Advanced SWAT Course Notes: Comparative Analysis of CHIRPS data and INUMET data

Willem Vervoort & Flora Mer

24-11-2017

Introduction

This is part of a series of teaching documents related to the "How do I use satellite and global reanalysis data for hydrological simulations in SWAT?" jointly organised by the University of Sydney, IRI (the University of Columbia) and INIA, Uruguay.

This part explains a comparative analysis between CHIRPS data and INUMET data to use as rainfall data.

INUMET Weather Station data

The daily precipitation data from weather stations were delivered by the National Institue of Meteoroglogy in Uruguay (INUMET).

Download INUMET data

The following script will show :

- how to load the INUMET rainfall data
- how to select the stations with good amount of data
- how to convert UTM coordinates to latitute and longitude used by SWAT
- how to create a unique dataframe with all information needed for further analysis

```
# Some useful packages to install
library(tidyverse)
library(zoo)
```

```
library(rgeos)
library(sp)
library(rgdal)
library(lubridate)
```

```
# Read in the INUMET station locations
Stations <- read.csv("Precipitacion_SantaLucia_inumet_coordinates.csv")

# Read in the precipitation data for each INUMET stations
Pdata <- read.csv("Precipitacion SantaLucia_inumet_stations_prcp.csv",
    na.strings = "NaN", header = T, nrows = 11322)


Dates2 <- as.Date(paste(Pdata[, 1], Pdata[, 2], Pdata[, 3], sep = "-"))

# Find the INUMET stations that have 90% of data after 2001
Pdata_2000 <- Pdata[Dates2 >= "2000-01-01", ]
Pdata_2000 <- Pdata_2000[-(1:5), ]
result <- rep(0, (ncol(Pdata_2000) - 3))
for (i in 4:ncol(Pdata_2000)) {
    result[i - 3] <- sum(ifelse(is.na(Pdata_2000[, i]), 1, 0))/nrow(Pdata_2000)
}

# result indicates the fraction of NA data for the stations
# Throw out all the columns and rows where result >0.1
Pdata_new <- Pdata_2000[, -(which(result > 0.1) + 3)]
Stations <- Stations[-which(result > 0.1), ]
Dates2000 <- Dates2[Dates2 >= as.Date("2000-01-01")]


# Zoo the Pdata (rainfall)
Pdata_z <- zoo(Pdata_new[, 4:ncol(Pdata_new)], order.by = Dates2000,
    frequency = 1)
head(Pdata_z)
```

```
##            X2673 X2826 X2819 X2748 X2725 X2715 X2683 X2680 X2670 X2498
## 2000-01-01     0     0     0     0     0     0     0     0     0     0
## 2000-01-02     0     0     0     0     0     0     0     0     0     0
## 2000-01-03     0     0     0     0     0     0     0     0     0     0
## 2000-01-04     0     0     0     0     0     0     0     0     0     0
## 2000-01-05     0     0     0     0     0     0     0     0     0     0
## 2000-01-06     0     0     0     0     0     0     0     0     0     0
##            X2549 X86545 X2588 X2632
## 2000-01-01     0      0     0     0
## 2000-01-02     0      0     0     0
## 2000-01-03     0      0     0     0
## 2000-01-04     0      0     0     0
## 2000-01-05     0      0     0     0
## 2000-01-06     0      0     0     0
```

```
# convert UTM to latlong

Stat_UTM <- SpatialPoints(cbind(Stations$X_UTM, Stations$Y_UTM),
    proj4string = CRS("+proj=utm +zone=21 +south +datum=WGS84 +units=m +no_defs"))
```

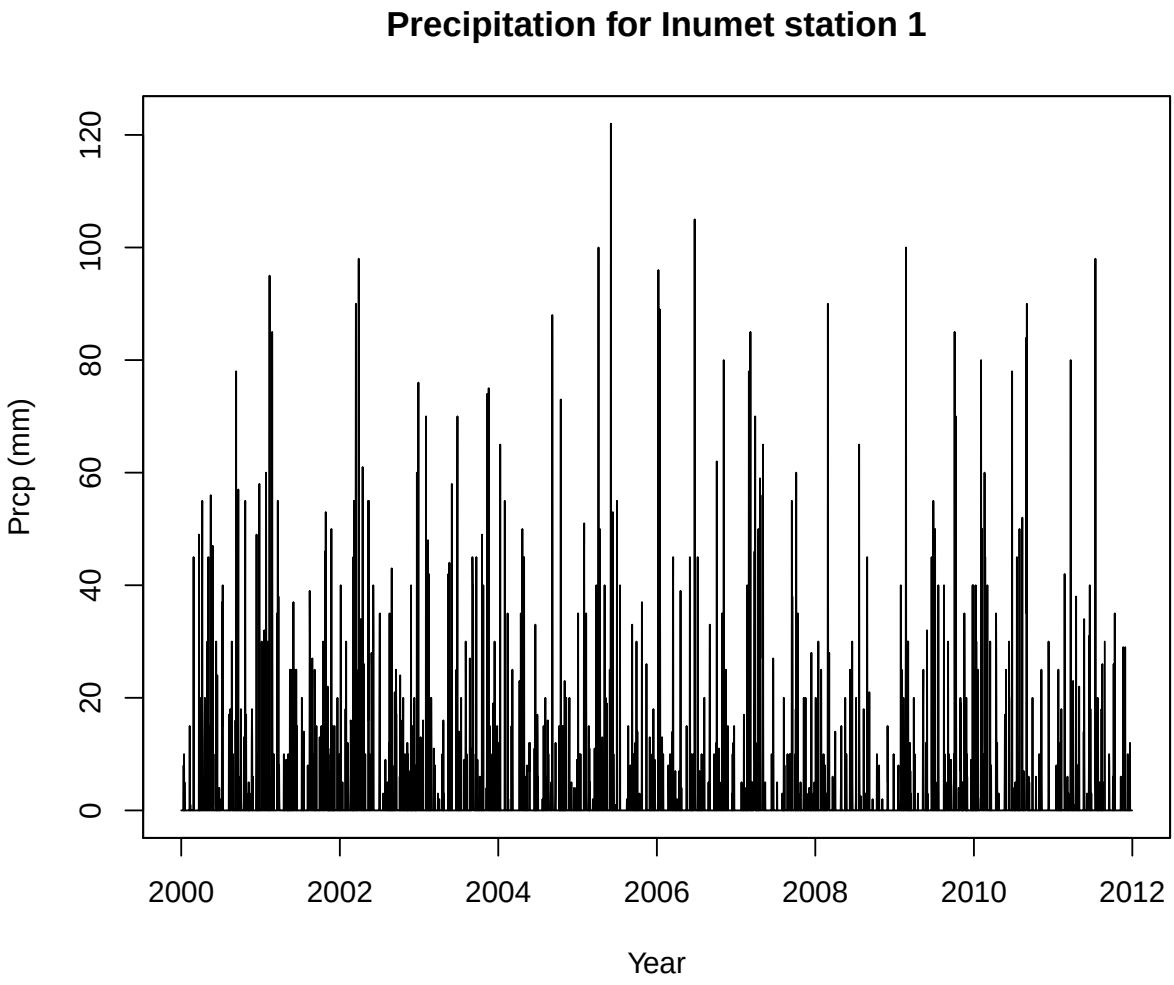
```r
longlatcoor <- spTransform(Stat_UTM, CRS("+proj=longlat +datum=WGS84 +no_defs"))
elev <- rep(50, nrow(Stations))


# create a dataframe for the stations file with the following
# columns: Stations name, Latitute, Longitude, Elevation
foo <- data.frame(colnames(Pdata_z), round(coordinates(longlatcoor)[,
    2], 2), round(coordinates(longlatcoor)[, 1], 2), round(elev,
    2))
colnames(foo) <- c("Stations", "Lat", "Long", "Elev")
head(foo)
```

```
##   Stations    Lat   Long Elev
## 1    X2673 -34.24 -55.92   50
## 2    X2826 -34.63 -55.05   50
## 3    X2819 -34.54 -55.87   50
## 4    X2748 -34.38 -56.54   50
## 5    X2725 -34.35 -54.79   50
## 6    X2715 -34.34 -55.76   50
```

```r
# create a list for each inumet station witn Longitude,
# Latitude, Precipitation data
outlist <- list()

for (i in 1:ncol(Pdata_z)) {
    outlist[[i]] <- list(x = foo$Long[i], y = foo$Lat[i], prcp = Pdata_z[,
        i])

}

names(outlist) <- colnames(Pdata_z)
# show the element of the list for Inumet station number 1
str(outlist[[1]])
```

```
## List of 3
##  $ x   : num -55.9
##  $ y   : num -34.2
##  $ prcp:'zooreg' series from 2000-01-01 to 2011-12-31
##   Data: Named num [1:4383] 0 0 0 0 0 0 0 0 0 0 ...
##   ..- attr(*, "names")= chr [1:4383] "6940" "6941" "6942" "6943" ...
##   Index:  Date[1:4383], format: "2000-01-01" "2000-01-02" ...
##   Frequency: 1
```

```r
# test the loop by plotting precipitation data against time
# for Inumet station number 1
plot(outlist[[1]]$prcp, main = "Precipitation for Inumet station 1",
    xlab = "Year", ylab = "Prcp (mm)")
```



