

ESCENARIOS CLIMÁTICOS DE CORTO PLAZO

Un Caso de Estudio en Argentina y Uruguay

Paso a Paso

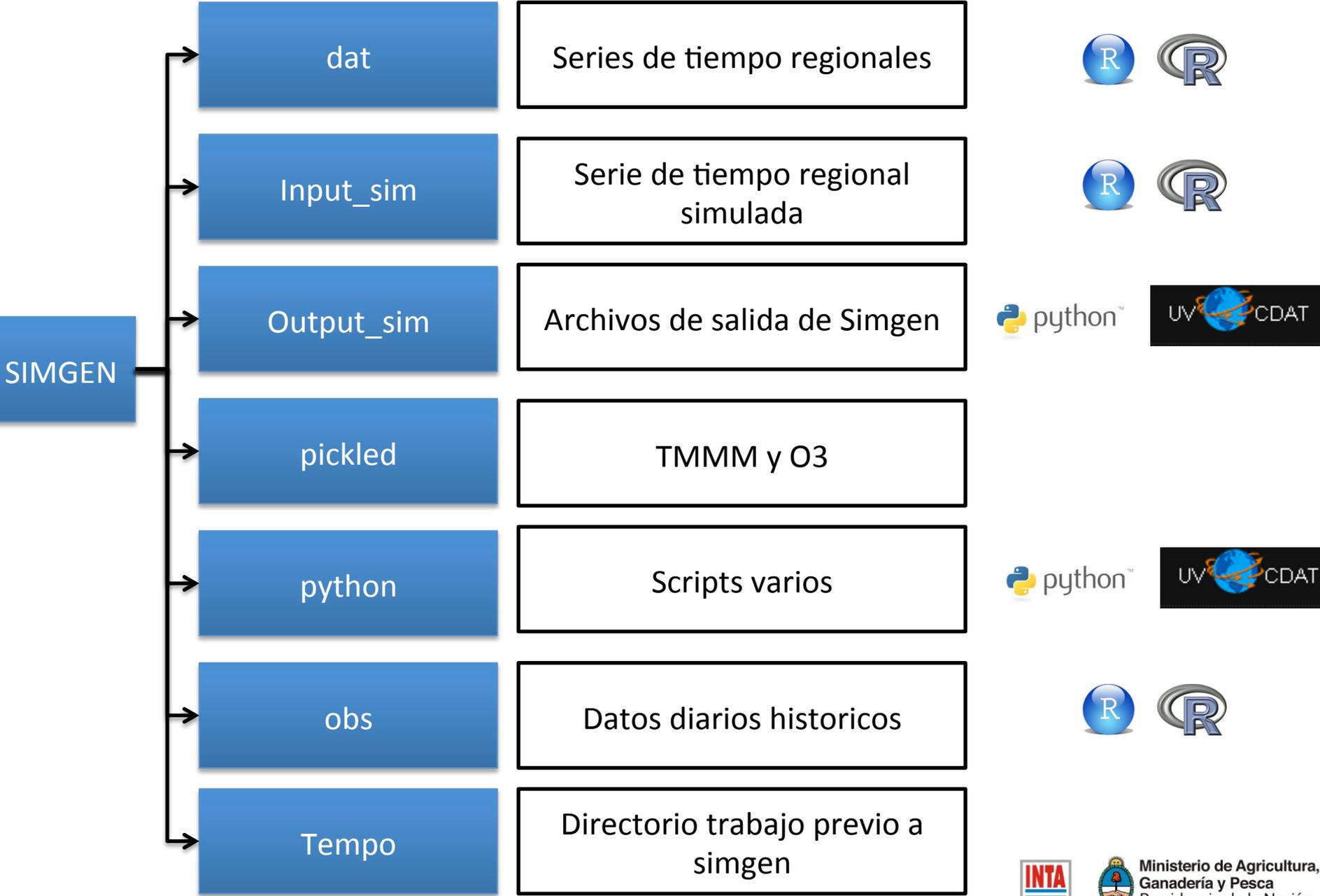


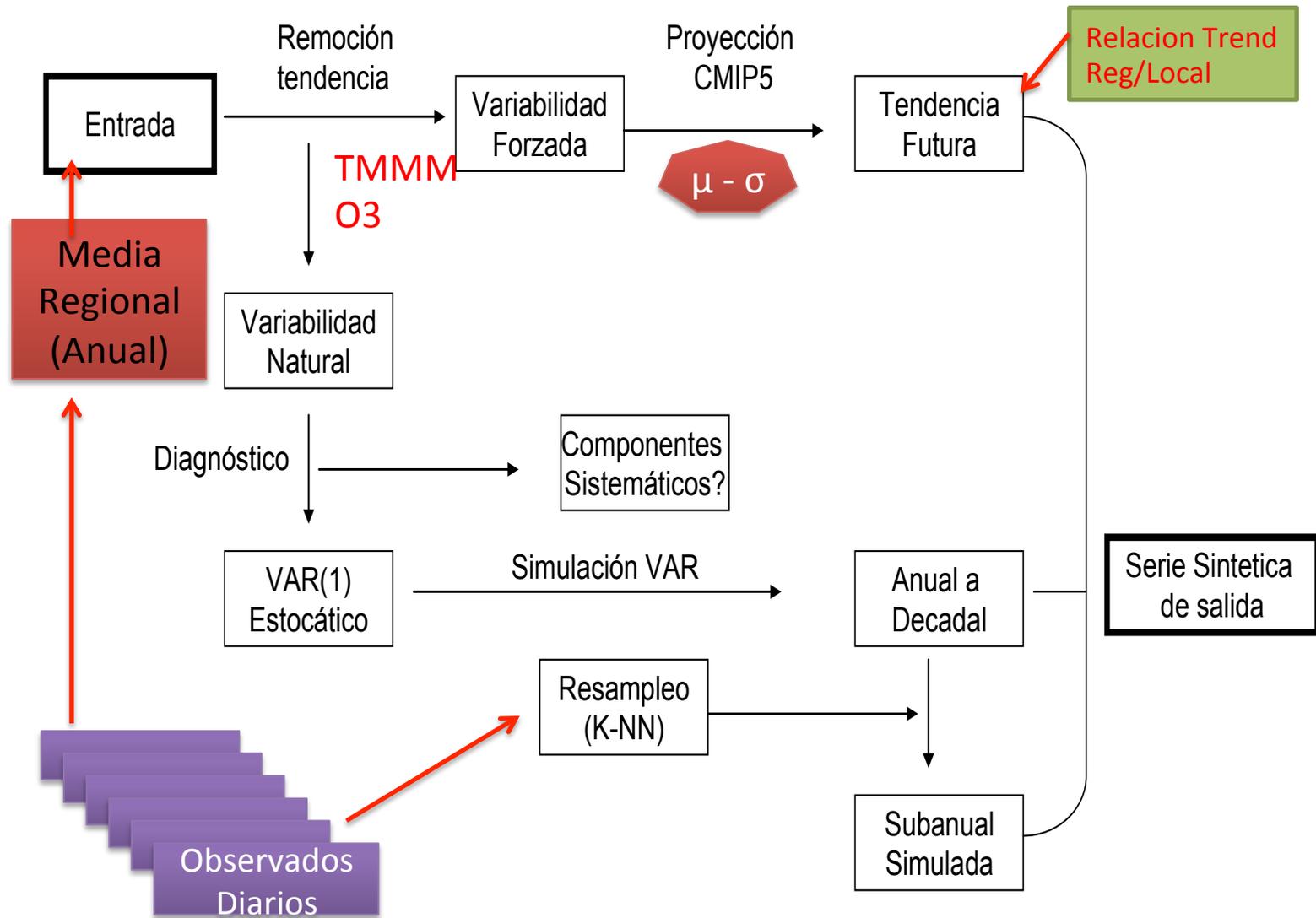
Ministerio de Agricultura,
Ganadería y Pesca
Presidencia de la Nación

