Series Temporales y Minería de flujos de datos

Ejercicios Guiados

Parte I: Series Temporales

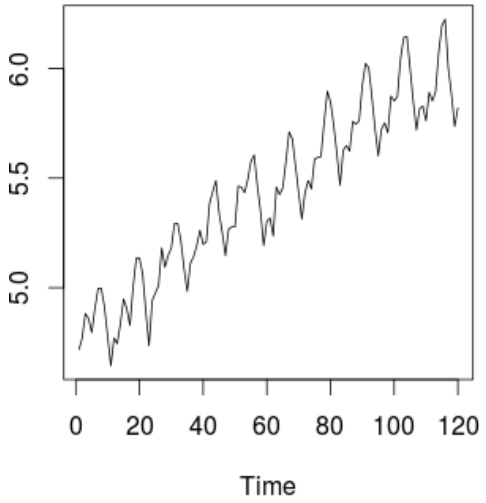
Manuel Pegalajar Cuéllar (manupc@decsai.ugr.es)
Dpto. Ciencias de la Computación e I.A., Universidad de Granada

Índice

- Introducción de la serie temporal a estudiar
- Análisis de la serie
- Ejercicio propuesto y condiciones de entrega
- Evaluación

Ejercicios propuestos

- Se propone estudiar la siguiente serie temporal:
  - Una compañía aérea nos proporciona el número de pasajeros de avión (en miles), anotados mensualmente, a lo largo de 11 años (concretamente, entre 1949 y 1959).
  - La compañía nos pide elaborar un modelo predictivo que le permita conocer el número de pasajeros estimado para todos los meses de 1960.



Manuel Pegalajar Cuéllar (manupc@decsai.ugr.es)
Dpto. Ciencias de la Computación e I.A., Universidad de Granada

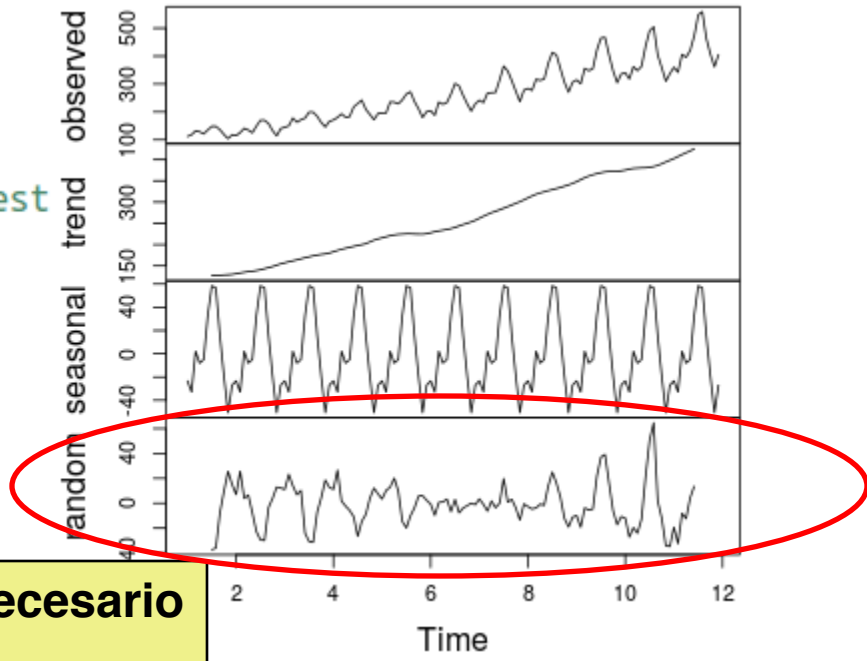
Índice

- Introducción de la serie temporal a estudiar
- **Análisis de la serie**
- Ejercicio propuesto y condiciones de entrega
- Evaluación

