ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS

ANÁLISIS CLÚSTER PARA SERIES DE TIEMPO ESTACIONALES Y MODELIZACIÓN DE CAUDALES DE RÍOS DEL BRASIL.

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERÍA MATEMÁTICA

PROYECTO DE INVESTIGACIÓN

CRISTIAN DAVID PACHACAMA SIMBAÑA
cristian.pachacama01@epn.edu.ec

Directora: UQUILLAS ANDRADE ADRIANA, PH.D.
adriana.uquillas@epn.edu.ec

OCTUBRE 2018

DECLARACIÓN

Yo, CRISTIAN DAVID PACHACAMA SIMBAÑA, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual, correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Cristian David Pachacama Simbaña

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por CRISTIAN DAVID PACHACAMA SIMBAÑA, bajo mi supervisión.

Uquillas Andrade Adriana, Ph.D.
Directora del Proyecto

AGRADECIMIENTOS

A mi familia, ya que su cariño y apoyo me llevaron a donde ahora estoy. A mis mejores amigos Miguel, Rubi, Luis y Pablo, por brindarme su amistad y tan gratos momentos.

A mi tutora Adriana por ser una guía y apoyarme desde el primer momento a alcanzar esta meta, gracias por depositar su confianza en mi. A los profesores Erwin Jimenez, Luis Horna, y de manera especial a Juan Carlos Trujillo quienes hicieron nacer en mi la pasión por la Matemática, pasión que espero inspirar a más generaciones de estudiantes.

Finalmente, a grandes matemáticos de la historia como George Cantor, Simeón Poisson , Abraham Wald, y Karl Pearson, cuyo trabajo me inspiró a profundizar en el conocimiento de esta bella ciencia.

DEDICATORIA

A mis padres Magdalena y Lucio, por su incondicional amor, sus sabios consejos y su paciencia, siempre lo tendré presente. A Isabel por su apoyo incondicional, y por aparecer en el momento exacto en mi vida para llenarla de felicidad. A Miguel, Rubi, Pablo y Luis por brindarme su sincera amistad.

Índice general

Índice de figuras

Índice de tablas

Resumen

En el presente trabajo se aborda la aplicación del Análisis Clúster para Series de Tiempo orientado al modelamiento de caudales de los principales ríos de Brasil, que se midieron en 150 estaciones repartidas en los mismos, esto a partir de variables climáticas y la combinación de técnicas de modelamiento como Análisis de Componentes Principales Funcional (ACPF), SARIMAX y STL-Loess.

Específicamente lo que se hace es crear un número pequeño de clústers (de 2 a 4 clústers) a partir de las 150 estaciones (donde se midieron los caudales), donde cada grupo contendrá a estaciones en la que sus caudales posean un comportamiento temporal lo más similar posible, luego para cada uno de estos clústers, mediante el uso de ACPF, hallaremos una sola serie de tiempo que resuma el comportamiento de los caudales del clúster. Finalmente se modela la serie de tiempo de cada clúster a partir de variables climáticas, usándolas como variables explicativas en el marco del modelamiento SARIMAX.

Mostraremos después las ventajas y la eficiencia de modelar una enorme cantidad de series de tiempo con el uso de estás técnicas, esto debido a que el modelo que explica cada clúster puede ser extendido (usando los mismos retardos y variables explicativas) a cada una de las series de tiempo que lo componen. Realizamos estudios comparativos entre un modelo (SARIMAX) individual para un caudal específico y el modelo del clúster al que pertenece, obteniendo resultados similares en cuanto a predictibilidad. Donde se obtuvo para el modelo individual un Error Cuadrático Medio (RMSE) del 0.3 % y un AIC de $-652{,}21$ mientras que para el modelo del clúster se obtuvo un RMSE de 0.4 %,y un AIC de $-763{,}23$.

Así mostramos que conseguimos pasar del problema de modelar 150 series de tiempo, a modelar las series de tiempo de unos cuantos clústers.

Palabras clave: Análisis Clúster para Series de Tiempo, descomposición STL-Loess,SARIMAX, Análisis de Componentes Principales Funcional.

Abstract

This paper deals with the application of the Cluster Analysis for Time Series oriented to the modeling of flows of the main rivers of Brazil, which were measured in 150 stations distributed in them, this from climatic variables and the combination of techniques of modeling as Principal Functional Components Analysis (FPCA), SARIMAX and STL-Loess.

Specifically what is done is to create a small number of clusters (from 2 to 4 clusters) from the 150 stations (where the flows were measured), where each group will contain stations in which their flows have a temporary behavior similar possible, then for each of these clusters, through the use of ACPF, we will find a single time series that summarizes the behavior of the flows of the cluster. Finally, the time series of each cluster is modeled from climatic variables, using them as explanatory variables in the SARIMAX modeling framework.

We will show later the advantages and the efficiency of modeling a huge amount of time series with the use of these techniques, this because the model that explains each cluster can be extended (using the same delays and explanatory variables) to each of the time series that compose it. We perform comparative studies between an individual model (SARIMAX) for a specific flow and the model of the cluster to which it belongs, obtaining similar results in terms of predictability. Where an Average Quadratic Error (RMSE) of 0.3 % and an AIC of $-652{,}21$ was obtained for the individual model, while for the cluster model an RMSE of 0.4 % was obtained, and an AIC of $-763{,}23$.

Thus we show that we managed to move from the problem of modeling 150 time series, to modeling the time series of a few clusters.

Keywords: Time Series Cluster Analysis, STL-Loess decomposition, Functional Principal Component Analysis

Capítulo 1

Introducción

Brasil tiene uno de los sistemas hidrológicos más complejos, diversos y extensos del mundo. A diferencia de la gran mayoría de los países desarrollados, Brasil tiene en los ríos su principal fuente de generación de electricidad, ocupando el tercer lugar dentro de los más grandes productores hidroeléctricos del mundo. Debido a la importancia del sector hidroeléctrico buscar formas de facilitar y mejorar el modelamiento de datos asociados a este sector es un problema prioritario. Problema provocado por la dificultad que supone lidiar con la enorme cantidad de datos (accesibles desde la web de instituciones como ANA, ONS, NOAA, CPTEC, etc.) asociados a mediciones de Caudales de los ríos que componen este sistema, que cuenta con alrededor de 150 estaciones de medición repartidas en todo Brasil. Dichos datos se presentan en forma de Series de Tiempo que posee tres características que dificultan su análisis, la primera es que estas series de tiempo poseen observaciones diarias de los caudales en un periodo de tiempo de alrededor de 30 años, es decir, son series muy extensas. La segunda característica es que estas series de tiempo son estacionales, y por último existe evidencia de que el ruido o error asociado a estas series no se distribuye normalmente, sino que su distribución posee colas más pesadas como las analizadas en teoría de valores extremos. En ese contexto, notamos que es posible disminuir la dimensión del problema a través la identificación de clústers o zonas representativas (no necesariamente geográficas) que resuman el comportamiento temporal que poseen los caudales de los ríos. Esto en términos de modelamiento esto se traduce en pasar del problema de modelar el nivel de caudal en todas las 150 estaciones, al problema de modelar únicamente 1 estación por cada clúster.

Ya que el problema se basa en identificar grupos de ríos cuyos Caudales se com-

portan de manera similar en el tiempo, se propone la utilización de el .Análisis Clúster de Series de Tiempo", que es una técnica de agrupamiento que considera una función de disimilitud.entre las series de tiempo (que mide que tan distintas son un par de series) y a partir de ella crea grupos de series, cada grupo contiene series de tiempo parecidas". Al elegir adecuadamente la función de disimilitud (diseñada para series de tiempo) es posible agrupar a los ríos en grupos basados en el comportamiento temporal de sus caudales. Esto con la finalidad de lidiar con la complejidad que supone analizar y modelar esta enorme cantidad de series de tiempo de caudales, pasando de analizar alrededor de 150 series a unas pocas (una serie por Clúster), sin dejar de lado la estructura y comportamiento estacional de cada una de ellas, partiendo de una adecuada elección de la función de disimilitud. Hay que destacar que el modelamiento de caudales juega un rol trascendental en la creación de políticas que adopta sector energético de Brasil, que como mencionamos anteriormente está alimentado en su mayoría por el sector hidroeléctrico en donde el análisis que planteamos permitiría profundizar en la planificación de las operaciones de plantas hidroeléctricas que depende directamente del comportamiento temporal de los ríos que las alimentan, esta planificación podría evitar por ejemplo eventos de déficit energético provocados por una deciencia estructural de la disponibilidad de energía, que a la larga tiene impacto económico y social mayor que los cortes de energía.

1.1. Descripción de los Datos

Contamos con una base de datos de 31588 observaciones diarias de 2383 variables, las variables se encuentran clasificadas en 5 Tipos:

1. Primero las variables correspondientes a Caudales (Vazoes), mismas que corresponden a series de tiempo de 150 estaciones georeferenciadas.
2. Además, tenemos variables referentes a Clima. Contamos con observaciones de 8 variables en 260 estaciones. Las variables son las siguientes:
    - Evaporacao_ Piche
    - Insolacao
    - Precipitacao_ 12H
    - Temp_ Comp_ Media
    - TempMaxima

- TempMinima_ 12H
- Umidade_ Relativa_ Media
- Velocidade_ do_ Vento_ Media

3. Contamos además con las variables globales que corresponden a 13 índices, que miden fenómenos meteorológicos y climatológicos a nivel mundial

    - AAO: Antartic Oscillation Index
    - AO: Artic Oscillation Index
    - MJO-RMM1: Oscilacao Madden-Jullian RMM1
    - MJO-RMM2: Oscilacao Madden-Jullian RMM2
    - NAO: North Atlantic Oscillation Index
    - Nino3: El Niño 3
    - Nino4: El Niño 4
    - Nino12: El Niño 1+2
    - Nino34: El Niño 3.4
    - SOI: Southern Oscillation Index
    - SOI_ DAR: Southern Oscillation Index
    - SOI_ TAH: Southern Oscillation Index
    - TSI: Total solar irradiance

Capítulo 2

Marco Teórico

2.1. Descomposición STL - Loess

STL es un procedimiento de filtrado propuesto en [Cleveland et al., 1990], sque permite descomponer una serie de tiempo en sus componentes estacional, tendencia y Residuo. STL tiene un diseño simple que consiste en una secuencia de aplicaciones del Loess Smoother; la simplicidad permite el análisis de las propiedades del procedimiento y permite un cálculo rápido, incluso para series de tiempo muy largas y grandes cantidades de tendencia y suavizado estacional. Otras características de STL son la especificación de cantidades de suavizado estacional y de tendencias que varían, de manera casi continua, desde una cantidad muy pequeña de suavizado hasta una cantidad muy grande; estimaciones robustas de la tendencia y los componentes estacionales que no están distorsionados por un comportamiento aberrante en los datos; especificación del período de componente estacional a cualquier múltiplo entero del intervalo de muestreo de tiempo mayor que uno; y la capacidad de descomponer series de tiempo con valores perdidos.

Definiciones

2.1.1. Loess-Regresión Local

Sean x_i y y_i (para $i = 1, 2, ..., n$) son observaciones de una variable independiente y dependiente respectivamente. La curva de regresión "Loess", $\hat{g}(x)$, es un suavizado de y dado x que puede calcularse para cualquier valor de dominio de la variable independiente. Así "Loess.está definida sobre cualquier valor no solamente sobre

x_i.Como veremos más adelante, esta es una importante característica que en STL nos permitirá lidiar con los valores perdidos y eliminar el componente estacional de manera sencilla. En realidad Loess puede ser usada para suavizar y en función de cualquier número de variables independientes, pero para STL, solo es necesario considerar una variable independiente.

Primero se calcula $\hat{g}(x)$ de la siguiente manera. Se escoge un entero positivo q. Supongamos $q \leq n$. Los q valores de x_i que son más cercanos a x se seleccionan, cada uno esta dado por el *Peso del Vecindario* basado en su distancia desde x. Sea $\lambda_q(x)$ la distancia de el *q-ésimo* x_i más lejano de x. Sea W la función de peso tricúbica definida por:

$$W(u) = \begin{cases} (1-u^3)^3 & \text{para } 0 \leq u < 1 \\ 0 & \text{para } u \geq 1 \end{cases}$$

El peso del vecindario para cualquier x_i es

$$v_i(x) = W\left(\frac{|x_i - x|}{\lambda_q(x)}\right)$$

Así un x_i cercano a x tiene el peso más grande; los pesos decrecen a medida que x_i se aleja de x, mientras que se aproxima a cero en el *q-ésimo* punto más lejano. El próximo paso es ajustar un polinomio de grado d a los datos con peso $v_i(x)$ en (x_i, y_i). El valor del polinomio ajustado localmente evaluado en x es $\hat{g}(x)$. En este caso solo analizaremos el caso en que $d = 1$ y 2, es decir, ajustando localmente un polinomio lineal o cuadrático.

