Advanced SWAT Course Notes: Basic Rainfall Analysis of CHIRPS data

Willem Vervoort & Flora Mer

23-11-2017

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 specific part explains a basic rainfall analysis of CHIRPS data. It is a basic comparison of CHIRPS data available in the Santa Lucia subcatchment.

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. Here we will demonstrate access to the daily v2.0 from the IRI data library for a period of 2000 - 2017.

Load CHIRPS data

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

Packages for the script

The following packages are needed for the all script.

```r
# Some useful packages to install
library(tidyverse)
library(zoo)
library(rgeos)
library(sp)
library(rgdal)
library(lubridate)
```

Download CHIRPS data and map CHIRPS data location in the subcatchment

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. After loading the CHIRPS data, we will convert precipitation data, latitute, longitude into dataframes.

Then, We will plot the chirps data location into the subcatchment. The script is the following:

```r
# Read in the chirps rainfall data once saved in the
# computer
chirpsSL <- readRDS("output.rds", refhook = NULL)

# convert prcp from list into a dataframe first extract
# the prcp elements into a list and then convert to a
# dataframe

SLprcp <- chirpsSL %>% map("prcp")
is.list(SLprcp)  # to check SLprcp is a list
```

```
## [1] TRUE
```

```r
SLprcp_df <- as.data.frame(do.call(cbind, SLprcp))
# head(SLprcp_df) # to have a look at the prcp dataframe

# Add Dates to precipitation data
Dates <- seq.Date(as.Date("2000-01-01"), as.Date("2017-09-30"),
    by = 1)
Dates_df <- data.frame(Dates)
Dates_df_ <- separate(Dates_df, "Dates", c("Year", "Month",
    "Day"), sep = "-")
SLprcp_dates <- cbind(Dates_df_, SLprcp_df)

# erase the 'V' caracter of the column names
names(SLprcp_dates) <- gsub("V", "", names(SLprcp_dates),
    fixed = TRUE)
# head(SLprcp_dates) # to have a look at the
# SLprcp_dates dataframe

# second extract the lat and long elements into a list
# and then convert to a dataframe

SLlon <- chirpsSL %>% map("x")
SLlon_df <- as.data.frame(do.call(c, SLlon))
colnames(SLlon_df)[1] <- "lon"
head(SLlon_df)
```

```
##       lon
## 1 -56.275
## 2 -56.275
## 3 -56.275
## 4 -56.275
## 5 -56.275
## 6 -56.275
```

```r
SLlat <- chirpsSL %>% map("y")
SLlat_df <- as.data.frame(do.call(c, SLlat))
colnames(SLlat_df)[1] <- "lat"
str(SLlat_df)
```

```
## 'data.frame':    338 obs. of  1 variable:
##  $ lat: num  -34.5 -34.5 -34.4 -34.4 -34.3 ...
```

```r
# add dataframe with the chirps point numbers
chirps_stations <- c(colnames(SLprcp_df))
chirps_stations <- c(1:nrow(SLlat_df))

# Create one dataframe with chirps point numbers, lat,
# long

chirpslatlong_df <- cbind(chirps_stations, SLlon_df, SLlat_df)
head(chirpslatlong_df)
```

```
##   chirps_stations     lon     lat
## 1               1 -56.275 -34.525
## 2               2 -56.275 -34.475
## 3               3 -56.275 -34.425
## 4               4 -56.275 -34.375
## 5               5 -56.275 -34.325
## 6               6 -56.275 -34.275
```

```r
# read in the shapefile
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
```

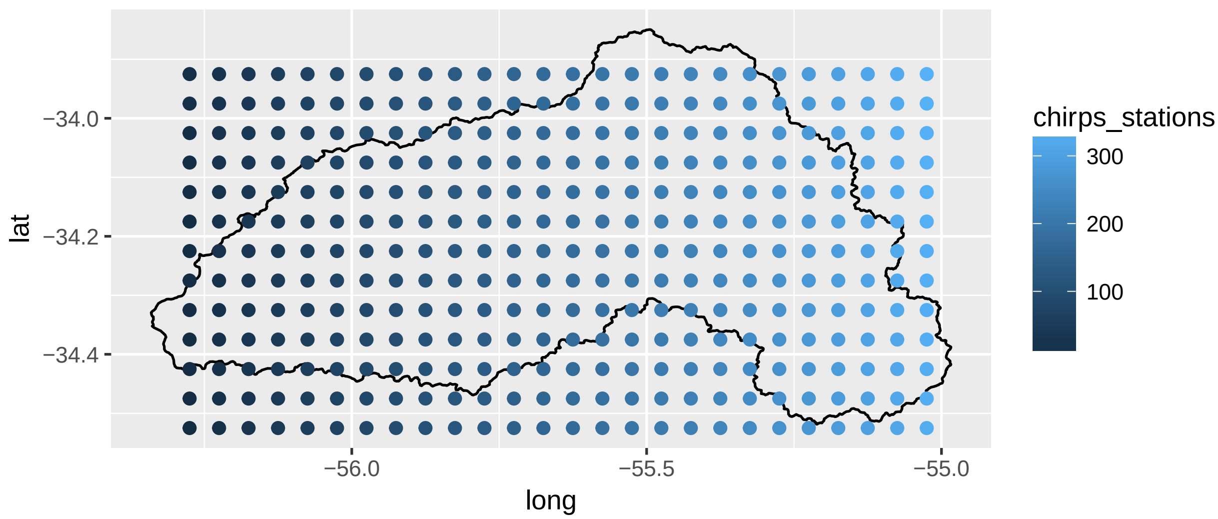
```r
map <- ggplot() + geom_polygon(data = SL, aes(x = long,
    y = lat, group = group), colour = "black", fill = NA)
```

