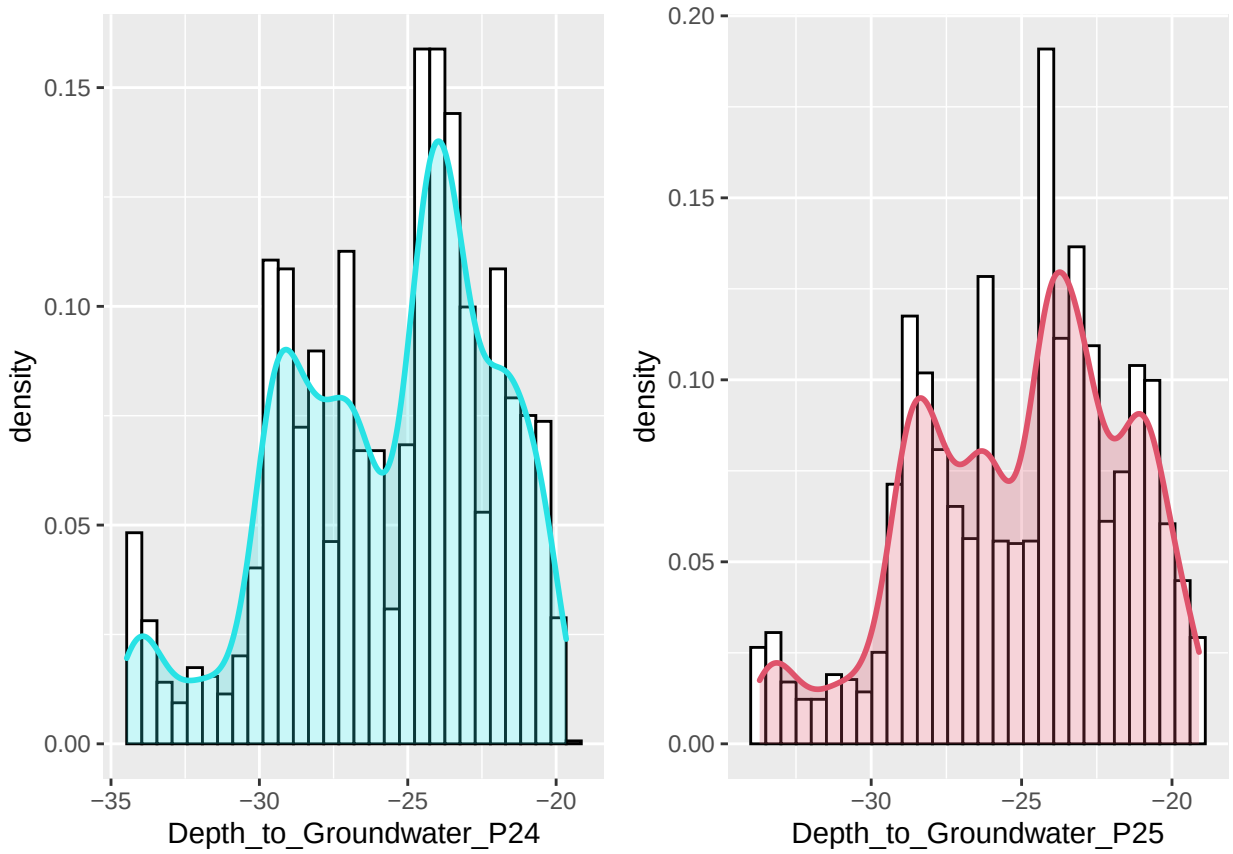

Figure 3: Distribution of Depth_to_Groundwater_{P24, P25}

We note that the measurements from the two devices slightly differ. This can be due to the fact that they are positioned in different areas of the aquifer or it is a case of two different models of device. This information is not available to us so we assume one of the two. Most importantly we know that they measure the same aquifer and the resulted values are similar to each other that we can consider both and explore them both.

Following is a figure to compare them more closely.

```
p1 <- ggplot(train) +
  geom_density(aes(x=Depth_to_Groundwater_P24, color="Depth_to_Groundwater_P24")) +
  geom_density(aes(x=Depth_to_Groundwater_P25, color="Depth_to_Groundwater_P25")) +
  xlab("Depth to Groundwater (m)")

p1
```

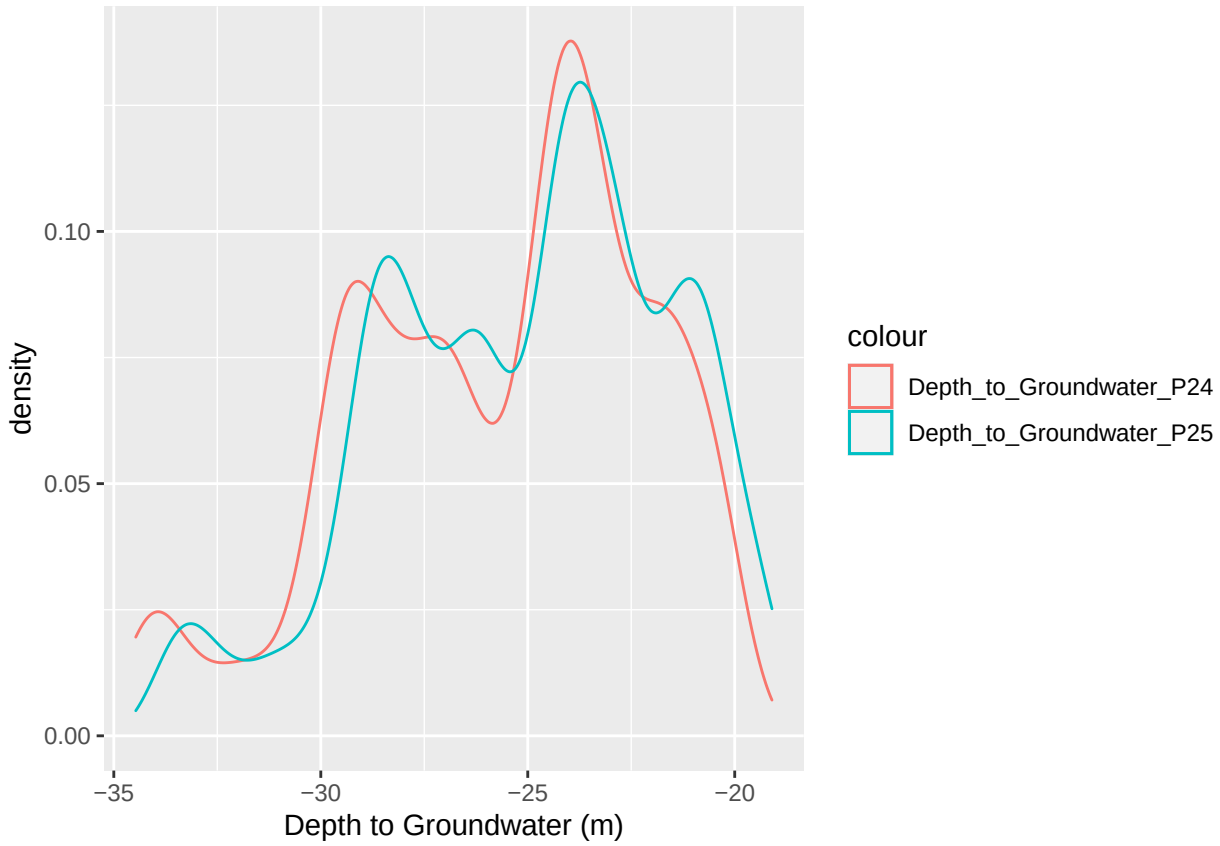

Figure 4: Comparison of Depth_to_Groundwater_{P24, P25} distributions

3.1.2 `Depth_to_Groundwater_P24` and `Depth_to_Groundwater_P25` trought the years

Another important relation to explore is the behavior of the depth to groundwater w.r.t the years. For example from **Figure 5** it is clear that 2012 can be considered an outlier year, but only in theoretical sense. The difference in the depth that happened in 2012 is a clear indication of the importance and impact of some features, which we will see why in the following sections.

```
p1 <- ggplot(train, aes(x=Year, y=Depth_to_Groundwater_P24)) +
    geom_boxplot(fill=5, alpha=0.2) +
    xlab("Year")

p2 <- ggplot(train, aes(x=Year, y=Depth_to_Groundwater_P25)) +
    geom_boxplot(fill=2, alpha=0.2) +
    xlab("Year")

grid.arrange(p1, p2, nrow = 1)
```

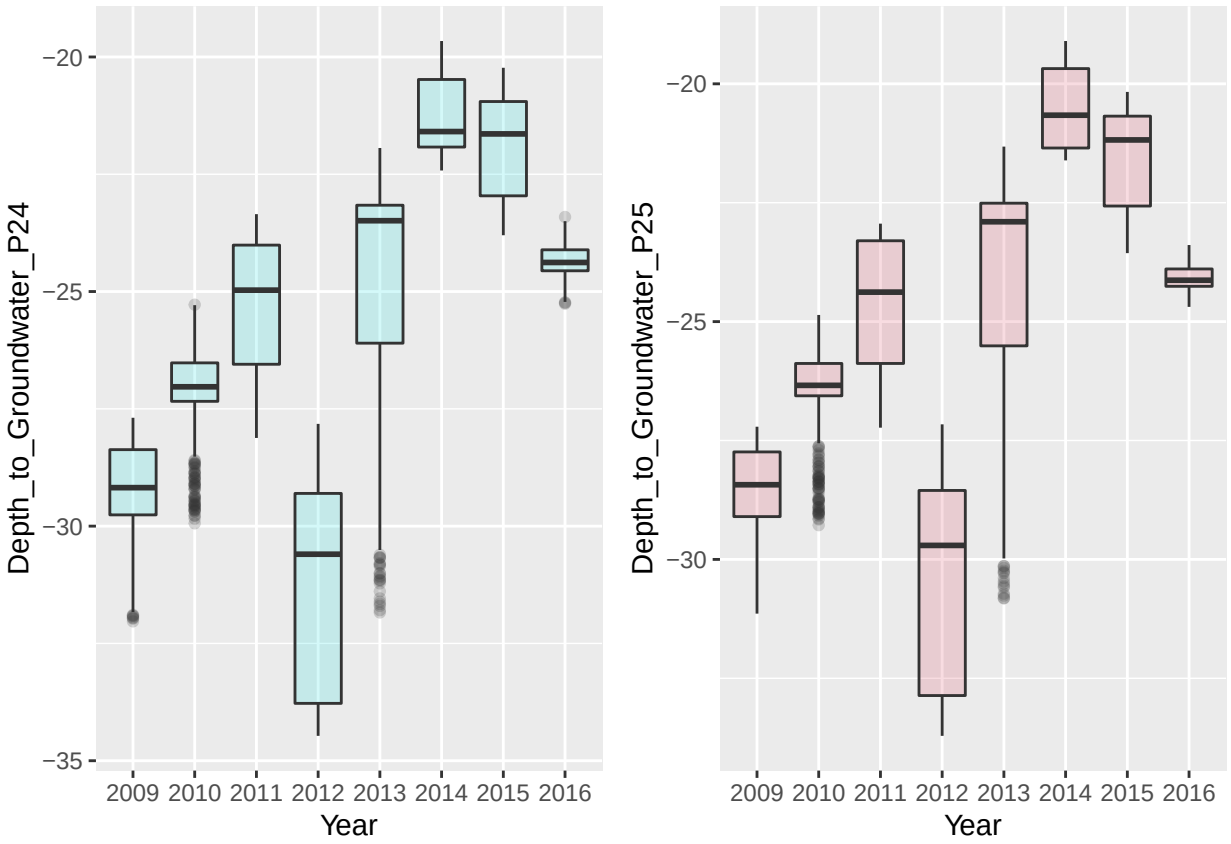

Figure 5: Boxplots Depth_to_Groundwater_{P24, P25} per Year

3.1.3 Depth_to_Groundwater_P24 and Depth_to_Groundwater_P25 trought the months

Furthermore what can be useful is to understand the behavior of the depth per each month. In order to return a proper summary, each month is represented by the mean of all the depths measured through the years.

```
#Depth to Groundwater P24
train_month_24 <- train %>%
  group_by(Month) %>%
  summarise(Depth_to_Groundwater_P24= round(mean(Depth_to_Groundwater_P24), 2))

month_24 <- train_month_24 %>%
  ggplot(aes(x = Month, y = Depth_to_Groundwater_P24)) +
  geom_point(color = "blue") +
  scale_x_discrete(labels=as.numeric(unique(train$Month))) +
  labs(title = "Depth - Month - Year", x="Month", y="Depth P25") + theme_bw(base_size = 15) +
  geom_line(aes(as.numeric(train_month_24$Month), Depth_to_Groundwater_P24), colour='orange')

#Depth to Groundwater P25
train_month_25 <- train %>%
  group_by(Month) %>%
  summarise(Depth_to_Groundwater_P25= round(mean(Depth_to_Groundwater_P25), 2))

month_25 <- train_month_25 %>%
  ggplot(aes(x = Month, y = Depth_to_Groundwater_P25)) +
  geom_point(color = "blue") +
```

```
  scale_x_discrete(labels=as.numeric(unique(train$Month))) +
  labs(title = "Depth - Month - Year", x="Month", y="Depth P25") + theme_bw(base_size = 15) +
  geom_line(aes(as.numeric(train_month_25$Month), Depth_to_Groundwater_P25), colour='orange')

grid.arrange(month_24, month_25)
```