Ahora supongamos que $q > n$. $\lambda_n(x)$ es la distancia de x al x_i más lejano. Para $q > n$ definimos $\lambda_q(x)$ por

$$\lambda_q(x) = \lambda_n(x)\frac{q}{n}$$

Luego de manera análoga a lo anterior, definimos los pesos de los vecindarios usando este valor de $\lambda_q(x)$.

Para usar Loess, d y q deben ser previamente elegidos. Las elecciones en el contexto de STL se discutirán a detalle más adelante. A medida que q crece, $\hat{g}(x)$ se hace más suave. Cuando q tiende a infinito, $v_i(x)$ tiende a 1 y $\hat{g}(x)$ tiende al polinomio de mínimos cuadrados ordinarios de grado d.

Supongamos que cada observación (x_i, y_i) tiene un peso ρ_i que expresa la confianza de la observación relativa a las otras. Por ejemplo, si y_i tiene varianza $\sigma^2 k_i$ donde k_i es conocido, luego ρ_i puede ser $1/k$. Así, podemos incorporar estos pesos

en el suvizamiento Loess en forma sencilla usando $\rho_i v_i(x)$ como los pesos en el ajuste de mínimos cuadrados. Esto provee un mecanismo mediante el cual podemos construir robustez en STL.

El diseño general.

STL consiste de dos procedimientos recursivos: un bucle interno anidado dentro de un bucle externo. En cada uno de los pasos del bucle interno, las componentes de tendencia y estacionalidad son actualizadas una vez; cada recorrido completo del bucle interno consiste de $n_{(i)}$ tales pasos. Cada paso del bucle externo consiste del bucle interno seguido por el calculo de pesos de robustez; estos pesos son usados en la siguiente corrida del bucle interno para reducir la influencia del comportamiento transitorio y aberrante en las componentes de tendencia y estacionalidad. Un paso inicial del bucle externo se realiza con todos los pesos de robustez iguales a 1, y luego $n_{(0)}$ pasos del bucle externo se llevan acabo. Las elecciones de $n_{(i)}$ y $n_{(0)}$ se discutirán más adelante.

Supongamos que el número de observaciones en cada periodo, o ciclo, de la componente estacional es $n_{(p)}$. Por ejemplo, si la serie es mensual con un año de periodicidad, entonces $n_{(p)} = 12$. Necesitamos poder referirnos a la subserie de valores en cada posición del ciclo estacional. Por ejemplo, para una serie mensual con $n_{(p)} = 12$, la primera subserie contiene los valores de Enero, la segunda tiene los valores de Febrero, y así sucesivamente. Nos referiremos a cada una de estas $n_{(p)}$ subseries como *subserie-ciclo*.

Bucle Interno

Cada paso de el bucle interno consiste de un suavizado estacional que actualiza la componente estacional, seguida por suavizado de tendencia que actualiza la componente de tendencia. Supongamos $S_v^{(k)}$ y $T_v^{(k)}$ para $v = 1, 2, ..., N$ son las componentes estacional y de tendencia al final del *k-ésimo* paso; estas dos componentes se definen para todos los tiempos $v = 1, 2, ..., N$, inclusive donde Y_v es un valor perdido. Las actualizaciones de el $(k+1)$ paso, $S_v^{(k+1)}$ y $T_v^{(k+1)}$, son calculadas de la siguiente manera.

Paso 1.

Quitar Tendencia.- Una serie sin tendencia $Y_v - T_v^{(k)}$ es calculada . Si Y_v es un valor perdido en un punto particular del tiempo, entonces la serie sin tendencia es también tiene un valor perdido en esa posición.

Paso 2.

Suavizado de Subseries-Ciclo.- Cada subserie-ciclo de la serie sin tendencia es suavizado mediante Loess considerando $q = n_{(s)}$ y $d = 1$. Los valores suavizados se calculan en todas las posiciones de tiempo de las subseries-ciclo, incluyendo aquellos con valores perdidos, y en las posiciones justo antes de la primera posición de la subserie y justo después del último. Por ejemplo, suponga que la serie es mensual, $n_{(p)} = 12$. La colección de los valores suavizados para todas las subseries-ciclo son series estacionales provisionales, $C_v^{(k+1)}$, consiste de $N + 2n_{(p)}$ valores que van desde $v = -n_{(p)} + 1$ hasta $N + n_{(p)}$.

Paso 3.

Paso-bajo Filtro de Suavizado de Subseries-ciclo.- Un filtro paso-bajo es aplicado a $C_v^{(k+1)}$. El filtro consiste de una media móvil de longitud $n_{(p)}$, seguido por otra media móvil de longitud $n_{(p)}$, seguida de una media móvil de longitud 3, seguida de un suavizado Loess con $d = 1$ y $q = n_{(l)}$. La salida. $L_v^{(k+1)}$, esta definida en las posiciones $v = 1$ hasta N porque las tres medias móviles no pueden extenderse hasta el final. El suavizado estacional del **Paso 2** fue extendido $n_{(p)}$ posiciones en cada final en anticipación de esta pérdida.

Paso 4.

Quitar tendencia de las Subseries-Ciclo suavizadas.- El componente estacional desde el $(k+1) - simo$ bucle es $S_v^{(k+1)} = C_v^{(k+1)} - L_v^{(k+1)}$ para $v = 1, 2, ..., N$. Se resta $L_v^{(k+1)}$ para evitar que la energía de baja frecuencia entre en el componente estacional.

Paso 5.

Desestacionalización.- Se calcula la serie desestacionalizada $Y_v - S_v^{(k+1)}$. Si Y_v es un dato perdido en una posición particular de tiempo, entonces también lo será en la

serie desestacionalizada.

Paso 6.

Suavizado en Tendencia.- La serie desestacionalizada es suavizada mediante Loess con los parámetros $q = n_{(t)}$ y $d = 1$. Los valores suavizados se calculan para todas las posiciones de tiempo $(v = 1, 2, ..., N)$, inclusive donde existen valores perdidos. La componente de tendencia del $(k+1) - simo$ bucle, $R_v^{(k+1)}$ para $v = 1, 2, ..., N$, es el conjunto de valores suavizados.

Así la porción suavizada estacional del bucle interno corresponde a los pasos 2,3,y 4, mientras que la porción de suavizado en tendencia corresponde alPaso 6

Para llevar a cabo el Paso 1 en el paso inicial a través del bucle interno necesitamos valores iniciales, $T_v^{(0)}$, para la componente de tendencia. Usando $T_v^{(0)} = 0$ funciona bastante bien. La tendencia se vuelve parte de la Subserie-Ciclo suavizada, $C_v^{(1)}$, pero se elimina en gran medida durante el Paso 4.

2.1.2. Bucle externo

Supongamos que hemos realizado una ejecución inicial del bucle interno para obtener estimaciones, T_v y S_v, de la componentes de tendencia y estacionalidad. Luego el residuo es

$$R_v = Y_v - T_v - S_v$$

(Notemos que el residuo, a diferencia de T_v y S_v, no está definido donde Y_v tiene valores perdidos.) Definiremos un peso a cada posición de tiempo en la que Y_v es observado. Estos *pesos de robustez* reflejan lo extremo que es R_v. Un valor atípico en los datos que resultan en un $|R_v|$ muy grande tendrá un peso pequeño o próximo a cero. Sea

$$h = 6 \text{ mediana}(|R_v|)$$

Luego los pesos de robustez en el tiempo v es

$$\rho_v = B\left(\frac{|R_v|}{h}\right)$$

Donde B es la función bicuadrática de pesos:

$$B(u) = \begin{cases} (1-u^2)^2 & \text{para } 0 \leq u < 1 \\ 0 & \text{para } u \geq 1 \end{cases}$$

Ahora el bucle interno se repite, pero en las series suavizadas de los Paso 2 y Paso 6, el peso del vecindario para el valor en el tiempo v se multiplica por el peso de robustez, ρ_v. Esto es solo un uso de los pesos de confiabilidad discutidos en la Loess. Estas iteraciones de robustez del bucle externo se llevan a cabo un total de $n_{(0)}$ veces. Cada vez que ingresamos al bucle interno después de la pasada inicial no establecemos $T_v^{(0)}$ como lo hicimos en la pasada inicial, sino que usamos el componente de tendencia del Paso 6 del bucle interno anterior.

2.1.3. Elección de Parámetros

El STL tiene 6 parámetros:

- $n_{(p)}$ = Número de observaciones en cada ciclo de la componente estacional,
- $n_{(i)}$ = Número de pasadas a través el bucle interno,
- $n_{(o)}$ = Número de iteraciones robustas del bucle externo,
- $n_{(l)}$ = Parámetro de suavizado para el filtro del paso inferior,
- $n_{(t)}$ = Parámetro de suavizado para la componente de tendencia,
- $n_{(s)}$ = Parámetro de suavizado para la componente estacional.

La elección de los 5 primero parámetros es sencilla. Sin embargo, $n_{(s)}$ debe elegirse cuidadosamente para cada aplicación.

El parámetro $n_{(p)}$ indica la periodicidad de la serie, por ejemplo para datos con periodicidad anual se toma $n_{(p)} = 365$ si los datos son diarios, mientras que para datos mensuales se tomará $n_{(p)} = 12$.

Para la elección de $n_{(i)}$ primero supongamos que no necesitamos las iteraciones robustas ($n_{(p)} = 0$), entonces se quiere escoger $n_{(i)}$ suficientemente grande para que las actualizaciones de las componentes estacional y de tendencia converjan, afortunadamente convergen bastante rápido, por lo que en la mayoría de los casos basta tomar $n_{(i)} = 1$, aunque se recomienda tomar $n_{(i)} = 2$.

En el caso en que necesitemos las iteraciones robustas, se elige $n_{(o)}$ lo suficientemente grande para que las estimaciones de las componentes estacional y de tendencia converjan. En este caso [Cleveland et al., 1990] sugiere fijar $n_{(i)} = 1$.

$n_{(l)}$ siempre se toma como el menor entero impar mayor que $n_{(p)}$.

Para elegir $n_{(s)}$, notamos que cada subserie ciclo se suaviza a medida que $n_{(s)}$ crece, otro criterio a considerar es que $n_{(s)}$ debe ser un impar mayor o igual que 7.

El valor recomendado de $n_{(t)}$ es:

$$n_{(t)} = \left[\frac{1{,}5 n_{(p)}}{(1 - 1{,}5 n_{(s)})} \right]_{impar}$$

2.2. Tratamiento de Valores perdidos

En este capítulo ilustraremos una propuesta para el tratamiento de valores perdidos en series de tiempo estacionales. Para ello consideremos una serie de tiempo (Y_t), de la que conocemos las observaciones

$$y_1, y_2, ..., y_{j-1}, y_j, y_k, y_{k+1}, ..., y_n$$

donde $1 < j < k < n$. Recordemos que la descomposición STL-Loess permité descomponer aditivamente la serie en sus componentes de tendencia y estacionalidad inclusive en aquellos valores de t para los que no conocemos y_t , es decir, para $t = j + 1, j + 2, ..., k - 1$.

Luego de descomoponer Y_t obtenemos su tendencia T_t, estacionalidad S_t (para $t = 1, 2, ..., n$), y el residuo U_t (para $t = 1, 2, ..., j, k, ..., n$). Además estás series cumplen la relación siguiente.