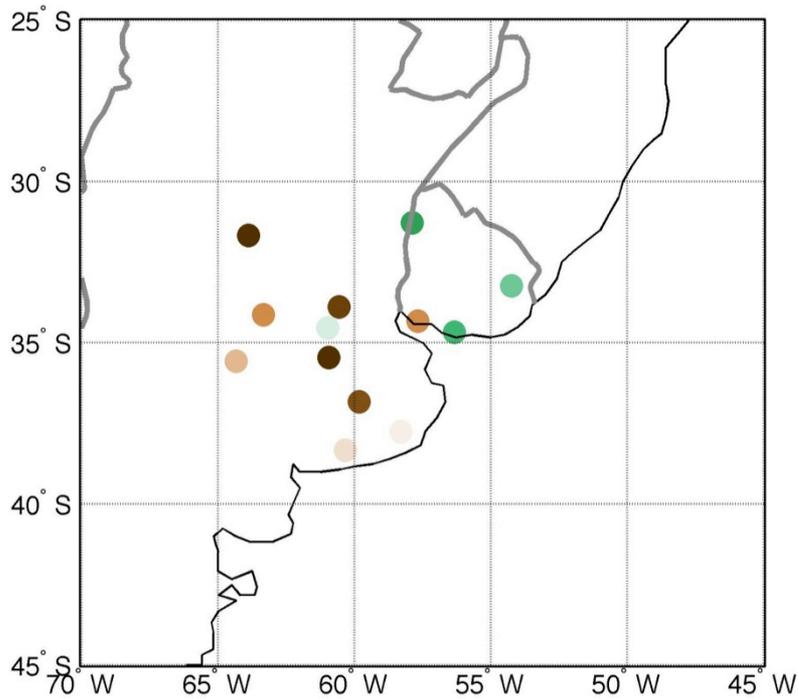
Hoja de ruta

- Armado estructura de directorios
- Preparación de datos de entrada
 - Formato de datos diarios de estaciones
 - Media regional
 - Descomposición de la variabilidad climática
- Ajuste modelo VAR
- Simulación estocástica









Argentina

- Azul
- Balcarce
- Junin
- Nueve de Julio
- Pergamino
- Tres Arroyos

Uruguay

- Las Brujas
- La Estanzuela
- Salto
- Treinta y Tres

Estructura de los datos diarios observados

2 archivos por estación (ST y DAT)

ST

DAT

[SITE]							
RAAZ		1931	1	13.3	30.2	0	29.2
[LAT, LON and ALT]		1931	2	16.9	27.8	11.7	23.4
		1931	3	17.4	27	1.5	22
		1931	4	18.2	30.7	0	25.1
	-36.833 -59.833 147	1931	5	12.7	20.8	0	20.1
[WEATHER FILES]		1931	6	6.8	21.9	0	27.5
RAAZOBS.DAT		1931	7	8.4	26.9	0	30.4
[FORMAT]		1931	8	12.2	28.4	0	28.4
YEAR JDAY MIN MAX RAIN RAD		1931	9	17	30.8	0	26.1
[END]		1931	10	14.3	28	6	26
		1931	11	17.1	34.2	0	29
		1931	12	16.8	32.2	16.2	27.5
		1931	13	17.6	27.4	8.6	21.9
		1931	14	17.8	30.8	4.7	25.1
		1931	15	17.3	26.2	7.6	20.7
		1931	16	17.8	28.3	0	22.5
		1931	17	12.6	34.6	0	32.5
		1931	18	17.5	29.4	0.1	23.8

Información general de la estación

Datos observados

En Rstudio: Abrir el script 01_prep_ntcc_chile.r

The screenshot shows the RStudio interface with the following components:

- Source Editor:** Displays the script `01_prep_ntcc_chile.R` with the following code:

```
1 ### Script para transformar los datos al formato ACRU
2 ## libraries#####
3 library(lattice)
4 library(dse)
5 library(ggplot2)
6 library(seas)
7 library(date)
8 #####
9 # Preparaci?n de datos para la media regional
10 # En este caso uso de ejemplo las estaciones de Argentina y Uruguay
11 #####
12 ### Leo los archivos diarios del LARS Observado
13
14 #setwd("C:/IRI_OCT/")
15 setwd("~/Compartido/singen")
16 files <- list.files("daily", pattern=".st")
17 codigo.estacion.inicial <- 2000 # debe comenzar en 1000 para manten
18
19 ### calculo el periodo en que coinciden todas las estaciones
20 inilist <- vector(length=length(files))
21
```
- Environment Pane:** Shows "Global Environment" with the message "Environment is empty".
- Files Pane:** Shows a list of installed packages with columns for Name, Description, and Version.

Name	Description	Version
colospace	Color Space Manipulation	1.2-4
dichromat	Color Schemes for Dichromats	2.0-0
digest	Create cryptographic hash digests of R objects	0.6.4
dse	Dynamic Systems Estimation (time series package)	2013.3-2
ggplot2	An implementation of the Grammar of Graphics	1.0.0
gtable	Arrange grobs in tables.	0.1.2
labeling	Axis Labeling	0.2
manipulate	Interactive Plots for RStudio	0.98.953
munsell	Munsell colour system	0.4.2
plyr	Tools for splitting, applying and combining data	1.8.1
proto	Prototype object-based programming	0.3-10
RColorBrewer	ColorBrewer palettes	1.0-5
Rcpp	Seamless R and C++ Integration	0.11.2
reshape2	Flexibly reshape data: a reboot of the	1.4

Instalar paquetes:

Seas, date y reshape

```
01_prep_ntcc_chile.R x
Source on Save
2 > ## Cargo las librerías necesarias #####
3 library(lattice); library(dse) ; library(ggplot2); library(seas); library(date)
4 > #####
5 # Preparación de datos para la media regional
6 # En este caso uso de ejemplo las estaciones de Argentina y Uruguay
7 > #####
8 ### Leo los archivos diarios del LARS Observado
9 # seteo el directorio de trabajo
10 setwd("/media/sf_Compartido/singen")
11 files <- list.files("daily", pattern=".st")
12 # debe comenzar en 1000 para mantener los 4 digitos
13 codigo.estacion.inicial <- 2000
14
15 ### calculo el periodo en que coinciden todas las estaciones
16 inilist <- vector(length=length(files))
17 endlis <- vector(length=length(files))
18 n <- 1
19 > for (eachfile in files) { # eachfile="RATROBS.st"
20   dat <- read.lars(paste("daily/", eachfile, sep=""))
21   temp <- subset(dat, dat$year==min(dat$year))
22   inilist[n] <- ifelse(length(temp$year) <=364, min(dat$year)+1, min(dat$year))
23   temp <- subset(dat, dat$year==max(dat$year))
24   endlis[n] <- ifelse(length(temp$year) <=364, max(dat$year)-1, max(dat$year))
25   n <- n+1
26 }
27
28 year_ini <- max(inilist)
29 year_end <- min(endlis)
30
```

