When we try to estimate the correlation coefficient between multiple variables, the task is more complicated in order to obtain a simple and tidy result. A simple solution is to use the tidy() function from the *{broom}* package. In this post we are going to estimate the correlation coefficients between the annual precipitation of several Spanish cities and climate teleconnections indices: [download](https://dominicroye.github.io/files/teleconnections_indices.zip). The data of the teleconnections are preprocessed, but can be downloaded directly from [crudata.uea.ac.uk](https://crudata.uea.ac.uk/cru/data/pci.htm). The daily precipitation data comes from [ECA&D](https://www.ecad.eu/dailydata/index.php).

**Packages**

In this post we will use the following packages:

| **Package** | **Description** |
| --- | --- |
| tidyverse | Collection of packages (visualization, manipulation): ggplot2, dplyr, purrr, etc. |
| broom | Convert results of statistical functions (lm, t.test, cor.test, etc.) into tidy tables |
| fs | Provides a cross-platform, uniform interface to file system operations |
| lubridate | Easy manipulation of dates and times |

#install the packages if necessary

if(!require("tidyverse")) install.packages("tidyverse")

if(!require("broom")) install.packages("broom")

if(!require("fs")) install.packages("fs")

if(!require("lubridate")) install.packages("lubridate")

#load libraries

library(tidyverse)

library(broom)

library(fs)

library(lubridate)

**Import data**

First we have to import the daily precipitation of the selected weather stations.

1. Create a vector with all precipitation files using the function dir\_ls() of the *{fs}* package.
2. Import the data using the map\_df() function of the *{purrr}* package that applies another function to a vector or list, and joins them together in a single *data.frame*.
   1. Select the columns that interest us, b) Convert the date string into a date object using the ymd() function of the *{lubridate}* package, c) Create a new column *yr* with the years, d) Divide the precipitation values by 10 and reclassify absent values -9999 by NA, e) Finally, reclassify the ID of each weather station creating a factor with new labels.

More details about the use of the dir\_ls() and map\_df() functions can be found in this previous [post](https://dominicroye.github.io/en/2019/import-excel-sheets-%20with-r%20/).

#precipitation files

files <- dir\_ls(regexp = "txt")

files

## RR\_STAID001393.txt RR\_STAID001394.txt RR\_STAID002969.txt

## RR\_STAID003946.txt RR\_STAID003969.txt

#import all files and join them together

pr <- files %>% map\_df(read\_csv,skip=20)

## Parsed with column specification:

## cols(

## STAID = col\_double(),

## SOUID = col\_double(),

## DATE = col\_double(),

## RR = col\_double(),

## Q\_RR = col\_double()

## )

## Parsed with column specification:

## cols(

## STAID = col\_double(),

## SOUID = col\_double(),

## DATE = col\_double(),

## RR = col\_double(),

## Q\_RR = col\_double()

## )

## Parsed with column specification:

## cols(

## STAID = col\_double(),

## SOUID = col\_double(),

## DATE = col\_double(),

## RR = col\_double(),

## Q\_RR = col\_double()

## )

## Parsed with column specification:

## cols(

## STAID = col\_double(),

## SOUID = col\_double(),

## DATE = col\_double(),

## RR = col\_double(),

## Q\_RR = col\_double()

## )

## Parsed with column specification:

## cols(

## STAID = col\_double(),

## SOUID = col\_double(),

## DATE = col\_double(),

## RR = col\_double(),

## Q\_RR = col\_double()

## )

pr

## # A tibble: 133,343 x 5

## STAID SOUID DATE RR Q\_RR

##

## 1 1393 20611 19470301 0 0

## 2 1393 20611 19470302 5 0

## 3 1393 20611 19470303 0 0

## 4 1393 20611 19470304 33 0

## 5 1393 20611 19470305 15 0

## 6 1393 20611 19470306 0 0

## 7 1393 20611 19470307 85 0

## 8 1393 20611 19470308 3 0

## 9 1393 20611 19470309 0 0

## 10 1393 20611 19470310 0 0

## # ... with 133,333 more rows

#create levels for the factor

id <- unique(pr$STAID)

#the corresponding labels

lab <- c("Bilbao","Santiago","Barcelona","Madrid","Valencia")