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

```r
# chirps Df - dataframe chirps points with lat, lont for
# spatial plot
Statchirps_df <- data.frame(Stations = chirps_stations,
    Long = chirpslatlong_df$lon, Lat = chirpslatlong_df$lat)

# spatial plot of chirps point locations into the
# subcatchment
pl <- ggplot() + geom_polygon(data = SL, aes(x = long, y = lat,
    group = group), colour = "black", fill = NA) + coord_equal() +
    geom_point(data = Statchirps_df, aes(Long, Lat, colour = chirps_stations),
        size = 2)  #+
```

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

```r
# geom_text(data = Statchirps_df,
# aes(Long,Lat,label=chirps_stations, vjust=1))
# #uncomment if want to plot the point numbers
pl
```



The resolution of chirps data is 5 km to 5 km. So, here we have 338 chirps point accross the whole subcatchment.

Calculate basic statistics based on daily and annual step

We will do some basic statistics to understand better the rainfall pattern accross all CHIRPS data into the catchment.

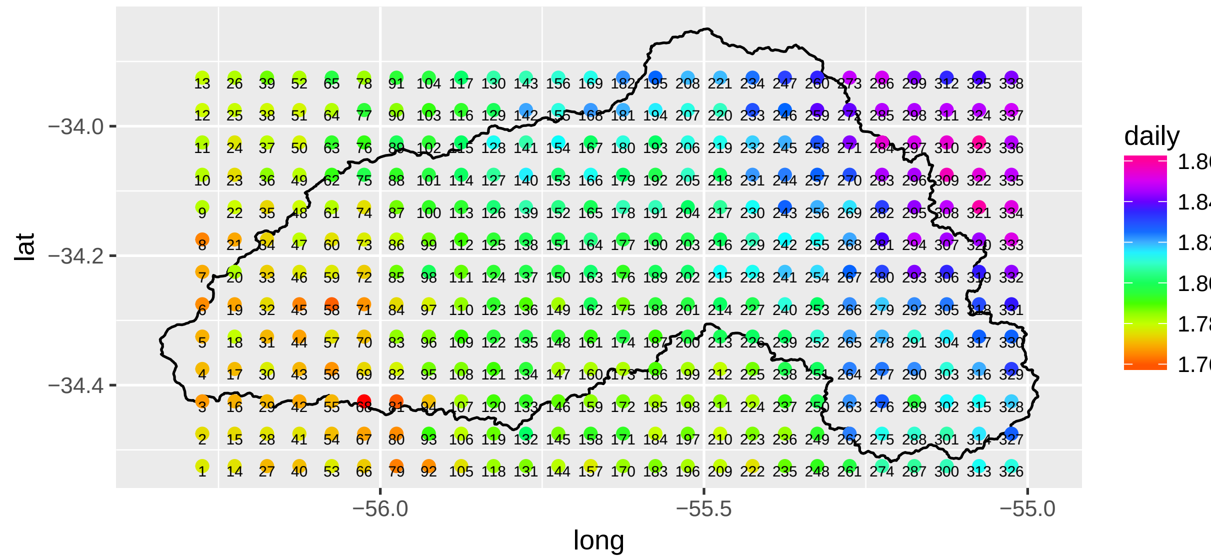
Mean daily rainfall

We first compare th mean daily rainfall of all the CHIRPS precipitation data, using the log of the daily mean. The script is the following:

```
# building the right dataframe format
SLprcp_dates_new <- zoo(SLprcp_dates[, 4:ncol(SLprcp_dates)],
    order.by = Dates, frequency = 1)
# log mean daily rainfall
mean_d_P_chirps <- apply(SLprcp_dates_new, 2, function(x) exp(mean(log(x +
    1), na.rm = T)))
meanchirps_df <- Statchirps_df
meanchirps_df$daily <- mean_d_P_chirps
# spatial plot of the daily mean rainfall of chirps data
pl <- ggplot() + geom_polygon(data = SL, aes(x = long, y = lat,
    group = group), colour = "black", fill = NA) + coord_equal() +
    geom_point(data = meanchirps_df, aes(Long, Lat, colour = daily),
        size = 2) + geom_text(data = meanchirps_df, aes(Long,
    Lat, label = Stations, vjust = 1), size = 2) + scale_colour_gradientn(colors = rainbow(10))
```

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

```
pl
```



The spatial plot of mean daily rainfall with CHIRPS data shows there is an increase in daily rainfall from south-west to north-est. However, there are no big difference accross the catchment as the log of mean daily rainfall is going from 1.76 to 1.86.