- Carga de paquetes R
- Setear directorio de trabajo
- Listar archivos disponibles
- Chequear períodos en común

```

31 for (eachfile in files) { # eachfile="RATROBS.st"
32   dat <- read.lars(paste("daily/", eachfile, sep=""))
33   dat <- subset(dat, dat$year >= year_ini & dat$year <= year_end)
34   dat$cod <- "00000007"
35   dat$datetex <- as.character(dat$date)
36   dat$mes <- substr(dat$datetex, 6,7)
37   dat$dia <- substr(dat$datetex, 9,10)
38   dat$col1 <- paste(dat$cod, dat$year, dat$mes, dat$dia, sep="")
39   dat$dummy <- " 4.9 26.33 95.00 47.00 138.2"
40   dat$precip <- as.numeric(dat$precip)
41   rain <- format(dat$precip, digits = 3, nsmall=1, justify = c("right"))
42   tmax <- format(dat$t_max, digits = 3, trim=FALSE, justify = c("right"), width=5)
43   tmin <- format(dat$t_min, digits = 3, nsmall=1, justify = c("right"), width=5)
44   acru <- cbind(dat$col1, rain, tmax, tmin, dat$dummy)
45   ##### Anualizacion Lluvia #####
46     datos.rain <- data.frame(cbind(dat$year, dat$precip))
47     datos.rain.na <- na.omit(datos.rain)
48     names(datos.rain.na) <- c("year", "rain")
49     rain.anual <- aggregate(datos.rain.na$rain, by = list(datos.rain.na$year), FUN = "sum", na.omit=TRUE)
50     names(rain.anual) <- c("year", "rain")
51     ## TMAX ##
52     datos.tmax <- data.frame(cbind(dat$year, dat$t_max))
53     datos.tmax.na <- na.omit(datos.tmax)
54     names(datos.tmax.na) <- c("year", "tmax")
55     tmax.anual <- aggregate(datos.tmax.na$tmax, by = list(datos.tmax.na$year), FUN = "mean", na.omit=TRUE)
56     names(tmax.anual) <- c("year", "tmax")
57     ## TMIN ##
58     datos.tmin <- data.frame(cbind(dat$year, dat$t_min))
59     datos.tmin.na <- na.omit(datos.tmin)
60     names(datos.tmin.na) <- c("year", "tmin")
61     tmin.anual <- aggregate(datos.tmin.na$tmin, by = list(datos.tmin.na$year), FUN = "mean", na.omit=TRUE)
62     names(tmin.anual) <- c("year", "tmin")
63     ## junto todo
64     names(rain.anual) <- c("year", "rain")
65     datos.anual <- merge(rain.anual, tmin.anual)
66     datos.anual2 <- merge(datos.anual, tmax.anual)
67     ##### los archivos de datos observados deben llamarse "obshis_XXXX.txt" donde XXXX son 4 digitos
68     ##### Creo directorio ACRU y guardo lista de estaciones, acru y anuales
69     dir.create(file.path("acru"), showWarnings = FALSE)
70     filesal.acru <- paste("acru/", "obshis_", codigo.estacion.inicial, ".txt", sep="")
71     filesal.anual <- paste("acru/", eachfile, "_ANUAL.TXT", sep="")
72     filesal.anual2 <- paste("acru/", "TODOS_ANUAL.TXT2", sep="")
73     codigos <- data.frame(eachfile, codigo.estacion.inicial)
74     names(codigos) <- c("estacion", "codigo_obshist")
75     write.table(acru, filesal.acru, quote = FALSE, col.names=FALSE, row.names=FALSE, sep=" ")
76     write.table(datos.anual2, filesal.anual, quote = FALSE, col.names=FALSE, row.names=FALSE, sep=" ")
77     write.table(codigos, "acru/codigos.txt", append=TRUE,col.names=FALSE, row.names=FALSE, sep=" ")
78     datos.anual2$estacion <- eachfile
79     write.table(datos.anual2, filesal.anual2, append= TRUE, quote = FALSE, col.names=FALSE, row.names=FALSE, sep=" ")
80     codigo.estacion.inicial <- codigo.estacion.inicial+1
81 }

```

← Da el formato que lee simgen

Anualiza las variables

Escribe los archivos ACRU



Como resultado del loop se crea el directorio ACRU

Archivo Editar Ver Ir Marcadores Ayuda

media sf_Compartido simgen **acru**

Lugares

- Recientes
- Carpeta personal
- Escritorio
- Descargas
- Documentos
- Imágenes
- Música
- Videos
- Papelera

Dispositivos

- Equipo

Red

- Examinar red
- Conectar al servidor

Nombre

- codigos.txt
- obshis_2000.txt
- obshis_2001.txt
- obshis_2002.txt
- obshis_2003.txt
- obshis_2004.txt
- obshis_2005.txt
- obshis_2006.txt
- obshis_2007.txt
- obshis_2008.txt
- obshis_2009.txt
- RAAZOBS.st_ANUAL.TXT
- RABAOBS.st_ANUAL.TXT
- RAJUOBS.st_ANUAL.TXT
- RANJOBS.st_ANUAL.TXT
- RAPEOBS.st_ANUAL.TXT
- RATROBS.st_ANUAL.TXT
- TODOS_ANUAL.TXT2
- UYLBOBS.st_ANUAL.TXT
- UYLEOBS.st_ANUAL.TXT
- UYSAOBS.st_ANUAL.TXT
- UYTTOBS.st_ANUAL.TXT

```
codigos.txt x
"RAAZOBS.st" 2000
"RABAOBS.st" 2001
"RAJUOBS.st" 2002
"RANJOBS.st" 2003
"RAPEOBS.st" 2004
"RATROBS.st" 2005
"UYLBOBS.st" 2006
"UYLEOBS.st" 2007
"UYSAOBS.st" 2008
"UYTTOBS.st" 2009
```

```
obshis_2000.txt x
```

00000007	19730101	0.0	31.4	11.5	4.9	26.33	95.00	47.00	138.2
00000007	19730102	0.0	28.8	9.0	4.9	26.33	95.00	47.00	138.2
00000007	19730103	0.0	31.1	17.5	4.9	26.33	95.00	47.00	138.2
00000007	19730104	0.0	25.0	16.5	4.9	26.33	95.00	47.00	138.2
00000007	19730105	0.0	25.8	5.0	4.9	26.33	95.00	47.00	138.2
00000007	19730106	0.0	25.7	14.0	4.9	26.33	95.00	47.00	138.2
00000007	19730107	0.0	25.5	5.1	4.9	26.33	95.00	47.00	138.2
00000007	19730108	0.0	33.3	11.3	4.9	26.33	95.00	47.00	138.2

```
RAAZOBS.st_ANUAL.TXT x
1973 927.8 7.41753424657534 20.6186301369863
1974 789.9 7.5372602739726 21.1087671232877
1975 957.5 7.88931506849315 21.0624657534247
1976 1103.6 7.06065573770492 20.4286885245902
1977 1110.3 8.53643835616438 21.1630136986301
1978 1117.8 8.50520547945206 20.4728767123288
1979 639.9 7.46739726027397 21.6054794520548
1980 1413.3 8.82868852459016 21.281693989071
1981 886.7 7.69506849315068 20.8750684931507
1982 1117.3 7.81452054794521 20.6950684931507
1983 847.1 7.67972602739726 21.1887671232877
```



```
--  
83 ### Calculo la media regional, leo el archivo TODOS_ANUAL.TXT2  
84  
85 dat <- read.table("acru/TODOS_ANUAL.TXT2")  
86 dat$V5 <- NULL  
87 names(dat) <- c("year", "rain", "tmin", "tmax")  
88 media.regional <- aggregate(dat, by = list(dat$year), FUN = "mean", na.omit=TRUE)  
89  
90 media.regional.print <- media.regional[,-(1:2)]  
91 write.table(media.regional.print, "acru/media_regional.txt", col.names=FALSE, row.names=FALSE )  
92  
93 media.regional.dia <- media.regional.print  
94 media.regional.dia$rain <- media.regional$rain/365  
95 write.table(media.regional.dia, "acru/media_regional_dia.txt", col.names=FALSE, row.names=FALSE )  
96
```