CHIRPS data

CHIRPS is a quasi global 30 year data set: Climate Hazards Group InfraRed Precipitation with Station data derived from blending satellite data and observational data. As a result this could be a good replacement of station input data in catchments were station data is lacking or missing. In addition it can deliver a better spatial coverage of rainfall data. ## download CHIRPS data

To download the CHIRPS data, refer to the document "Advanced SWAT Course Notes: Downloading and managing CHIRPS data".

read CHIRPS data

Once you followed the instruction of the document "Advanced SWAT Course Notes: Downloading and managing CHIRPS data", you would be able to load the CHIRPS data selected in R. Then, we will add dates to the CHIRPS data.

```
# Load the CHIRPS data in rds format
chirpsSL <- readRDS("output.rds", refhook = NULL)
```

```r
# insert the dates for the CHIRPS data and make zoo

C_dates <- seq.Date(as.Date("2000-01-01"), as.Date("2017-09-30"),
    by = 1)
```

Correlation analysis betweent CHIRPS data and INUMET data

We want to make a comparitive analysis of all CHIRPS data against all INUMET data based on pentads. This comparison will help us to understand how representative is the CHIRPS data relative to the local INUMET stations. The fist step is to write some functions to create pentad precipitation data:

```r
# You need to define the series of days (the pentads) So we
# want to write a function that generates the pentads for
# each year then we want to aggregate on the pentads
require(epiR)
```

```r
# bunch of functions to create pentad
pentad_fun <- function(Date) {
    d <- seq.Date(as.Date(Date), as.Date(Date) + days_in_month(as.Date(Date)),
        by = 5)
    d1 <- d[1:(length(d) - 1)]  # need to not have the last day
}

long_pentad <- function(startDate, endDate) {
    months <- seq.Date(as.Date(startDate), as.Date(endDate),
        by = "month")
    pentads <- sapply(months, pentad_fun)
    out <- c(do.call(c, pentads), as.Date(endDate))
    return(out)
}

rep_fun <- function(x) {
    out <- sort(rep(x, times = c(diff(x), 1)))
    return(out)
}
```

The second step is to create a loop to calculate correlation between each INUMET station and each CHIRPS point

```r
# loop to calculate correlation between each inumet point and
# each chirps point create first the output table
cor1 <- matrix(0, nrow = length(chirpsSL), ncol = nrow(foo))

result <- data.frame(Long = numeric(length = length(chirpsSL)),
    Lat = numeric(length = length(chirpsSL)), cor1)

# fill the table output with the loop

for (j in 1:nrow(foo)) {
    for (i in 1:length(chirpsSL)) {
        c_precip <- zoo(chirpsSL[[i]]$prcp, order.by = C_dates)

        # merge in a same table precipitation data from INUMET and
```

```
        # precipitation data from CHIRPS
        test_merge <- merge.zoo(c_precip, outlist[[j]]$prcp,
            all = F)


        # now apply the pentad function
        chirps_pentad <- rep_fun(long_pentad(time(test_merge)[1],
            time(test_merge)[nrow(test_merge)]))


        # now aggregate the data
        chirps_rain_pentad <- aggregate(coredata(test_merge),
            list(pentad = chirps_pentad), sum)

        # calculate correlation between CHIRPS pentad precipitation
        # and INUMET pentad precipitation
        result[i, j + 2] <- cor(chirps_rain_pentad[, 2], chirps_rain_pentad[,
            3], use = "complete.obs")
        result$Long[i] <- chirpsSL[[i]]$x
        result$Lat[i] <- chirpsSL[[i]]$y
    }
}
```

Once we calculate the correlation, we can plot the results. Here is a plot example of the result.The plot can be done for each INUMET station (changing X1 by X2,...).

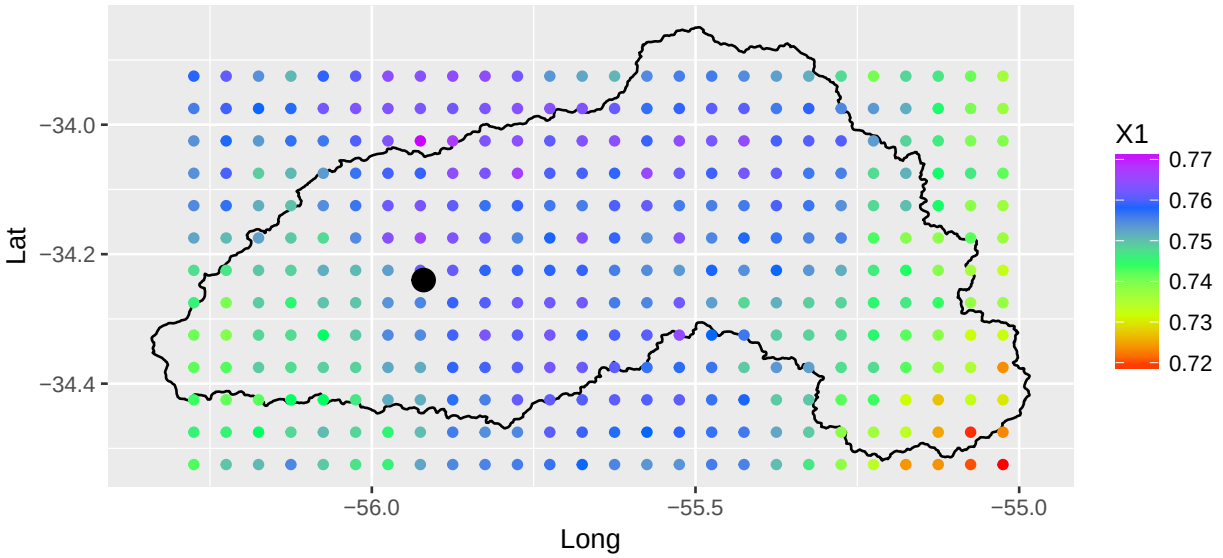
```
# ggplot of the correlation of INUMET station X1 against all
# CHIRPS data in the catchment Load in catchment shape
SL <- readOGR("sl_shape/subcuencaSantaLuciahastariostaluciachico.shp")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "sl_shape/subcuencaSantaLuciahastariostaluciachico.shp", layer: "subcuencaSantaLuciahastarios
## with 1 features
## It has 1 fields
```

```
pl <- ggplot(result, aes(x = Long, y = Lat)) + geom_polygon(data = SL,
    aes(x = long, y = lat, group = group), colour = "black",
    fill = NA) + coord_equal() + # add result of correlation
geom_point(aes(colour = X1)) + scale_color_gradientn(colours = rainbow(5)) +
    # add inumet station location
geom_point(data = foo[1, ], aes(colour = X1), colour = "black",
    size = 4, fill = NA)
```

```
## Regions defined for each Polygons
```

```
pl
```