Análisis de la serie

- Lo primero que hacemos es cargar la serie y mostrar su descomposición, para hacernos una idea visual del trabajo.
- Como son datos anuales, inicialmente se podría asumir algún tipo de estacionalidad anual.

```
5  NPred= 12; # Valores a predecir
6  NTest= 12; # Valores que vamos a dejar para test
7
8  serie<-scan("pasajeros_1949_1959.dat")
9  serie.ts<- ts(serie, frequency = 12)
10 plot(decompose(serie.ts))
```



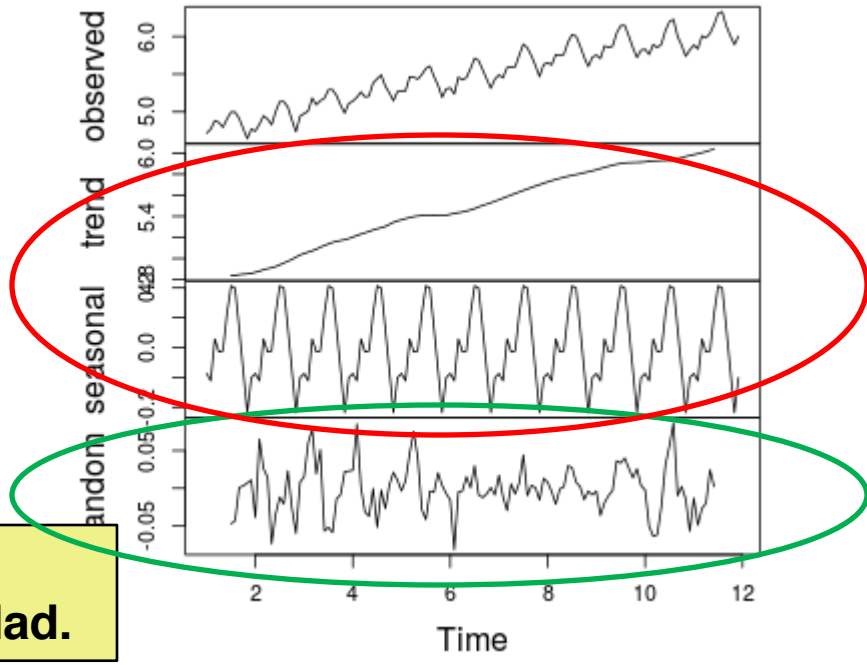
Hay variación en la varianza. Será necesario reducirla (transformación log)

Manuel Pegalajar Cuéllar (manupc@decsai.ugr.es)
Dpto. Ciencias de la Computación e I.A., Universidad de Granada

Análisis de la serie

- Lo primero que hacemos es cargar la serie y mostrar su descomposición, para hacernos una idea visual del trabajo.
- Como son datos anuales, inicialmente se podría asumir algún tipo de estacionalidad anual.

```
28  serie.ts<- log(serie.ts);
29  serie.log<- log(serie);
30
31  # Visualizamos de nuevo
32  plot(decompose(serie.ts))
```



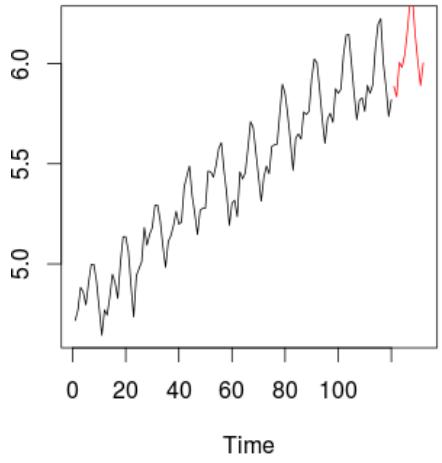
Solucionado el problema de la varianza.
Vemos que hay tendencia y estacionalidad.