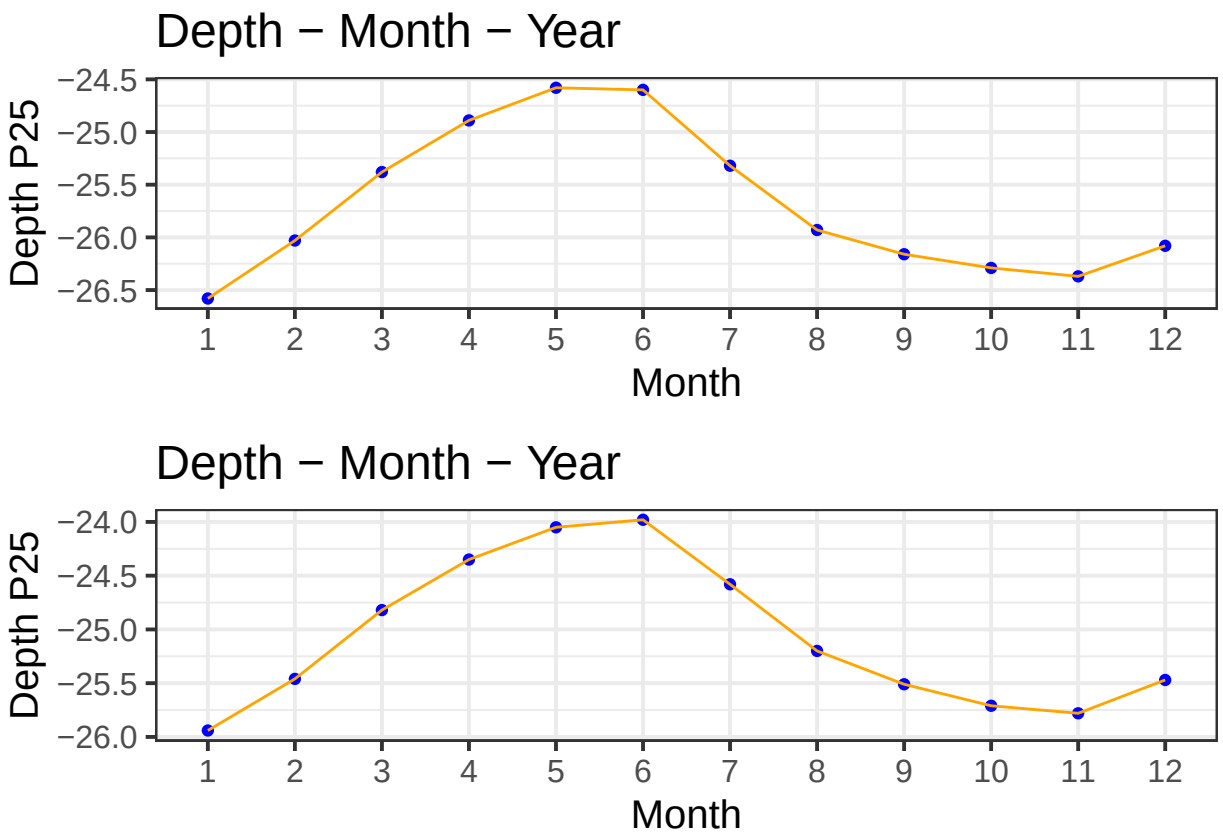

Figure 6: Mean of each month through the years of Depth_to_Groundwater_{P24, P25} distributions

It's easy to observe from **Figure 6** that both distributions tend to decrease its depths to groundwater until the first half of the year, more or less during the period of the end of the winter and the entire spring, when the weather becomes warmer. As the summer begins, the depth increases for the rest of the year with a slight decrease in the last month. What comes out is that the relationship between the depth and the seasons is crucial. This will also be discussed later when we use season as one of the predictors to asses the water level direction.

3.1.4 Depth_to_Groundwater_P24 and Depth_to_Groundwater_P25 trought the seasons

Observing the data when divided by years and months, we conclude what was already obvious: this type of problem and data is strongly dependent of all year rounds weather patterns. This led us to explore also the behavior of the data when divided into seasons. In the following we do the separation using the previously defined months to create our season column.

```
p1 <- ggplot(train) +
  aes(x=Date, y=Depth_to_Groundwater_P24, colour=season) +
  geom_point()+
  ggtitle("Depth to GroundWater p24 through the seasons")


p2 <- ggplot(train) +
```

```
  aes(x=Date, y=Depth_to_Groundwater_P25, colour=season) +
  geom_point()+
  ggtitle("Depth to GroundWater p25 through the seasons")

grid.arrange(p1, p2, ncol=1)
```

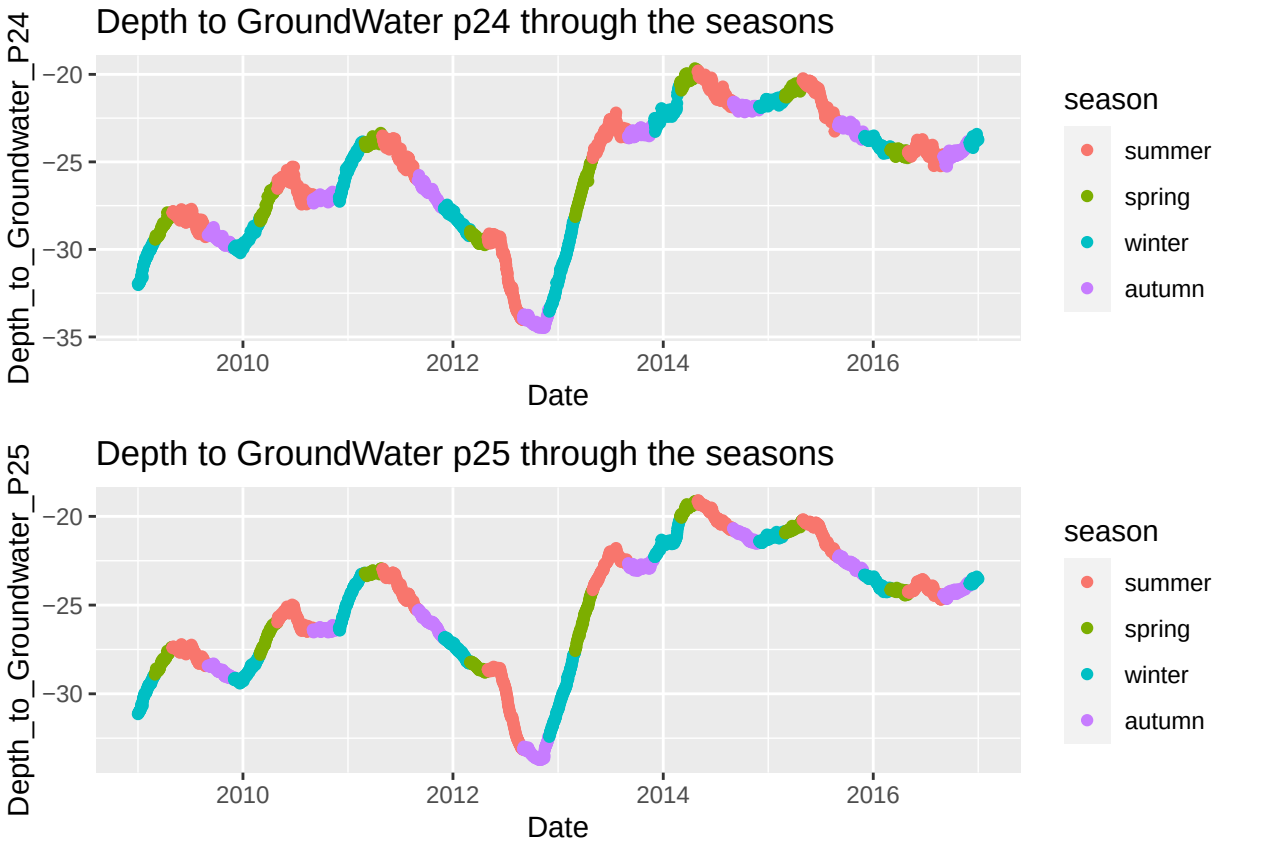

Figure 7: The behavior of Depth_to_Groundwater_{P24, P25} through the seasons

From **Figure 7** we can observe an interesting pattern. Primarily let us define that when we speak about the direction that our data follows, we mean the following: when our values become less negative, we declare a positive direction. In plain terms this is due to the fact that when the values are closer to 0, it means our aquifer has more water.

For winter and spring periods we recognize mostly a rise in our values which consequently means that we have a positive direction, while for summer in general the case is opposite. However, another amusing part of the plot is the purple representation which indicates the autumn period. In this case we see that the data is more concentrated and not very spread, which leads us to believe that not a lot of changes occur in the depth during this period. But let us continue absorbing the behavior with respect to the seasons.

3.2 Exploring temperature columns

We assume that temperature plays a big role in assessing the current levels of water. Before studying it's relationship we are going to study both temperature features `Temperature_Bastia_Umbra` and `Temperature_Petrignano` to understand their differences and distributions.