#first changes

pr <- select(pr,STAID,DATE,RR)%>%

mutate(DATE=ymd(DATE),

RR=ifelse(RR==-9999,NA,RR/10),

STAID=factor(STAID,id,lab),

yr=year(DATE))

pr

## # A tibble: 133,343 x 4

## STAID DATE RR yr

##

## 1 Bilbao 1947-03-01 0 1947

## 2 Bilbao 1947-03-02 0.5 1947

## 3 Bilbao 1947-03-03 0 1947

## 4 Bilbao 1947-03-04 3.3 1947

## 5 Bilbao 1947-03-05 1.5 1947

## 6 Bilbao 1947-03-06 0 1947

## 7 Bilbao 1947-03-07 8.5 1947

## 8 Bilbao 1947-03-08 0.3 1947

## 9 Bilbao 1947-03-09 0 1947

## 10 Bilbao 1947-03-10 0 1947

## # ... with 133,333 more rows

We still need to filter and calculate the annual amount of precipitation. Actually, it is not correct to sum up precipitation without taking into account that there are missing values, but it should be enough for this practice. Then, we change the table format with the spread() function, passing from a long to a wide table, that is, we want to obtain one column per weather station.

pr\_yr <- filter(pr,DATE>="1950-01-01",DATE<"2018-01-01")%>%

group\_by(STAID,yr)%>%

summarise(pr=sum(RR,na.rm = TRUE))

pr\_yr

## # A tibble: 324 x 3

## # Groups: STAID [5]

## STAID yr pr

##

## 1 Bilbao 1950 1342

## 2 Bilbao 1951 1306.

## 3 Bilbao 1952 1355.

## 4 Bilbao 1953 1372.

## 5 Bilbao 1954 1428.

## 6 Bilbao 1955 1062.

## 7 Bilbao 1956 1254.

## 8 Bilbao 1957 968.

## 9 Bilbao 1958 1272.

## 10 Bilbao 1959 1450.

## # ... with 314 more rows

pr\_yr <- spread(pr\_yr,STAID,pr)

pr\_yr

## # A tibble: 68 x 6

## yr Bilbao Santiago Barcelona Madrid Valencia

##

## 1 1950 1342 1800. 345 NA NA

## 2 1951 1306. 2344. 1072. 798. NA

## 3 1952 1355. 1973. 415. 524. NA

## 4 1953 1372. 973. 683. 365. NA

## 5 1954 1428. 1348. 581. 246. NA

## 6 1955 1062. 1769. 530. 473. NA

## 7 1956 1254. 1533. 695. 480. NA

## 8 1957 968. 1599. 635. 424. NA

## 9 1958 1272. 2658. 479. 482. NA

## 10 1959 1450. 2847. 1006 665. NA

## # ... with 58 more rows

The next step is to import the climate teleconnection indices.

#teleconnections

telecon <- read\_csv("teleconnections\_indices.csv")

## Parsed with column specification:

## cols(

## yr = col\_double(),

## NAO = col\_double(),

## WeMO = col\_double(),

## EA = col\_double(),

## `POL-EUAS` = col\_double(),

## `EATL/WRUS` = col\_double(),

## MO = col\_double(),

## SCAND = col\_double(),

## AO = col\_double()

## )

telecon

## # A tibble: 68 x 9

## yr NAO WeMO EA `POL-EUAS` `EATL/WRUS` MO SCAND AO

##

## 1 1950 0.49 0.555 -0.332 0.0217 -0.0567 0.335 0.301 -1.99e-1

## 2 1951 -0.07 0.379 -0.372 0.402 -0.419 0.149 -0.00667 -3.65e-1

## 3 1952 -0.37 0.693 -0.688 -0.0117 -0.711 0.282 0.0642 -6.75e-1

## 4 1953 0.4 -0.213 -0.727 -0.0567 -0.0508 0.216 0.0233 -1.64e-2

## 5 1954 0.51 1.20 -0.912 0.142 -0.318 0.386 0.458 -5.83e-4

## 6 1955 -0.64 0.138 -0.824 -0.0267 0.154 0.134 0.0392 -3.62e-1

## 7 1956 0.17 0.617 -1.29 -0.197 0.0617 0.256 0.302 -1.63e-1

## 8 1957 -0.02 0.321 -0.952 -0.638 -0.167 0.322 -0.134 -3.42e-1

## 9 1958 0.12 0.941 -0.243 0.138 0.661 0.296 0.279 -8.68e-1

## 10 1959 0.49 -0.055 -0.23 -0.0142 0.631 0.316 0.725 -7.62e-2

## # ... with 58 more rows

Finally we need to join both tables by year.