Coefficient of variation of daily rainfall

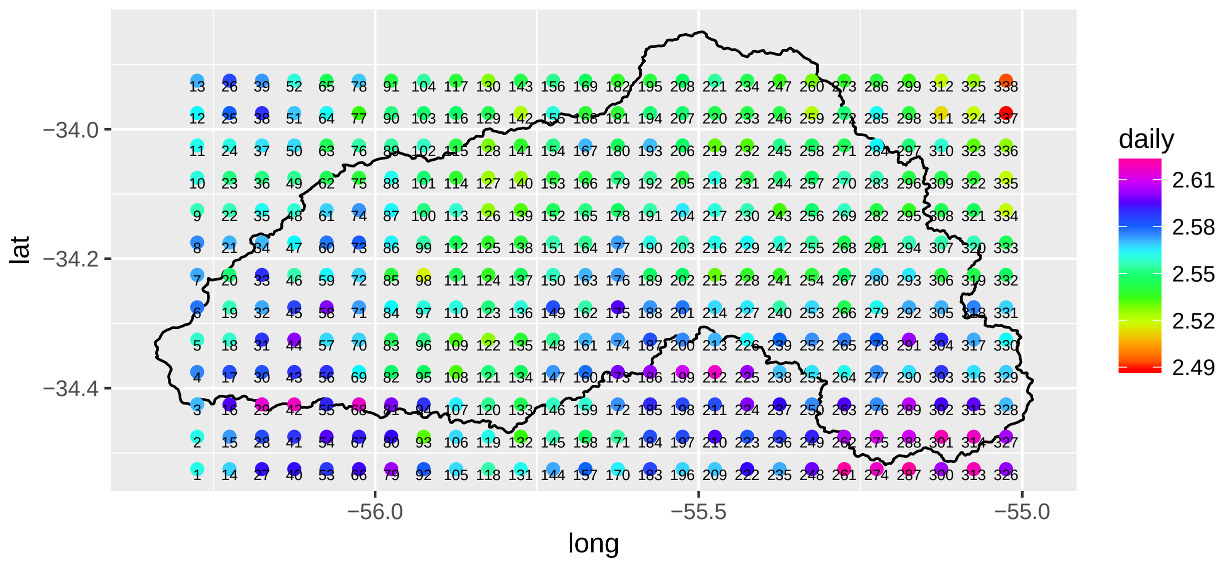
Now we will have a look at the coefficient of variation (CV) of daily rainfall of each chirps point available. The CV is defined as the ratio of the standard deviation (sd) to the mean. In this case, it shows the extent of variability in relation to the mean of the daily rainfall.

```
# CV daily rainfall
CV_d_P_chirps <- apply(SLprcp_dates_new, 2, function(x) sd(x,
    na.rm = T)/mean(x, na.rm = T))
CV_df_chirps <- Statchirps_df
CV_df_chirps$daily <- CV_d_P_chirps

pl <- ggplot() + geom_polygon(data = SL, aes(x = long, y = lat,
    group = group), colour = "black", fill = NA) + coord_equal() +
    geom_point(data = CV_df_chirps, aes(Long, Lat, colour = daily),
        size = 2) + geom_text(data = CV_df_chirps, aes(Long,
```

```
    Lat, label = Stations, vjust = 1), size = 2) + scale_colour_gradientn(colors = rainbow(10))
```

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

```
pl
```



The variability of the daily rainfall pattern is increasing from north-east to south-west, in contrary to the daily mean rainfall. However, the values of the coefficient of varibility are not really different accross the catchment.

Mean annual rainfall

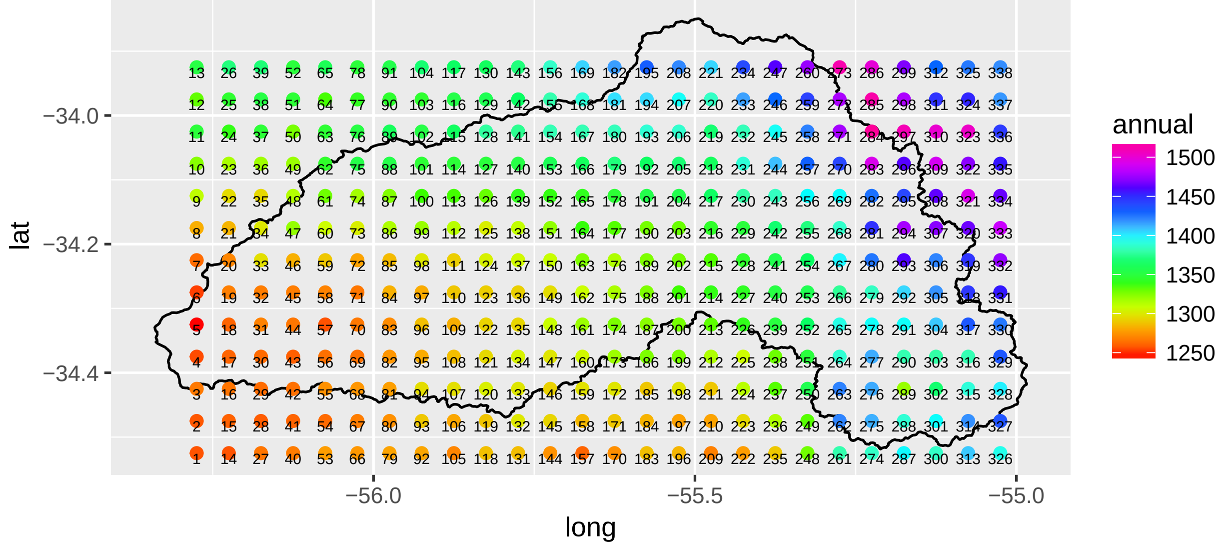
After looking at daily step, We will compare the mean annual rainfall data of CHIRPS precipitation data. The R script is the following:

```
# mean annual rainfall

SLprcp_dates_new <- zoo(SLprcp_dates[, 4:ncol(SLprcp_dates)],
    order.by = Dates, frequency = 1)
annualchirps <- aggregate(SLprcp_dates_new, list(year = format(time(SLprcp_dates_new),
    "%Y")), sum, na.rm = T)
```

```r
mean_a_P_chirps <- apply(annualchirps, 2, mean, na.rm = T)
meanchirps_df <- Statchirps_df
meanchirps_df$annual <- mean_a_P_chirps

## spatial plot of the mean annual rainfall of CHIRPS
## data
pl <- ggplot() + geom_polygon(data = SL, aes(x = long, y = lat,
    group = group), colour = "black", fill = NA) + coord_equal() +
    geom_point(data = meanchirps_df, aes(Long, Lat, colour = annual),
        size = 2) + geom_text(data = meanchirps_df, aes(Long,
    Lat, label = Stations, vjust = 1), size = 2, check_overlap = FALSE) +
    scale_colour_gradientn(colors = rainbow(10))
```