3.2.1 How does Temperature_Bastia_Umbra and Temperature_Petrignano differ from each other?

```
p1 <- ggplot(train, aes(x=Temperature_Bastia_Umbra)) +
  geom_histogram(aes(y = ..density..),
                 colour = 1, fill = "white") +
  geom_density(lwd = 1, colour = 5,
               fill = 5, alpha = 0.25)


p2 <- ggplot(train, aes(x=Temperature_Petrignano)) +
  geom_histogram(aes(y = ..density..),
                 colour = 1, fill = "white") +
  geom_density(lwd = 1, colour = 2,
               fill = 2, alpha = 0.25)

grid.arrange(p1, p2, nrow = 1)
```

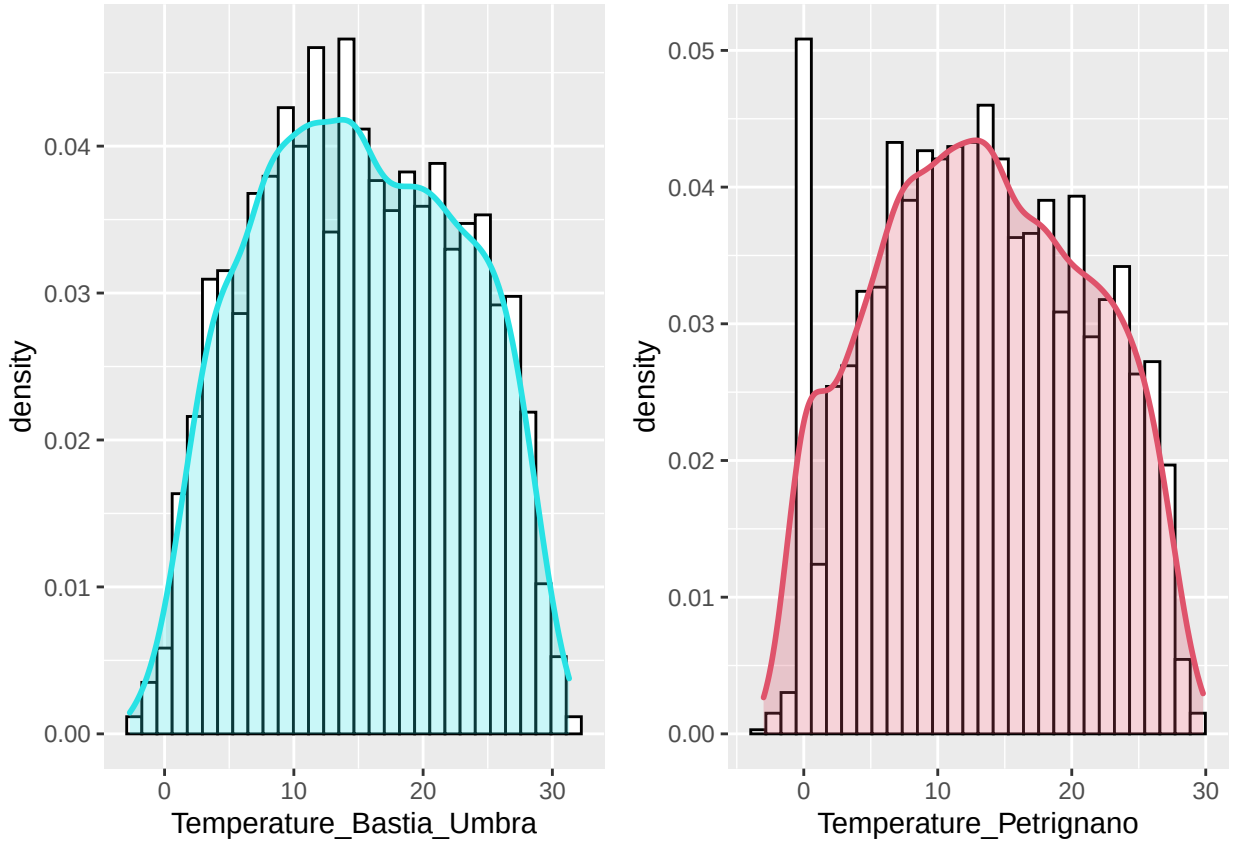

Figure 8: Distribution of Tempeture in Bastia Umbra and Petrignano

```
p1 <- ggplot(train) +
  geom_density(aes(x=Temperature_Bastia_Umbra, color="Temperature_Bastia_Umbra")) +
  geom_density(aes(x=Temperature_Petrignano, color="Temperature_Petrignano")) +
  xlab("Temperature columns [ºC]")

p1
```

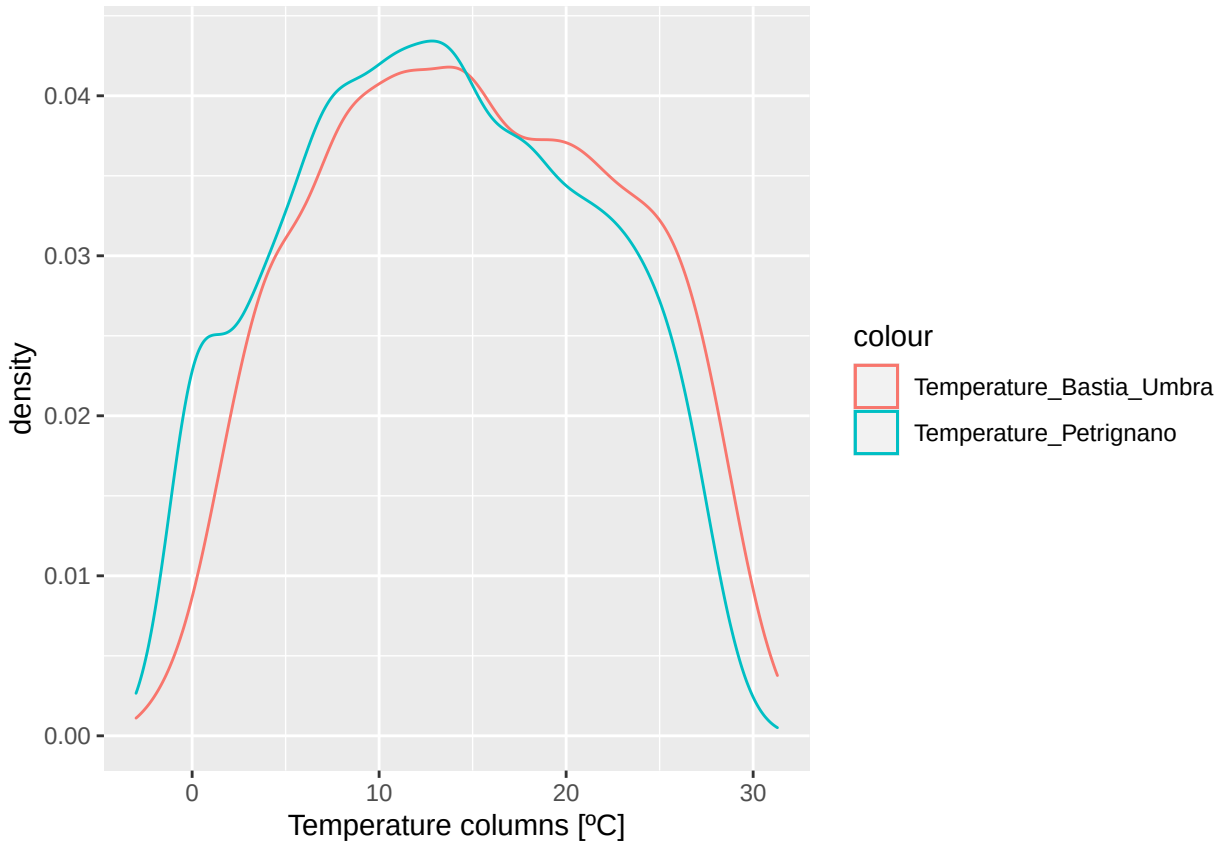

Figure 9: Comparison of Temperature distributions

From **Figure 8** and **Figure 9** we see that temperature in Bastia Umbra and Petrignano are not that different. This makes sense since these two points are only approximately just 5 km apart from each other.

3.2.2 How much do `Temperature_Bastia_Umbra` and `Temperature_Petrignano` varies per year?

```
p1 <- ggplot(train, aes(x=Year, y=Temperature_Bastia_Umbra)) +
    geom_boxplot(fill="slateblue", alpha=0.2) +
    xlab("Year")

p2 <- ggplot(train, aes(x=Year, y=Temperature_Petrignano)) +
    geom_boxplot(fill="slateblue", alpha=0.2) +
    xlab("Year")

grid.arrange(p1, p2, nrow = 1)
```

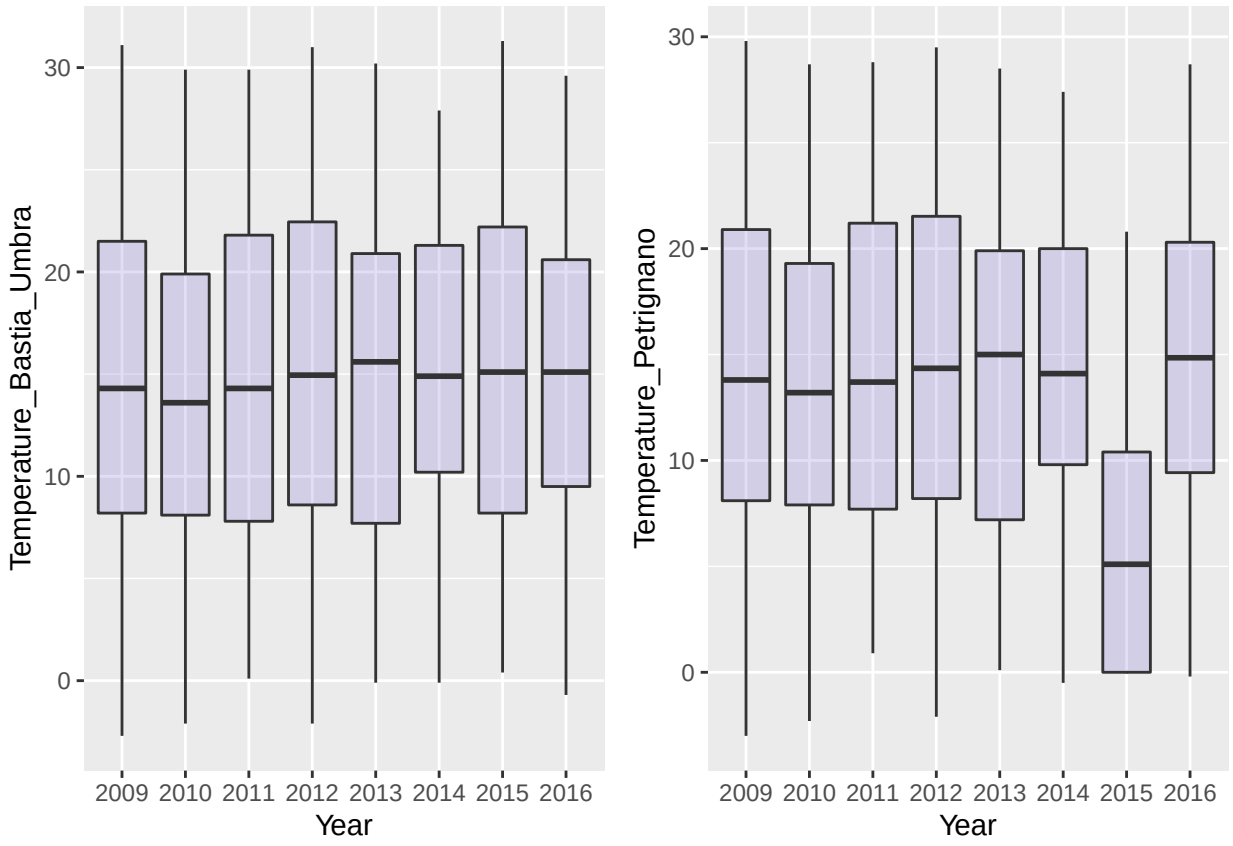

Figure 10: Summaries Bastia Umbra and Petrignano temperatures per Year

Temperatures seems to have a similar behavior over the past years. Except for Petrignano that in 2015 we see a sudden drop in temperature. This anomaly doesn't seems right since we don't see the same drop for Bastia Umbra.