data\_all <- left\_join(pr\_yr,telecon,by="yr")

data\_all

## # A tibble: 68 x 14

## yr Bilbao Santiago Barcelona Madrid Valencia NAO WeMO EA

##

## 1 1950 1342 1800. 345 NA NA 0.49 0.555 -0.332

## 2 1951 1306. 2344. 1072. 798. NA -0.07 0.379 -0.372

## 3 1952 1355. 1973. 415. 524. NA -0.37 0.693 -0.688

## 4 1953 1372. 973. 683. 365. NA 0.4 -0.213 -0.727

## 5 1954 1428. 1348. 581. 246. NA 0.51 1.20 -0.912

## 6 1955 1062. 1769. 530. 473. NA -0.64 0.138 -0.824

## 7 1956 1254. 1533. 695. 480. NA 0.17 0.617 -1.29

## 8 1957 968. 1599. 635. 424. NA -0.02 0.321 -0.952

## 9 1958 1272. 2658. 479. 482. NA 0.12 0.941 -0.243

## 10 1959 1450. 2847. 1006 665. NA 0.49 -0.055 -0.23

## # ... with 58 more rows, and 5 more variables: `POL-EUAS` ,

## # `EATL/WRUS` , MO , SCAND , AO

**Correlation test**

A correlation test between paired samples can be done with the cor.test() function of R Base. In this case between the annual precipitation of Bilbao and the NAO index.

cor\_nao\_bil <- cor.test(data\_all$Bilbao,data\_all$NAO,

method="spearman")

## Warning in cor.test.default(data\_all$Bilbao, data\_all$NAO, method =

## "spearman"): Cannot compute exact p-value with ties

cor\_nao\_bil

##

## Spearman's rank correlation rho

##

## data: data\_all$Bilbao and data\_all$NAO

## S = 44372, p-value = 0.2126

## alternative hypothesis: true rho is not equal to 0

## sample estimates:

## rho

## 0.1531149

str(cor\_nao\_bil)

## List of 8

## $ statistic : Named num 44372

## ..- attr(\*, "names")= chr "S"

## $ parameter : NULL

## $ p.value : num 0.213

## $ estimate : Named num 0.153

## ..- attr(\*, "names")= chr "rho"

## $ null.value : Named num 0

## ..- attr(\*, "names")= chr "rho"

## $ alternative: chr "two.sided"

## $ method : chr "Spearman's rank correlation rho"

## $ data.name : chr "data\_all$Bilbao and data\_all$NAO"

## - attr(\*, "class")= chr "htest"

We see that the result is in an unmanageable and untidy format. It is a console summary of the correlation with all the statistical parameters necessary to get a conclusion about the relationship. The orginal structure is a list of vectors. However, the tidy() function of the *{broom}* package allows us to convert the result into a table format.

tidy(cor\_nao\_bil)

## # A tibble: 1 x 5

## estimate statistic p.value method alternative

##

## 1 0.153 44372. 0.213 Spearman's rank correlation rho two.sided

**Apply the correlation test to multiple variables**

The objective is to apply the correlation test to all weather stations and climate teleconnection indices.

First, we must pass the table to the long format, that is, create a column/variable for the city and for the value of the corresponding precipitation. Then we repeat the same for the teleconnections indices.

data <- gather(data\_all,city,pr,Bilbao:Valencia)%>%

gather(telecon,index,NAO:AO)

data

## # A tibble: 2,720 x 5

## yr city pr telecon index

##

## 1 1950 Bilbao 1342 NAO 0.49

## 2 1951 Bilbao 1306. NAO -0.07

## 3 1952 Bilbao 1355. NAO -0.37

## 4 1953 Bilbao 1372. NAO 0.4

## 5 1954 Bilbao 1428. NAO 0.51

## 6 1955 Bilbao 1062. NAO -0.64

## 7 1956 Bilbao 1254. NAO 0.17

## 8 1957 Bilbao 968. NAO -0.02

## 9 1958 Bilbao 1272. NAO 0.12

## 10 1959 Bilbao 1450. NAO 0.49

## # ... with 2,710 more rows

To apply the test to all cities, we need the corresponding groupings. Therefore, we use the group\_by() function for indicating the two groups: *city* and *telecon*. In addition, we apply the nest() function of the *{tidyr}* package (*{tidyverse}* collection) with the aim of creating lists of tables nested per row. In other words, in each row of each city and teleconnection index we will have a new table that contains the year, the precipitation value and the value of each teleconection, correspondingly.

data\_nest <- group\_by(data,city,telecon)%>%nest()

data\_nest

## # A tibble: 40 x 3

## city telecon data

##

## 1 Bilbao NAO

## 2 Santiago NAO

## 3 Barcelona NAO

## 4 Madrid NAO

## 5 Valencia NAO

## 6 Bilbao WeMO

## 7 Santiago WeMO

## 8 Barcelona WeMO

## 9 Madrid WeMO

## 10 Valencia WeMO

## # ... with 30 more rows

str(slice(data\_nest,1))