Para ajustar el modelo VAR los datos no deben presentar tendencia



Para remover las tendencias utilizamos la metodología de descomposición de la variabilidad climática en escalas temporales

```
### remuevo la tendencia con la temp global|
#####
periodo <- 8                                     ## Per?odo m?nimo a retener
icone=1                                         ## tipo de relleno serie
iconb=1                                         ## tipo de relleno serie
year <- seq(1901, 2095, by=1)
tsm <- read.table("daily/serie_cmip5.txt", quote="\")
trend <- cbind(year, tsm)
names(trend) <- c("year", "tsm")
data <- merge(media.regional, trend)
## CALCULO DE LA VARIABILIDAD DE LARGO PLAZO (CC)
## Calculo parametro de correlacion Chi-Sq - CORR Pearson (Serie TSM : Serie Obs) * SD(obs)/SD(TSM)
base <- data.frame(na.omit(cbind(data$year, data$tsm, data$rain/365)))
names(base) <- c("year", "tsm", "rain")
corr <- cor.test (base$tsm, base$rain)
coef <- corr$estimate * sd(base$rain)/sd(base$tsm)
# calculo de la serie con media de cc TrendCC = Media(obs) + K * TSM(i)-Media(TSM)
resultado <- data.frame(base$year, base$rain)
names(resultado) <- c("year", "rain")
resultado$trendccM <- round(mean(base$rain) + coef * (base$tsm-(mean(base$tsm))),2)
resultado$trendcc <- round(resultado$trendccM - mean(base$rain),2)
resultado$detrendcc <- round(base$rain - resultado$trendccM,2)
resultado.rain <- data.frame(resultado, "rain")
## Figura 3 del paper para tmin en este ejemplo

plot(resultado$year, resultado$rain, type="p")
lines(resultado$year, resultado$trendccM, col="red")
xyplot(detrendcc ~ year, data = resultado, type=c("l", "p"))
```

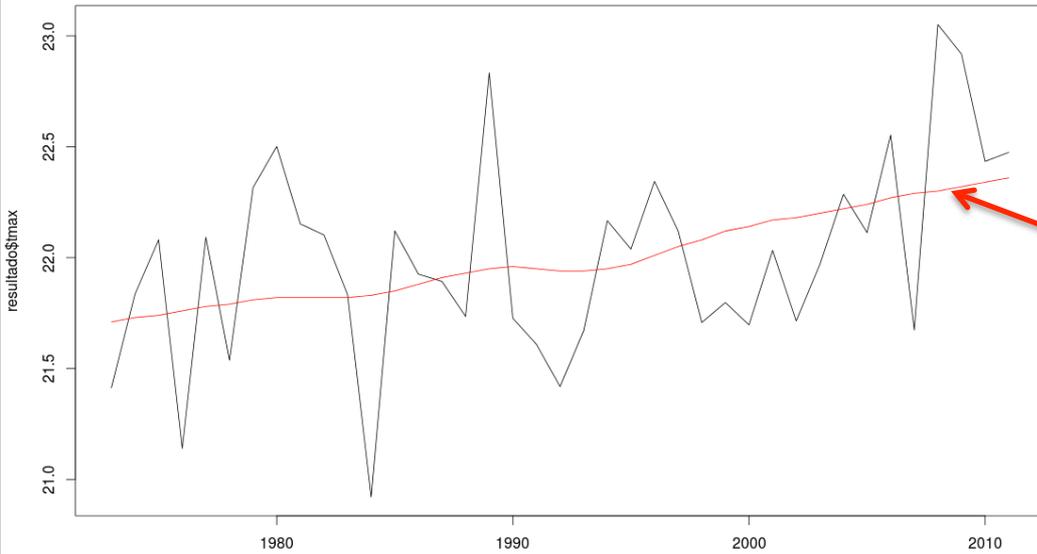


Gráficos diagnóstico

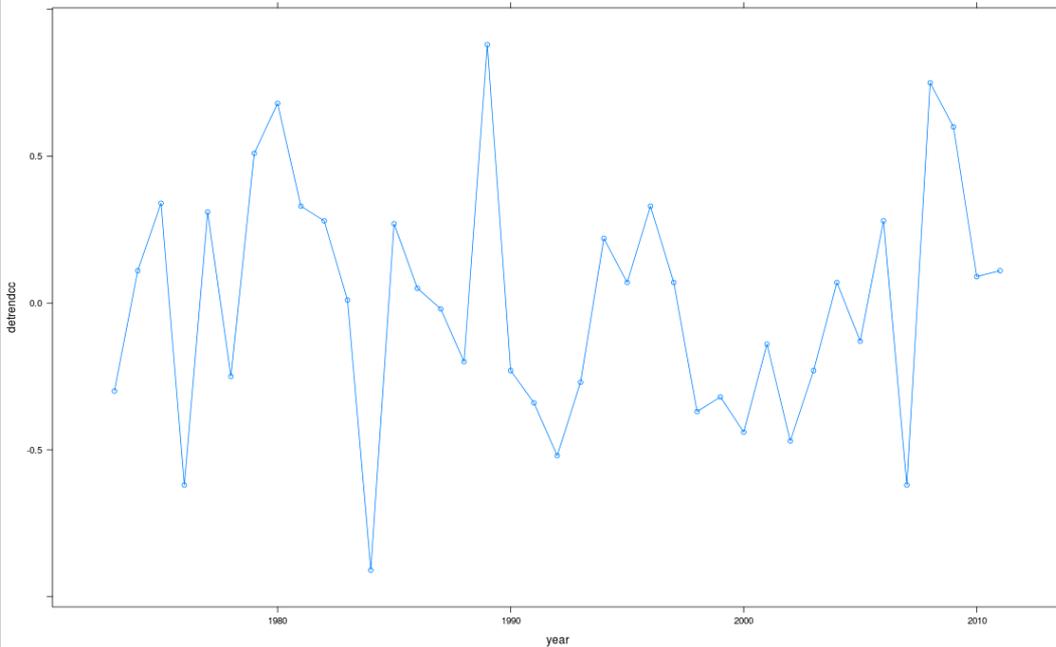
Exactamente lo mismo para las temperaturas máximas y mínimas