Plotting the temperature in Petrignano for 2015 we see (**Figure 11**) that a third of the data points registered a temperature of 0 ºC. We think this is more related to a data feeding problem perhaps due to sensors malfunctioning than to a real change in temperatures. For this reason and due to the fact that Bastia Umbra is a good estimator of Petrignano temperature we will no longer consider `Temperature_Petrignano` as one of the predictors for the direction of `Depth_to_Groundwater_{P24, P25}`

```
train_2015 <- train %>% filter(Year == 2015)
p1 <- ggplot(train_2015, aes(x=Date, y=Temperature_Petrignano)) +
        geom_line()

p1
```

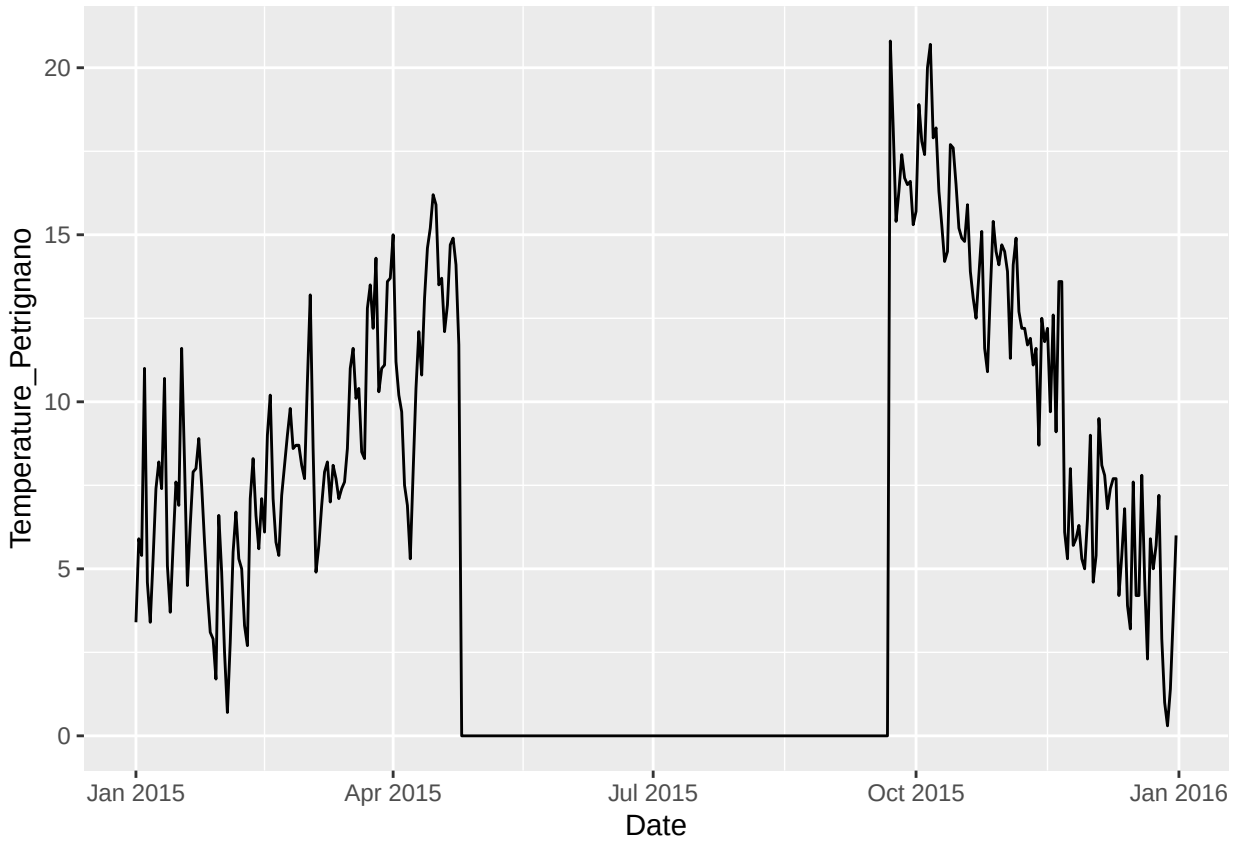

Figure 11: Petrignano Temperature for 2015

3.2.3 Relationship between `Depth_to_Groundwater` and `Temperature`

When we analyse the relationship between depth to ground water and temperature as a whole we don't find any clear pattern between the two. As we can see in **Figure 12**.

```
p1 <-
  ggplot(train, aes(x=Depth_to_Groundwater_P24, y=Temperature_Bastia_Umbra)) +
    geom_point()
p1
```

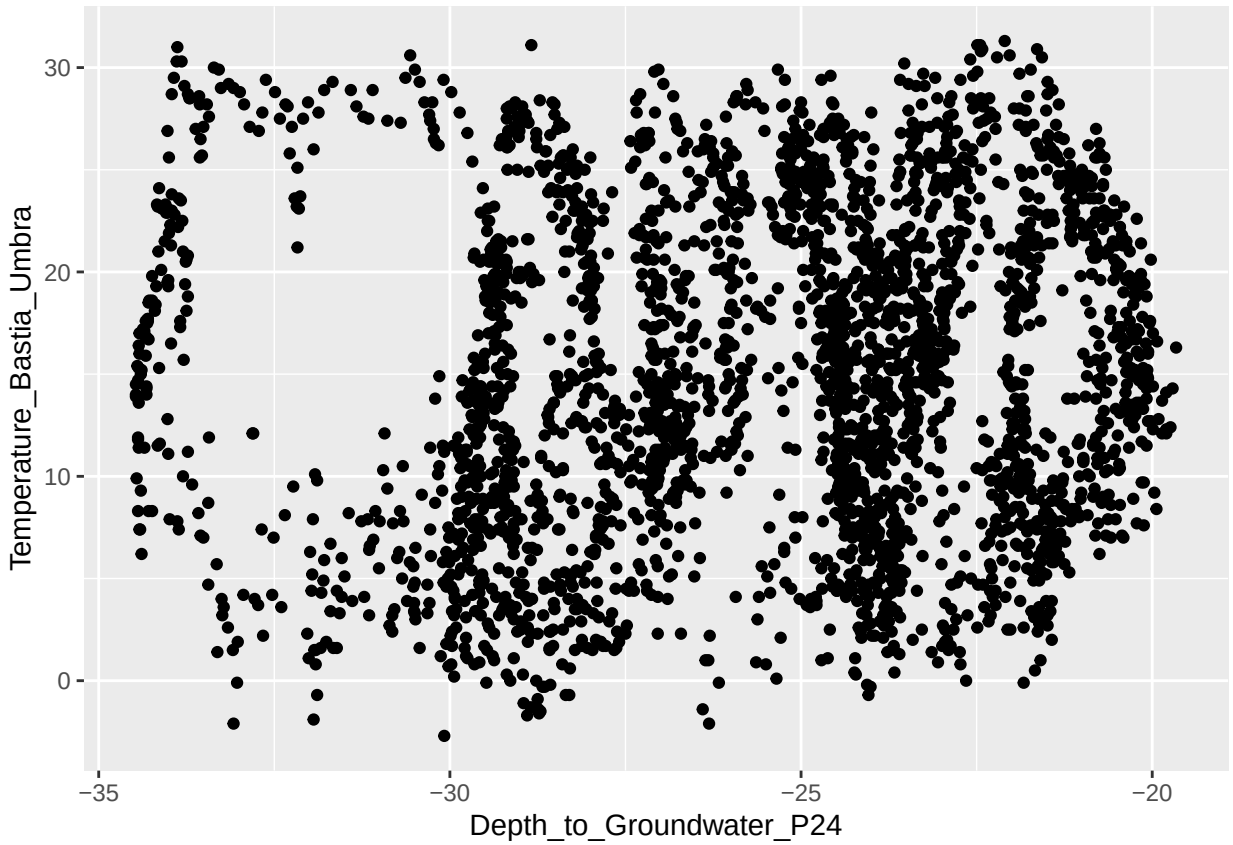

Figure 12: Depth to groundwater against temperature in Bastia Umbra

On the other hand when analyzing the relationship independently per year (**Figure 13**) for most of the cases we see a pattern saying that when the temperature increases depth to groundwater also tends to increase.

```
p1 <-
  ggplot(train, aes(x=Depth_to_Groundwater_P24, y=Temperature_Bastia_Umbra, color=Year)) +
    geom_point()
p1 <- p1 + facet_grid(rows = vars(Year), scales = "free")

p2 <-
  ggplot(train, aes(x=Depth_to_Groundwater_P25, y=Temperature_Bastia_Umbra, color=Year)) +
    geom_point()
p2 <- p2 + facet_grid(rows = vars(Year), scales = "free")


grid.arrange(p1, p2, nrow = 1)
```

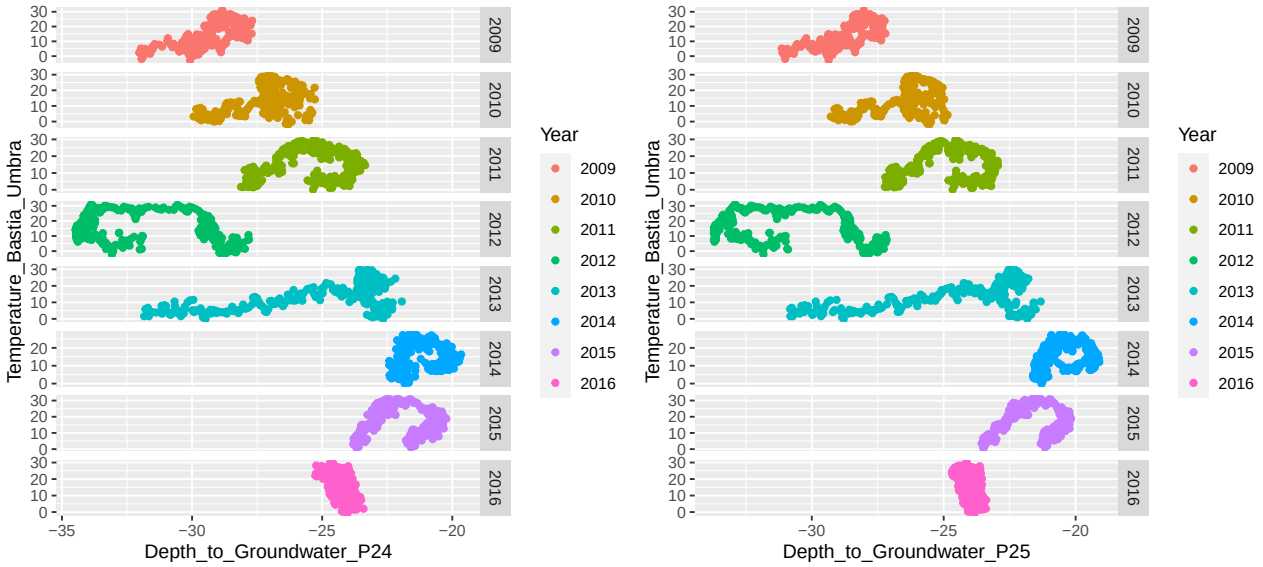

Figure 13: Depth to groundwater against temperature in Bastia Umbra per year

Some exceptions to the rule are seen for 2012 and 2016 where the trend is in the opposite direction (when temperature decreases depth of groundwater increases). This is an indicator that temperature alone is not enough to tell the behavior of depth to groundwater.

Let's move forward and perform a similar analysis with `Rainfall_Bastia_Umbra`, another important feature that may affect the groundwater levels.

3.3 Exploring Rainfall in Basta Umbra

```
p1 <- ggplot(train, aes(x=Date, y=Rainfall_Bastia_Umbra)) +
        geom_line()

p2 <- ggplot(train, aes(x=Rainfall_Bastia_Umbra)) +
  geom_histogram(aes(y = ..density..),
                 colour = 1, fill = "white") +
  geom_density(lwd = 1, colour = 5,
               fill = 5, alpha = 0.25)

grid.arrange(p1, p2, nrow=1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

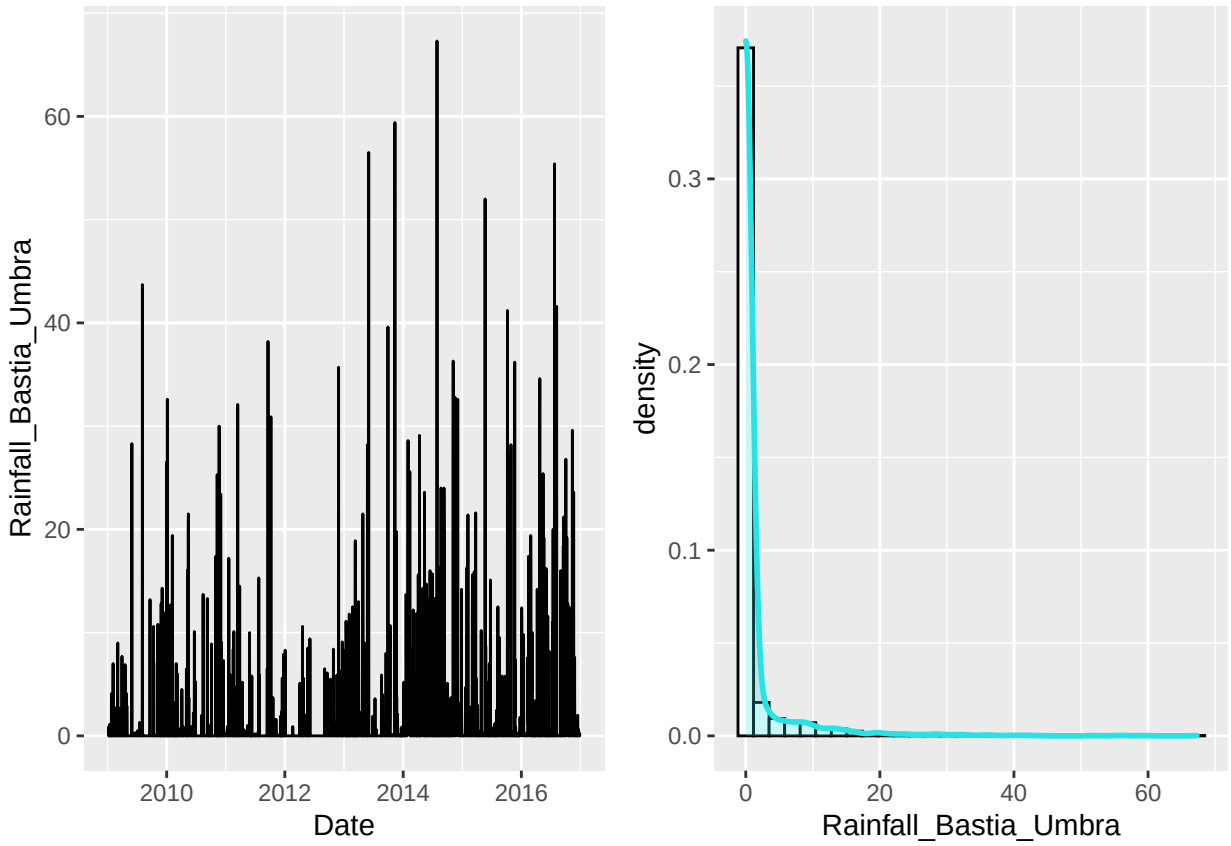

Figure 14: (left) : Rainfall in Bastia Umbra over time, (right): Distribution of Rainfall in Bastia Umbra

In **Figure 14** we see a very right skew distribution for Rainfall. Meaning that during the analysed period most of the days did not rained. Let's explore if we can tell any pattern of how this affected the depth to ground water.

3.3.1 Relationship between `Depth_to_Groundwater` and `Rainfall`