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

```r
pl
```



The spatial plot of mean annual rainfall with CHIRPS data shows there is an increase in rainfall from south-west to north-est, same pattern observed with mean daily rainfall data. There are more rainfall in the mountain area. The range of annual rainfall for the catchment is between 1250 mm per year and 1500 mm per year.

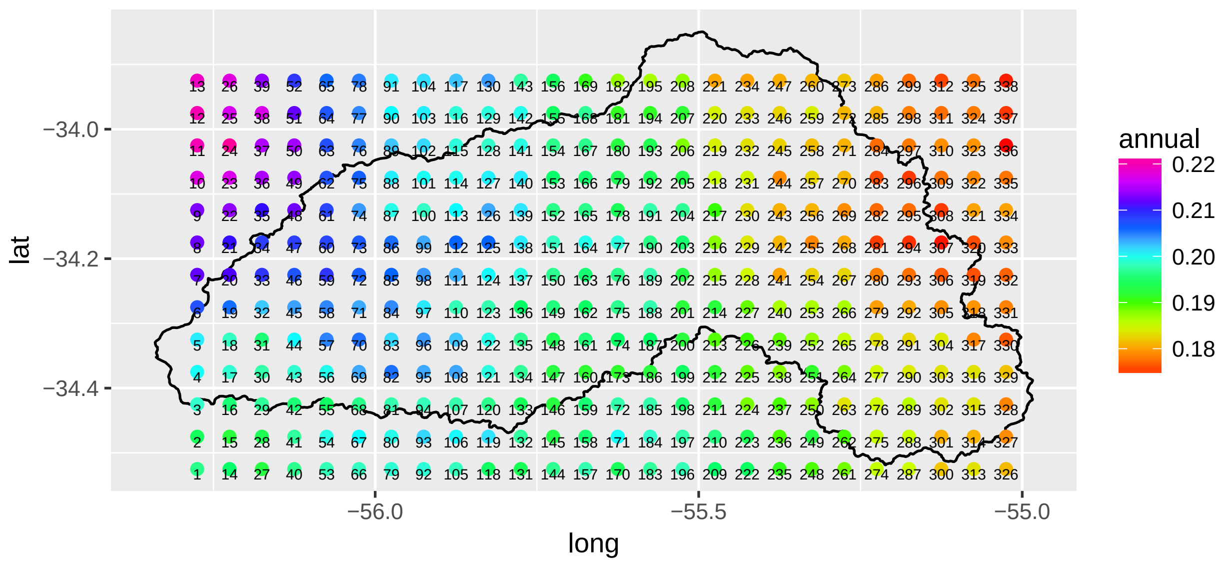
Coefficient of variation of annual rainfall

We will have a look at the coefficient of variation (CV) of annual rainfall. The CV is defined as the ratio of the standard deviation (sd) to the mean. In this case, it shows the extent of variability in relation to the mean of the annual rainfall. The script R is the following:

```
# CV annual rainfall
CV_a_P_chirps <- apply(annualchirps, 2, function(x) sd(x)/mean(x))
CV_df_chirps <- Statchirps_df
CV_df_chirps$annual <- CV_a_P_chirps

## spatial plot of CV rainfall of CHIRPS data
pl <- ggplot() + geom_polygon(data = SL, aes(x = long, y = lat,
    group = group), colour = "black", fill = NA) + coord_equal() +
    geom_point(data = CV_df_chirps, aes(Long, Lat, colour = annual),
        size = 2) + geom_text(data = CV_df_chirps, aes(Long,
    Lat, label = Stations, vjust = 1), size = 2) + scale_colour_gradientn(colors = rainbow(10))
```

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

```
pl
```



The CV annual rainfall map show a higher variability of annual rainfall in the west than in the east of the subcatchment. The annual rainfall is less variable in the mountains than in the plains.

Calculate basic statistics based on seasonality

After analyzing the chirps rainfall data at annual step, we will do some statistic calculation at seasonal step.

Function for seasonal analysis

The following R script define the function with the division of the 4 seasons period.

```
# Seasonal analysis
seasonschirps <- sapply(time(SLprcp_dates_new), function(x) ifelse(month(x) >
    11 || month(x) < 3, "Summer", ifelse(month(x) < 5, "Autumn",
    ifelse(month(x) < 8, "Winter", "Spring"))))
```