## Classes 'tbl\_df', 'tbl' and 'data.frame': 1 obs. of 3 variables:

## $ city : chr "Bilbao"

## $ telecon: chr "NAO"

## $ data :List of 1

## ..$ :Classes 'tbl\_df', 'tbl' and 'data.frame': 68 obs. of 3 variables:

## .. ..$ yr : num 1950 1951 1952 1953 1954 ...

## .. ..$ pr : num 1342 1306 1355 1372 1428 ...

## .. ..$ index: num 0.49 -0.07 -0.37 0.4 0.51 -0.64 0.17 -0.02 0.12 0.49 ...

The next step is to create a function, in which we define the correlation test and pass it to the clean format using the tidy() function, which we apply to each groupings.

cor\_fun <- function(df) cor.test(df$pr,df$index,method="spearman")%>%tidy()

Now we only have to apply our function to the column that contains the tables for each combination between city and teleconnection. To do this, we use the map() function that applies another function to a vector or list. What we do is create a new column that contains the result, a statistical summary table, for each combination.

data\_nest <- mutate(data\_nest, model = map(data,cor\_fun))

data\_nest

## # A tibble: 40 x 4

## city telecon data model

##

## 1 Bilbao NAO

## 2 Santiago NAO

## 3 Barcelona NAO

## 4 Madrid NAO

## 5 Valencia NAO

## 6 Bilbao WeMO

## 7 Santiago WeMO

## 8 Barcelona WeMO

## 9 Madrid WeMO

## 10 Valencia WeMO

## # ... with 30 more rows

str(slice(data\_nest,1))

## Classes 'tbl\_df', 'tbl' and 'data.frame': 1 obs. of 4 variables:

## $ city : chr "Bilbao"

## $ telecon: chr "NAO"

## $ data :List of 1

## ..$ :Classes 'tbl\_df', 'tbl' and 'data.frame': 68 obs. of 3 variables:

## .. ..$ yr : num 1950 1951 1952 1953 1954 ...

## .. ..$ pr : num 1342 1306 1355 1372 1428 ...

## .. ..$ index: num 0.49 -0.07 -0.37 0.4 0.51 -0.64 0.17 -0.02 0.12 0.49 ...

## $ model :List of 1

## ..$ :Classes 'tbl\_df', 'tbl' and 'data.frame': 1 obs. of 5 variables:

## .. ..$ estimate : num 0.153

## .. ..$ statistic : num 44372

## .. ..$ p.value : num 0.213

## .. ..$ method : chr "Spearman's rank correlation rho"

## .. ..$ alternative: chr "two.sided"

How can we undo the list of tables in each row of our table?

First we eliminate the column with the data and then simply we can apply the unnest() function.

corr\_pr <- select(data\_nest,-data)%>%unnest()

corr\_pr

## # A tibble: 40 x 7

## city telecon estimate statistic p.value method alternative

##

## 1 Bilbao NAO 0.153 44372. 0.213 Spearman's rank~ two.sided

## 2 Santia~ NAO -0.181 61902. 0.139 Spearman's rank~ two.sided

## 3 Barcel~ NAO -0.0203 53460. 0.869 Spearman's rank~ two.sided

## 4 Madrid NAO -0.291 64692. 0.0169 Spearman's rank~ two.sided

## 5 Valenc~ NAO -0.113 27600. 0.422 Spearman's rank~ two.sided

## 6 Bilbao WeMO 0.404 31242. 0.000706 Spearman's rank~ two.sided

## 7 Santia~ WeMO 0.332 35014. 0.00594 Spearman's rank~ two.sided

## 8 Barcel~ WeMO 0.0292 50862 0.813 Spearman's rank~ two.sided

## 9 Madrid WeMO 0.109 44660 0.380 Spearman's rank~ two.sided

## 10 Valenc~ WeMO -0.252 31056. 0.0688 Spearman's rank~ two.sided

## # ... with 30 more rows

The result is a table in which we can see the correlations and their statistical significance for each city and teleconnection index.

**Heatmap of the results**

Finally, we make a heatmap of the obtained result. But, previously we create a column that indicates whether the correlation is significant with p-value less than 0.05.

corr\_pr <- mutate(corr\_pr,sig=ifelse(p.value<0.05,"Sig.","Non Sig."))

ggplot()+

geom\_tile(data=corr\_pr,

aes(city,telecon,fill=estimate),

size=1,

colour="white")+

geom\_tile(data=filter(corr\_pr,sig=="Sig."),

aes(city,telecon),

size=1,

colour="black",

fill="transparent")+

geom\_text(data=corr\_pr,

aes(city,telecon,label=round(estimate,2),

fontface=ifelse(sig=="Sig.","bold","plain")))+

scale\_fill\_gradient2(breaks=seq(-1,1,0.2))+

labs(x="",y="",fill="",p.value="")+

theme\_minimal()+

theme(panel.grid.major = element\_blank(),

panel.border = element\_blank(),

panel.background = element\_blank(),

axis.ticks = element\_blank())