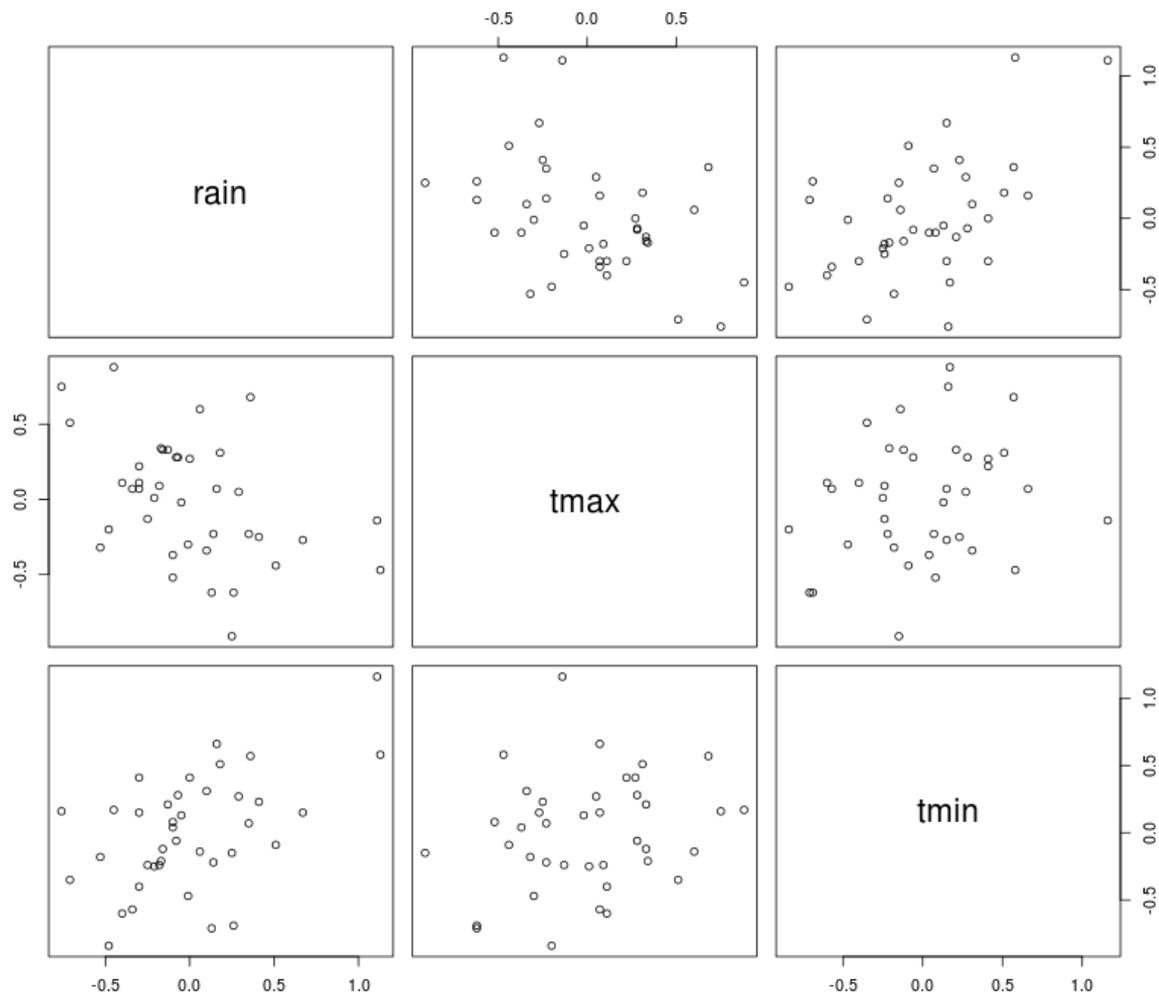
Temperatura Máxima



Tendencia de largo plazo
(asociada al CC)



Datos sin tendencia



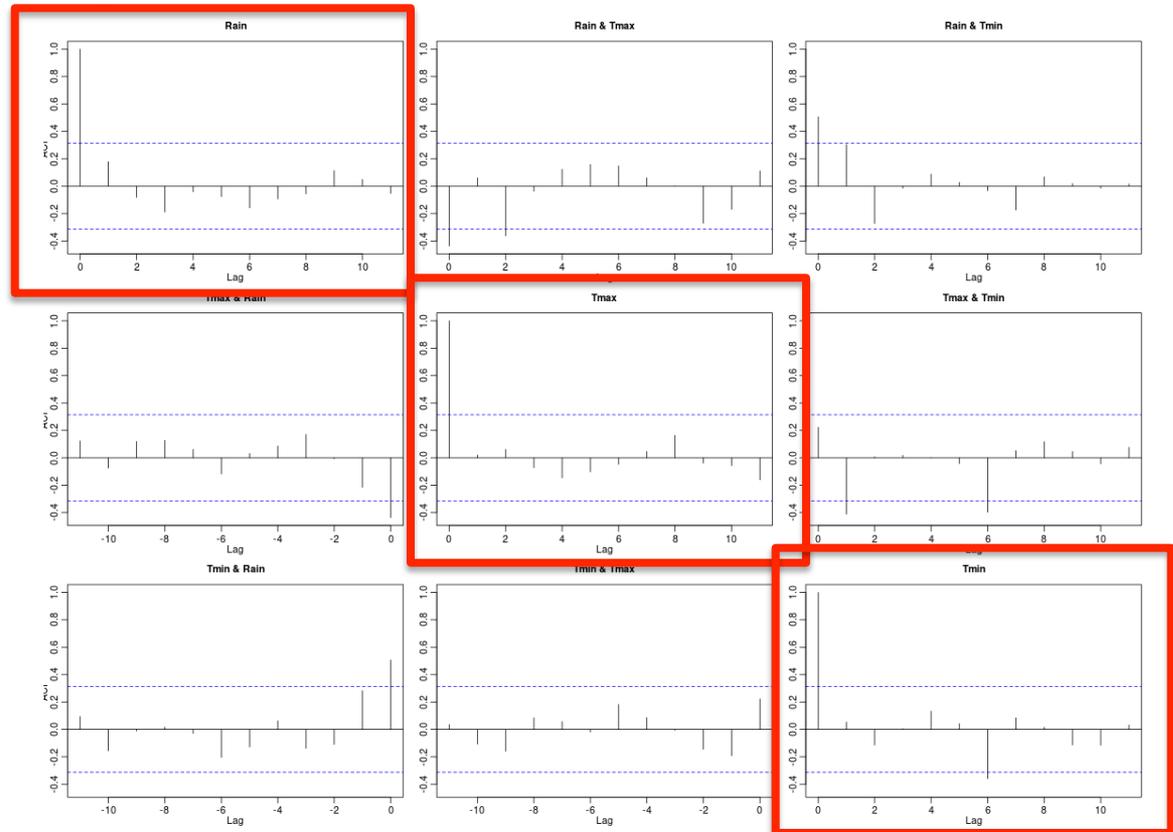
```
## Ahora debo aplicar el paquete DSE a la serie sin tendencia  
## primero estimo el modelo con estVARXls y se genera la serie con simulate  
##
```

```
ts <- TSdata(output= detrended)  
seriesNamesOutput(ts) <- c("Rain","Tmax", "Tmin")  
acf(ts) # los datos no parecen autocorrelacionados
```

```
model01 <- estVARXls(ts, max.lag=1)  
model02 <- estVARXls(ts, max.lag=2)  
model03 <- estVARXls(ts, max.lag=3)  
model04 <- estVARXls(ts, max.lag=4)  
model05 <- estVARXls(ts, max.lag=5)  
model06 <- estVARXls(ts, max.lag=6)  
model07 <- estVARXls(ts, max.lag=7)  
model08 <- estVARXls(ts, max.lag=8)  
model09 <- estVARXls(ts, max.lag=9)  
model10 <- estVARXls(ts, max.lag=9)
```

El paquete "DSE" utiliza los datos en el formato TSdata.

La mayoría de los análisis de series de tiempo utilizan este tipo de estructuras



Estimo el modelo VAR para varios lags, aunque mas allá del lag(1) es bastante mas complicado.