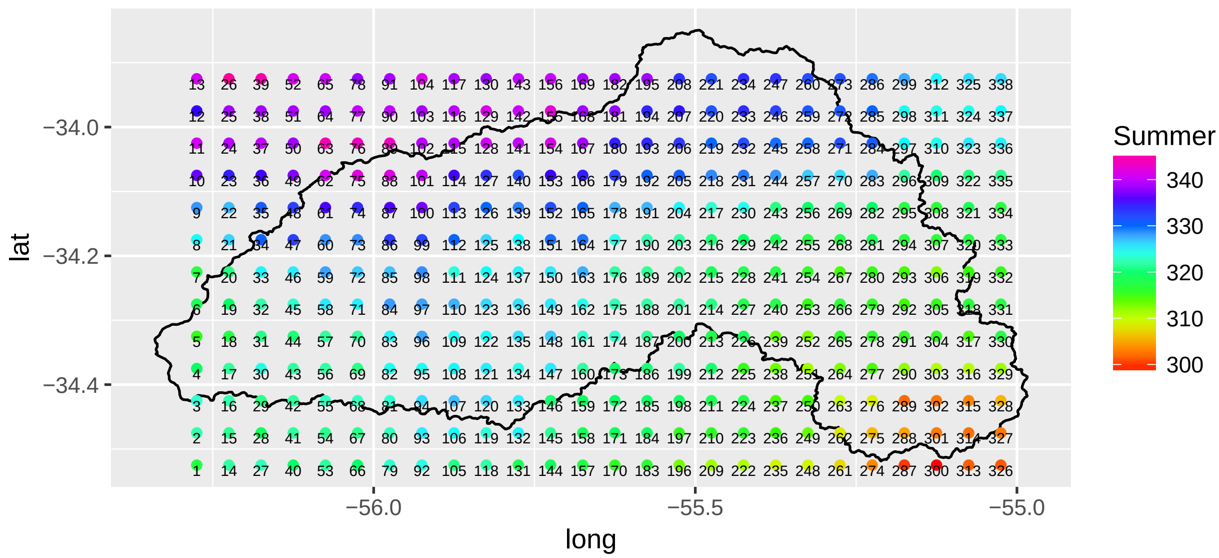
Calculate mean by season

We will analyze the chirps rainfall pattern looking at annual rainfall by season. The script is the following:

```
# summarise by season
season_P_chirps <- aggregate(coredata(SLprcp_dates_new),
    list(year = format(time(SLprcp_dates_new), "%Y"), season = seasonschirps),
    sum, na.rm = T)
# calculate mean by season
mean_s_P_chirps <- aggregate(season_P_chirps[, 3:340], list(season = season_P_chirps$season),
    mean)
for (i in 1:nrow(mean_s_P_chirps)) {
    meanchirps_df[, i + 4] <- t(mean_s_P_chirps[i, 2:339])  #error
    colnames(meanchirps_df)[i + 4] <- mean_s_P_chirps$season[i]
}
# spatial plot Summer
pl <- ggplot() + geom_polygon(data = SL, aes(x = long, y = lat,
    group = group), colour = "black", fill = NA) + coord_equal() +
    geom_point(data = meanchirps_df, aes(Long, Lat, colour = Summer)) +
    geom_text(data = meanchirps_df, aes(Long, Lat, label = Stations,
        vjust = 1), size = 2) + scale_colour_gradientn(colors = rainbow(10))
```

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

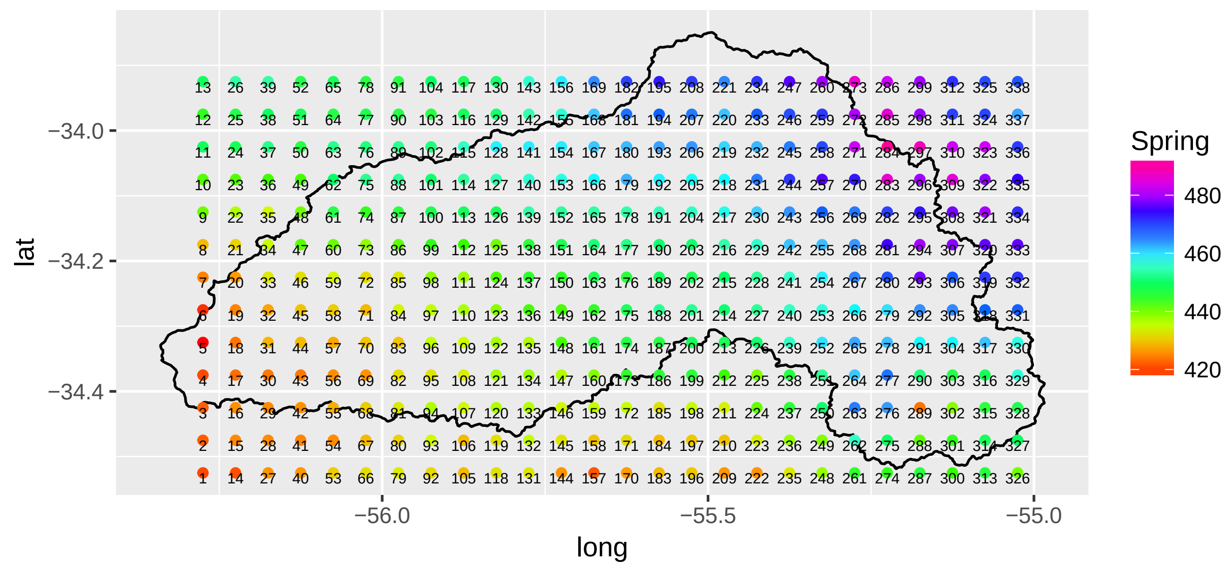
```
pl
```




```
# Spring

pl <- ggplot() + geom_polygon(data = SL, aes(x = long, y = lat,
    group = group), colour = "black", fill = NA) + coord_equal() +
    geom_point(data = meanchirps_df, aes(Long, Lat, colour = Spring)) +
    geom_text(data = meanchirps_df, aes(Long, Lat, label = Stations,
        vjust = 1), size = 2) + scale_colour_gradientn(colors = rainbow(10))
```