```
rainfall <- data.frame(train$Rainfall_Bastia_Umbra)

a=min(train$Depth_to_Groundwater_P24)
b=max(train$Rainfall_Bastia_Umbra)
ggplot() +
  geom_col(aes(x=train$Date, y=train$Rainfall_Bastia_Umbra), colour="orange") +
  geom_line(aes(x=train$Date, y=train$Depth_to_Groundwater_P24+50), colour="blue") +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  ggtitle("Depth and Rainfall") +
  xlab("Date") +
  ylim(c(a,b)) +
  scale_y_continuous(
    "depth (-m)",
    sec.axis = sec_axis(~ . * 1.20, name = "rainfall (mm)"))
```

```
## Scale for 'y' is already present. Adding another scale for 'y', which will
## replace the existing scale.
```

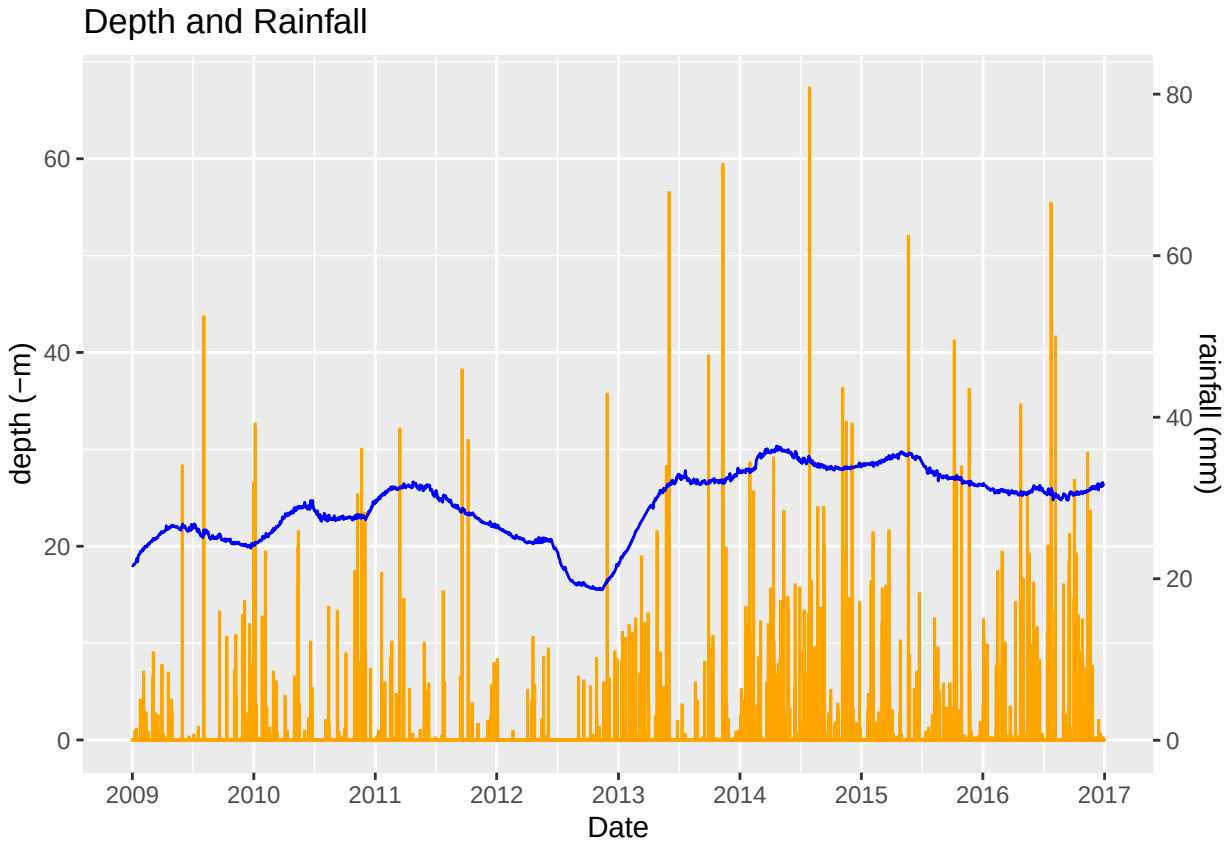

Figure 15: Groundwater depth and Rainfall trends

Although the scales are different (rainfall is in millimeters while the depth is in meters), the dependency between the variables looks nice. In **Figure 15** it's clear that in the first 5 years, so until 2014, the rainfall ratio is much more occasional that the recent years. If we look at the most recent years, the depth is steady but at an higher level if compared to the first ones, with the exception of the 2012-2013 year gap. In that last one, it's clear that the rainfall was very small or non existent for a big range of time (approximately around summer) leading after that to an increase of the depth (that means a decrease of the water capacity - hydrometry).

From logical thinking we are doing our analysis with the idea that the amount of rainfall directly affects the depth to groundwater values. For this purpose we focus a bit more on that relationship. In **Figure 16** we visualize the amount of days without any rainfall at all.

What is intriguing to mention is that most of the year 2012 there was no registered rainfall. And if we go back to the depth to groundwater analysis and **Figure 5**, we see that this information complements our findings. It gives us a direct dependency of the rainfall and depth to groundwater data. In the following plot where we visualize the amount of rainfall during the years, we notice an eminent jump from the year 2012 to 2013 and it continuous to the following years.

The last plot shows us the difference between rainfall values with respect to the seasons and this contributes to our findings plotted in **Figure 7**.

```
p1 <-
  ggplot(train %>% filter(Rainfall_Bastia_Umbra == 0), aes(x=Year)) +
    geom_bar() +
  ggtitle("Amount of days w/ 0 Rainfall") +
  labs(y = "Days (count)")

train_rain = train %>%
```

```
  group_by(Year) %>%
  summarize(Rainfall = sum(Rainfall_Bastia_Umbra))

train_rain_season = train %>%
  group_by(season) %>%
  summarize(Rainfall = sum(Rainfall_Bastia_Umbra))

p2 <-
  ggplot(train_rain, aes(x=Year, y=Rainfall)) +
    geom_col() +
  ggtitle("Rainfall (mm) per Year") +
  labs(y = "Rainfall (mm)")

p3 <-
  ggplot(train_rain_season, aes(x=season, y=Rainfall)) +
    geom_col() +
  ggtitle("Rainfall (mm) per season") +
  labs(y = "Rainfall (mm)")

grid.arrange(p1, p2, p3, ncol = 3)
```

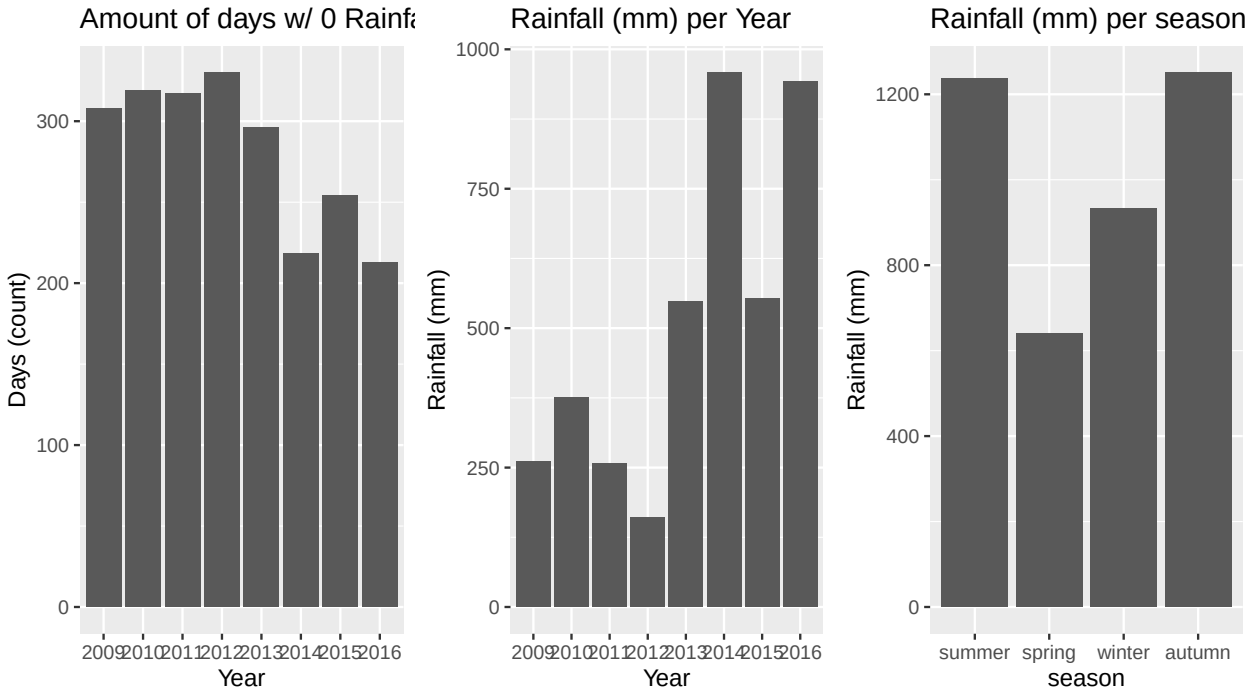

Figure 16:
(left): Amount of days without rain per year. (middle): mm of rainfall per year. (right): mm of rainfall per season

3.5 Correlations Between Features

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
par(mfrow=c(1,1))
C <- cor(train[colnames(train)[2:8]])
```

```
C2 <- C
colnames(C2) <- c("Rainfall", "Depth P24", "Depth P25",
                  "Temp Bastia", "Temp Petrignano",
                  "Volume", "Hydrometry")
rownames(C2) <- colnames(C2)
corrplot(10*C2, is.corr = FALSE, method="shade",
         type='lower', tl.col = 'black', tl.srt = 45,
         col = COL2('PuOr', 10), cl.ratio=.3, col.lim = c(-10, 10))
```