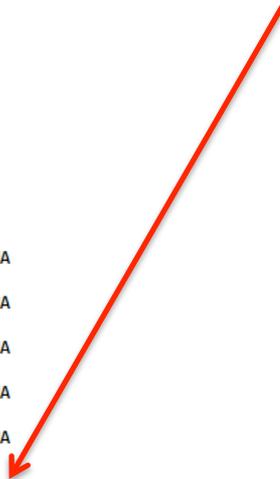
```

> informationTests(model01, model02, model03, model04, model05, model06, model07, model08, model09, model10)
      based on no.of parameters      based on theoretical parameter dim.
      PORT -ln(L)  AIC  BIC  GVC  RICE  FPE  AIC  BIC  GVC  RICE  FPE
[1] 81.95 34.52 87.03 111.89 87.76 88.58 87.07 NA NA NA NA NA
[1] 62.70 27.12 90.24 139.96 93.33 97.27 90.53 NA NA NA NA NA
[1] 56.97 22.74 99.48 174.06 106.88 117.91 100.47 NA NA NA NA NA
[1] 54.94 20.06 112.13 211.57 126.18 151.92 114.54 NA NA NA NA NA
[1] 53.36 17.58 125.16 249.46 148.77 206.73 130.04 NA NA NA NA NA
[1] 63.432 5.474 118.948 268.106 155.804 311.047 127.776 NA NA NA NA NA
[1] 47.522 -4.571 116.859 290.876 171.785 NaN 131.724 NA NA NA NA NA
[1] 46.248 -8.878 126.243 325.120 205.833 NaN 150.148 NA NA NA NA NA
[1] 119.696 -6.902 148.197 371.933 262.002 NaN 185.652 NA NA NA NA NA
[1] 119.696 -6.902 148.197 371.933 262.002 NaN 185.652 NA NA NA NA NA

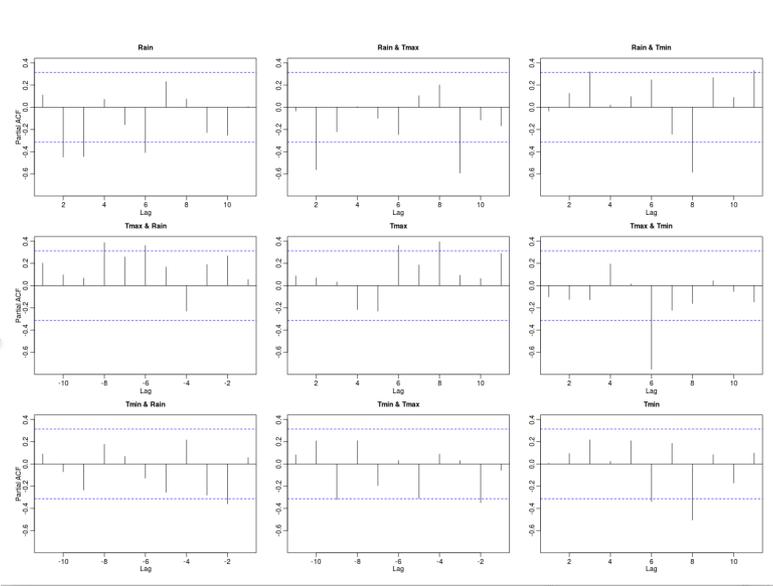
opt 8 8 1 1 1 1 1 1 1 1 1 1
PORT - Portmanteau test -ln(L)- neg. log likelihood
AIC - neg. Akaike Information Criterion BIC - neg. Bayes Information Criterion
GVC - Generalized Cross Validation RICE - Rice Criterion
FPE - Final Prediction Error
WARNING - These calculations do not account for trend parameters in ARMA models.
> |

```

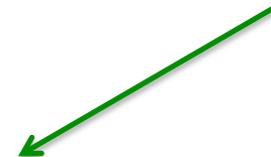
Determina el modelo óptimo de acuerdo a ese criterio



checkResiduals(modelXX)



Generación secuencia simulada



```
201 # genero la long.seq con el modelo lag(1)
202 long.seq <- simulate(model01, start=c(1973,1), freq=1,sampleT=100000)
203 # calculo las correlaciones en los detrended(obs) y los simulados
204 cor(detrended)
205 cor(long.seq$output[])
206
207 # Tabla 3 del paper Correlaci??n entre variables
208 # (detrended)
209 #          rain          tmax          tmin
210 # rain  1.0000000 -0.4366612  0.506614
211 # tmax -0.4366612  1.0000000  0.223449
212 # tmin  0.5066140  0.2234490  1.000000
213 # (long.seq$output[])
214 #          Rain          Tmax          Tmin
215 # Rain  1.0000000 -0.4400227  0.5083784
216 # Tmax -0.4400227  1.0000000  0.2193674
217 # Tmin  0.5083784  0.2193674  1.0000000
218
```

Las correlaciones entre variables son muy similares entre observados y simulados

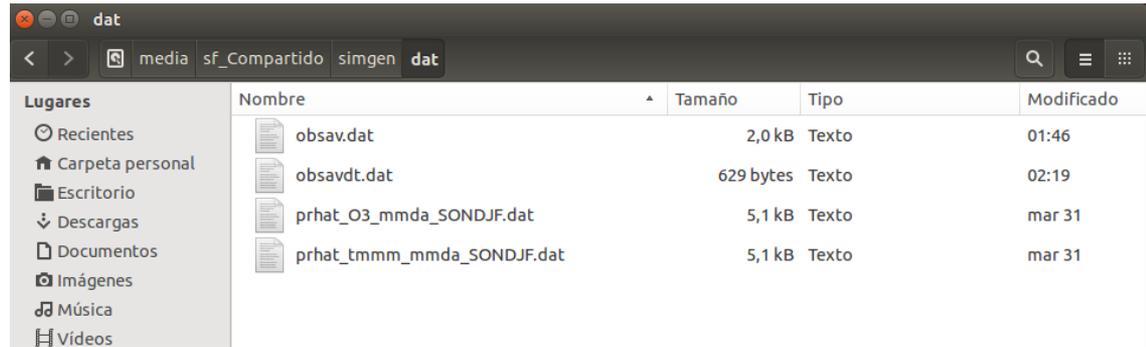


El modelo es ACEPTABLE

Ahora debemos distribuir los archivos en los directorios que corresponden para que los pueda leer SIMGEN

Media regional y media regional "detrended". Deben renombrarse como :
obsav.dat y obsavdt.dat