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

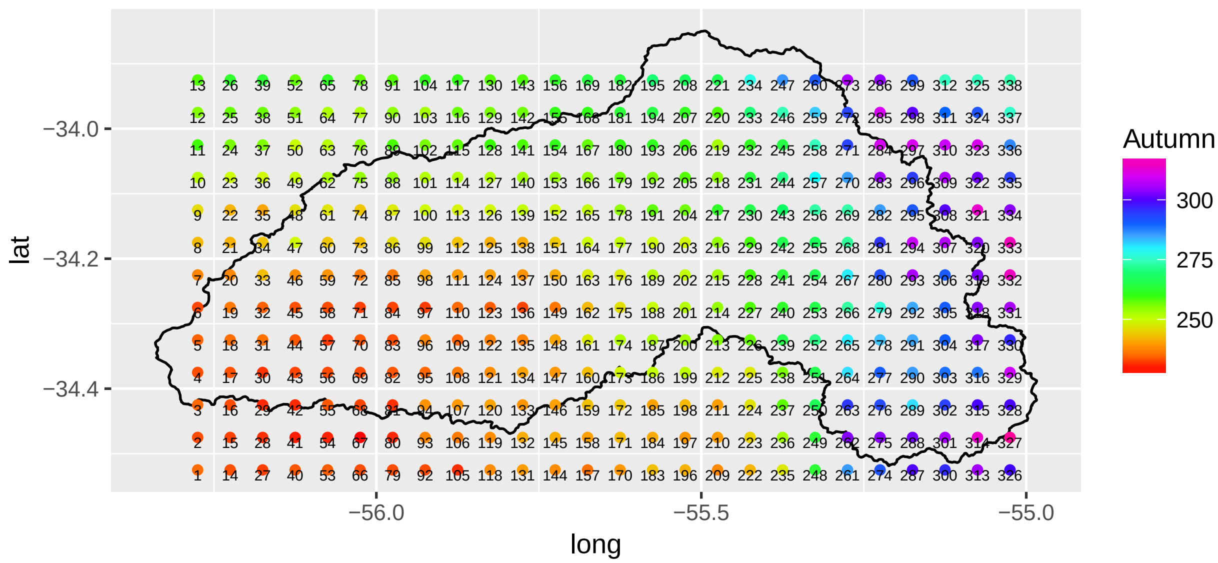
```
pl
```




```r
# Autumn
pl <- ggplot() + geom_polygon(data = SL, aes(x = long, y = lat,
    group = group), colour = "black", fill = NA) + coord_equal() +
    geom_point(data = meanchirps_df, aes(Long, Lat, colour = Autumn)) +
    geom_text(data = meanchirps_df, aes(Long, Lat, label = Stations,
        vjust = 1), size = 2) + scale_colour_gradientn(colors = rainbow(10))
```

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

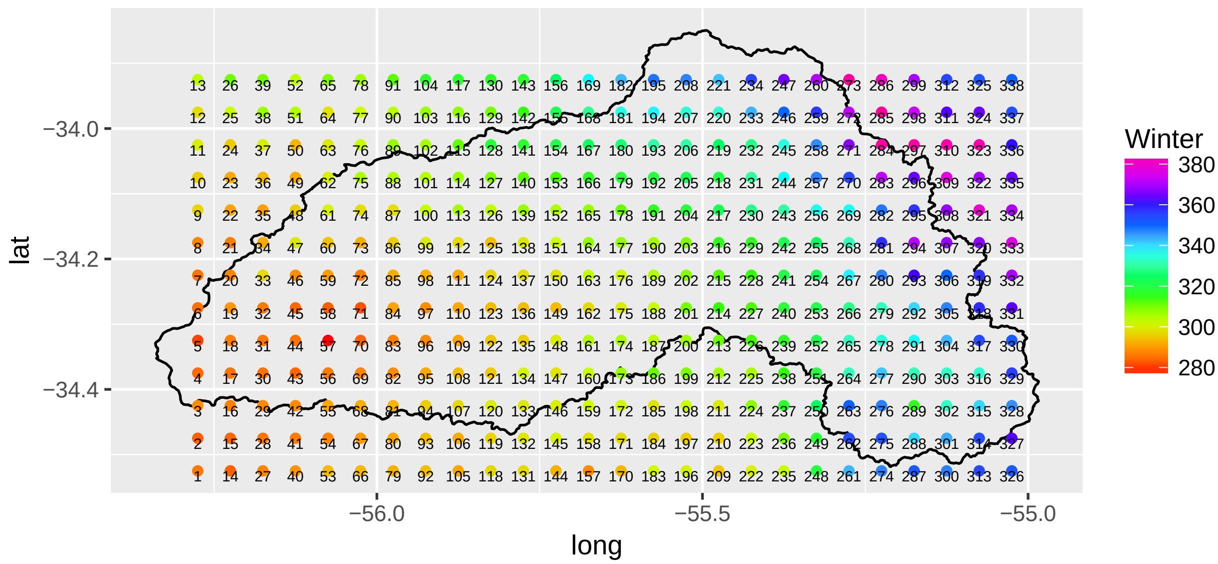
```r
pl
```




```r
# Winter
pl <- ggplot() + geom_polygon(data = SL, aes(x = long, y = lat,
    group = group), colour = "black", fill = NA) + coord_equal() +
    geom_point(data = meanchirps_df, aes(Long, Lat, colour = Winter)) +
    geom_text(data = meanchirps_df, aes(Long, Lat, label = Stations,
        vjust = 1), size = 2) + scale_colour_gradientn(colors = rainbow(10))
```