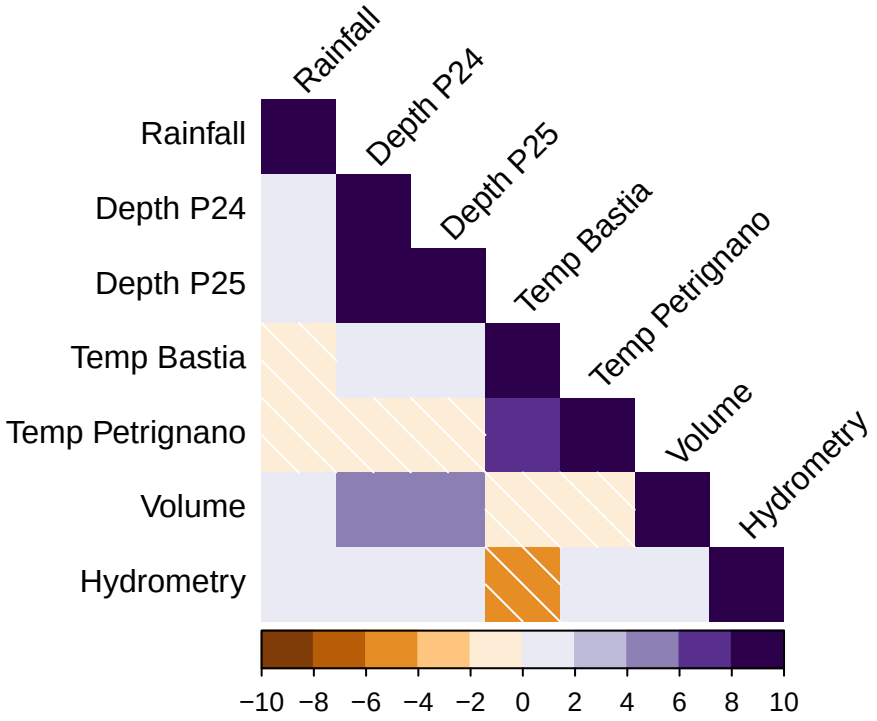

Figure 17: Correlation between features

In addition to the diagonal and trivial correlations (like depth 24 and 25 or temperature between the cities), there are some interesting results to look at. As expected, the volume and the depth (both 24 and 25 distributions) are positively correlated: if the depth is increasing it means that the aquifer is increasing the space for water and so we would have a bigger volume, viceversa if we have a decreasing depth.

3.5.1 Correlations Matrices through Months

As shown before, the selection by months it's useful to catch more correlations. By using the same statistics strategy done for the depth (the mean), the same will be evaluated for the other variables.

We then first create a new dataframe of 12 rows (months) and as many columns as many features we want to exploit.

```
rain_month <- train %>%
  group_by(Month) %>%
  summarise(Rainfall_Bastia_Umbra= round(sum(Rainfall_Bastia_Umbra)/96, 2))
#96 is the total number of months in the train set

hydro_month <- train %>%
  group_by(Month) %>%
  summarise(Hydrometry_Fiume_Chiascio_Petrignano=
              round(mean(Hydrometry_Fiume_Chiascio_Petrignano), 2))
```

```r
vol_month <- train %>%
  group_by(Month) %>%
  summarise(Volume_C10_Petrignano= round(mean(Volume_C10_Petrignano), 2))

temp_month <- train %>%
  group_by(Month) %>%
  summarise(Temperature_Bastia_Umbra= round(mean(Temperature_Bastia_Umbra), 2))

monthly_df <- data.frame(Month= train_month_24[1],
                         Depth_to_Groundwater_P24=train_month_24[2],
                         Depth_to_Groundwater_P25=train_month_25[2],
                         Rainfall_Bastia_Umbra=rain_month[2],
                         Hydrometry_Fiume_Chiascio_Petrignano=hydro_month[2],
                         Volume_C10_Petrignano=vol_month[2],
                         Temperature_Bastia_Umbra=temp_month[2])
```

```r
D <- cor(monthly_df[colnames(monthly_df)[-1]])
D2 <- D
colnames(D2) <- c("Depth P24", "Depth P25", "Rainfall",
                  "Hydrometry", "Volume",
                  "Temp Bastia")
rownames(D2) <- colnames(D2)
corrplot(10*D2, is.corr = FALSE, method="shade",
         type='lower', tl.col = 'black', tl.srt = 45,
         col = COL2('PuOr', 10), cl.ratio=.3, col.lim = c(-10, 10))
```

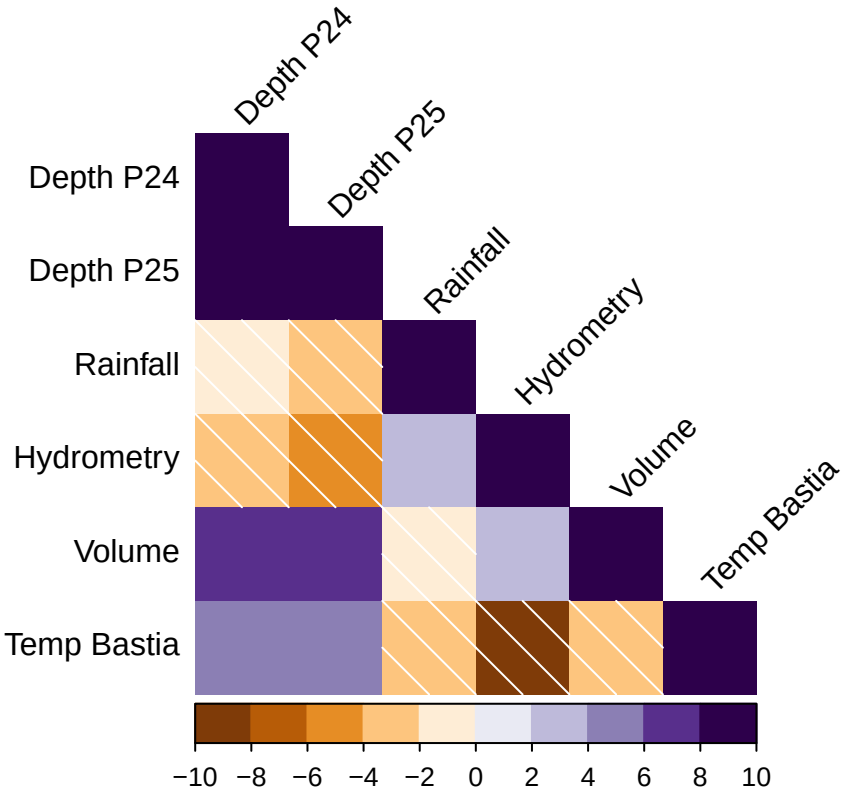

Figure 18: Correlation between features when agregated by month

Compared to the previous point, the month matrix in **Figure 18** tells us more information about the

variables. We can see that the temperature and the depth are properly correlated through each other, even if we already explained how this is not enough to forecast the problem. Another slight correlation is given by the hydrometry and the rainfall, that could be useful to understand the variables and their influences to each other.

3.5.2 Correlations Matrices through Seasons

In addition we explore the correlations w.r.t the seasons. For this purpose we first create a new - seasons data frame.

```
train_season24 <- train %>%
  group_by(season) %>%
  summarise(Depth_to_Groundwater_P24= round(mean(Depth_to_Groundwater_P24), 2))

train_season25 <- train %>%
  group_by(season) %>%
  summarise(Depth_to_Groundwater_P25= round(mean(Depth_to_Groundwater_P25), 2))

rain_season <- train %>%
  group_by(season) %>%
  summarise(Rainfall_Bastia_Umbra= round(sum(Rainfall_Bastia_Umbra)/4, 2))



hydro_season <- train %>%
  group_by(season) %>%
  summarise(Hydrometry_Fiume_Chiascio_Petrignano=
              round(mean(Hydrometry_Fiume_Chiascio_Petrignano), 2))

vol_season <- train %>%
  group_by(season) %>%
  summarise(Volume_C10_Petrignano= round(mean(Volume_C10_Petrignano), 2))

temp_season <- train %>%
  group_by(season) %>%
  summarise(Temperature_Bastia_Umbra= round(mean(Temperature_Bastia_Umbra), 2))

season_df <- data.frame(season = train_season24[1],
                        Depth_to_Groundwater_P24=train_season24[2],
                        Depth_to_Groundwater_P25=train_season25[2],
                        Rainfall_Bastia_Umbra=rain_season[2],
                        Hydrometry_Fiume_Chiascio_Petrignano=hydro_season[2],
                        Volume_C10_Petrignano=vol_season[2],
                        Temperature_Bastia_Umbra=temp_season[2])

correlation <- cor(season_df[colnames(season_df)[-1]])
C1 <- correlation
colnames(C1) <- c("Depth P24","Depth P25", "Rainfall",
                  "Hydrometry", "Volume",
                  "Temp Bastia")
rownames(C1) <- colnames(C1)
corrplot(10*C1, is.corr = FALSE, method="shade",
         type='lower', tl.col = 'black', tl.srt = 45,
         col = COL2('RdBu', 10), cl.ratio=.3, col.lim = c(-10, 10))
```

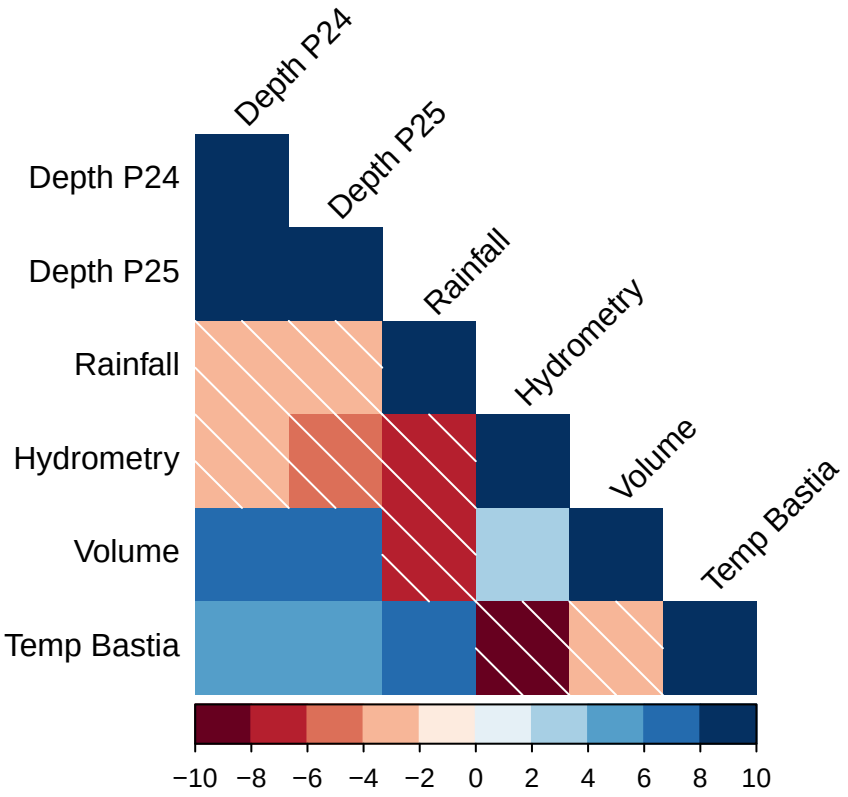

Figure 19: Correlation between features when agregated by seasons

4. Experiments

In this section we will experiment first with a (dummy) baseline that always predicts the day before direction. In theory any other more complex model should perform better than this simple heuristic. As a second type of model we will experiment with logistic regression using different predictors. Such as only waterbody related features (temperature, rainfall, volume, hydrometry), addition of lag variables and season. For every model we will track it's accuracy and F1-score.