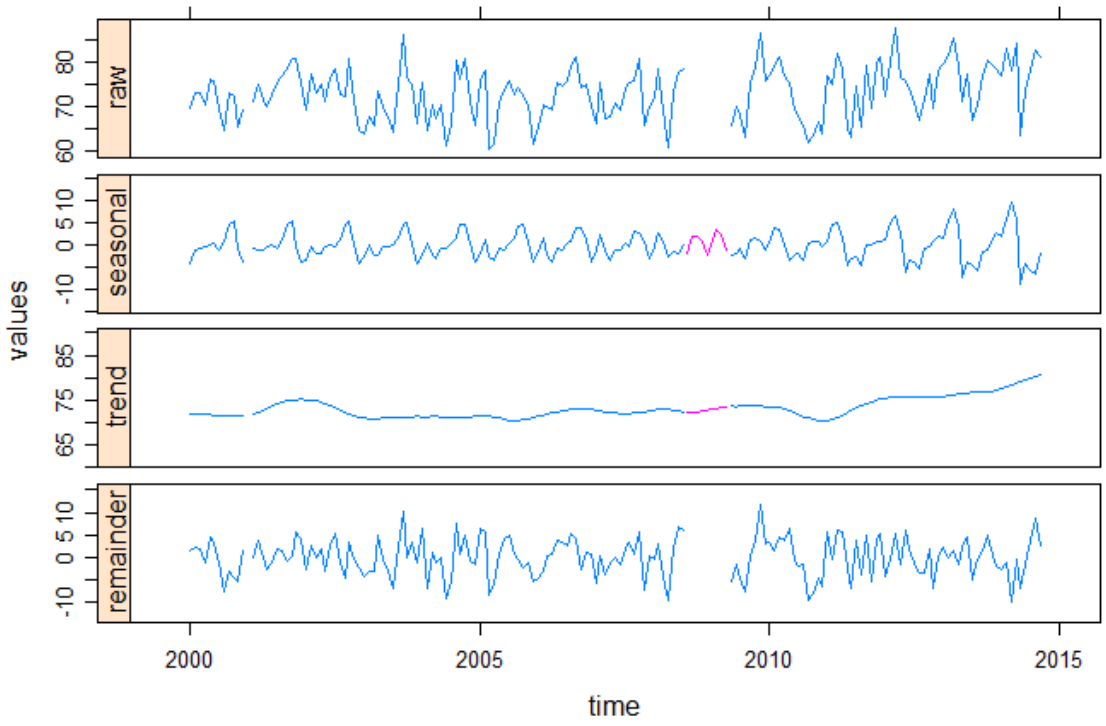
$$Y_t = T_t + S_t + R_t \tag{2.1}$$

Ilustramos lo antes mencionado en el siguiente gráfico (Ver 2.1) , que corresponde a la descomposición STL-Loess de una serie de datos mensuales de precipitación (lluvias) medidos en cierta zona de Brasil, está serie tiene estacionalidad anual (12 meses); notemos que es necesario fijar los parámetros de la descomposición adecuadamente ya que están asociados al número de retardos considerados al estimar tanto la componente estacional como la tendencia.

El gráfico muestra en la primera fila la serie climática, en segundo lugar muestra

su componente estacional, en tercera fila encontramos su componente de tendencia, y finalmente el residuo. Como podemos notar las componentes de tendencia y estacionalidad están definidas en todo el dominio de tiempo.

Figura 2.1: Descomposición STL-Loess de Serie



Notemos que bastaría conocer los valores de R_t en para todo t e inmediatamente conoceríamos los de Y_t gracias a la ecuación (2.1).

Así, proponemos simular los valores perdidos de R_t. Una forma simple de simular dichos valores, es usando el método simulación de la Transformada Inversa (Ver [Ross, 2006]) partiendo de la función de distribución empírica de los Residuos. Esto debido a que los residuos tienden a comportarse como un proceso estacionario, suponiendo que se especificaron bien los parámetros de la descomposición STL.

Pues bien, simulamos los valores perdidos de los residuos usando el siguiente algoritmo.

1. Primero calculamos la función de distribución empírica $\hat{F}_0(u)$ de los residuos de la descomposición.

$$\hat{F}_0(u) := \frac{1}{n}\sum_{i=1}^{n} I_{(y_i \leq u)}$$

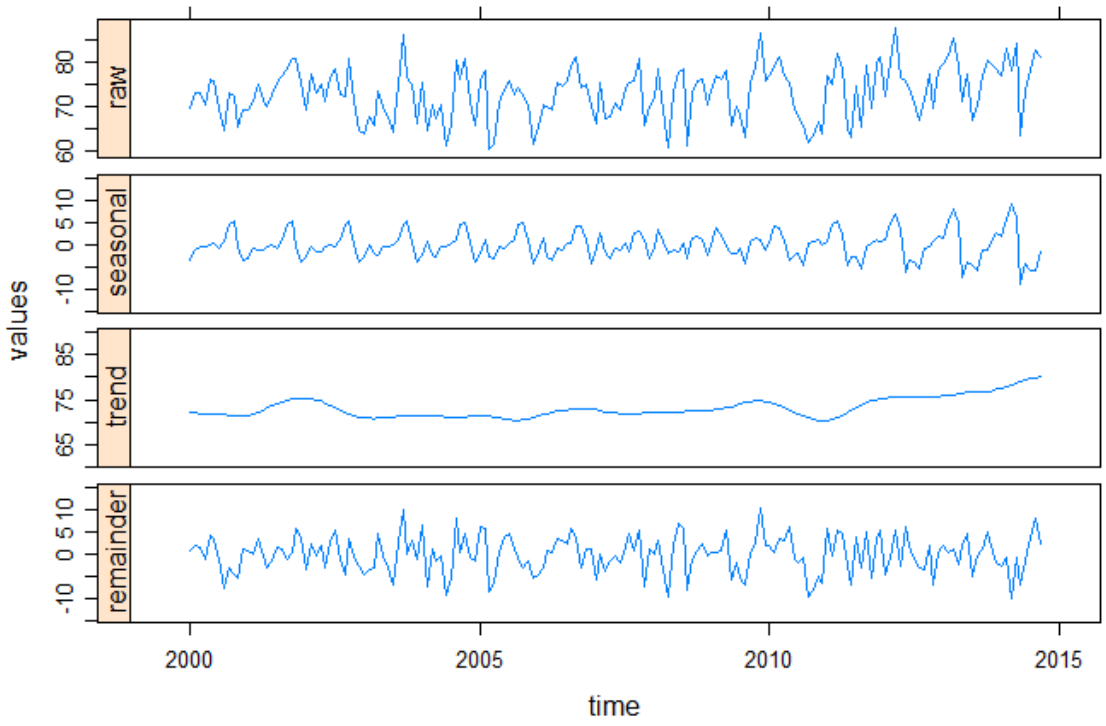
2. Simular $u \sim U[0,1]$

3. Calcular $y := \hat{F}_0^{-1}(u) = inf\{t \in \mathbb{R} : u \leq \hat{F}_0(t)\}$

De esta manera y sigue tiene distribución $\hat{F}_0$ (la distribución de los Residuos)

Volviendo al ejemplo antes mostrado (Ver 2.1), luego de simular los valores perdidos de R_t obtenemos la serie completa Y_t como se puede observar en (2.2)

Figura 2.2: Serie Corregida



2.3. Análisis de Conglomerados (Clúster)

El Análisis Clúster es un técnica de aprendizaje no supervisada que tiene como objetivo dividir un conjunto de objetos en grupos homogéneos (clústers). La partición se realiza de tal manera que los objetos en el mismo clúster son más similares entre sí que los objetos en diferentes grupos según un criterio definido. En muchas aplicaciones reales, el análisis de clúster debe realizarse con datos asociados a series de tiempo. De hecho, los problemas de agrupamiento de series de tiempo surgen de manera natural en una amplia variedad de campos, incluyendo economía, finanzas,

medicina, ecología, estudios ambientales, ingeniería y muchos otros. Con frecuencia, la agrupación de series de tiempo desempeña un papel central en el problema estudiado. Estos argumentos motivan el creciente interés en la literatura sobre la agrupación de series de tiempo, especialmente en las últimas dos décadas, donde se ha proporcionado una gran cantidad de contribuciones sobre este tema. En [Liao, 2005] se puede encontrar un excelente estudio sobre la agrupación de series de tiempo, aunque posteriormente se han realizado nuevas contribuciones significativas. Particularmente importante en la última década ha sido la explosión de documentos sobre el tema provenientes tanto de comunidades de minería de datos como de reconocimiento de patrones. [Fu, 2011] proporciona una visión general completa y exhaustiva de las últimas orientaciones de minería de datos de series de tiempo, incluida una gama de problemas clave como representación, indexación y segmentación de series de tiempo, medidas de disimilitud, procedimientos de agrupamiento y herramientas de visualización.

Una pregunta crucial en el Análisis Clúster es establecer lo que queremos decir con objetos de datos "similares", es decir, determinar una medida de similitud (o disimilitud) adecuada entre dos objetos. En el contexto específico de los datos asociados a series de tiempo, el concepto de disimilitud es particularmente complejo debido al carácter dinámico de la serie. Las diferencias generalmente consideradas en la agrupación convencional no podrían funcionar adecuadamente con los datos dependientes del tiempo porque ignoran la relación de interdependencia entre los valores.

De esta manera, diferentes enfoques para definir una función de disimilitud entre series de tiempo han sido propuestos en la literatura pero nos centraremos en aquellas medidas asociadas a la autocorrelación (simple, e inversa), correlación cruzada y periodograma de las series (Ver: [Struzik and Siebes, 1999]; [Galeano and Peña, 2000]; [Caiado et al., 2006]; [Chouakria and Nagabhushan, 2007]). Estos enfoques basados en características tienen como objetivo representar la estructura dinámica de cada serie mediante un vector de características de menor dimensión, lo que permite una reducción de dimensionalidad (las series temporales son esencialmente datos de alta dimensionalidad) y un ahorro significativo en el tiempo de cálculo, además de que nos ayudan a alcanzar el objetivo central por el que usaremos el Análisis Clúster que es el de la modelización de series de tiempo.

2.3.1. Métricas y Funciones de Disimilitud

Desde un punto de vista general el término proximidad indica el concepto de cercanía en espacio, tiempo o cualquier otro contexto. Desde un punto de vista matemático, ese término hace referencia al concepto de disimilitud o similaridad entre dos elementos. Sea O un conjunto finito o infinito de elementos (individuos, estímulos sujetos u objetos) sobre los que queremos definir una proximidad.

Definición 2.3.1. Dados dos puntos $o_i, o_j \in O$ y δ es una función real de $O \times O \to \mathbb{R}$, con $\delta_{ij} = \delta(o_i, o_j)$. Se diría que δ es una disimilitud si verifica

- $\delta_{ij} = \delta_{ji}, \forall i, j.$
- $\delta_{ii} \leq \delta_{ij}, \forall i, j.$
- $\delta_{ii} = \delta_o, \forall i.$

La primera condición podría eliminarse, aunque resulta necesaria si se desea comparar con una distancia. No obstante, esa condición suele violarse cuando las disimilitud provienen de juicios emitidos por sujetos, ya que éstos no siempre califican igual al par (i, j) que al par (j, i). Las condiciones segunda y tercera suelen establecerse igualmente para $\delta_o = 0$, aunque también es conocido que cuando a un individuo le son presentados dos estímulos idénticos, éste tiende a asignarles algún valor de disimilitud no nulo y generalmente positivo, y además no siempre se define $\delta_o \geq 0$ ya que, si por ejemplo las disimilitud provienen de una transformación, éstas podrían ser negativas.

Existen diferentes medidas para el cálculo de disimilitud entre un par de variables o individuos. Si consideramos una matriz de datos x_{ri}, obtenida de n objetos sobre p variables, algunos ejemplos de medidas son:

- *Distancia euclídea ponderada*

$$\delta_{rs} = \left(\sum_i w_i (x_{ri} - x_{si}) \right)^{1/2}$$

- *Métrica de Minkowski*

$$\delta_{rs} = \left(\sum_i x_{ri} - {x_{si}}^{\lambda} \right)^{1/\lambda}, \quad \lambda \geq 1$$

- *Separación angular*

$$\delta_{rs} = 1 - \frac{\sum_i x_{ri} x_{si}}{(\sum_i x_{ri}^2 \sum_i x_{si}^2)^{1/2}}$$

2.3.2. Métricas para Series de Tiempo

El problema de medir similitudes o diferencias entre datos asociados a series de tiempo ha sido estudiado ampliamente por autores como [Johnson and Wichern, 2004], además de [Galeano and Peña, 2000] propusieron compara las funciones de autocorrelación de las series, [Diggle and Fisher, 1991]con enfoques no paramétricos comparando el espectro de las series, [Piccolo, 1990] que dió una métrica basada en modelos ARIMA, [Diggle and Al Wasel, 1997] quien desarrollo métodos basados en análisis espectral, y [Maharaj, 2000] quien comparó dos series estacionarias basándose en sus parámetros autoregresivos . A continuación se muestran un par de ejemplos de estas métricas

- [Galeano and Peña, 2000] propone una métrica que se basa en la estimación de la función de autocorrelación de las series. Sean (x_t), (y_t) dos series de tiempo, y $\hat{\rho} = (\hat{\rho}_1, \hat{\rho}_2, ..., \hat{\rho}_k)$ el vector de coeficientes de autocorrelación estimados hasta el retardo k (que supondremos es el mayor retardo significativo). Así, se define la distancia entre las series de tiempo x e y como sigue.

$$d_{ACF}(x, y) = \sqrt{(\hat{\rho}_x - \hat{\rho}_y)' \Omega (\hat{\rho}_x - \hat{\rho}_y)}$$

  Donde Ω es una matriz de pesos (simétrica y semidefinida positiva) usualmente se considera $\Omega = I_k$

- [Piccolo, 1990] define una métrica para series de tiempo que pueden representarse como un ARIMA(p,0,q) es decir, series que puedan escribirse en su forma Autoregresiva AR(∞) mediante el operador $\pi(L) = 1 - \pi_1 L - \pi_2 L^2 - ....$ Bajo esas condiciones, se define la métrica siguiente.

$$d_{PIC}(x, y) = \sqrt{\sum_{j=1}^{\infty} (\pi_{(x,j)} - \pi_{(y,j)})^2}$$

2.3.3. Algoritmos de Agrupamiento

Una vez que se determina la medida de disimilitud, se obtiene una matriz de disimilitud inicial (que contiene la disimilitud entre parejas de series), y luego se usa

un algoritmo de agrupamiento convencional para formar los clústers (grupos) con las series. De hecho, la mayoría de los enfoques de agrupamiento de series de tiempo revisados por [Liao, 2005] son variaciones de procedimientos generales como por ejemplo: K-Means, K-Medoids, PAM, CLARA [Kaufman and Rousseeuw, 1986] o de Clúster jerárquico que utilizan una gama de disimilitudes específicamente diseñadas para tratar con series de tiempo y algunas de sus características.