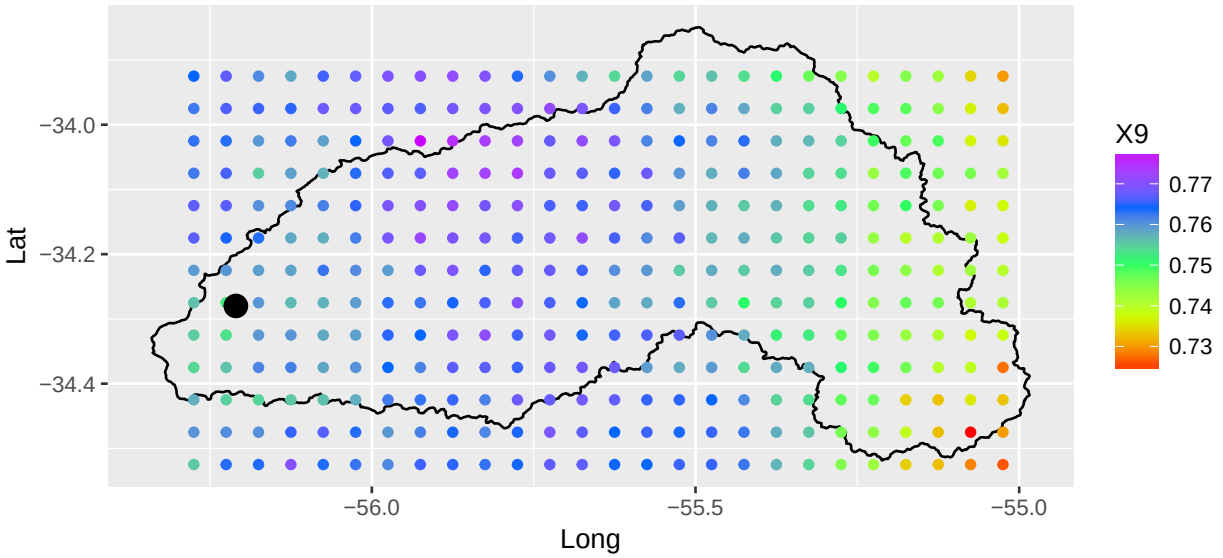


The plot shows the correlation between one INUMET station, the station X1 (black point), and all CHIRPS data of the catchment. The correlation values are varying from 0.72 to 0.77 accross the catchment.

```
# ggplot of the correlation of INUMET station X9 against all
# CHIRPS data in the catchment
pl <- ggplot(result, aes(x = Long, y = Lat)) + geom_polygon(data = SL,
    aes(x = long, y = lat, group = group), colour = "black",
    fill = NA) + coord_equal() + # add result of correlation
geom_point(aes(colour = X9)) + scale_color_gradientn(colours = rainbow(5)) +
    # add inumet station location
geom_point(data = foo[9, ], aes(colour = X9), colour = "black",
    size = 4, fill = NA)
```

```
## Regions defined for each Polygons
```

```
pl
```



The plot shows the correlation between one INUMET station, the station X9 (black point), and all CHIRPS data of the catchment. This plot can be done for the 14 INUMET stations.

Spatial auto-correlation analysis betweent CHIRPS data and INUMET data

The next question we want to answer is what is the space time variation in rainfall accross the catchment and is this different based on the INUMET stations and the CHIRPS data. To analyse the spatial variation accross the catchment, we used the Moran's I. We did 2 different calculation using the Moran's with the 2 sets of precipitation data (INUMET & CHIRPS):

1- at daily step
2- at seasonal step

Moran's I at daily step using INUMET precipitation data

We calculate the Moran's I for all INUMET precipitation data accross the catchment using daily step precipitation accross year. The result will show one Moran's I for each day accross years. The script is the

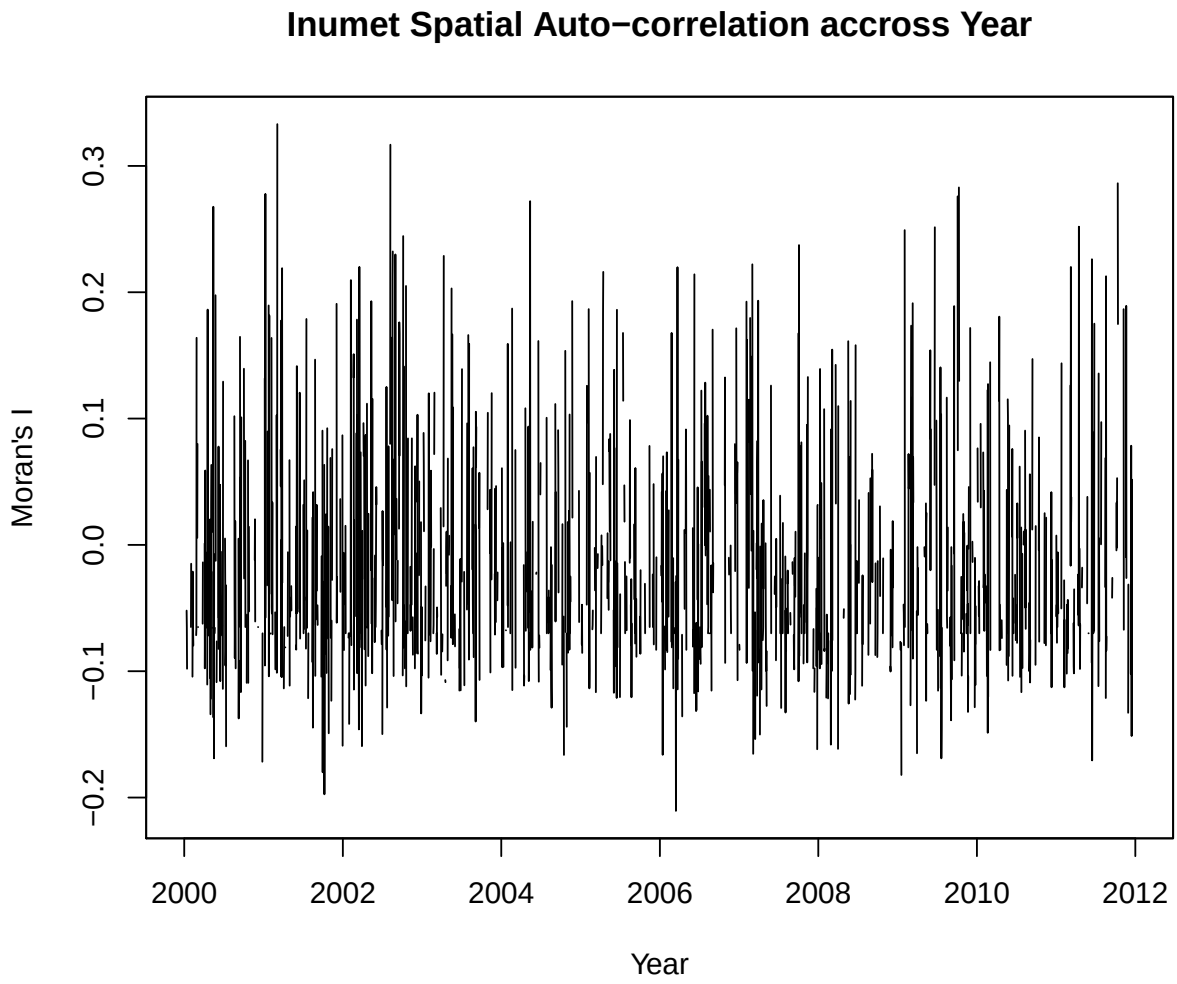
following:

```
# Calculate Moran's I accross year for INUMET stations
# package required and library install.packages('ape')
library(ape)

# create an empty vector
moran_inumet <- rep(0, nrow(Pdata_z))

# in foo, clear elevation column which will be replace by
# precipitation for one day and repeat that for all day
foo <- foo[, -4]

# need to generate a matrix of inverse distance weights to
# calculate Moran's I
inumet.dists <- as.matrix(dist(cbind(foo$Long, foo$Lat)))

inumet.dists.inv <- 1/inumet.dists
diag(inumet.dists.inv) <- 0

inumet.dists.inv[1:14, 1:14]  # 14 represents the number of rows within foo
```

```
##            1         2         3         4         5        6         7
## 1  0.0000000 1.0488613 3.2879797 1.5732919 0.8807924 5.299989 1.0307088
## 2  1.0488613 0.0000000 1.2122325 0.6618889 2.6171196 1.303880 2.6090903
## 3  3.2879797 1.2122325 0.0000000 1.4517168 0.9119215 4.381080 1.0398629
## 4  1.5732919 0.6618889 1.4517168 0.0000000 0.5713446 1.280369 0.6271472
## 5  0.8807924 2.6171196 0.9119215 0.5713446 0.0000000 1.030873 5.4473471
## 6  5.2999894 1.3038796 4.3810795 1.2803688 1.0308731 0.000000 1.2285902
## 7  1.0307088 2.6090903 1.0398629 0.6271472 5.4473471 1.228590 0.0000000
## 8  1.3328595 2.3408229 1.2992493 0.7249994 2.4899052 1.660910 4.4721360
## 9  3.4159349 0.8253196 2.3363466 2.9000740 0.7033712 2.202729 0.7935508
## 10 1.8841114 1.1230285 1.3469311 0.8873216 1.1407174 1.944407 1.4421744
## 11 1.1056645 1.4923711 1.0037712 0.6500088 2.1081851 1.255505 3.1295877
## 12 2.8295592 0.7652339 1.7165011 2.3966279 0.6788286 1.847738 0.7685489
## 13 3.2879797 1.2491220 2.0327891 1.0718046 1.1165695 3.787770 1.3926550
## 14 5.1502620 1.2429315 2.7196415 1.2051692 1.0505384 6.984303 1.2782750
##            8         9        10        11        12       13       14
## 1  1.3328595 3.4159349 1.8841114 1.1056645 2.8295592 3.287980 5.150262
## 2  2.3408229 0.8253196 1.1230285 1.4923711 0.7652339 1.249122 1.242932
## 3  1.2992493 2.3363466 1.3469311 1.0037712 1.7165011 2.032789 2.719641
## 4  0.7249994 2.9000740 0.8873216 0.6500088 2.3966279 1.071805 1.205169
## 5  2.4899052 0.7033712 1.1407174 2.1081851 0.6788286 1.116569 1.050538
## 6  1.6609096 2.2027287 1.9444069 1.2555049 1.8477377 3.787770 6.984303
## 7  4.4721360 0.7935508 1.4421744 3.1295877 0.7685489 1.392655 1.278275
## 8  0.0000000 0.9599423 2.0194810 3.5421817 0.9277572 2.021130 1.784577
## 9  0.9599423 0.0000000 1.2675503 0.8377372 5.1987524 1.700051 2.054987
## 10 2.0194810 1.2675503 0.0000000 2.0974919 1.3506116 3.990434 2.649995
## 11 3.5421817 0.8377372 2.0974919 0.0000000 0.8423610 1.649124 1.405110
## 12 0.9277572 5.1987524 1.3506116 0.8423610 0.0000000 1.694672 1.916708
## 13 2.0211302 1.7000510 3.9904344 1.6491235 1.6946719 0.000000 7.808688
## 14 1.7845765 2.0549873 2.6499947 1.4051098 1.9167079 7.808688 0.000000
```

```
# loop to rearrange foo data (which are INUMET data) and
# calculate Moran's I
```

```r
for (i in 1:nrow(Pdata_z)) {

    # rearrange foo by transposition
    foo$prcp <- t(Pdata_z[i, ])
    if (all(foo$prcp == 0, na.rm = T)) {
        moran_inumet[i] <- NA
    } else {
        # calculate Moran's I for each day
        moran_inumet[i] <- Moran.I(foo$prcp[, 1], inumet.dists.inv,
            na.rm = T)$observed
    }
}

moran_i_z <- zoo(moran_inumet, order.by = time(Pdata_z))

# plot moran indice against time
plot(moran_i_z, type = "l", xlab = "Year", ylab = "Moran's I",
    main = "Inumet Spatial Auto-correlation accross Year")
```