Manuel Pegalajar Cuéllar (manupc@decsai.ugr.es)
Dpto. Ciencias de la Computación e I.A., Universidad de Granada

Análisis de la serie

- El siguiente paso será dividir los datos para ajuste y test. Cogeremos, por ejemplo, los 12 últimos valores para el test dado que también necesitaremos predecir 12 valores de la serie.

```
39 serieTr<- serie.log[1:(length(serie.log)-NTest)];
40 tiempoTr<- 1:length(serieTr)
41 serieTs<- serie.log[(length(serie.log)-NTest+1):length(serie)];
42 tiempoTs<- (tiempoTr[length(tiempoTr)]+1):(tiempoTr[length(tiempoTr)]+NTest);
43
44 plot.ts(serieTr, xlim=c(1, tiempoTs[length(tiempoTs)]))
45 lines(tiempoTs, serieTs, col="red")
```

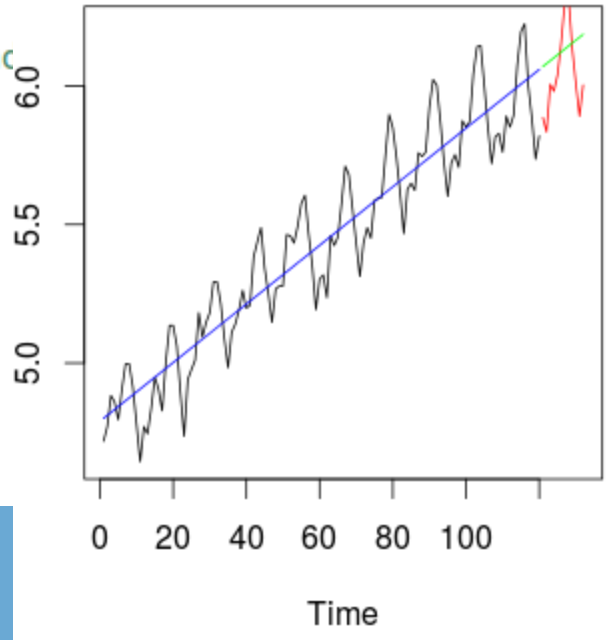


Manuel Pegalajar Cuéllar (manupc@decsai.ugr.es)
Dpto. Ciencias de la Computación e I.A., Universidad de Granada

Análisis de la serie

- A continuación, modelaremos la tendencia. Asumiremos un comportamiento lineal en este caso como hipótesis.

```
54 parametros.H1 <- lm (serieTr ~ tiempoTr) # Ajustamos modelo
55
56 # Calculamos la estimación de la tendencia
57 TendEstimadaTr.H1<-parametros.H1$coefficients[1]+tiempoTr*parametros.H1$coefficients[2]
58 TendEstimadaTs.H1<-parametros.H1$coefficients[1]+tiempoTs*parametros.H1$coefficients[2]
59
60
61 # Mostramos en la misma figura la serie y la tendencia estimac
62 plot.ts(serieTr, xlim=c(1, tiempoTs[length(tiempoTs)]))
63 lines(tiempoTr, TendEstimadaTr.H1, col="blue")
64 lines(tiempoTs, serieTs, col="red")
65 lines(tiempoTs, TendEstimadaTs.H1, col="green")
```



Manuel Pegalajar Cuéllar (manupc@decsai.ugr.es)
Dpto. Ciencias de la Computación e I.A., Universidad de Granada

Análisis de la serie

- Comprobamos que la hipótesis de tendencia lineal es válida. Para ello se aplica un T-test, asumiendo normalidad en los datos (u otro test no paramétrico si los datos no son normales), que compare los residuos del ajuste con los errores del modelo en test.
- En nuestro caso, todos los tests de normalidad dan p-value>0.05. Asumimos normalidad.
- También el T-test da un p-value>0.05. No existen diferencias significativas en los datos.