Particionamiento alrededor de Medoides (PAM)

El algoritmo PAM fue propuesto en [Rousseeuw and Kaufman, 1990], tiene por objetivo hallar k grupos (clústers), esto mediante la identificación de objetos representativos que están lo más centralmente localizados dentro de cada grupo, estos objetos se conocen como "medoides". Una vez identificados los medoides, los objetos se agrupan al medoide más similar.

Observación. La çalidad de agrupamiento"del método se mide como la distancia promedio entre los objetos y sus respectivos medoides.

La manera en la que PAM halla los k medoides es partir de un conjunto arbitrario de objetos para luego intercambiarlos sucesivamente de tal manera de que en cada paso se mejore la calidad de agrupamiento.

Por ejemplo, para medir el efecto de intercambiar un objeto O_{i1} por O_{i2} el algoritmo PAM calcula el costo $C_j(i_1, i_2)$ (para todo objeto O_j no selecionado. $C_j(i_1, i_2)$ se calcula según cada uno de los siguientes casos:

1. Supongamos que O_j pertenece al grupo representado por el medoide O_i. Luego supongamos que O_j es más parecido a O_k que O_h. Así, si reemplazamos O_i por O_h como medoide del grupo, entonces O_j pertenecería al grupo representado por O_k. Por lo tanto el costo de intercambio de medoides respecto de O_j es :

$$C_j(i, h) = d(O_j, O_k) - d(O_j, O_i)$$

   Notemos que $C_j(i, h) \geq 0$

2. Supongamos que O_j pertenece al grupo representado por el medoide O_i. Pero esta vez O_j es menos parecido a O_k que O_h. Así, el costo de reemplazar O_i por O_h viene dado por:

$$C_j(i,h) = d(O_j, O_h) - d(O_j, O_i)$$

En este caso $C_j(i,h)$ puede ser positivo o negativo.

3. Supongamos que O_j pertenece a un grupo distinto al representado por el medoide O_i . Sea O_k el medoide de ese grupo. Luego supongamos que O_j es más similar a O_k que a O_h, entonces:

$$C_j(i,h) = 0$$

4. Supongamos que O_j pertenece al grupo representado por el medoide O_i. Entonces reeplazar O_i con O_h provocaría que O_j pase del grupo representado por O_h al grupo representado por O_k. Así, el costo viene dado por:

$$C_j(i,h) = d(O_j, O_h) - d(O_j, O_k)$$

Notemos que $C_j(i,h) < 0$

5. Finalmente el costo total de reemplazar O_i por O_h está dado por:

$$T(i,h) = \sum_i C_j(i,h)$$

Algoritmo

1. Seleccionar k objetos arbitrariamente
2. Calcular $T(i,h)$ para todos los pares de objetos, tales que O_i está seleccionado y O_h no.
3. Seleccionar el par O_i, O_h que minimice $T(i,h)$. Si el mínimo $T(i,h)$ es negativo, reemplazar O_i con O_h y vuelva al paso 2.
4. Caso contrario, para cada objeto no seleccionado, hallar el medoide más parecido.

Nota. Resultados experimentales muestran que PAM funciona adecuadamente con conjuntos de datos pequeños (100 objetos), pero no es eficiente para grandes conjuntos de datos, lo que es evidente al analizar la complejidad del algoritmo PAM donde vemos que cada iteración del algoritmo tiene un orden de complejidad de $O(k(n-k)^2)$.

CLARA

CLARA (Clustering Large Aplications) es un método desarrollado por Kaufman y Rousseuw con la finalidad de lidiar con un gran número de datos. El algoritmo CLARA consiste básicamente en aplicar PAM sobre una muestra aleatoria de objetos, en lugar de aplicarlo a todos los objetos. Esto debido a que los medoides de una muestra aproximaría a los medoides de todos los objetos. Para mejorar esta aproximación CLARA toma varias muestras y devuelve la mejor agrupación. En este caso, la calidad de agrupamiento se mide como la distancia promedio entre todos los objetos y sus medoides (no solo los de la muestra).

Algoritmo

1. Para i de 1 a L realizar:
2. Tomar una muestra de m objetos aleatoriamente, y ejecutar el algoritmo PAM para hallar los k medoides de la muestra.
3. Para cada objeto O_j en la data entera, determinar cual de los k medoides es el más similar.
4. Calcular la distancia (o disimilitud) promedio del agrupamiento obtenido en el paso anterior. Si este valor es menor al mínimo anterior, actualizamos el valor mínimo y guardar los k medoides del paso 2 como los mejores medoides obtenidos hasta el momento.

Corridas experimentales realizadas en [Rousseeuw and Kaufman, 1990] muestran que tomar $L = 5$ muestras de tamaño $m = 40 + 2k$ da buenos resultados.

Observación. Se puede corroborar que el orden de complejidad del algoritmo CLARA es $O(k(40 + k)^2 + k(n - k))$, esto explica porque CLARA es más eficiente que PAM para valores grandes de n.

2.3.4. Validación

Una etapa adicional dentro del análisis clúster consiste en determinar la cantidad de clústers que es más apropiada para los datos. Idealmente, los clústers resultantes no solo deberían tener buenas propiedades estadísticas (compactas, bien separadas, conectadas y estables), sino también resultados relevantes. Se han propuesto una variedad de medidas y métodos para validar los resultados de un análisis clúster y

determinar tanto el número de clústers, así como identificar qué algoritmo de agrupamiento ofrece el mejor rendimiento, algunas de estas ellas pueden encontrarse en [Fraley and Raftery, 1998]; [Duda et al., 2001] ; [Salvador and Chan, 2004] ; [Kerr and Churchill, 2001]. Esta validación puede basarse únicamente en las propiedades internas de los datos o en alguna referencia externa.

2.4. Análisis de Componentes Principales Funcional

2.4.1. Estadísticos para Datos Funcionales

A continuación se muestra un resumen de los estadísticos clásicos aplicados a datos funcionales. El primero de ellos corresponde a la función Media que toma los valores:

$$\bar{x}(t) = \frac{1}{n}\sum_{i=1}^{n} x_i(t)$$

De igual manera la función Varianza toma los valores

$$var_x(t) = \frac{1}{n-1}\sum_{i=1}^{n} [x(t) - \bar{x}(t)]^2$$

Y la función Desviación Estándar es la raíz cuadrada de la función Varianza.

Un estadístico también importante es la covarianza y la correlación. En este caso la función de covarianza nos muestra entre las observaciones de $x(t)$ en distintos valores de t, se define como sigue:

$$cov_x(t_1, t_2) = \frac{1}{n-1}\sum_{i=1}^{n} [x_i(t_1) - \bar{x}(t_1)][x_i(t_2) - \bar{x}(t_2)]$$

Luego, la función de correlación viene dada por:

$$corr_x(t_1, t_2) = \frac{cov_x(t_1, t_2)}{\sqrt{var_x(t_1) var_x(t_2)}}$$

Como vemos las versiones funcionales de los estadísticos clásicos son análogas a sus definiciones tradicionales.

2.4.2. ACP clásico

El concepto central explotado una y otra vez en estadística multivariante es el de tomar una combinación lineal de variables por ejemplo:

$$f_i = \sum_{j=1}^{k} \beta_j x_{ij}, \quad i = 1, 2, ..., N$$

donde β_j es un coeficiente de ponderación (pesos) aplicado a los valores observados x_{ij} de la variable $j - sima$. Podemos escribir la ecuación anterior en su forma vectorial como:

$$f_i = \langle \beta, x_i \rangle \quad i = 1, 2, ..., N$$

Donde $\beta = (\beta_1, \beta_2, ..., \beta_k)$ y $x_i = (x_{i1}, x_{i2}, ..., x_{ik})$

En el caso multivariante elegimos los pesos para mostrar los tipos de variación que están fuertemente representados en los datos. El análisis de componentes principales se puede definir en términos del siguiente procedimiento paso a paso, que define conjuntos de ponderaciones normalizadas que maximizan la variación de f_i.

1. Primero se halla el vector de pesos $\phi_1 = (\phi_{11}, \phi_{21}, ..., \phi_{k1})'$ para el cual la combinación:

   $$f_{i1} := \sum_{j=1}^{k} \phi_{j1} x_{ij} = \langle \phi_1, x_i \rangle$$

   se maximice $\frac{1}{N} \sum_i f_{i1}^2$ (media cuadrática) , sujeto a la condición de que

   $$\sum_{j=1}^{k} \phi_{j1}^2 = ||\phi_1||^2 = 1$$

2. En las siguientes iteraciones (de la 2 hasta k como máximo) hacemos seguimos un procedimiento parecido. Por ejemplo para $m - simo$ iteración calculamos el vector de pesos ϕ_m y los nuevos valores $f_i m = \langle \phi_m, x_i \rangle$ que maximicen su media cuadrática $\frac{1}{N} \sum_i f_{im}^2$, sujetos a la condición de que $||\phi_m||^2 = 1$ y las $m - 1$ condiciones siguientes:

   $$\sum_{j=1}^{k} \phi_{jr} \phi_{jm} = \langle \phi_r, \phi_m \rangle = 0, \quad r < m$$

La motivación para el primer paso es que al maximizar el cuadrado medio, estamos identificando el modo de variación más fuerte y más importante en las variables. La restricción de suma de cuadrados de unidades en los pesos es esencial para que el problema esté bien definido; sin él, los cuadrados medios de los valores de combinación lineal podrían hacerse arbitrariamente grandes. En los pasos segundo y subsiguientes, buscamos nuevamente los modos de variación más importantes, pero requerimos que los pesos que los definen sean ortogonales a los identificados anteriormente, de modo que indiquen algo nuevo. Por supuesto, la cantidad de variación medida en términos de $\frac{1}{N}\sum_i f_{im}^2$ decrece en cada iteración.

Los coeficientes de las combinaciones lineales f_{im} se las conoce como *scores de la componente principal* y son útiles en el sentido que indican cuanto de la variabilidad en los datos proviene de la componente principal asociada.

2.4.3. ACP Funcional

Análogamente al ACP clásico, las contrapartes de los valores variables son los valores de función $X_i(s)$, de modo que el índice discreto j en el contexto clásico se reemplaza por el índice continuo s. Las sumas sobre j se reemplazan por integrales sobre s para definir la combinación lineal:

$$f_i = \int \beta(s)x_i(s)\mathrm{ds} = \langle \beta, x_i \rangle$$

Los pesos β_j ahora se convierten en funciones de ponderación con valores $\beta(s)$.

De igual manera el primer paso del ACP funcional consiste en hallar la función de pesos ϕ_1 que maximice $\frac{1}{N}\sum_i f_{i1}^2 = \frac{1}{N}\sum_i \langle \phi_1, x_i \rangle$ sujeto a la condición de que $||\phi_1||^2 \int \phi_1^2(s)\mathrm{ds} = 1$.