The plot shows the spatial variability of INUMET rainfall observation accross catchment and accross time.

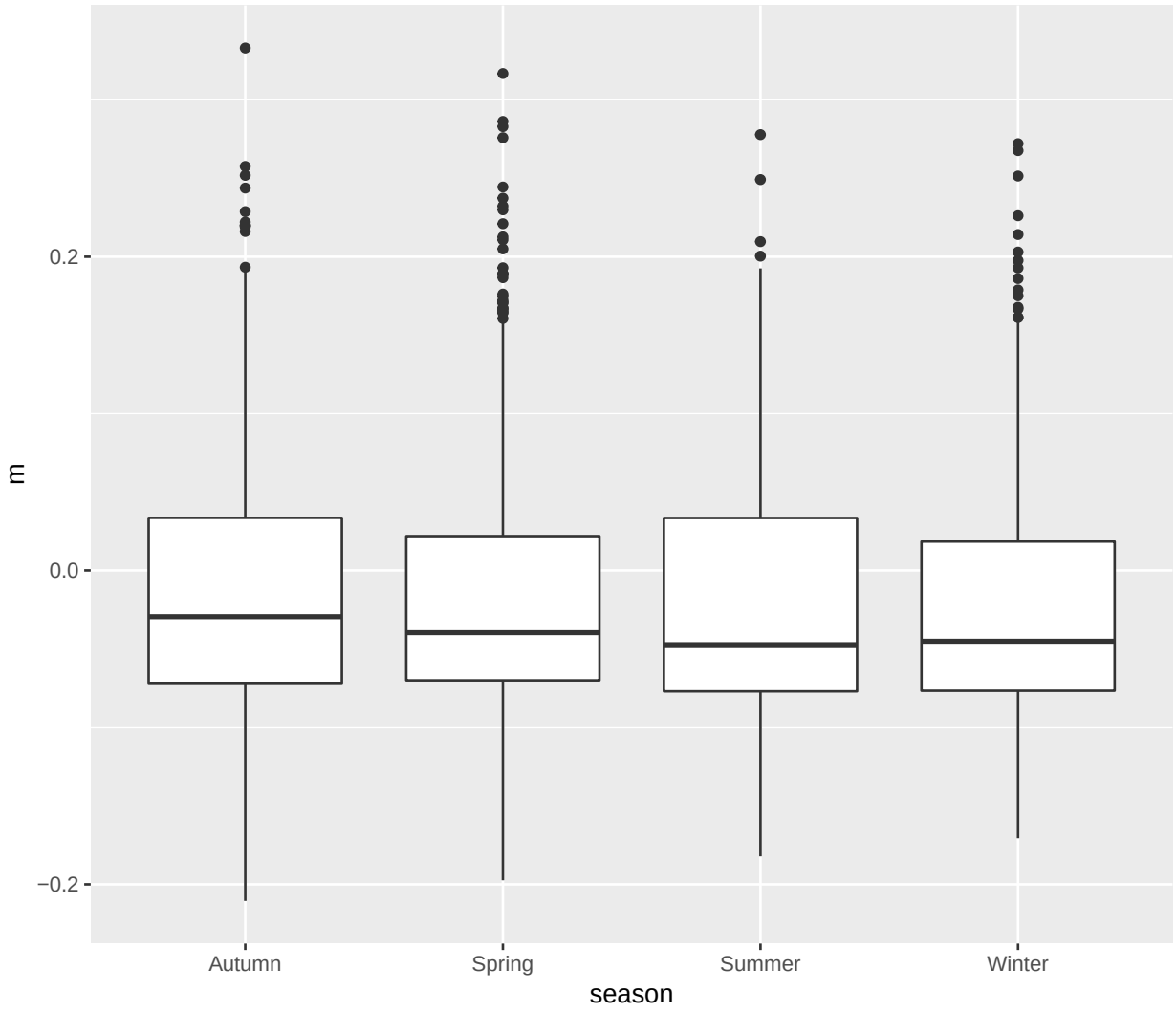
The Moran's I is pretty low which means there is high variability of precipitation data accross the catchment.

Moran's I at seasonal step using INUMET precipitation data

Then, we calculate the Moran's i at seasonal step to see if the autocorrelation for INUMET stations is variable accross the seasons.
The script is the following:

```
# Calculate Moran's I for each season - aggregate date of
# each season generating a function to put together data into
# each season
season <- sapply(time(moran_i_z), function(x) ifelse(month(x) >
    11 || month(x) < 3, "Summer", ifelse(month(x) < 5, "Autumn",
    ifelse(month(x) < 8, "Winter", "Spring"))))
# create a new data frame by season
moran_season <- data.frame(m = coredata(moran_i_z), season = season)

# boxplot of Moran's I for each season
pl <- ggplot(moran_season, aes(season, m)) + geom_boxplot()
pl
```

```
## Warning: Removed 2840 rows containing non-finite values (stat_boxplot).
```




```
# statistical parameters
summary(lm(m ~ season, data = moran_season))
```

```
##
## Call:
## lm(formula = m ~ season, data = moran_season)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.19960 -0.05708 -0.02419  0.04377  0.34403
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.010959   0.005259  -2.084   0.0373 *
## seasonSpring -0.004736   0.006524  -0.726   0.4680
## seasonSummer -0.008934   0.006816  -1.311   0.1901
## seasonWinter -0.010534   0.006795  -1.550   0.1213
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.08561 on 1539 degrees of freedom
##   (2840 observations deleted due to missingness)
## Multiple R-squared:  0.0019, Adjusted R-squared:  -4.514e-05
## F-statistic: 0.9768 on 3 and 1539 DF,  p-value: 0.4028
```

The boxplot and the p-value within the summary show that there is no significative variation accross the season using INUMET data. For every season it shows a quite high variation of precipitation data. So, to try another analysis and be able to find some difference accross season, we calculate the number of day where Moran's I is superior or inferior to a specific value for each season. The script is the following:

```
# aggregate number of day where Moran's I is superior or
# inferior to a specific value index value less than 0
LessThan0 <- aggregate(moran_season$m, list(season = moran_season$season),
    function(x) sum(ifelse(x <= 0, 1, 0), na.rm = T))
colnames(LessThan0)[2] <- "NumberDaysAutocorr<0"
pander(LessThan0)
```

season	NumberDaysAutocorr<0
Autumn	169
Spring	336
Summer	265
Winter	268

```
# index value more than 0.2
MoreThan0.2 <- aggregate(moran_season$m, list(season = moran_season$season),
    function(x) sum(ifelse(x > 0.2, 1, 0), na.rm = T))
colnames(MoreThan0.2)[2] <- "NumberDaysAutocorr>0.2"
pander(MoreThan0.2)
```

season	NumberDaysAutocorr>0.2
Autumn	11
Spring	12
Summer	4
Winter	6

Looking at the results, spring has the higher number of day where Moran's I is inferior to 0. This could mean that this season has the most variable precipitation accross space. However, spring (in addition to automn) is also the season where there is the higher number of day where Moran's I is greater or equal to 0.2, which is saying that spring also has the most auto-correlate precipitation values accross space. So, with those first analysis regarding seasonality, it is hard to make a clear conclusion.