Before experimenting we will create the lag variables and target column (`Direction_{P24, P25}`) for the validation set.

```
# extra features for validation set
val <- create_lag(val, "Depth_to_Groundwater_P24", 1)
val <- create_lag(val, "Depth_to_Groundwater_P25", 1)

val <- create_lag(val, "Depth_to_Groundwater_P24", 2)
val <- create_lag(val, "Depth_to_Groundwater_P25", 2)

val <- create_direction(val, "Depth_to_Groundwater_P24")
val <- create_direction(val, "Depth_to_Groundwater_P25")

val <- create_season(val)
```

4.1 Baseline: Using previous day direction as prediction

1. Predicting Direction_P24

From the output below we can see that using the day before direction to predict `Direction_P24` is not good. We could get even better results by just random guessing.

```
library(caret)
```

```
## Loading required package: lattice
```

```
val_lag <- create_lag(data=val, col_name="Direction_P24", step=1)
baseline_preds <- val_lag$lag_1_Direction_P24
confusionMatrix(as.factor(baseline_preds),
                as.factor(val$Direction_P24),
                mode = "everything",
                positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 225 184
##          1 184 136
##
##                Accuracy : 0.4952
##                  95% CI : (0.4583, 0.5321)
##     No Information Rate : 0.561
##     P-Value [Acc > NIR] : 0.9998
##
##                   Kappa : -0.0249
##
##  Mcnemar's Test P-Value : 1.0000
##
##             Sensitivity : 0.4250
##             Specificity : 0.5501
##          Pos Pred Value : 0.4250
##          Neg Pred Value : 0.5501
##               Precision : 0.4250
##                  Recall : 0.4250
##                      F1 : 0.4250
##              Prevalence : 0.4390
##          Detection Rate : 0.1866
##    Detection Prevalence : 0.4390
##       Balanced Accuracy : 0.4876
##
##        'Positive' Class : 1
##
```

2. Predicting Direction_P25 The baseline heuristic works a bit better for `Direction_P25` than it did for `Direction_P24`. But still the results are very close to random guessing. These two basic models are going to be our baselines to measure against the future more sophisticated models.

```
val_lag <- create_lag(data=val, col_name="Direction_P25", step=1)
baseline_preds <- val_lag$lag_1_Direction_P25
confusionMatrix(as.factor(baseline_preds),
                as.factor(val$Direction_P24),
                mode = "everything",
                positive = "1")
```

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction   0   1
##          0 245 183
##          1 164 137
##
##                Accuracy : 0.524
##                  95% CI : (0.487, 0.5608)
##     No Information Rate : 0.561
##     P-Value [Acc > NIR] : 0.9797
##
##                   Kappa : 0.0273
##
##  Mcnemar's Test P-Value : 0.3339
##
##             Sensitivity : 0.4281
##             Specificity : 0.5990
##          Pos Pred Value : 0.4551
##          Neg Pred Value : 0.5724
##               Precision : 0.4551
##                  Recall : 0.4281
##                      F1 : 0.4412
##              Prevalence : 0.4390
##          Detection Rate : 0.1879
##    Detection Prevalence : 0.4129
##       Balanced Accuracy : 0.5136
##
##        'Positive' Class : 1
##
```

4.2 Logistic Regression

4.2.1 Model 1: Predicting Direction using only river features

For these two experiments we will only use `Rainfall_Bastia_Umbra`, `Temperature_Bastia_Umbra` and `Volume_C10_Petrignano`. No lag or season variables are added here.

1. Predicting Direction_P24

Looking at the p-values of the output below we see that all the variables can be considered significant. Being `Rainfall_Bastia_Umbra` the least significant one with a p-value of 0.00186.

We also achieved a better performance than the baseline. With an accuracy of 0.609 and an F1-score of 0.591.

```
glm.fits <- glm (Direction_P24 ~
                   Rainfall_Bastia_Umbra +
                   Temperature_Bastia_Umbra +
                   Volume_C10_Petrignano,
                 data = train, family = binomial)
summary(glm.fits)
```

```
##
## Call:
## glm(formula = Direction_P24 ~ Rainfall_Bastia_Umbra + Temperature_Bastia_Umbra +
##     Volume_C10_Petrignano, family = binomial, data = train)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -2.4037  -1.0890  -0.7228   1.1343   1.8524
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                3.502e+00  2.945e-01  11.892  < 2e-16 ***
## Rainfall_Bastia_Umbra      2.674e-02  8.592e-03   3.112  0.00186 **
## Temperature_Bastia_Umbra -3.522e-02  5.035e-03  -6.995 2.65e-12 ***
## Volume_C10_Petrignano      1.066e-04  9.632e-06  11.065  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4040.6  on 2921  degrees of freedom
## Residual deviance: 3816.3  on 2918  degrees of freedom
## AIC: 3824.3
##
## Number of Fisher Scoring iterations: 4
```

```
library(recipes)
```

```
##
## Attaching package: 'recipes'
```

```
## The following object is masked from 'package:stats':
##
##     step
```

```
glm.probs <- predict(glm.fits, val, type="response")
glm.pred <- rep (0, 730)
glm.pred[glm.probs > .5] = 1

confusionMatrix(as.factor(glm.pred),
                as.factor(val$Direction_P24),
                mode = "everything",
                positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 239 114
##          1 171 206
##
##                Accuracy : 0.6096
##                  95% CI : (0.5731, 0.6452)
##     No Information Rate : 0.5616
##     P-Value [Acc > NIR] : 0.0048852
##
##                   Kappa : 0.2223
##
##  Mcnemar's Test P-Value : 0.0009094
##
##             Sensitivity : 0.6438
##             Specificity : 0.5829
```

```
##          Pos Pred Value : 0.5464
##          Neg Pred Value : 0.6771
##              Precision : 0.5464
##                 Recall : 0.6438
##                     F1 : 0.5911
##             Prevalence : 0.4384
##         Detection Rate : 0.2822
##    Detection Prevalence : 0.5164
##       Balanced Accuracy : 0.6133
##
##        'Positive' Class : 1
##
```

2. Predicting Direction_P25

Let's perform a similar analysis but this time with the `Direction_P25` target.

Looking at the p-values of the output below we see that all the variables can be considered significant. For `Direction_P25`, `Rainfall_Bastia_Umbra` seems to be more significant than for `Direction_P25`.

We also achieved a better performance than the baseline. With an accuracy of 0.630 and an F1-score of 0.587.

```
glm.fits <- glm (Direction_P25 ~
                   Rainfall_Bastia_Umbra +
                   Temperature_Bastia_Umbra +
                   Volume_C10_Petrignano,
                 data = train, family = binomial)
summary(glm.fits)
```

```
##
## Call:
## glm(formula = Direction_P25 ~ Rainfall_Bastia_Umbra + Temperature_Bastia_Umbra +
##     Volume_C10_Petrignano, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4268  -1.0305  -0.6349   1.1088   2.0325
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 3.953e+00  3.048e-01  12.968  < 2e-16 ***
## Rainfall_Bastia_Umbra       3.829e-02  9.185e-03   4.169 3.06e-05 ***
## Temperature_Bastia_Umbra -5.573e-02  5.221e-03 -10.673  < 2e-16 ***
## Volume_C10_Petrignano       1.151e-04  9.911e-06  11.613  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4023.5  on 2921  degrees of freedom
## Residual deviance: 3692.8  on 2918  degrees of freedom
## AIC: 3700.8
##
## Number of Fisher Scoring iterations: 4
```

```
library(recipes)
```

```
glm.probs <- predict(glm.fits, val, type="response")
glm.pred <- rep (0, 730)
glm.pred[glm.probs > .5] = 1

confusionMatrix(as.factor(glm.pred),
                as.factor(val$Direction_P25),
                mode = "everything",
                positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 268 109
##          1 161 192
##
##                Accuracy : 0.6301
##                  95% CI : (0.594, 0.6653)
##     No Information Rate : 0.5877
##     P-Value [Acc > NIR] : 0.010614
##
##                   Kappa : 0.256
##
##  Mcnemar's Test P-Value : 0.001911
##
##             Sensitivity : 0.6379
##             Specificity : 0.6247
##          Pos Pred Value : 0.5439
##          Neg Pred Value : 0.7109
##               Precision : 0.5439
##                  Recall : 0.6379
##                      F1 : 0.5872
##              Prevalence : 0.4123
##          Detection Rate : 0.2630
##    Detection Prevalence : 0.4836
##       Balanced Accuracy : 0.6313
##
##        'Positive' Class : 1
##
```

4.2.2 Model 2: Predicting direction using river features + lags

Let's see if by adding lag predictors we are able to improve the model performance.

Predicting Direction_P24

Looking a the p-values below both the lag_1 and lag_2 seems to be significant. We also got a better F1-score performance of 0.601

```
glm.fits <- glm (Direction_P24 ~ lag_1_Depth_to_Groundwater_P24 +
                   lag_2_Depth_to_Groundwater_P24 +
                   Rainfall_Bastia_Umbra +
                   Temperature_Bastia_Umbra +
                   Volume_C10_Petrignano,
                 data = train, family = binomial)
```

```
summary(glm.fits)
```

```
##
## Call:
## glm(formula = Direction_P24 ~ lag_1_Depth_to_Groundwater_P24 +
##     lag_2_Depth_to_Groundwater_P24 + Rainfall_Bastia_Umbra +
##     Temperature_Bastia_Umbra + Volume_C10_Petrignano, family = binomial,
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9271  -1.0600  -0.6838   1.1157   2.0853
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     2.606e+00  3.449e-01   7.554 4.22e-14 ***
## lag_1_Depth_to_Groundwater_P24 -1.540e+00  3.259e-01  -4.726 2.29e-06 ***
## lag_2_Depth_to_Groundwater_P24  1.453e+00  3.242e-01   4.481 7.43e-06 ***
## Rainfall_Bastia_Umbra           2.944e-02  8.815e-03   3.340 0.000839 ***
## Temperature_Bastia_Umbra       -3.290e-02  5.172e-03  -6.361 2.01e-10 ***
## Volume_C10_Petrignano           1.530e-04  1.177e-05  13.004  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4038.1  on 2919  degrees of freedom
## Residual deviance: 3752.6  on 2914  degrees of freedom
##   (2 observations deleted due to missingness)
## AIC: 3764.6
##
## Number of Fisher Scoring iterations: 4
```

```
glm.probs <- predict(glm.fits, val, type="response")
glm.pred <- rep (0, 730)
glm.pred[glm.probs > .5] = 1

confusionMatrix(as.factor(glm.pred),
                as.factor(val$Direction_P24),
                mode = "everything",
                positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 216  99
##          1 194 221
##
##                Accuracy : 0.5986
##                  95% CI : (0.562, 0.6344)
##     No Information Rate : 0.5616
##     P-Value [Acc > NIR] : 0.02375
##
```

```
##                   Kappa : 0.2106
##
##  Mcnemar's Test P-Value : 3.984e-08
##
##             Sensitivity : 0.6906
##             Specificity : 0.5268
##          Pos Pred Value : 0.5325
##          Neg Pred Value : 0.6857
##               Precision : 0.5325
##                  Recall : 0.6906
##                      F1 : 0.6014
##              Prevalence : 0.4384
##          Detection Rate : 0.3027
##    Detection Prevalence : 0.5685
##       Balanced Accuracy : 0.6087
##
##        'Positive' Class : 1
##
```

Predicting Direction_P25

Similar than the case for `Direction_P24` both the lag_1 and lag_2 seems to be significant. And this translates to a better F1-score of 0.6110.