Luego en las siguientes iteraciones calculamos la función de ponderación ϕ_m elegida de modo que maximice $\frac{1}{N}\sum_i \langle \phi_m, x_i \rangle$ sujeto a la condición de que $||\phi_m||^2 = 1$ y las $m-1$ condiciones de ortogonalidad siguientes:

$$\langle \phi_r, \phi_m \rangle = 0, \quad r < m$$

Cada función de ponderación tiene la tarea de definir el modo de variación más importante en las curvas sujetas a que cada modo sea ortogonal a todos los modos definidos en los pasos anteriores

2.4.4. Bases ortonormales óptimas

Hay varias otras formas de motivar ACP, y una es definir el siguiente problema: Queremos encontrar un conjunto de K funciones ortonormales ϕ_m para que la expansión de cada curva en términos de estas funciones básicas se aproxime lo más posible a la curva . Dado que estas funciones básicas son ortonormales, se deduce que la expansión será de la forma

$$\hat{x}_i(t) = \sum_{j=1}^{K} f_{ij}\phi_j(t)$$

donde

$$f_{ij} = \langle x_i, \phi_j \rangle$$

. Como criterio de ajuste en cada curva usaremos el error cuadrático definido como:

$$||x_i - \hat{x}_i||^2 = \int [x_i(s) - \hat{x}_i(s)]^2 \mathrm{ds}$$

Y por lo tanto una media global de ajuste viene dada por

$$SSE = \sum_{i=1}^{N} ||x_i - \hat{x}_i||^2$$

Así el problema se reduce a hallar la base que minimice SSE, y análogamente al ACP clásico, este problema resulta ser equivalente al de hallar los valores y vectores propios de la matriz de covarianzas, que para el caso funcional resulta en hallar funciones propias a partir de la función de covarianzas, es decir, resolver la ecuación-propia siguiente:

$$\int v(s,t)\phi(t)\mathrm{dt} = \langle v(s,.), \phi \rangle = \rho\phi(s) \tag{2.2}$$

donde v es la función de covarianza dada por

$$v(s,t) = \frac{1}{N}\sum_{i=1}^{N} x_i(s)x_i(t)$$

Notemos que el lado izquierdo de la ecuación (2.2) puede expresarse como una transformación integral digamos V definida por:

$$V\phi = \int v(.,t)\phi(t)\mathrm{dt}$$

A esta transformación integral se la conoce como 'Operador de Covarianza". Así, obtenemos de la ecuación (2.2) la ecuación propia:

$$V\phi = \rho\phi$$

Donde ϕ resulta ser una función propia asociada al operador de covarianza.

2.5. Modelos SARIMAX

El modelo SARIMAX es una extensión del modelo SARIMA (Seasonal Autoregressive Integrated Moving Average), mejorado con la capacidad de integrar variables exógenas (explicativas) para aumentar su rendimiento de pronóstico. Esta versión multivariable del modelo SARIMA$(p,d,q)(P,D,Q)_s$, llamada ARIMA estacional con factor exógeno (es decir, SARIMAX), que se expresa formalmente como:

$$\varphi_p(B) = \Phi_P(B^s)\nabla^d\nabla_s^D y_t = \sum_{k=1}^{K} \beta_k x_t^{(k)} + \theta_q(B)\Theta_Q(B^s)\varepsilon_t$$

Donde:

- y_t es la variable de estudio,
- $x_t^{(k)}$ es la k-ésima variable exógena,
- B es el operador de retardos ($B^k y_t = y_{t-k}$),
- $\varphi_p(B)$ es el polinomio AR de retardos de orden p,
- $\theta_q(B)$ es el polinomio MA de retardos de orden p,
- $\Phi_P(B)$ es el polinomio AR de retardos de orden p,
- $\Theta_Q(B)$ es el polinomio MA de retardos de orden p,
- $\nabla^d := (1-B)^d$ es el operador de diferenciación de orden d
- $\nabla_s^D := (1-B^s)^D$ es el operador de diferenciación estacional de orden D (y estacionalidad s) .

Capítulo 3

Metodología

Descripción General

3.1. Elección de Métrica

Se selecciona la métrica (en general se usa una función de disimilitud) asociada la ?Autocorrelación? (relación con sus propios retardos), ya que compara el comportamiento Temporal de una pareja de series por lo que es útil para una posterior modelamiento (SARIMAX por ejemplo, que considera un modelo dependiente del pasado de la serie). A partir de esta pseudo-métrica se genera una matriz de distancias entre todas las estaciones de ?Vazoes?.

$$d_{ACF}(x,y) = \sqrt{(\hat{\rho}_x - \hat{\rho}_y)'\Omega(\hat{\rho}_x - \hat{\rho}_y)}$$

donde $\hat{\rho}$ es el vector de coeficientes de autocorrelación estimados, mientras que Ω es una matriz de pesos usualmente $\Omega = I$ y así se obtiene la distancia Euclidea entre los coeficientes de autocorrelación, o $\Omega = [cov(\hat{\rho})]^{-1}$ y en este caso se obtiene la distancia de Mahalanobis entre los coeficientes de autocorrelación.

3.2. Elección del Método de Clústerización

En esta etapa se elige una técnica de clústerización o agrupamiento que típicamente parte de una matriz de distancias entre objetos (en este caso las series de Vazoes), y considerando estas distancias agrupa estos objetos de tal manera que en

cada grupo se encuentren objetos muy cercanos entre si (es decir busca un grupo homogeneo), pero distantes a objetos pertenecientes a otros grupos. Entre las técnicas que se consideraron tenemos al Clúster Jerárquico que genera un árbol llamado ?Dendograma? que muestra paso a paso como se forman los grupos. Una segunda técnica más eficiente que se considera es el algoritmo CLARA de clusterización, que es usado cuando se necesita agrupar una gran cantidad de objetos.

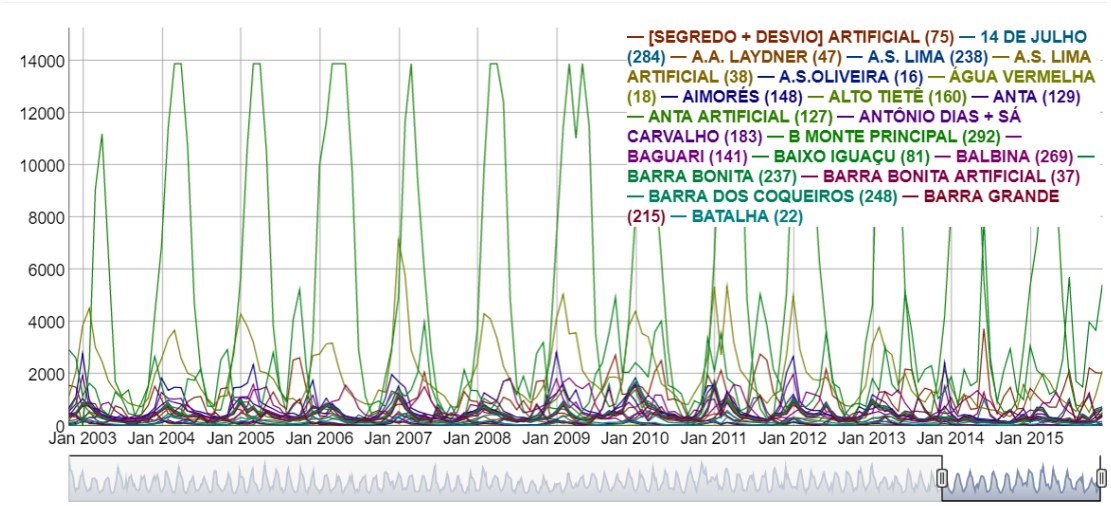
Observación. La elección del número de grupos en los datos es a veces subjetiva y depende de la experiencia del investigador. Sin embargo, podemos encontrar una partición natural en el conjunto de datos mediante el llamado coeficiente de inconsistencia

Figura 3.1: Mapa de Clústers

3.3. Análisis de Componente Principales Funcional

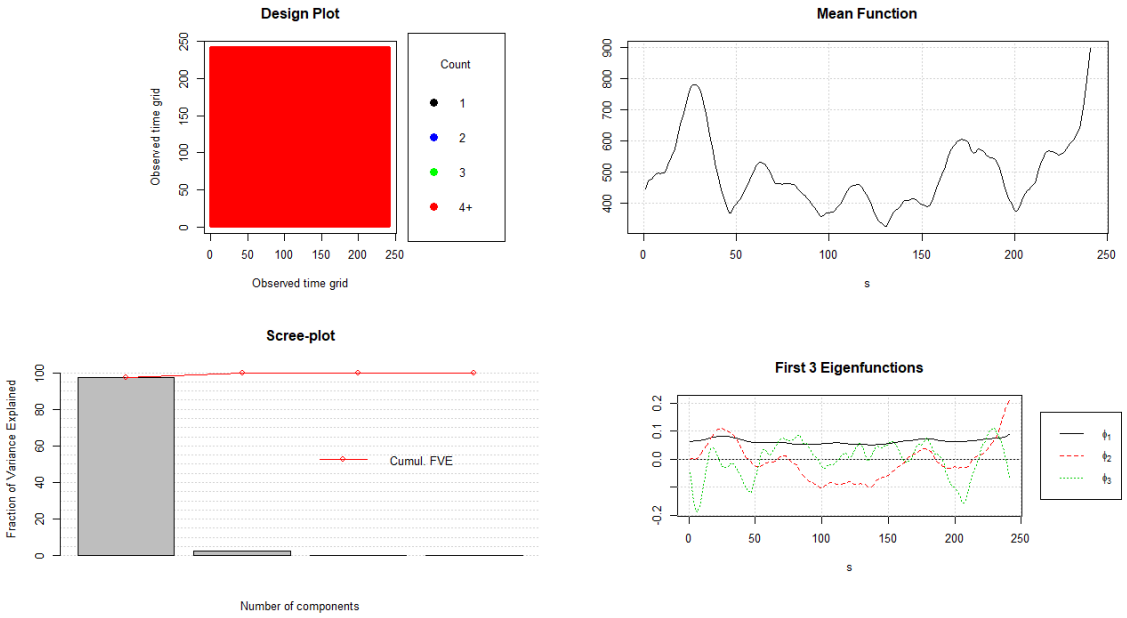
El ACP típico se encarga de reducir la dimensión de un conjunto de datos mediante el cálculo de un grupo mucho menor de variables ortogonales que mejor representan el conjunto original de datos. Análogamente el análisis de componentes principales funcionales (ACPF) es una extensión del ACP clásico en el que las componentes principales están representadas por funciones y no por vectores (Ramsay Sylverman, 2005). La filosofía principal del análisis de datos funcionales es la creencia de que la mejor fuente de información es la función observada y no un arreglo

Figura 3.2: Series del Clúster 1



de números. Así podemos usar todas las series de Vazoes de un clúster para hallar esta función (o funciones) que representan el comportamiento de todo el clúster.

Figura 3.3: Análisis de Componentes Principales Funcional



3.4. Limpieza de Datos de Clima

Un punto importante previo al modelamiento, es la limpieza de los datos. En este caso contamos con una alta presencia de valores perdidos especialmente en las series asociadas a variables Climáticas. Por lo que consideramos retirar todas

aquellas series que contengan más del 10Mientras que para las restantes, diseñamos un algoritmo de corrección de los valores perdidos de las series. Dicho sea de paso que estas series tienen la peculiaridad de ser Estacionales.