- dat
- Input_sim
- Output_sim
- pickled
- python
- obs

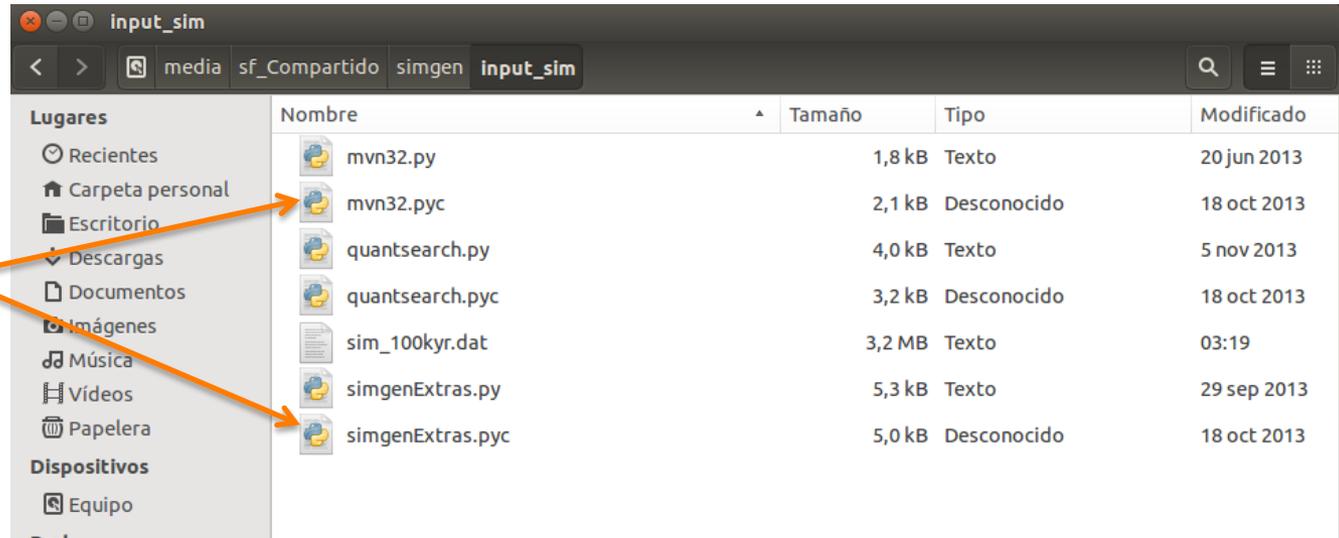


Ahora debemos distribuir los archivos en los directorios que corresponden para que los pueda leer SIMGEN

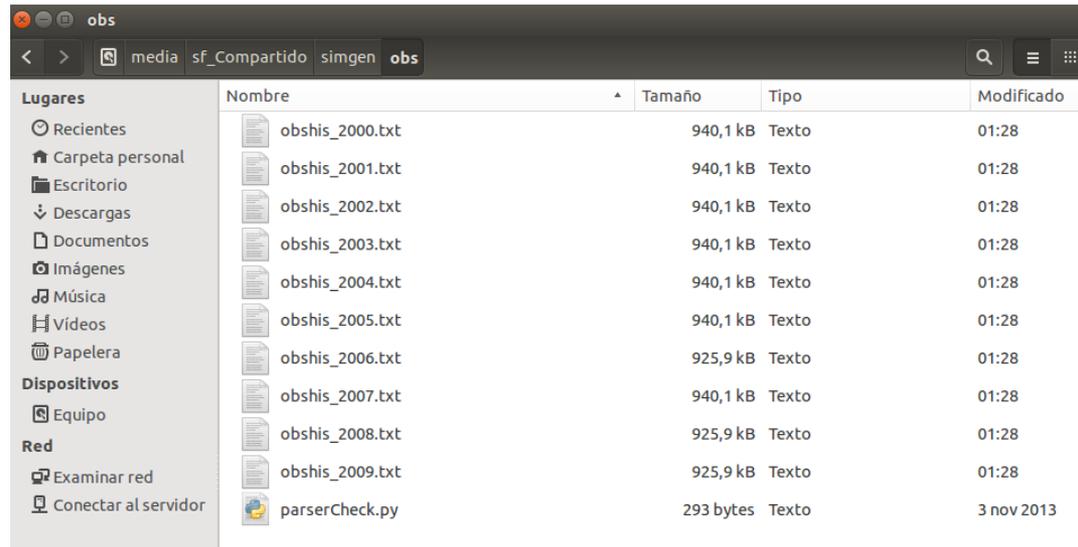
- dat
- Input_sim
- Output_sim
- pickled
- python
- obs

Secuencia simulada, renombrarse como sim_100kyr.dat

Scripts auxiliares a simgen



Ahora debemos distribuir los archivos en los directorios que corresponden para que los pueda leer SIMGEN



dat

Input_sim

Output_sim

pickled

python

obs

Van los archivos obshist_ xxxx.txt

Comenzamos a utilizar Python

```
simgen@simgen-VirtualBox: /media/sf_Compartido/simgen
simgen@simgen-VirtualBox:~$ cd /media/sf_Compartido/simgen
simgen@simgen-VirtualBox:/media/sf_Compartido/simgen$ ls -l
total 92
-rwxrwx--- 1 root vboxsf 14255 jul 16 03:20 01_prep_ntcc_chile.R
drwxrwx--- 1 root vboxsf  952 jul 16 03:18 acru
drwxrwx--- 1 root vboxsf 1020 nov  5 2013 daily
drwxrwx--- 1 root vboxsf  238 jul 16 14:58 dat
-rwxrwx--- 1 root vboxsf  853 jun  5 2013 detrend2.py
-rwxrwx--- 1 root vboxsf 1148 nov  5 2013 detrend2.pyc
drwxrwx--- 1 root vboxsf  340 jul 16 14:53 input_sim
drwxrwx--- 1 root vboxsf  476 jul 16 14:06 obs
drwxrwx--- 1 root vboxsf  102 jul 16 14:54 output_sim
drwxrwx--- 1 root vboxsf  102 jul 30 2013 pickled
drwxrwx--- 1 root vboxsf  204 jul 16 15:21 python
-rwxrwx--- 1 root vboxsf 4003 nov  3 2013 quantsearch.py
-rwxrwx--- 1 root vboxsf 2150 sep 29 2013 readqc.py
-rwxrwx--- 1 root vboxsf 2386 oct 18 2013 readqc.pyc
-rwxrwx--- 1 root vboxsf 37715 mar 31 16:41 simgen9s.py
-rwxrwx--- 1 root vboxsf 5256 sep 29 2013 simgenExtras.py
-rwxrwx--- 1 root vboxsf 4612 oct 18 2013 simgenExtras.pyc
simgen@simgen-VirtualBox:/media/sf_Compartido/simgen$
```