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

```r
pl
```



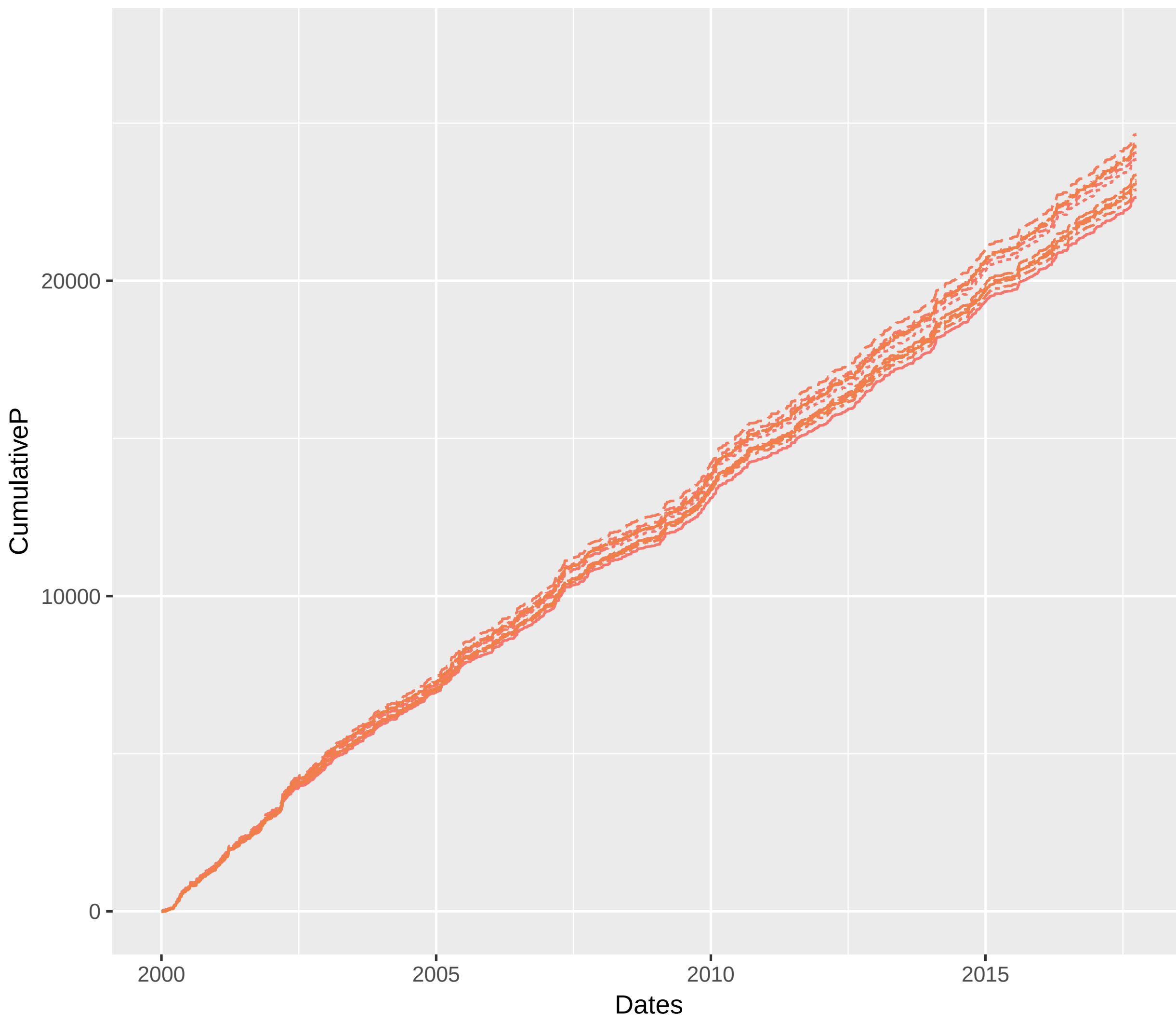
Summer shows a different rainfall pattern than the 3 others seasons. Indeed, the rainfall increase from south-east to nort-west, while the 3 others seasons shows an increase from south-west to north-east such as the annual pattern. The rainfall variability seems bigger in winter accross the catchment.

Calculate double mass curves

The double mass analysis is a common data analysis approach for investigating the behaviour of records made of meteoroligical data at a number of different locations. The R script is the following:

```r
# double mass curves set all missing data to 0
SLprcp_dates_cor <- SLprcp_dates_new
SLprcp_dates_NA <- SLprcp_dates_new
SLprcp_dates_NA[] <- 0
for (i in 1:338) {
    SLprcp_dates_NA[is.na(SLprcp_dates_cor[, i]), i] <- 1
    SLprcp_dates_cor[is.na(SLprcp_dates_cor[, i]), i] <- 0
}

NAcountchirps <- apply(SLprcp_dates_NA, 2, sum)
```