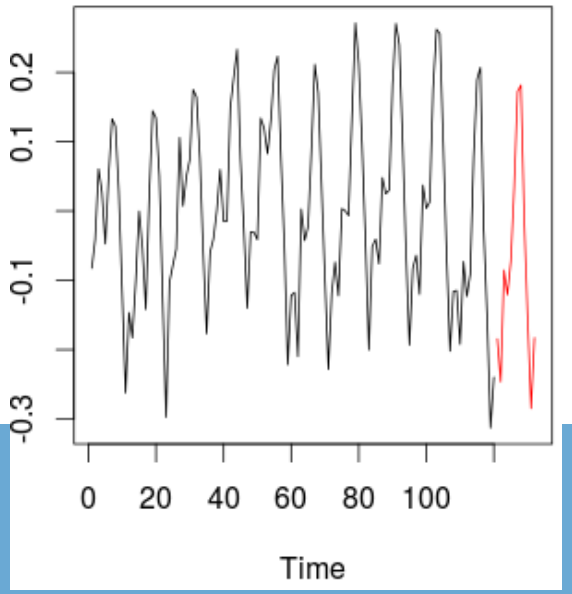
```
# Test de normalidad de Jarque Bera
JB<- jarque.bera.test(parametros.H1$residuals);
JB<- jarque.bera.test( (TendEstimadaTs.H1-serieTs) );

# Test de Student
TT<- t.test(c(parametros.H1$residuals, TendEstimadaTs.H1-serieTs));
```

Análisis de la serie

- Por tanto, al no encontrar diferencias significativas en errores de ajuste y test, asumimos que la hipótesis de modelado lineal es factible y la aceptamos.
- Eliminamos tendencia en la serie, como paso siguiente.

```
88  # Eliminamos la tendencia
89  serieTr.SinTend.H1<- serieTr-TendEstimadaTr.H1;
90  serieTs.SinTend.H1<- serieTs-TendEstimadaTs.H1;
91  plot.ts(serieTr.SinTend.H1, xlim=c(1, tiempoTs[length(tiempoTs)]))
92  lines(tiempoTs, serieTs.SinTend.H1, col="red")
```



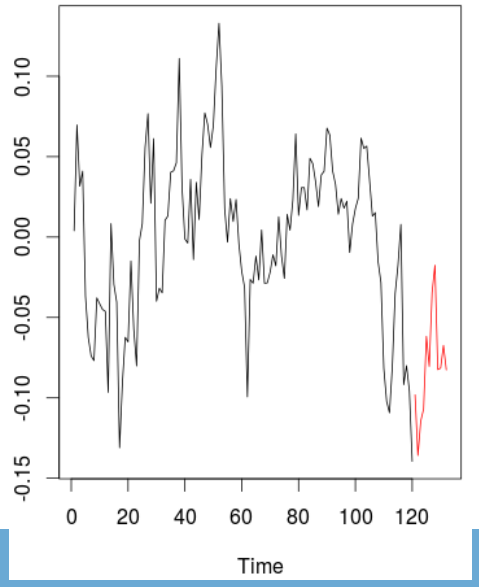
Manuel Pegalajar Cuéllar (manupc@decsai.ugr.es)
Dpto. Ciencias de la Computación e I.A., Universidad de Granada

Análisis de la serie

- El siguiente paso es eliminar la estacionalidad. Al comienzo, asumimos una estacionalidad anual (12 meses/valores de la serie) al crear el objeto ts (diapositiva 5):
  - serie.ts<- ts(serie, frequency=12)
- Para eliminar la estacionalidad, podemos hacer uso de las salidas de la función “decompose”:

```
# Calculamos y eliminamos la estacionalidad
k<- 12; # Asumimos periodo de estacionalidad k= 12
estacionalidad.H1<- decompose(serie.ts)$seasonal[1:k];

#Eliminamos estacionalidad para el modelo
aux<-rep(estacionalidad.H1, length(serieTr)/length(estacionalidad.H1));
serieTr.SinTendEst.H1<- serieTr.SinTend.H1-aux;
serieTs.SinTendEst.H1<- serieTs.SinTend.H1-estacionalidad.H1;
plot.ts(serieTr.SinTendEst.H1, xlim=c(1, tiempoTs[length(tiempoTs)]))
lines(tiempoTs, serieTs.SinTendEst.H1, col="red")
```