Pues bien aplicando el algoritmo de de Limpieza de Datos expuesto en el capítulo anterior obtenemos los siguiente

Figura 3.4: Serie de Tiempo Climática

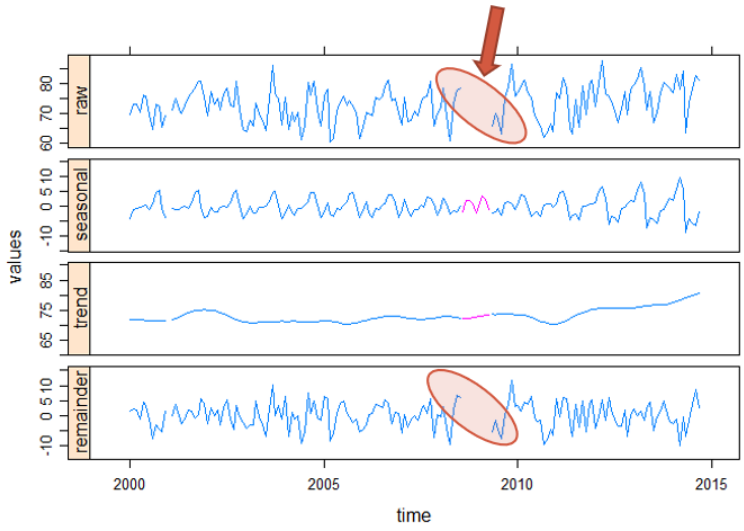
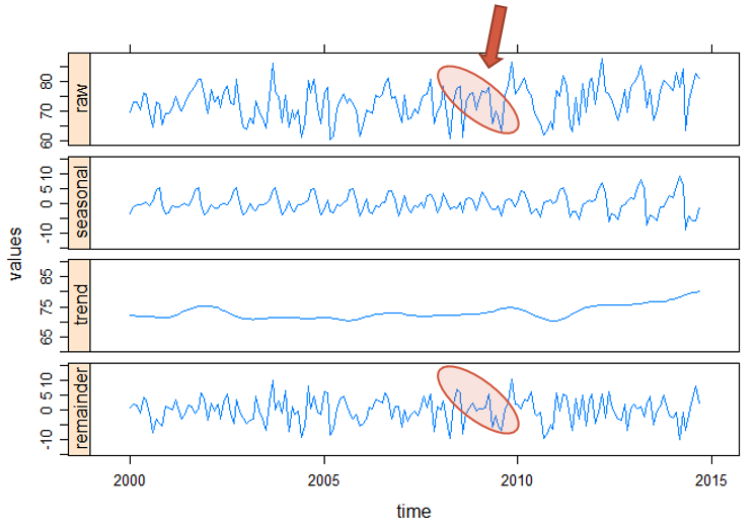


Figura 3.5: Serie de Tiempo Climática Corregida



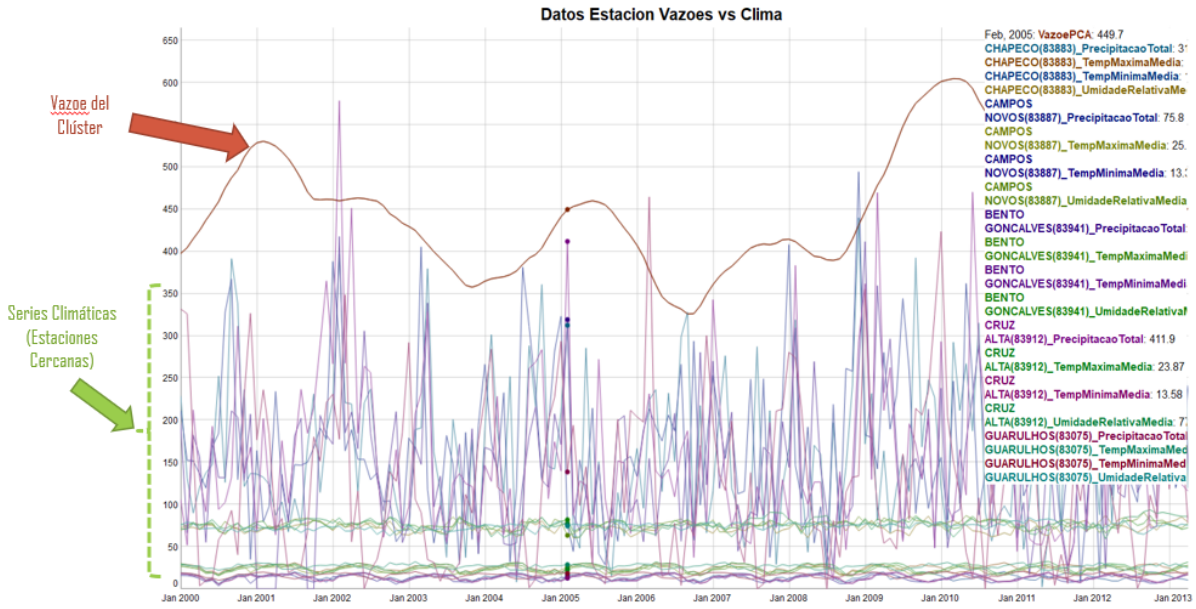
3.5. Agrupar datos de Vazoes y Clima

Una vez corregidas las series climáticas, el criterio para agrupar las series climáticas a series de Vazoes de determinado clúster es el siguiente: Para cada serie de

Vazoe del clúster se busca la serie climática de la estación más cercana, al final tenemos un listado de estaciones climáticas por clúster (donde podría esta repetida una o más estaciones pero luego se quitan las estaciones repetidas).

Finalmete esta lista de estaciones climáticas (y las 6 variables que la conforman) constituyen las variables explicativas del modelo que plantearemos adelante para poder explicar el comportamiento del Flujo (Vazoe) de cada Clúster, es decir de la serie Vazoe que representa el clúster obtenida del ACP-Funcional.

Figura 3.6: Serie de Tiempo Climática Corregida



3.6. Modelamiento de cada clúster

Finalmente formulamos un modelo SARIMAX, mismo que considera la parte estacional de los Flujos (Vazoes) representantes de cada Clúster, y además los relaciona con las series climáticas asociadas a dicho clúster. El modelo propuesto es SARIMAX$(p,0,q)(P,D,Q)_s$

$$\varphi_p(L)\Psi_P(L^s) \bigtriangledown_s^D V_t = \sum_{k=1}^{w} \beta_k C_{kt} + \phi_q(L)\Phi_Q(L^s)e_t$$

Donde V_t es el caudal estimado del clúster en el tiempo t, C_{kt} son las variables de Clima de la estación k en el tiempo t.

Una modificación del modelo propuesto es usar un SARIMAX$(p,0,q)(P,D,Q)_s$, pero modelando está vez los Logaritmos tanto de Vazoes como de las Series Climáticas.

$$\varphi_p(L)\Psi_P(L^s)\bigtriangledown_s^D \log(V_t) = \sum_{k=1}^{w} \beta_k \log(C_{kt}) + \phi_q(L)\Phi_Q(L^s)e_t$$

Figura 3.7: Ajuste y Predicción del Modelo SARIMAX

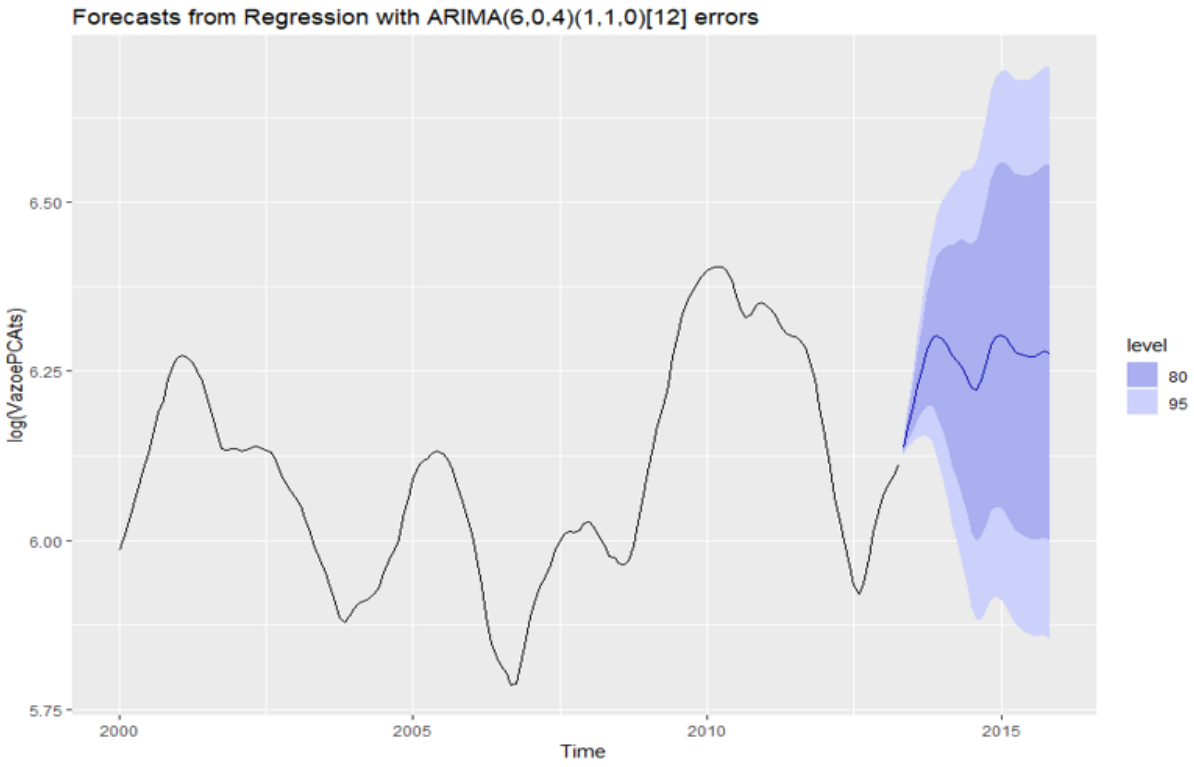
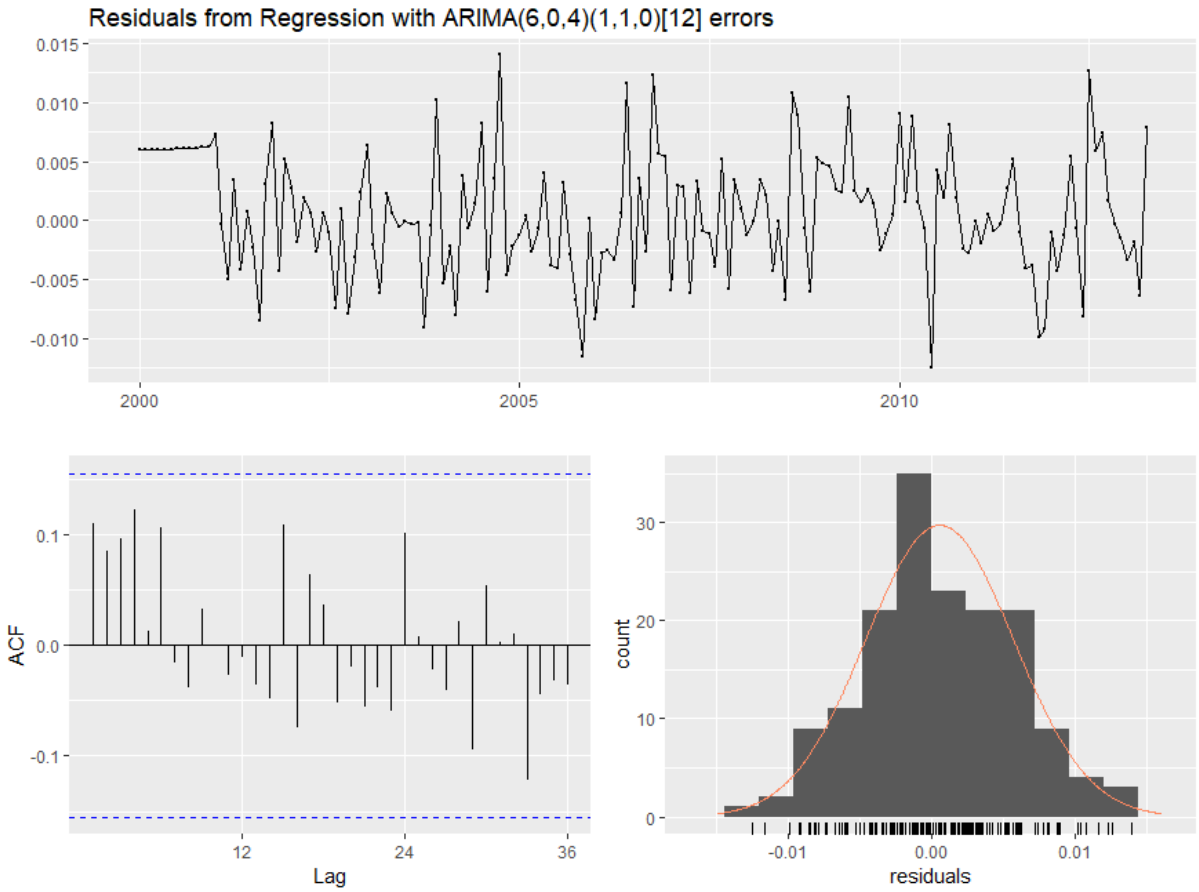


Figura 3.8: Residuos del Modelo SARIMAX



Capítulo 4

Conclusiones y Recomendaciones

Apéndice A

Aplicación Web

A continuación, mostramos la implementación en código R de la Aplicación Web desarrollada con el paquete *Shiny*, [RStudio, Inc, 2013], que contiene el análisis completo de las series de tiempo de flujos de rios de Brasil. Este aplicacion esta compuesta principalmente por tres archivos: *global* , *ui* y *server* .