Moran's I at daily step using CHIRPS data

We will do the same analysis as above but with the CHIRPS data. The script is the following:

```
# Calculate Moran's I accross year with chirps data some
# packages required
require(tidyverse)
require(zoo)

# As chirpsSL is a list, we need to convert prcp from a list
```

```
# into a dataframe first extract the prcp elements into a
# list
SLprcp <- chirpsSL %>% map("prcp")
# is.list(SLprcp) # to be sure prcp is a list
SLprcp_df <- as.data.frame(do.call(cbind, SLprcp))

# second extract the lat and long elements into a dataframe
SLlon <- chirpsSL %>% map("x")
SLlon_df <- as.data.frame(do.call(c, SLlon))
str(SLlon_df)
```

```
## 'data.frame':    338 obs. of  1 variable:
##  $ do.call(c, SLlon): num  -56.3 -56.3 -56.3 -56.3 -56.3 ...
```

```
SLlat <- chirpsSL %>% map("y")
SLlat_df <- as.data.frame(do.call(c, SLlat))
str(SLlat_df)
```

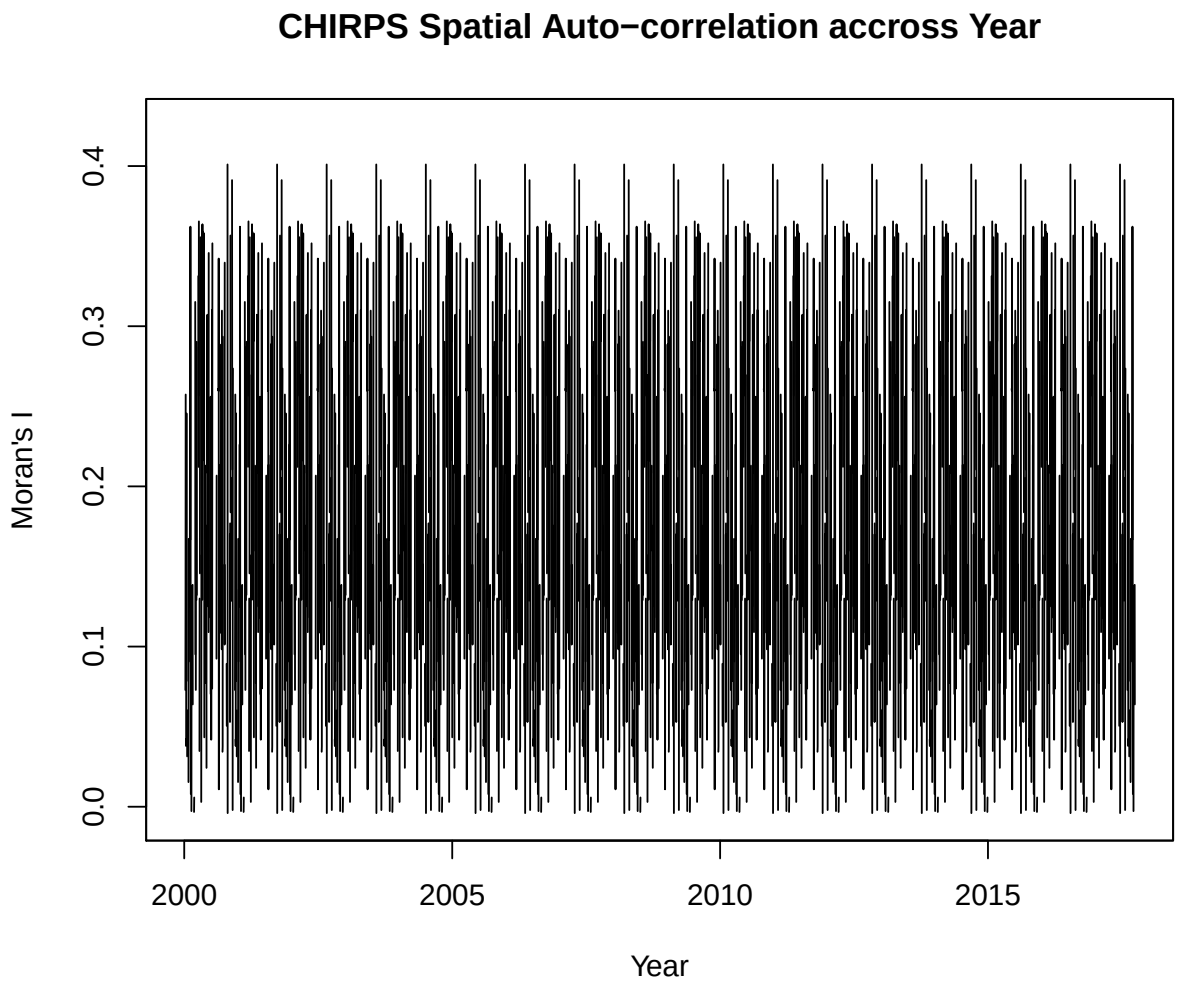
```
## 'data.frame':    338 obs. of  1 variable:
##  $ do.call(c, SLlat): num  -34.5 -34.5 -34.4 -34.4 -34.3 ...
```

```
# need to generate a matrix of inverse distance weights to
# calculate Moran's I
chirps.dists <- as.matrix(dist(cbind(SLlon_df[, 1], SLlat_df[,
    1])))

chirps.dists.inv <- 1/chirps.dists
diag(chirps.dists.inv) <- 0
# to have a look at the matrix
chirps.dists.inv[1:5, 1:5]
```

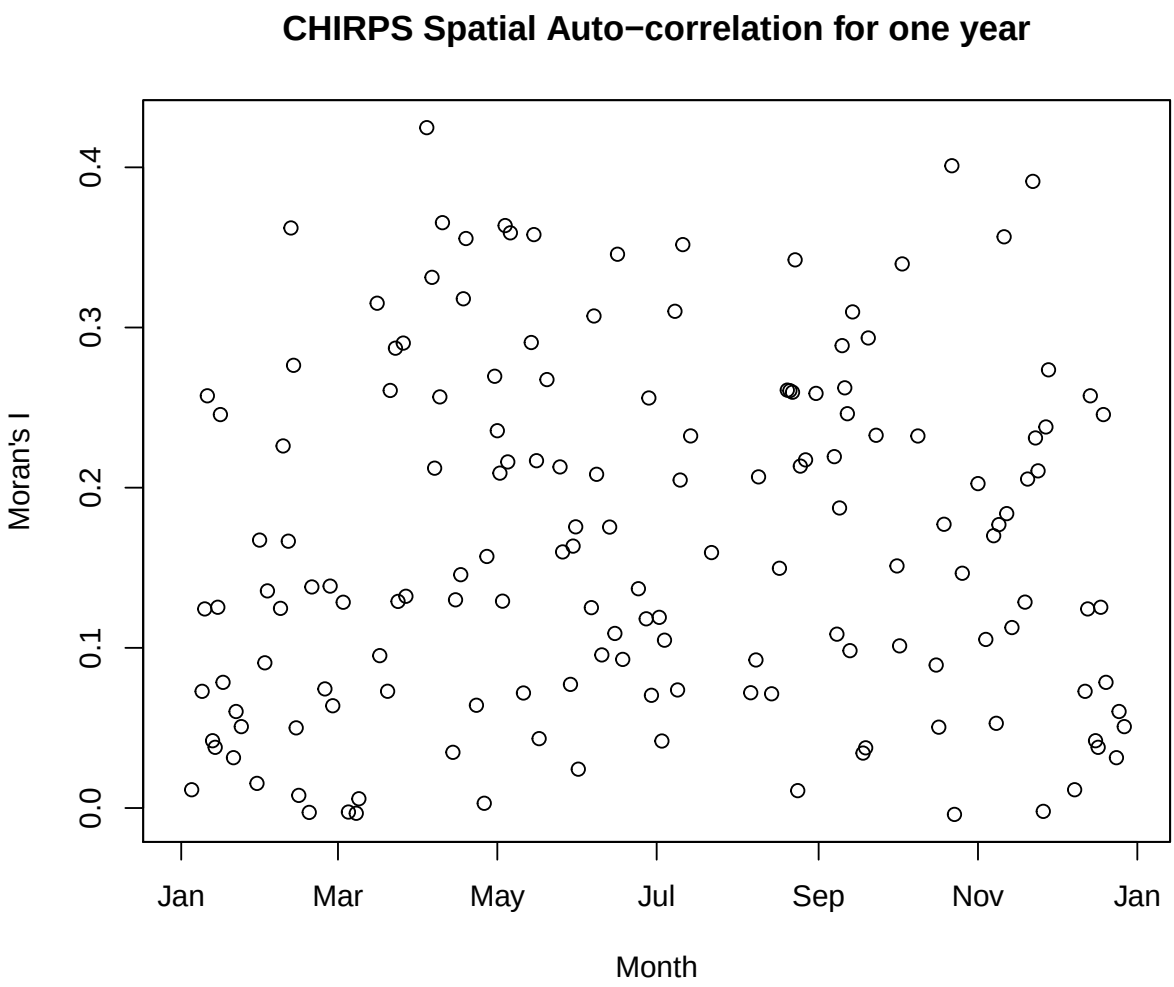
```
##           1         2        3         4         5
## 1  0.000000 20.000305 10.00015  6.666768  5.000076
## 2 20.000305  0.000000 20.00031 10.000153  6.666768
## 3 10.000153 20.000305  0.00000 20.000305 10.000153
## 4  6.666768 10.000153 20.00031  0.000000 20.000305
## 5  5.000076  6.666768 10.00015 20.000305  0.000000
```

```
# loop to rearrange chirpsSL data and calculate Moran's I
moran_chirps <- rep(0, nrow(SLlon_df))

for (i in 1:length(SLprcp_df)) {

    # rearrange chirps data SLprcp_df <- t(Pdata_z[i,])
    if (sum(ifelse(SLprcp_df[i, ] > 0, 1, 0)) < 1) {
        moran_chirps[i] <- NA
    } else {
        # caculate Moran's I for each day
        moran_chirps[i] <- Moran.I(as.numeric(SLprcp_df[i, ]),
            chirps.dists.inv, na.rm = T)$observed
    }
}
Dates <- seq.Date(as.Date("2000-01-01"), as.Date("2017-09-30"),
    by = 1)

moran_chirps_i_z <- zoo(moran_chirps, order.by = Dates)
```

```r
# plot Moran's I against time for chirps data
plot(moran_chirps_i_z, type = "l", xlab = "Year", ylab = "Moran's I",
    main = "CHIRPS Spatial Auto-correlation accross Year")
```




```r
# plot Moran's I for one year
plot(moran_chirps_i_z[1:365], type = "p", xlab = "Month", ylab = "Moran's I",
    main = "CHIRPS Spatial Auto-correlation for one year")
```