Manuel Pegalajar Cuéllar (manupc@decsai.ugr.es)
Dpto. Ciencias de la Computación e I.A., Universidad de Granada

Análisis de la serie

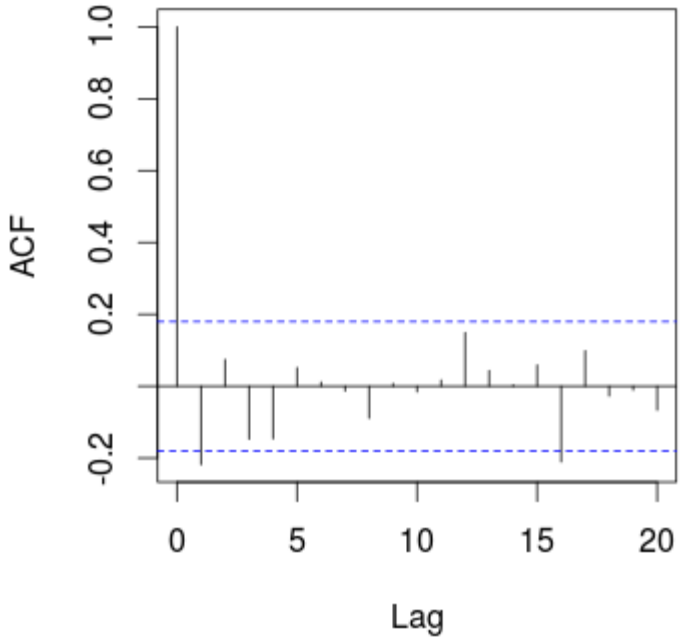
- Con la serie sin tendencia ni estacionalidad, debemos comprobar si es estacionaria antes de hipotetizar modelos de predicción. Usamos el Test de Dickey-Fuller aumentado.
- Diferenciamos la serie tras aplicar el test, que falla.
- Volvemos a aplicar el test a la serie diferenciada (esta vez sí se pasa el test).

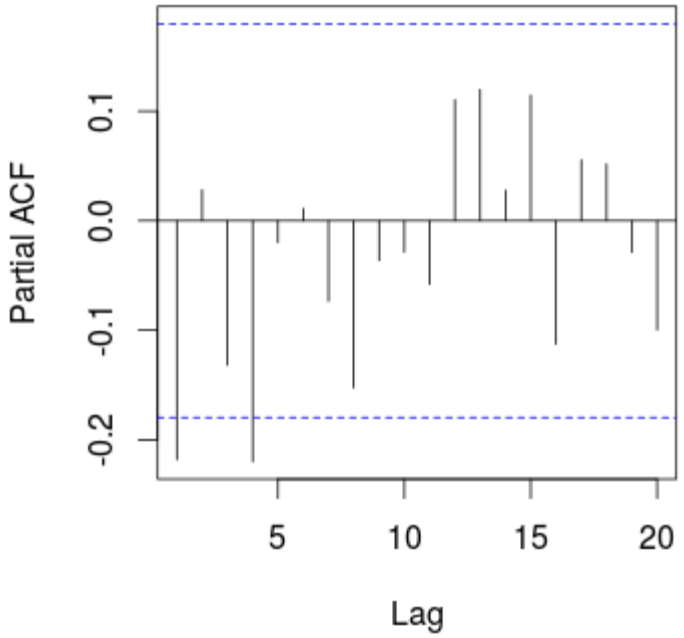
```
132 # Comprobamos el test de Dickey-Fuller aumentado para estacionaridad
133 adftest.H1<- adf.test(serieTr.SinTendEst.H1);
134
135 # Como no se supera (valor>0.05), diferenciamos la serie
136 serieTr.SinTendEstDiff.H1<- diff(serieTr.SinTendEst.H1);
137 serieTs.SinTendEstDiff.H1<- diff(serieTs.SinTendEst.H1);
138
139 # Volvemos a aplicar el test
140 adftest.H1<- adf.test(serieTr.SinTendEstDiff.H1);
```