Nota. La aplicación Web con el análisis completo, se encuentra disponible para su uso en la siguiente dirección: https://cristianpachacama.shinyapps.io/Tesis/. Además puede encontrar el código fuente de la misma en el repositorio de GitHub: https://github.com/CristianPachacama/AppTesis/

A.1. Paquetes (global.R)

Este archivo contiene la declaración de los paquetes extras, que contienen tods las funciones que se usaran en la aplicación web, y que son necesarios para su correcto funcionamiento.

```
#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
#------------------        global.R       ------------------
#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

pkgTest <- function(x){
  if (!require(x,character.only = TRUE)){
    install.packages(x,dep=TRUE)
    if(!require(x,character.only = TRUE)) stop("Paquete no encontrado")
  }
}
```

```r
#Descarga de Paquetes ================================
pkgTest("shinydashboard")
pkgTest("ggplot2")
pkgTest("dygraphs")
pkgTest("TSstudio")
pkgTest("leaflet")
pkgTest("htmltools")
pkgTest("rgdal")
pkgTest("readr")
pkgTest("DT")
pkgTest("dplyr")
pkgTest("reshape2")
pkgTest("lmtest")
pkgTest("TSdist")
pkgTest("xts")
pkgTest("stlplus")
pkgTest("TSA")
pkgTest("forecast")
pkgTest("smacof")
pkgTest("cluster")
pkgTest("ks")
pkgTest("fpca")
pkgTest("fdapace")

# Paquetes Necesarios ===============
library(shiny)
library(shinythemes)
library(shinydashboard)
#Graficos
library(ggplot2)
library(dygraphs)
library(TSstudio)
#Mapas
library(leaflet)
library(htmltools)
library(rgdal)
#Tablas
library(readr)
library(DT)
library(dplyr)
library(reshape2)
#Estadisticos
library(lmtest)
```

```
#Series de Tiempo
library(TSdist)
library(xts)
library(TSA)
library(forecast)
# STL - Loess
library(stlplus)
#MDS y Cluster
library(smacof)
library(cluster)
#ACP Funcional
library(ks)
library(fpca)
library(fdapace)

#» Carga de Datos
load('Data/Actual/InterfazMes.RData')
load('Data/Actual/DataVazoes.RData')
load("Data/Actual/VazoesCode.RData")
clima_dat=clima_dat2
particion = 0.20 #Particion Entrenamiento
set.seed(2)
# Matiz de Distancias
source(file ="Code/SARIMAX/Extras/DistanciasAVazoes.R",local = TRUE)
```

A.2. Interfaz de Usuario (ui.R)

La interfaz de usuario está compuesta por todos los elementos visuales desde donde el usuario de la aplicación puede interactuar con la misma, en este caso está destinada a que el usuario fije los parámetros que posteriormente son usados como insumos para la ejecución de los análisis (en el "sever"), finalmente muestra a los usuarios los resultados del análisis realizado.

```
# ================================================================
# !!!!!!!!!!!!!!!!!!!!    USER INTERFACE    !!!!!!!!!!!!!!!!!!!!
# ================================================================

navbarPage(
  id = 'tesis' ,
  title = "Tesis",
  header = tags$h2(" - ", tags$head(
```

```
  tags$link(rel = 'shortcut icon',
            href = 'epn.ico',
            type = 'image/x-icon')
)),
position = "fixed-top",
#theme=shinytheme('flatly'),#theme = 'estilo.css',
footer = fluidRow(
  column(
    12,
    img(src = 'epn_logo.png', width = '30px', align = 'center'),
    tags$b('Proyecto: '),
    ' "Extreme low Levels of setreamflow in Hydropower Plants".' ,
    '-',
    tags$a('Departamento de Matemática - EPN (2018)',
           href = 'http://www.epn.edu.ec'),
    tags$b('  ||  '),
    tags$b('Desarrollado por: '),
    tags$a('Cristian Pachacama', href =
             'http://www.linkedin.com/in/cristian-david-pachacama')
  )
),

#INTRODUCCION E INFORMACION DEL PROYECTO ---------------------------
tabPanel(
  'Introducción',
  icon = icon('home'),

  fluidRow(
    sidebarPanel(
      img(src = 'epn_logo2.png', width = '90%', align = 'center'),
      fluidRow(' '),
      hr(),
      fluidRow(
        column(3, tags$b('Proyecto Titulación:')),
        column(1),
        column(
          8,
          'Análisis Clúster para series de tiempo estacionales
          y modelización de caudales de ríos del Brasil.'
        )
        ),
      hr(),
      fluidRow(column(3, tags$b('Proyecto Semilla:')), column(1),
```

```
            column(8, 'PIS-16-14')),
    hr(),
    fluidRow(
      column(3, tags$b('Linea de Investigación:')),
      column(1),
      column(8, 'Modelos Econométricos')
    ),
    hr(),
    fluidRow(column(3, tags$b('Departamento:')), column(1),
            column(8, 'Matemática')),
    hr(),
    fluidRow(
      column(3, tags$b('Directora:')),
      column(1),
      column(8, 'PhD. Adriana Uquillas')
    ),
    hr(),
    fluidRow(column(3, tags$b('Autor:')), column(1),
            column(8, 'Cristian Pachacama'))


    ),

  mainPanel(
    h3(
      'Análisis Clúster para series de tiempo estacionales
      y modelización de caudales de ríos del Brasil.'
    ),
    hr(),
    h4('Resume:'),
    fluidRow(' '),
    p(
      'This paper deals with the application of the
      Cluster Analysis for Time Series
      oriented to the modeling of flows of the main
      rivers of Brazil, which were measured
      in 150 stations distributed in them, this from
      climatic variables and the combination
      of techniques of modeling as Principal
      Functional Components Analysis (FPCA),
      SARIMAX and STL-Loess.'
    ),
    p(
```

```
      'Specifically what is done is to create a
      small number of clusters (from 2 to 4 clusters)
      from the 150 stations (where the flows were
      measured), where each group will
      contain stations in which their flows have
      a temporary behavior similar possible,
      then for each of these clusters, through the
      use of ACPF, we will find a single time
      series that summarizes the behavior of the
      flows of the cluster. Finally, the time series
      of each cluster is modeled from climatic
      variables, using them as explanatory
      variables in the SARIMAX modeling framework.'
    ),
    p(
      'We will show later the advantages and the
      efficiency of modeling a huge amount
      of time series with the use of these techniques,
      this because the model that explains
      each cluster can be extended (using the
      same delays and explanatory variables) to
      each of the time series that compose it.
      We perform comparative studies between
      an individual model (SARIMAX) for a specific
      flow and the model of the cluster
      to which it belongs, obtaining similar results
      in terms of predictability. Where an
      Average Quadratic Error (RMSE) of 0.3 % and
      an AIC of 652,21 was obtained for
      the individual model, while for the cluster
      model an RMSE of 0.4 % was obtained,
      and an AIC of 762,32'
    ),
    p(
      'Thus we show that we managed to move from
      the problem of modeling 150 time
      series, to modeling the time series of a few clusters.'
    ),
    br(),
    p(
      tags$b('Keywords:'),
      tags$i(
        "Time Series Cluster Analysis,
```

```
          STL-Loess decomposition, Functional
          Principal Component Analysis"
        )
        )

        )

      ),
  hr()


      ),

# ANALISIS CLUSTER DE SERIES VAZOES ============================
tabPanel(
  'Clusters',

  fluidRow(
    # Panel Lateral  -----------
    sidebarPanel(
      h4('Cluster de Series de Tiempo'),
      p(
        'Primero selecciona una de las Métricas
        definidas para series de tiempo.'
      ),
      selectInput(
        'vaz_clus_metric',
        label = 'Selecciona Métrica',
        selected = 'D_acf',
        list(
          'Correlación Cruzada' = 'D_ccor',
          'Autocorrelación' = 'D_acf',
          'Correlación de Pearson' = 'D_cor',
          'Correlación Temporal' = 'D_cort',
          'Métrica Euclidea' = 'D_euc',
          'Métrica de Fourier' = 'D_fourier',
          'Métrica Infinito' = 'D_ifnrm',
          'Métrica Manhatan' = 'D_manh',
          'Métrica de Minkwoski' = 'D_mink',
          'Autocorrelación Parcial' = 'D_pacf',
          'Periodograma' = 'D_per'
        )
      ),
```

```
        p('Luego elige un método de clusterización
          (agrupamiento).'),
        selectInput(
          'vaz_clus_metod',
          label = 'Selecciona Método',
          selected = 'clara',
          list(
            'K-Medias' = 'kmedias',
            'K-Medoid (CLARA)' = 'clara',
            'Cluster Gerárquico' = 'gerarquico'
          )
        ),
        p('Finalmente elige el número de clusters
          que quieres que se formen.'),
        sliderInput(
          'vaz_clus_k',
          label = 'Número de Clusters',
          min = 2,
          max = 8,
          value = 4
        ),
        actionButton(
          'vaz_clus_boton',
          label = 'Clusterizar',
          icon = icon('braille')
        ),
        hr(),
        h4('Gráfico de Series'),
        p(
          'Para graficar una o varias series,
          primero clusteriza las estaciones, luego
          seleccione los nombres de las estaciones
          correspondientes en la Tabla que
          se encuentra en la parte inferior derecha'
        ),
        hr(),
        #Link a pestaña ACP Funcional
        p(
          'Si desea puede seguir con el Análisis
          de Componentes Principales Funcional
          de las Series de Flujos en la pestaña',
          actionLink(inputId = "pestania_acpf", label = "ACP Funcional")
        )
```

```
      ),
    # Panel Principal -----------
    mainPanel(
      h3('Mapa de Estaciones Clusterizadas: Vazoes '),
      hr(),
      leafletOutput("mapa_cluster", width = "100%", height = "450px"),
      hr(),
      h4('Tabla de Estaciones por Cluster'),
      fluidRow(
        dataTableOutput(outputId = "tabla_cluster"),
        #, width = "50%")),
        hr(),
        h4("Grafico de las Series"),
        dygraphOutput('vaz_clu_grf')
      )
    ),
    hr()
      ),

  # Analisis Comp. Princip Funcional  =================
  tabPanel(
    "ACP Funcional",

    h3(align = "center", "Análisis de Componentes Principales Funcional"),

    p(
      'Primero realice el Análisis Clúster en la pestaña anterior (',
      actionLink(inputId = "pestania_cluster2", label = "Clusters"),
      ') fijando adecuadamente los parámetros. Luego, selecciona
      que Clúster deseas analizar usando ACP Funcional.'
    ),
    # Número de Clúster a Analizar
    selectInput(
      'n_clus_acpf2',
      label = 'Selecciona Clúster',
      selected = "1",
      choices = 1:4
    ),

    p(
      "En el siguiente gráfico se muestran las
      series de Flujos que componene el clúster,
      así como una listado de las mismas."
```

```
    ),
    #Grafico de Series Vazoes por Cluster
    tabsetPanel(
      #Tabla Vazoes por Cluster
      tabPanel("Listado", br(),
               dataTableOutput(outputId = "tab_vaz_clus2")),
      #Grafico de Vazoes del Cluster
      tabPanel(
        "Gráfico",
        br(),
        br(),
        dygraphOutput(
          outputId = "graf_vaz_clus2",
          width = "98%",
          height = "300px"
        )

      )
    ),
    br(),

    #Resultados ACPF
    h4("Resultados del ACP Funcional"),
    p(
      "A continuación, se muestra un conjunto
      de gráficos resultado de haber relizado el
      ACP Funcional de las series de Flujos del
      Cluster. Es decir, la función media del
      proceso, las funciones propias, y el
      porcentaje que aporta cada componente a la
      variabilidad del proceso."
    ),
    #Grafico ACOF
    tabsetPanel(
      tabPanel("Gráfico Resumen",

               plotOutput(outputId = "graf_acpf1")),
      tabPanel("Gráficos de Presición",

               plotOutput(outputId = "graf_acpf2")),
      tabPanel("BoxPlot Funcional",

               plotOutput(outputId = "graf_acpf3"))
```

```
  )

  ),

# MODELAMIENTO SARIMAX   ============================
tabPanel(
  "SARIMAX",
  h3(align = "center", "Modelamiento SARIMAX"),
  p(
    "En esta sección modelaremos una serie de
    tiempo asociada a Flujos, usando como
    variables regresoras a variables Climáticas y
    las componentes principales del Clúster obtenidas
    a partir del ACP Funcional)."
),


withMathJax(),
p(
  "Se plantea un modelo \\(SARIMAX(p,d,q,P,D,Q) \\),
  que tiene la siguiente forma:"
),
p(
  align = "center",
  "\\( \\varphi_p(L)\\Psi_P(L^s) \\bigtriangledown _s^D V_t =
  \\sum_{k=1}^{w}\\beta_k C_{kt} + \\phi_q(L) \\Phi_Q(L^s)e_t \\)"
),

#Especificaciones
p(
  "Donde \\(V_t\\) es el caudal estimado del clúster en el
  tiempo \\(t\\), \\(C_{kt}\\) son las variables de Clima
  de la estación \\(k\\) en el tiempo \\(t\\)."
),


p(
  "Para ello primero seleccione el Clúster
  que desea analizar. Recuerde haber realizado
  primero el Análisis respectivo en la primera pestaña (",
  actionLink(inputId = "pestania_cluster3", label = "Clusters"),
  ').'
  ),
```

```
#Seleccionar Numero de Cluster
selectInput(
  'n_clus_acpf3',
  label = 'Selecciona Clúster',
  selected = "1",
  choices = 1:4
),
p(
  "A continuación, seleccione la estación
  correspondiente a la serie de tiempo de
  Flujos que desea modelar."
),
#Seleccionar Estacion Vazoe
selectInput(
  'nomb_est_vaz3',
  label = 'Selecciona Estación',
  selected = "1",
  choices = 1:4
),
#Grafico Estacion Seleccionada
dygraphOutput(outputId = "graf_vaz_estacion", width = "98%"),
p(
  "Luego, elije las variables regresoras del
  modelo, en este caso contamos con variables Climáticas."
),
br(),
# Variables Climáticas
h4("Variables Regresoras"),
p(
  "En la siguiente tabla se muestran las
  series climáticas asociadas a las estaciones
  de medición más cercanas a las estaciones
  donde se midieron los Flujos que componen
  el Clúster. Además, podemos encontrar
  la gráfica de dichas series, así como un mapa
  donde podemos observar las estaciones de
  medición de Flujos y sus correspondientes
  estaciones de medición de Clima."
),
p(
  tags$b("Nota:"),
  "Es posible que sea necesario
  desestacionalizar las series de clima,
```

```
    antes de ser usadas en el modelo."
  ),
  checkboxInput("ruidoClimaBox",
                label = "Desestacionalizar Series de Clima.",
                value = FALSE),
  p(
    "Puede incluir en el modelo además, variables
    como la serie de Fujos representante
    del Clúster, así como las series que
    representan a las variables Climáticas:
    Precipitación, Temperatura Máxima,
    Temperatura Mínima, y Humedad (halladas a partir
    de ACP Funcional)."
  ),
  # Eleccion Variables Extras
  checkboxInput("flujoBox",
                label = "Flujo del Clúster",
                value = FALSE),
  checkboxInput("precipBox",
                label = "Precipitación del Clúster",
                value = FALSE),
  checkboxInput("tempMaxBox",
                label = "Temperatura Máxima del Clúster",
                value = FALSE),
  checkboxInput("tempMinBox",
                label = "Temperatura Mínima del Clúster",
                value = FALSE),
  checkboxInput("humedBox",
                label = "Humedad del Clúster",
                value = FALSE),

  #Pestañas
  tabsetPanel(
    tabPanel(
      "Variables Regresoras",
      br(),
      #Tabla Variables Clima del Clúster
      dataTableOutput(outputId = "tab_clim_clus3")
    ),
    tabPanel(
      "Gráfico",
      br(),
      p(
```

```
        "Primero selecciona las variables de
        la tabla anterior para que sean graficadas."
      ),
      #Grafico de las Series Climaticas
      dygraphOutput(
        outputId = "graf_clim_clus3",
        width = "98%",
        height = "400px"
      )

      ),
    tabPanel("Mapa",
             #Mapa de estaciones de Clima
             leafletOutput(outputId = "map_clim_clus3"))

  ),
  br(),
  p(
    "Nota: Si no selecciona ninguna
    de las variables de la tabla anterior, por
    defecto se consideran todas las variable climáticas."
  ),
  br(),

  #Parámetros del Modelo
  h4("Selección del Parámetros"),
  p(
    "A continuación puede elegir los parámetros
    \\( (p,d,q,P,D,Q)\\) del modelo (asociados
    a los retardos y diferencias)."
  ),

  fluidRow(
    column(
      4,
      selectInput(
        inputId = "par_p",
        label = "p",
        choices = 0:12,
        selected = 6
      ),
      selectInput(
        inputId = "par_P",
```

```
          label = "P",
          choices = 0:12,
          selected = 1
        )
      ),
      column(
        4,
        selectInput(
          inputId = "par_d",
          label = "d",
          choices = 0:3,
          selected = 0
        ),
        selectInput(
          inputId = "par_D",
          label = "D",
          choices = 0:3,
          selected = 1
        )
      ),
      column(
        4,
        selectInput(
          inputId = "par_q",
          label = "q",
          choices = 0:8,
          selected = 4
        ),
        selectInput(
          inputId = "par_Q",
          label = "Q",
          choices = 0:8,
          selected = 0
        )
      )

    ),
    br(),
    p(
      align = "center",
      actionButton(
        inputId = "boton_modelo",
        label = "Ejecutar Análisis",
```

```
      icon = icon("cog", lib = "glyphicon")
    )
  ),

  hr(),
  h3("Resultados", align = "center"),
  h4(textOutput(outputId = "estacion_modelada")),

  tabsetPanel(
    #Coeficientes Estimados
    tabPanel(
      "Coeficientes",
      br(),
      p(
        "En esta sección presentamos un
        resumen general del modelo estimado
        a partir de los parámetros antes fijados."
      ),
      verbatimTextOutput("coeficientes")
      ),
    #Residuos
    tabPanel(
      "Residuos",
      br(),
      p(
        "A continuación podemos ver el gráfico
        de los residuos, su distribución, así como
        la función de autocorrelación de los mismos."
      ),
      plotOutput("resid_graf")
      ),
    #Prediccion
    tabPanel("Predicción")



    ),

  br()


    )
```

```
    )
    )
```