The plot shows us the spatial variability accross year for CHIRPS data accross the catchment is quite high (generally higher than with INUMET data) as the Moran auto-correlation index is low. However, there is no inverse correlation (Moran'I <0) as it was the case with INUMET data.

If we look at the plot of Moran's I for one year, we can see that the daily precipitation pattern is quite different accross the catchment.

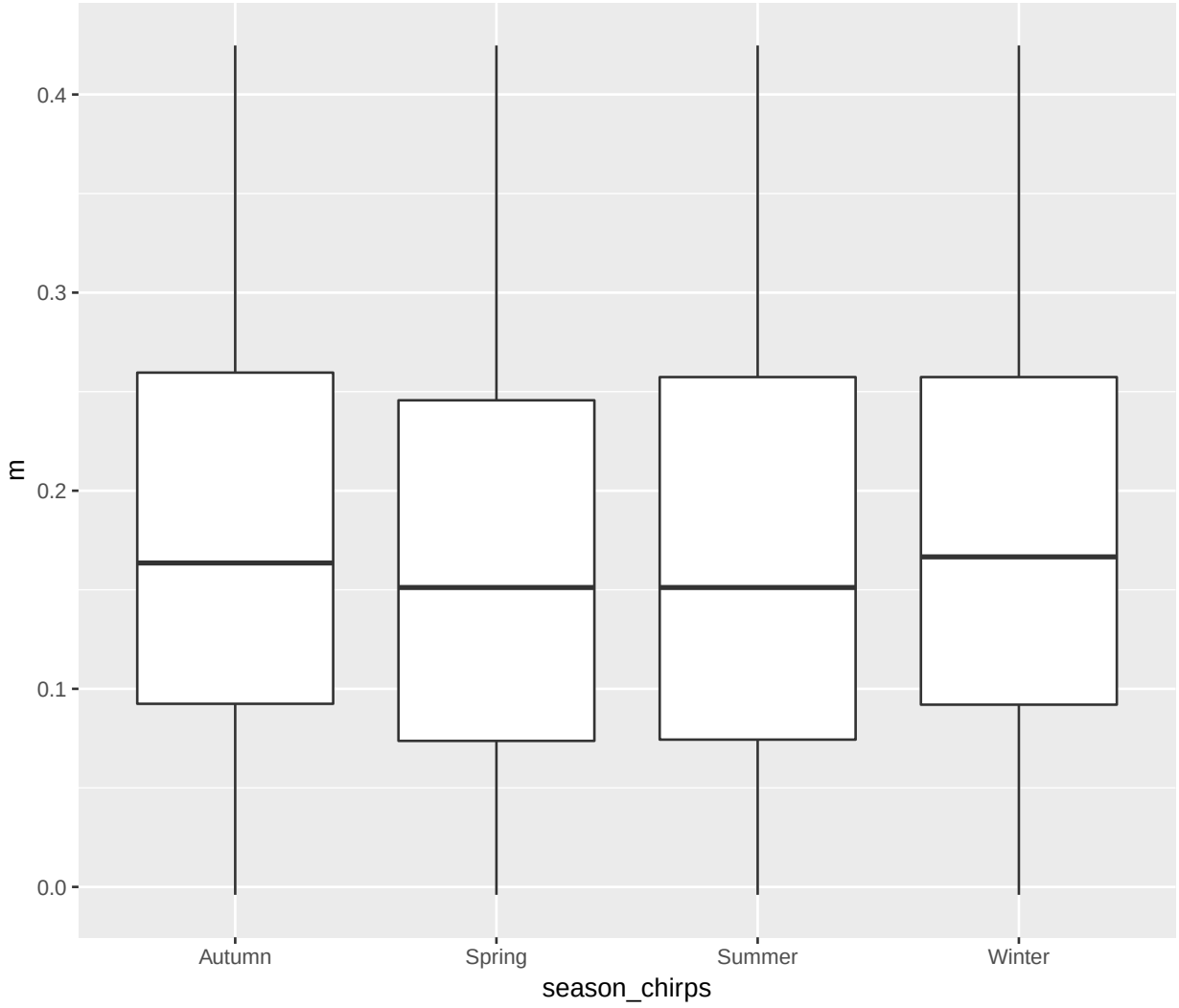
Moran's I at seasonal step using CHIRPS data

As we did the analysis with INUMET data, we will also have a look at the seasonal variability pattern of CHIRPS data.

```
# Calculate Moran's I with CHIRPS for each season - aggregate
# date of each season

# generating a function to put together data into each season
season_chirps <- sapply(time(moran_chirps_i_z), function(x) ifelse(month(x) >
    11 || month(x) < 3, "Summer", ifelse(month(x) < 5, "Autumn",
    ifelse(month(x) < 8, "Winter", "Spring"))))
# create new data frame by season
```

```
moran_chirps_season <- data.frame(m = coredata(moran_chirps_i_z),
    season = season_chirps)

# boxplot of Moran's I for each season
pl_chirps <- ggplot(moran_chirps_season, aes(season_chirps, m)) +
    geom_boxplot()
pl_chirps
```

```
## Warning: Removed 3719 rows containing non-finite values (stat_boxplot).
```




```
# statistical parameters

summary(lm(m ~ season, data = moran_chirps_season))
```

```
##
## Call:
## lm(formula = m ~ season, data = moran_chirps_season)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -0.17977 -0.09157 -0.01268  0.08492  0.25885
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.175786   0.005006  35.113   <2e-16 ***
## seasonSpring -0.009864   0.006137  -1.607    0.108
## seasonSummer -0.006050   0.006451  -0.938    0.348
## seasonWinter -0.002393   0.006470  -0.370    0.712
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1075 on 2760 degrees of freedom
##   (3719 observations deleted due to missingness)
## Multiple R-squared:  0.001198,   Adjusted R-squared:  0.000112
## F-statistic: 1.103 on 3 and 2760 DF,  p-value: 0.3465
```

The boxplot and the p-value within the summary show there is no significative variation accross the season using CHIRPS data. However, comparing to INUMET data the Moran's index is always higher than 0.