```
glm.fits <- glm (Direction_P25 ~ lag_1_Depth_to_Groundwater_P25 +
                   lag_2_Depth_to_Groundwater_P25 +
                   Rainfall_Bastia_Umbra +
                   Temperature_Bastia_Umbra +
                   Volume_C10_Petrignano,
                 data = train, family = binomial)
summary(glm.fits)
```

```
##
## Call:
## glm(formula = Direction_P25 ~ lag_1_Depth_to_Groundwater_P25 +
##     lag_2_Depth_to_Groundwater_P25 + Rainfall_Bastia_Umbra +
##     Temperature_Bastia_Umbra + Volume_C10_Petrignano, family = binomial,
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6256  -1.0119  -0.6226   1.0901   2.0814
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    2.698e+00  3.593e-01   7.508 6.00e-14 ***
## lag_1_Depth_to_Groundwater_P25  2.832e+00  6.635e-01   4.269 1.96e-05 ***
## lag_2_Depth_to_Groundwater_P25 -2.905e+00  6.621e-01  -4.387 1.15e-05 ***
## Rainfall_Bastia_Umbra           3.883e-02  9.285e-03   4.182 2.89e-05 ***
## Temperature_Bastia_Umbra       -4.728e-02  5.387e-03  -8.777  < 2e-16 ***
## Volume_C10_Petrignano           1.385e-04  1.173e-05  11.804  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 4020.7  on 2919  degrees of freedom
## Residual deviance: 3635.9  on 2914  degrees of freedom
##   (2 observations deleted due to missingness)
## AIC: 3647.9
##
## Number of Fisher Scoring iterations: 4
```

```
glm.probs <- predict(glm.fits, val, type="response")
glm.pred <- rep (0, 730)
glm.pred[glm.probs > .5] = 1

confusionMatrix(as.factor(glm.pred),
                as.factor(val$Direction_P25),
                mode = "everything",
                positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 264  96
##          1 165 205
##
##                Accuracy : 0.6425
##                  95% CI : (0.6065, 0.6773)
##     No Information Rate : 0.5877
##     P-Value [Acc > NIR] : 0.001394
##
##                   Kappa : 0.2866
##
##  Mcnemar's Test P-Value : 2.564e-05
##
##             Sensitivity : 0.6811
##             Specificity : 0.6154
##          Pos Pred Value : 0.5541
##          Neg Pred Value : 0.7333
##               Precision : 0.5541
##                  Recall : 0.6811
##                      F1 : 0.6110
##              Prevalence : 0.4123
##          Detection Rate : 0.2808
##    Detection Prevalence : 0.5068
##       Balanced Accuracy : 0.6482
##
##        'Positive' Class : 1
##
```

4.2.3 Model 3: Predicting direction using river features + lags + season

Finally let's experiment with what we have build up from the previous models but this time adding the `season` predictor.

Predicting Direction_P24

When compared with the previous model without `season` we see an slightly increment of the F1-score from 0.601 to 0.618.

```r
glm.fits_P24 <- glm (Direction_P24 ~ lag_1_Depth_to_Groundwater_P24 +
                    lag_2_Depth_to_Groundwater_P24 +
                    Rainfall_Bastia_Umbra +
                    Temperature_Bastia_Umbra +
                    Volume_C10_Petrignano +
                    season,
                  data = train, family = binomial)
summary(glm.fits_P24)
```

```
##
## Call:
## glm(formula = Direction_P24 ~ lag_1_Depth_to_Groundwater_P24 +
##     lag_2_Depth_to_Groundwater_P24 + Rainfall_Bastia_Umbra +
##     Temperature_Bastia_Umbra + Volume_C10_Petrignano + season,
##     family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8998  -1.0652  -0.6847   1.1076   2.1138
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     2.235e+00  3.897e-01   5.736 9.72e-09 ***
## lag_1_Depth_to_Groundwater_P24 -1.573e+00  3.271e-01  -4.809 1.51e-06 ***
## lag_2_Depth_to_Groundwater_P24  1.484e+00  3.254e-01   4.559 5.14e-06 ***
## Rainfall_Bastia_Umbra           3.011e-02  8.821e-03   3.413 0.000642 ***
## Temperature_Bastia_Umbra       -2.093e-02  9.163e-03  -2.284 0.022365 *
## Volume_C10_Petrignano           1.551e-04  1.189e-05  13.041  < 2e-16 ***
## seasonspring                    4.932e-01  1.506e-01   3.275 0.001056 **
## seasonwinter                    2.245e-01  1.830e-01   1.227 0.219835
## seasonautumn                    2.045e-01  1.233e-01   1.659 0.097142 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4038.1  on 2919  degrees of freedom
## Residual deviance: 3739.7  on 2911  degrees of freedom
##   (2 observations deleted due to missingness)
## AIC: 3757.7
##
## Number of Fisher Scoring iterations: 4
```

```r
glm.probs <- predict(glm.fits_P24, val, type="response")
glm.pred <- rep (0, 730)
glm.pred[glm.probs > .5] = 1

confusionMatrix(as.factor(glm.pred),
                as.factor(val$Direction_P24),
                mode="everything",
                positive="1")
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction   0   1
##          0 220  92
##          1 190 228
##
##                Accuracy : 0.6137
##                  95% CI : (0.5773, 0.6492)
##     No Information Rate : 0.5616
##     P-Value [Acc > NIR] : 0.002477
##
##                   Kappa : 0.241
##
##  Mcnemar's Test P-Value : 7.638e-09
##
##             Sensitivity : 0.7125
##             Specificity : 0.5366
##          Pos Pred Value : 0.5455
##          Neg Pred Value : 0.7051
##               Precision : 0.5455
##                  Recall : 0.7125
##                      F1 : 0.6179
##              Prevalence : 0.4384
##          Detection Rate : 0.3123
##    Detection Prevalence : 0.5726
##       Balanced Accuracy : 0.6245
##
##        'Positive' Class : 1
##
```

Predicting Direction_P25

When compared with the previous model with out `season` we see an slightly decrease of the F1-score from 0.6110 to 0.6073.

```
glm.fits_P25 <- glm (Direction_P25 ~ lag_1_Depth_to_Groundwater_P25 +
                   lag_2_Depth_to_Groundwater_P25 +
                   Rainfall_Bastia_Umbra +
                   Temperature_Bastia_Umbra +
                   Volume_C10_Petrignano +
                   season,
                 data = train, family = binomial)
summary(glm.fits_P25)
```

```
##
## Call:
## glm(formula = Direction_P25 ~ lag_1_Depth_to_Groundwater_P25 +
##     lag_2_Depth_to_Groundwater_P25 + Rainfall_Bastia_Umbra +
##     Temperature_Bastia_Umbra + Volume_C10_Petrignano + season,
##     family = binomial, data = train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.601  -1.009  -0.620   1.088   2.086
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)                     2.396e+00  4.037e-01   5.936 2.92e-09 ***
## lag_1_Depth_to_Groundwater_P25  2.718e+00  6.651e-01   4.086 4.38e-05 ***
## lag_2_Depth_to_Groundwater_P25 -2.792e+00  6.636e-01  -4.208 2.58e-05 ***
## Rainfall_Bastia_Umbra           3.981e-02  9.291e-03   4.285 1.83e-05 ***
## Temperature_Bastia_Umbra       -3.814e-02  9.395e-03  -4.060 4.91e-05 ***
## Volume_C10_Petrignano           1.390e-04  1.183e-05  11.755  < 2e-16 ***
## seasonspring                    3.600e-01  1.516e-01   2.374   0.0176 *
## seasonwinter                    1.712e-01  1.855e-01   0.923   0.3559
## seasonautumn                    6.571e-02  1.262e-01   0.521   0.6027
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4020.7  on 2919  degrees of freedom
## Residual deviance: 3628.7  on 2911  degrees of freedom
##   (2 observations deleted due to missingness)
## AIC: 3646.7
##
## Number of Fisher Scoring iterations: 4
```

```
glm.probs <- predict(glm.fits_P25, val, type="response")
glm.pred <- rep (0, 730)
glm.pred[glm.probs > .5] = 1

confusionMatrix(as.factor(glm.pred),
                as.factor(val$Direction_P25),
                mode="everything",
                positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 269 100
##          1 160 201
##
##                Accuracy : 0.6438
##                  95% CI : (0.6079, 0.6786)
##     No Information Rate : 0.5877
##     P-Value [Acc > NIR] : 0.0010824
##
##                   Kappa : 0.2863
##
##  Mcnemar's Test P-Value : 0.0002532
##
##             Sensitivity : 0.6678
##             Specificity : 0.6270
##          Pos Pred Value : 0.5568
##          Neg Pred Value : 0.7290
##               Precision : 0.5568
##                  Recall : 0.6678
##                      F1 : 0.6073
##              Prevalence : 0.4123
##          Detection Rate : 0.2753
```

```
##    Detection Prevalence : 0.4945
##       Balanced Accuracy : 0.6474
##
##        'Positive' Class : 1
##
```

4.3 Final Predictions

In this section we will use our last model. To run the predictions on the previously mentioned hidden test set.

```
# extra features for test set
test <- create_lag(test, "Depth_to_Groundwater_P24", 1)
test <- create_lag(test, "Depth_to_Groundwater_P25", 1)

test <- create_lag(test, "Depth_to_Groundwater_P24", 2)
test <- create_lag(test, "Depth_to_Groundwater_P25", 2)

test <- create_direction(test, "Depth_to_Groundwater_P24")
test <- create_direction(test, "Depth_to_Groundwater_P25")

test <- create_season(test)
```

1. Predicting Direction_P24 on the Test set

Final predictions for Direction_P24

```
glm.probs <- predict(glm.fits_P24, test, type="response")
glm.pred <- rep (0, nrow(test))
glm.pred[glm.probs > .5] = 1

confusionMatrix(as.factor(glm.pred),
                as.factor(test$Direction_P24),
                mode="everything",
                positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  69  25
##          1 212 241
##
##                Accuracy : 0.5667
##                  95% CI : (0.524, 0.6087)
##     No Information Rate : 0.5137
##     P-Value [Acc > NIR] : 0.007306
##
##                   Kappa : 0.1488
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9060
##             Specificity : 0.2456
##          Pos Pred Value : 0.5320
##          Neg Pred Value : 0.7340
##               Precision : 0.5320
```

```
##                  Recall : 0.9060
##                      F1 : 0.6704
##              Prevalence : 0.4863
##          Detection Rate : 0.4406
##    Detection Prevalence : 0.8282
##       Balanced Accuracy : 0.5758
##
##        'Positive' Class : 1
##
```

2. Predicting Direction_P25 on the Test set

```
glm.probs <- predict(glm.fits_P25, test, type="response")
glm.pred <- rep (0, nrow(test))
glm.pred[glm.probs > .5] = 1

confusionMatrix(as.factor(glm.pred),
                as.factor(test$Direction_P25),
                mode="everything",
                positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  95  39
##          1 191 222
##
##                Accuracy : 0.5795
##                  95% CI : (0.5369, 0.6213)
##     No Information Rate : 0.5229
##     P-Value [Acc > NIR] : 0.004435
##
##                   Kappa : 0.1782
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8506
##             Specificity : 0.3322
##          Pos Pred Value : 0.5375
##          Neg Pred Value : 0.7090
##               Precision : 0.5375
##                  Recall : 0.8506
##                      F1 : 0.6588
##              Prevalence : 0.4771
##          Detection Rate : 0.4059
##    Detection Prevalence : 0.7550
##       Balanced Accuracy : 0.5914
##
##        'Positive' Class : 1
##
```

	Accuracy		F1-Score	
Model	Direction_P24	Direction_P25	Direction_P24	Direction_P25
Baseline	0.495	0.524	0.425	0.441
Model 1	0.601	0.630	0.587	0.587
Model 2	0.599	0.643	0.601	0.611
Model 3	0.612	0.644	0.618	0.607

Table 2: Results: Accuracy and F1-score from the validation set

	Accuracy (Test)		F1-Score (Test)	
Model	Direction_P24	Direction_P25	Direction_P24	Direction_P25
Model 3	0.567	0.580	0.6704	0.659

Table 3: Results: Accuracy and F1-score from the test set

5. Results

We represent our results in the **Table 2**. From there we can see that the Baseline model, which uses the previous day direction as prediction, is just slightly better than random guessing. When speaking about our 3 models, we contemplate that Model 2 and Model 3 have similar results for accuracy and F1 score. While Model 2 uses lags and information from our original data frame and Model 3 has seasons added to its combination. It is notable that these two models perform better than Model 1, which has as input only the lags and original data frame. From these findings we can conclude that the lags and seasons were an appropriate tool to asses the trend direction.

From our so far findings we decided to go forward with Model 3 and use it to run the predictions on the hidden test set. The results from this have been displayed in **Table 3**

6. Conclusion

In conclusion the task of predicting the groundwater level is not a simple one. Because of that, the task of the problem changed to the prediction of the trend direction of the water. We were able to obtain descent results (**Table 3**) with an logistic regression whose predictors contained information from the Petrignano aquifer, lag variables and seasons. We confirmed that temperature and the lack of rainfall plays a big role in the behavior of the groundwater levels. We also learned that the depth to groundwater trends are not stationary ones and because of this the task is a complex one.