A.3. Ejecución de Tareas (server.R)

En el "server"se ejecutan todas las funciones y calculos que hacen parte del análisis, para posteriormente mostrar los resultados del mismo en la interfáz de usuario.

```
# ================================================================
# !!!!!!!!!!!!!!!!!!!!!!     SERVER      !!!!!!!!!!!!!!!!!!!!!!!
# ================================================================
function(input, output,session) {

  #Analisis Cluster  ------------------------------
  # source("Code/Clusters/MapaEstaciones.R",local = TRUE)
  source("Code/Clusters/Analisis.R",local = TRUE)
  source("Code/Clusters/Mapa.R",local = TRUE)
  source("Code/Clusters/Tabla.R",local = TRUE)
  source("Code/Clusters/Grafico.R",local = TRUE)
  #Link a panel Modelamiento
  observeEvent(input$pestania_acpf, {
    updateNavbarPage(session, "tesis", "ACP Funcional")
  })


  # Analisis Componentes Principales --------------
  observe({
    updateSelectInput(session,inputId = "n_clus_acpf2",
                      choices = 1:as.numeric(input$vaz_clus_k) )
  })
  source("Code/ACP Funcional/1_TablaVaz.R",local = TRUE)
  source("Code/ACP Funcional/1_GraficoVaz.R",local = TRUE)
  source("Code/ACP Funcional/2_ACP Funcional.R",local = TRUE)
  source("Code/ACP Funcional/2_ACPF Graficos.R",local = TRUE)
  observeEvent(input$pestania_cluster2, {
    updateNavbarPage(session, "tesis", "Clusters")
  })



  # Modelamiento SARIMAX  -------------------------
```

```
  observe({
    updateSelectInput(session,inputId = "n_clus_acpf3",
                      choices = 1:as.numeric(input$vaz_clus_k) )
  })
  source("Code/SARIMAX/0_Datos Clima.R",local = TRUE)
  source("Code/SARIMAX/1_Lista Vazoes.R",local = TRUE)
  source("Code/SARIMAX/1_Grafico Vazoe.R",local = TRUE)

  source("Code/ACP Funcional/Clima/ACP Clima.R",local = TRUE)
  source("Code/SARIMAX/2_Datos Reactivos.R",local = TRUE)
  source("Code/SARIMAX/2_Tabla Regresoras.R",local = TRUE)
  source("Code/SARIMAX/2_Grafico Regresoras.R",local = TRUE)

  source("Code/SARIMAX/Extras/TrainTest.R",local = TRUE)
  source("Code/SARIMAX/3_Modelo.R",local = TRUE)
  source("Code/SARIMAX/3_Resultados.R",local = TRUE)
  observeEvent(input$pestania_cluster3, {
    updateNavbarPage(session, "tesis", "Clusters")
  })
  #Subtitutlo Resultados
  output$estacion_modelada = renderText({
    paste("Modelamiento Estación:",input$nomb_est_vaz3)
  })
}
```

Apéndice B

Apendice 2

Now we show all the code chunks:

Apéndice C

Apendice 3

Now we show all the code chunks:

Bibliografía

[Caiado et al., 2006] Caiado, J., Crato, N., and Peña, D. (2006). A periodogram-based metric for time series classification. *Computational Statistics & Data Analysis*, 50(10):2668–2684.

[Chouakria and Nagabhushan, 2007] Chouakria, A. D. and Nagabhushan, P. N. (2007). Adaptive dissimilarity index for measuring time series proximity. *Advances in Data Analysis and Classification*, 1(1):5–21.

[Cleveland et al., 1990] Cleveland, R. B., Cleveland, W. S., McRae, J. E., and Terpenning, I. (1990). Stl: A seasonal-trend decomposition. *Journal of Official Statistics*, 6(1):3–73.

[Diggle and Al Wasel, 1997] Diggle, P. J. and Al Wasel, I. (1997). Spectral analysis of replicated biomedical time series. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 46(1):31–71.

[Diggle and Fisher, 1991] Diggle, P. J. and Fisher, N. I. (1991). Nonparametric comparison of cumulative periodograms. *Applied Statistics*, pages 423–434.

[Duda et al., 2001] Duda, R. O., Hart, P. E., Stork, D. G., et al. (2001). Pattern classification. *International Journal of Computational Intelligence and Applications*, 1:335–339.

[Fraley and Raftery, 1998] Fraley, C. and Raftery, A. E. (1998). How many clusters? which clustering method? answers via model-based cluster analysis. *The computer journal*, 41(8):578–588.

[Fu, 2011] Fu, T.-c. (2011). A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164–181.

[Galeano and Peña, 2000] Galeano, P. and Peña, D. P. (2000). Multivariate analysis in vector time series. *Resenhas do Instituto de Matemática e Estatística da Universidade de São Paulo*, 4(4):383–403.

[Johnson and Wichern, 2004] Johnson, R. A. and Wichern, D. W. (2004). Multivariate analysis. *Encyclopedia of Statistical Sciences*, 8.

[Kaufman and Rousseeuw, 1986] Kaufman, L. and Rousseeuw, P. J. (1986). Clustering large data sets. In *Pattern Recognition in Practice, Volume II*, pages 425–437. Elsevier.

[Kerr and Churchill, 2001] Kerr, M. K. and Churchill, G. A. (2001). Bootstrapping cluster analysis: assessing the reliability of conclusions from microarray experiments. *Proceedings of the National Academy of Sciences*, 98(16):8961–8965.

[Liao, 2005] Liao, T. W. (2005). Clustering of time series data survey. *Pattern recognition*, 38(11):1857–1874.

[Maharaj, 2000] Maharaj, E. A. (2000). Cluster of time series. *Journal of Classification*, 17(2):297–314.

[Piccolo, 1990] Piccolo, D. (1990). A distance measure for classifying arima models. *Journal of Time Series Analysis*, 11(2):153–164.

[Ross, 2006] Ross, S. (2006). Simulation.

[Rousseeuw and Kaufman, 1990] Rousseeuw, P. J. and Kaufman, L. (1990). Finding groups in data. *Series in Probability & Mathematical Statistics 1990-34 (1)*, pages 111–112.

[RStudio, Inc, 2013] RStudio, Inc (2013). *Easy web applications in R*. URL: `http://www.rstudio.com/shiny/`.

[Salvador and Chan, 2004] Salvador, S. and Chan, P. (2004). Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*, pages 576–584. IEEE.

[Struzik and Siebes, 1999] Struzik, Z. R. and Siebes, A. (1999). The haar wavelet transform in the time series similarity paradigm. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 12–22. Springer.