We calculate the mean and the standard deviation for each season with the following script to have a better idea about the exact values:

```
# Mean
Mean <- aggregate(moran_chirps_season$m, list(season = moran_chirps_season$season),
    mean, na.rm = T)
colnames(Mean)[2] <- "MeanAutocorr"
pander(Mean)
```

season	MeanAutocorr
Autumn	0.1758
Spring	0.1659
Summer	0.1697
Winter	0.1734

```
# Standard deviation
Standarddeviation <- aggregate(moran_chirps_season$m, list(season = moran_chirps_season$season),
    sd, na.rm = T)
colnames(Standarddeviation)[2] <- "Standard deviation Autocorr"
pander(Standarddeviation)
```

season	Standard deviation Autocorr
Autumn	0.1072
Spring	0.1078
Summer	0.1103
Winter	0.1042

Mean values and standard derivation values are not that different regarding seasonality.

Below, we calculate the number of day where Moran's I is superior or inferior to a specific value for each season. The script is the following:

```
# aggregate number of day where Moran's I is superior or
# inferior to a specific value index value less than 0.1
```

```
LessThan0.1 <- aggregate(moran_chirps_season$m, list(season = moran_chirps_season$season),
    function(x) sum(ifelse(x < 0.1, 1, 0), na.rm = T))
colnames(LessThan0.1)[2] <- "LessThan0.1AutoCorr"
pander(LessThan0.1)
```

season	LessThan0.1AutoCorr
Autumn	135
Spring	294
Summer	223
Winter	199

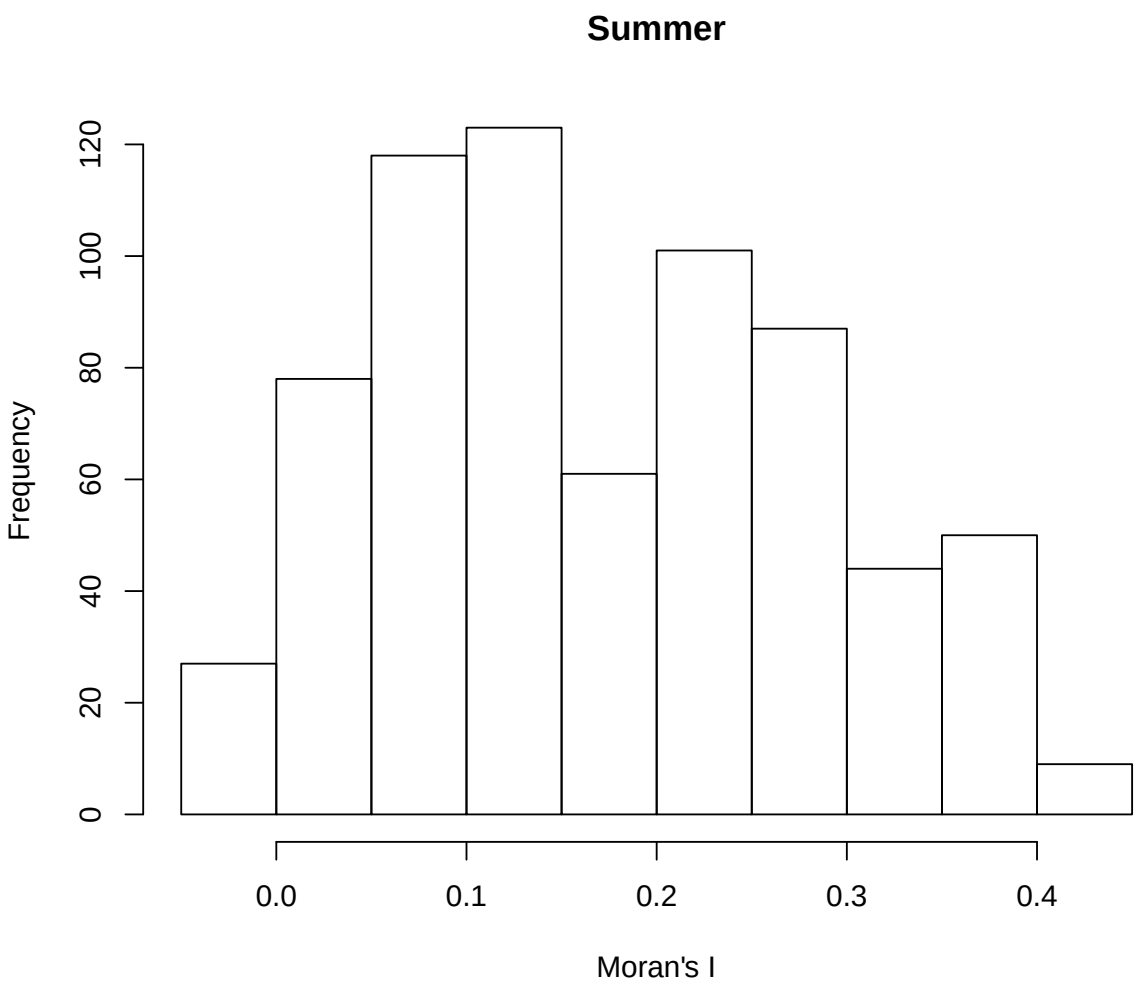
```
# index value more than 0.3
MoreThan0.3 <- aggregate(moran_chirps_season$m, list(season = moran_chirps_season$season),
    function(x) sum(ifelse(x > 0.3, 1, 0), na.rm = T))
colnames(MoreThan0.3)[2] <- "MoreThan0.3AutoCorr"
pander(MoreThan0.3)
```

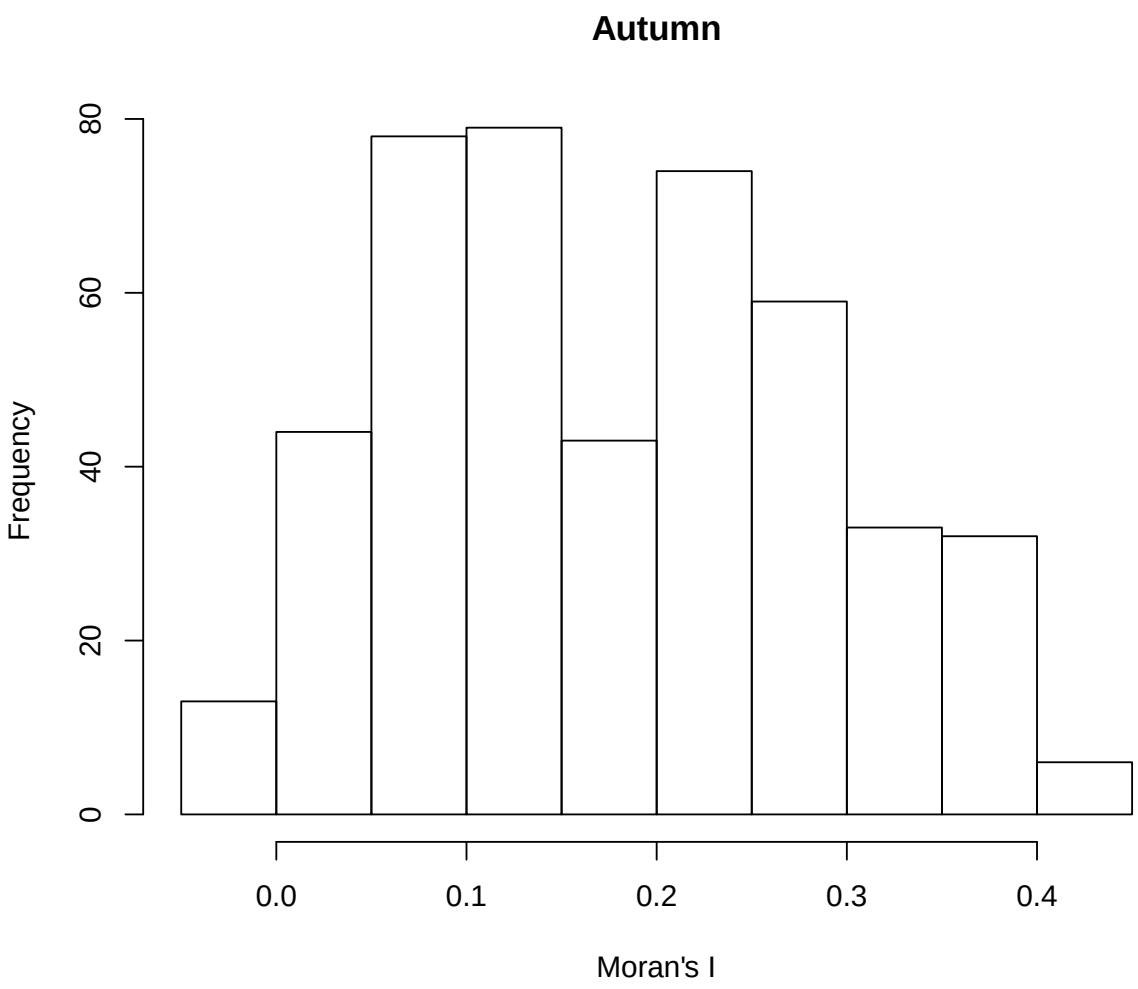
season	MoreThan0.3AutoCorr
Autumn	71
Spring	116
Summer	103
Winter	91