Abrir Terminal y
situarse en el
directorio de trabajo

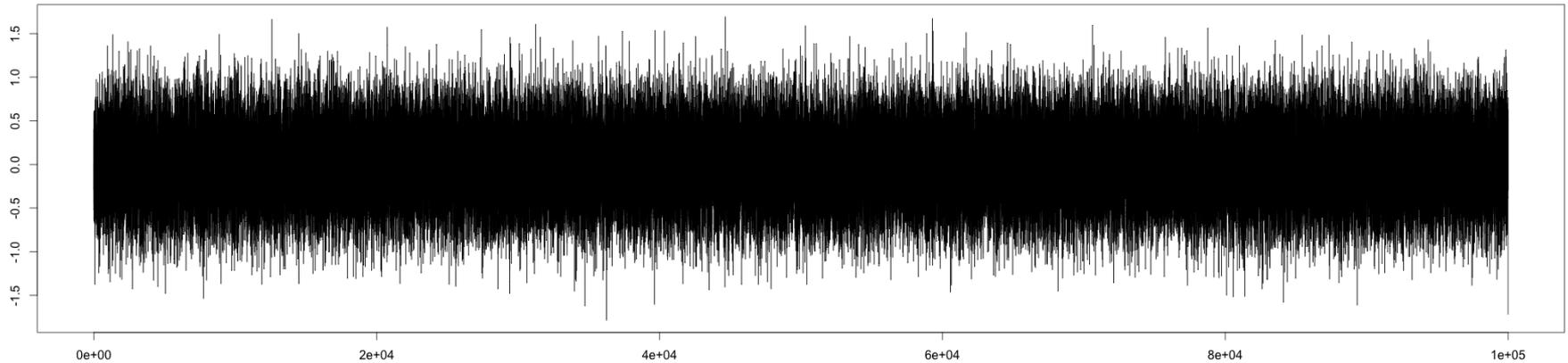
```
cd /media/sf_Compartido/simgen
```

`output=gen(obsix, simix, trendq, write=1, fname, simlen, locate, xval, M=1)`

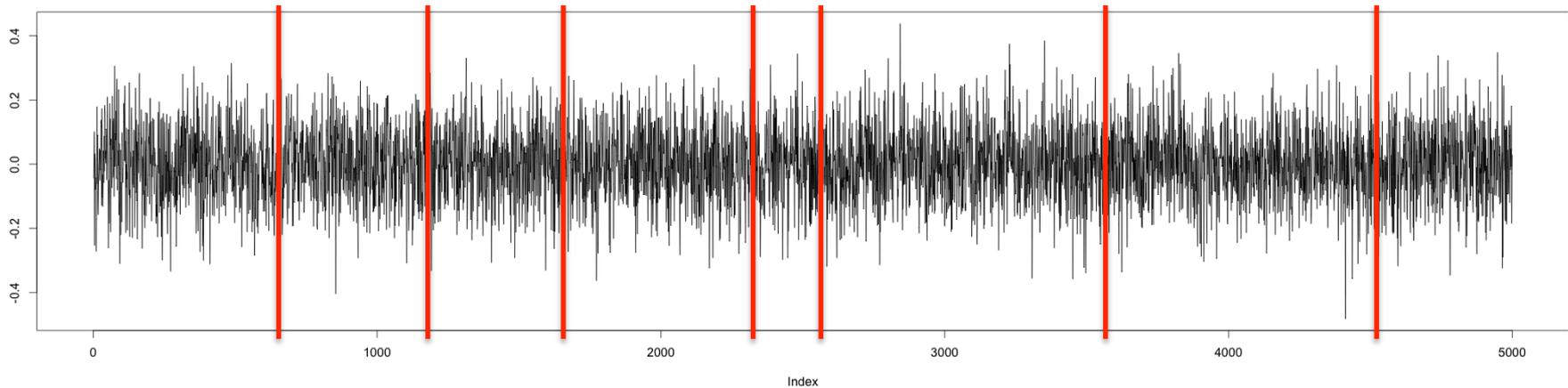


`output=gen([2000,2001], 2955, 0.5, write=1, fname='sim_100kyr.dat', simlen=60, locate=2041, xval=0, M=1)`

Secuencia simulada



Medias por décadas



SIMIX = ???



quantsearch.py

```
simgen@simgen-VirtualBox: /media/sf_Compartido/simgen/input_sim
simgen@simgen-VirtualBox:/media/sf_Compartido/simgen/input_sim$ python
Python 2.7.4 (default, Apr  8 2014, 16:09:57)
[GCC 4.8.1] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> from quantsearch import generar
>>> generar(10, 10)
Args to mvn32: [ 0.00078243 -0.00064908  0.00172764] [[ 0.02515893 -0.01357273
0.01926087]
 [-0.01357273  0.02003911 -0.00749087]
 [ 0.01926087 -0.00749087  0.02192334]] -0.200119672158 0
ppr,munew: -0.2001 0.1077 -0.1521
prlims: -0.2160 -0.1843
malims: -0.0338 0.2493
milims: -0.3001 -0.0040
prelims:-0.1578 0.1594
sort 1 completed...
sort 2 completed
Should be about equal to ppr: -0.2001
362 realizations to search, not a problem!
11 instances found!
[8999, 8111, 5705, 6784, 4974, 2250, 9443, 6229, 8341, 5318, 2319]
>>>
```

SIMIX = [8999, 8111, 5705, 6784, 4974, 2250, 9443, 6229, 8341, 5318, 2319]

```

simgen@simgen-VirtualBox: /media/sf_Compartido/simgen
simgen@simgen-VirtualBox: /media/sf_Compartido/simgen$ python
Python 2.7.4 (default, Apr 8 2014, 16:09:57)
[GCC 4.8.1] on linux2
Type "help()" on copyright, "credits" or "license" for more information
>>> from simgen9s import gen
>>> output=gen([2000], 8999, 0.5, write=1,fname='sim_100kyr.dat', simlen=60, locate=2041, xval=0, M=1)
Now fetching simulation file...
File is read and sliced!
Now fetching tMMM, for forced component...
File is read!
Size x21C! 60
SIMlen! 60
infoSerie.lenSerie 39
Computed(or fetched) CI!
Size of pr21forced 60
Size of simpr 60
Processing station file obs/obshis_2000.txt
14244 days in file...
Done!
0 filled values, of 14244
obs/obshis_2000.txt is read!
Size pr20forced: 39
Size yrpr (STA): 39
Regression coeffs, catchment on regional sig: 0.95 1.02 1.31
Variance adjustment factors: 0.3888 0.2557 0.2607
Variances for obs and sim...
pr: 0.2635 0.2128
ma: 0.2282 0.2233
mi: 0.3562 0.3660
Strip, inject means: 1.0000 1.0000
Strip, inject rmse (first 50 yr): 1.6348
Detected improper Tmax,Tmin values, attempting to fix...
Offending indices, values, diff:
11024 19.0 21.8 -2.8
New values:
22.05 21.8

Offending indices, values, diff:
11417 13.8 16.0 -2.2
New values:
16.25 16.0

Offending indices, values, diff:
24904 19.5928 20.3647 -0.771914
New values:
20.6147 20.3647

Offending indices, values, diff:
29148 17.1998 17.1395 0.0603523
New values:
17.3895 17.1395

Wrote 36159 lines!
>>>
>>>
  
```

output=gen([2000,2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009], 8999, 0.1, write=1,fname='sim_100kyr.dat', simlen=84, locate=2021, xval=0, M=1)



Nombre	Tamaño	Tipo	Modificado
sim_100k_obshis_2000_008999.txt	976,3 kB	Texto	18:17
sim_100k_obshis_2001_008999.txt	976,3 kB	Texto	18:17
sim_100k_obshis_2002_008999.txt	976,3 kB	Texto	18:17
sim_100k_obshis_2003_008999.txt	976,3 kB	Texto	18:17
sim_100k_obshis_2004_008999.txt	976,3 kB	Texto	18:17
sim_100k_obshis_2005_008999.txt	976,3 kB	Texto	18:17
sim_100k_obshis_2006_008999.txt	976,3 kB	Texto	18:17
sim_100k_obshis_2007_008999.txt	976,3 kB	Texto	18:17
sim_100k_obshis_2008_008999.txt	976,3 kB	Texto	18:17
sim_100k_obshis_2009_008999.txt	976,3 kB	Texto	18:17
yearly_simulation_obshis_2000_008999.txt	7,4 kB	Texto	18:17
yearly_simulation_obshis_2001_008999.txt	7,4 kB	Texto	18:17
yearly_simulation_obshis_2002_008999.txt	7,4 kB	Texto	18:17
yearly_simulation_obshis_2003_008999.txt	7,4 kB	Texto	18:17
yearly_simulation_obshis_2004_008999.txt	7,4 kB	Texto	18:17
yearly_simulation_obshis_2005_008999.txt	7,4 kB	Texto	18:17
yearly_simulation_obshis_2006_008999.txt	7,4 kB	Texto	18:17
yearly_simulation_obshis_2007_008999.txt	7,4 kB	Texto	18:17
yearly_simulation_obshis_2008_008999.txt	7,4 kB	Texto	18:17
yearly_simulation_obshis_2009_008999.txt	7,4 kB	Texto	18:17

Para obtener las repeticiones para el ensemble hay que repetir la operación tantas veces como “simix” se quieran utilizar