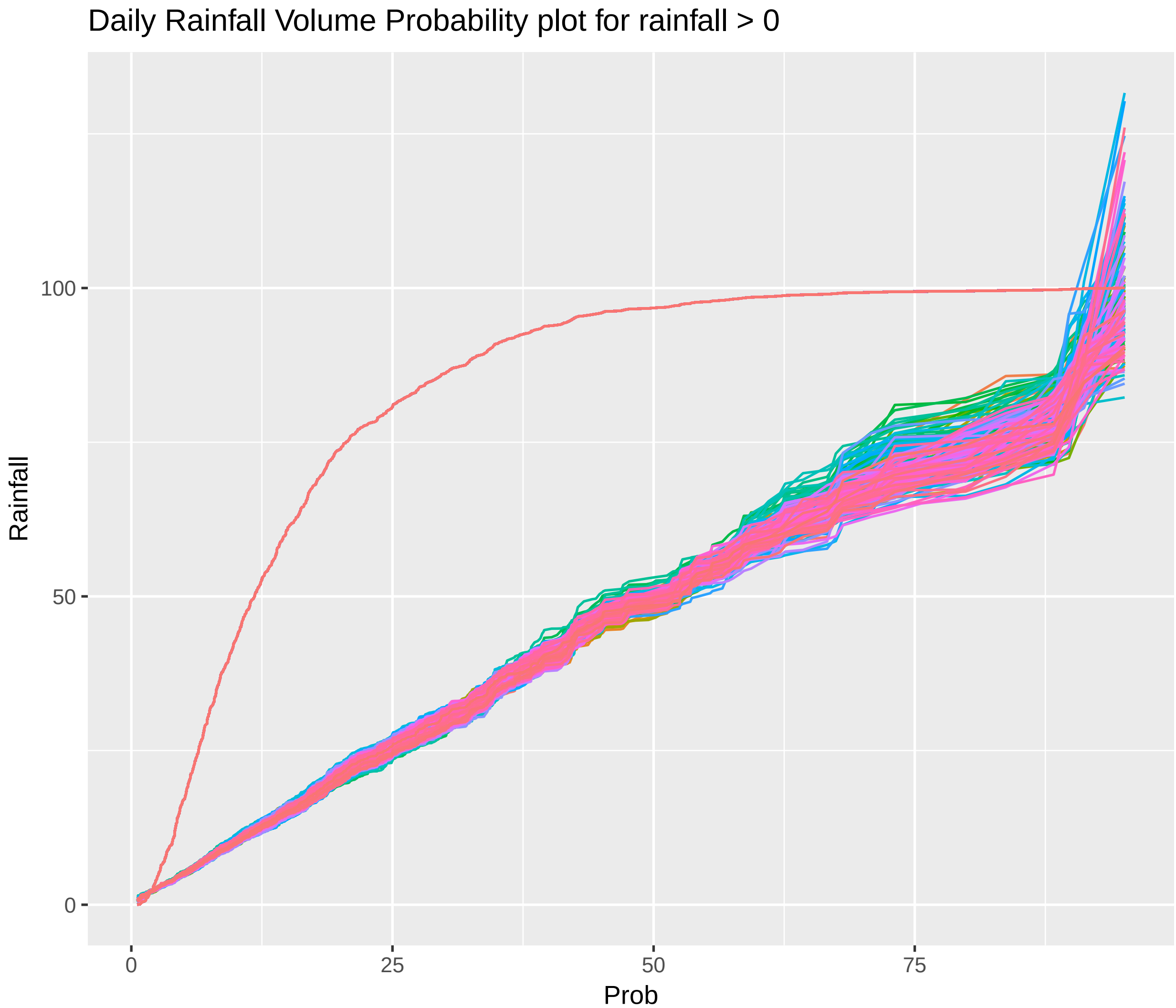
```r
# step 1 calculate cumulative mass curves
Cum_mass_chirps <- apply(SLprcp_dates_cor, 2, cumsum)

plot_df_chirps <- gather(as.data.frame(Cum_mass_chirps),
    key = Station, value = CumulativeP, 1:338)
plot_df_chirps$Dates <- rep(time(SLprcp_dates_new), 338)
maxP <- Cum_mass_chirps[nrow(Cum_mass_chirps), 1:338]
plot_df_chirps$maxP <- rep(maxP, each = nrow(SLprcp_dates_new))

# step 2 plot the cumulative mass curvess
pl <- ggplot(plot_df_chirps, aes(Dates, CumulativeP)) +
    geom_line(aes(colour = Station, linetype = Station)) +
    theme(legend.position = "none")
pl
```



As there are 338 chirps points, we did not plot the legend to have a better plot visualization.

The following R script is for plotting the frequency curves for rainfall greater than 0.The frequency curves for rainfall give us the probability of daily rain volume when rainfall is greater than 0.

```
# frequency curves for rainfall > 0
FDC_gen <- function(DATA) {
    FDC <- data.frame(probs = seq(0, 1, length = 1000) *
        100, rain = quantile(DATA[DATA > 0], probs = seq(0,
        1, length = 1000), na.rm = T))
    return(FDC)
}

Fcurves <- apply(SLprcp_dates_new, 2, FDC_gen)
# this is a list with a data.frame for each station
# str(Fcurves) # to have a look at Fcurves
F_df_chirps <- do.call(cbind, Fcurves)
F_df_chirps <- F_df_chirps[, -seq(3, 27, by = 2)]

F_df_pl_chirps <- gather(F_df_chirps, value = Rainfall,
    key = Station, 1:338)
colnames(F_df_pl_chirps)[1] <- "Prob"
F_df_pl_chirps$Station <- sapply(F_df_pl_chirps$Station,
    function(x) gsub(".rain", "", x))
f_plot_chirps <- ggplot(F_df_pl_chirps, aes(Prob, Rainfall)) +
    geom_line(aes(colour = Station)) + ggtitle("Daily Rainfall Volume Probability plot for rainfall > 0
    theme(legend.position = "none")

f_plot_chirps
```



PLOT: Not normal to have on data having another pattern ??!