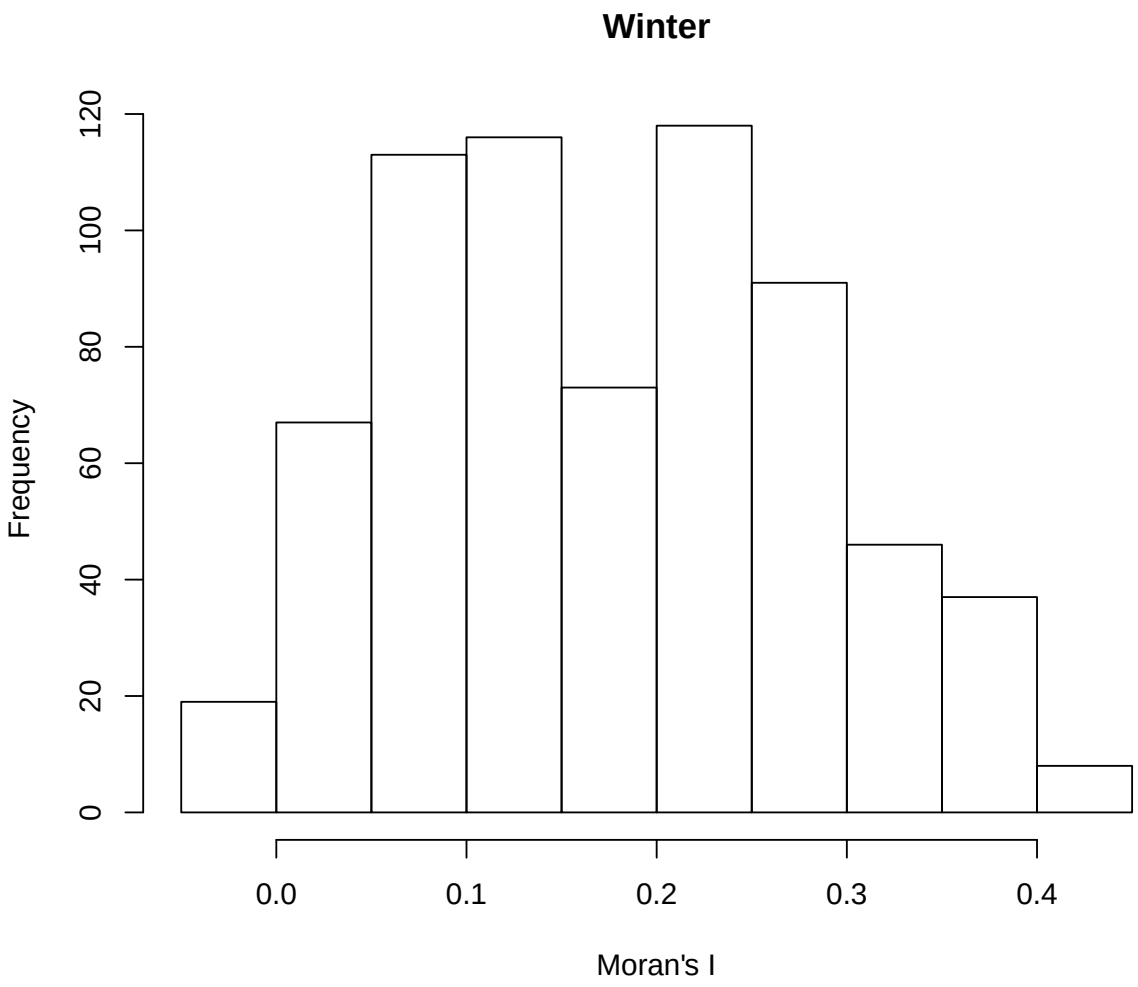
It shows the same kind of pattern than with INUMET data. Spring is at the same time showing the higher number of days with low autocorrelation (high variability) but also the higher number of day with higher autocorrelation (less variability).
Spring could be the season where the rainfalls are more convective. This element would need to be confirm with meteorology expert in Uruguay.

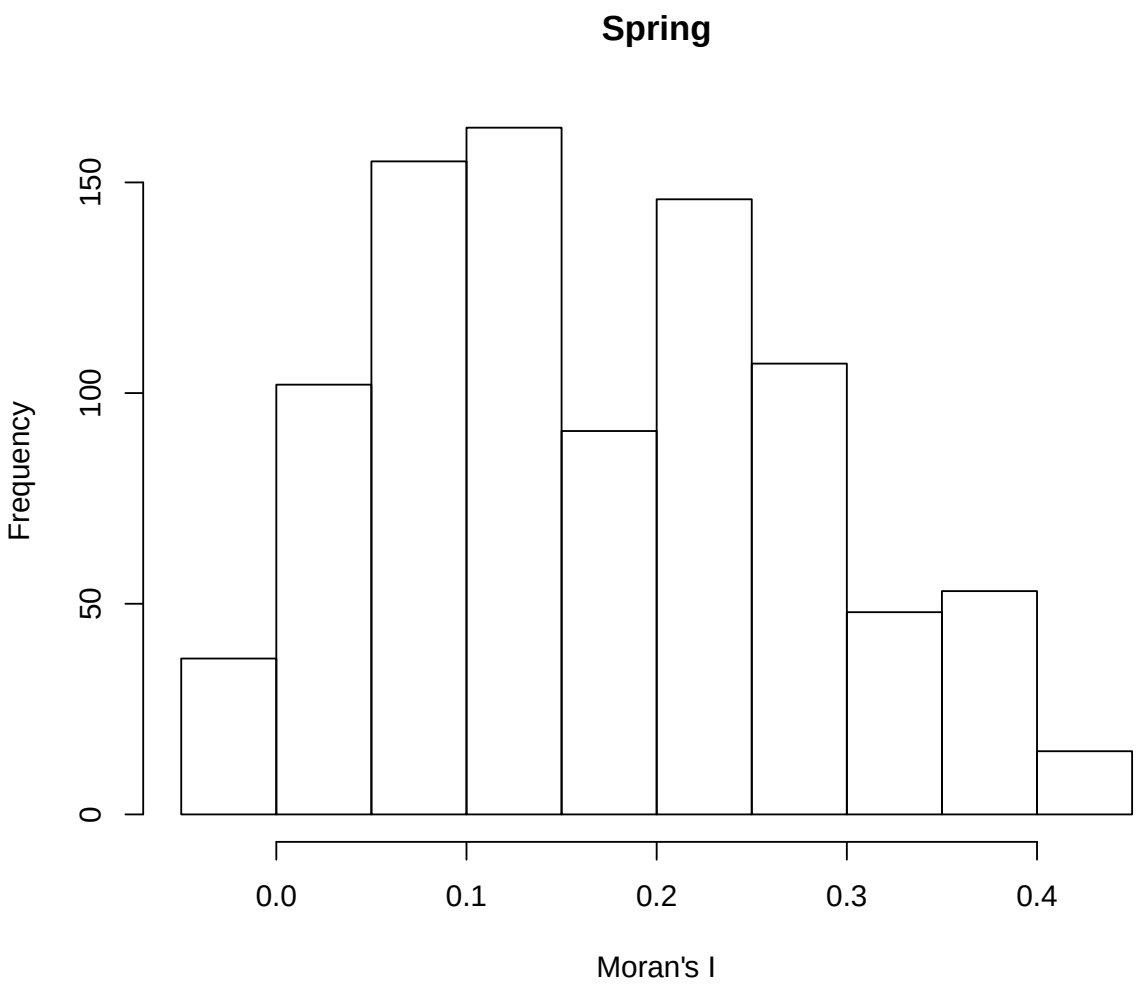
The following script line is plotting histogram for the 4 seasons.

```
# histogram
seasons <- unique(moran_chirps_season$season)
for (i in 1:4) {
    hist(moran_chirps_season$m[moran_chirps_season$season ==
        seasons[i]], main = seasons[i], xlab = "Moran's I")
}
```


Summer
Frequency
0
20
40
60
80
100
120
0.0
0.1
0.2
0.3
0.4
Moran's I


Autumn
Frequency
0
20
40
60
80
0.0
0.1
0.2
0.3
0.4
Moran's I


Winter
Frequency
0
20
40
60
80
100
120
0.0
0.1
0.2
0.3
0.4
Moran's I



The histogram for each season is showing how Moran's I values are frequent. Those results confirm the high variability of Moran's I accross seasons.