Análisis de la serie

- Con una serie ya estacionaria, vamos a visualizar ACF y PACF para poder dar hipótesis de modelos.

```
147  acf(serieTr.SinTendEstDiff.H1) # Mostramos ACF
148  pacf(serieTr.SinTendEstDiff.H1) # Mostramos PACF
```

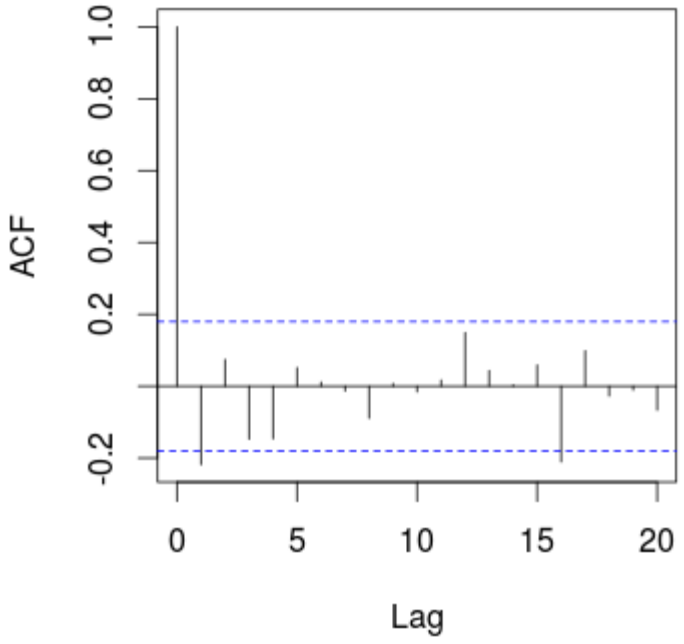


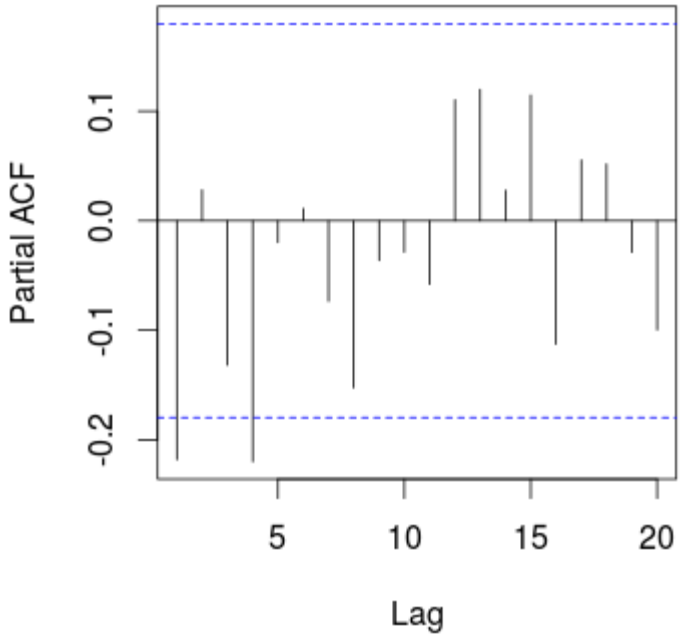


Manuel Pegalajar Cuéllar (manupc@decsai.ugr.es)
Dpto. Ciencias de la Computación e I.A., Universidad de Granada

Análisis de la serie

- Visualizando ACF y PACF, podríamos estar ante un modelo AR(4), un modelo MA(1). La intensidad de los picos es muy baja, por lo que podríamos estar cerca, también, de un error de distribución normal con media cero.





Análisis de la serie

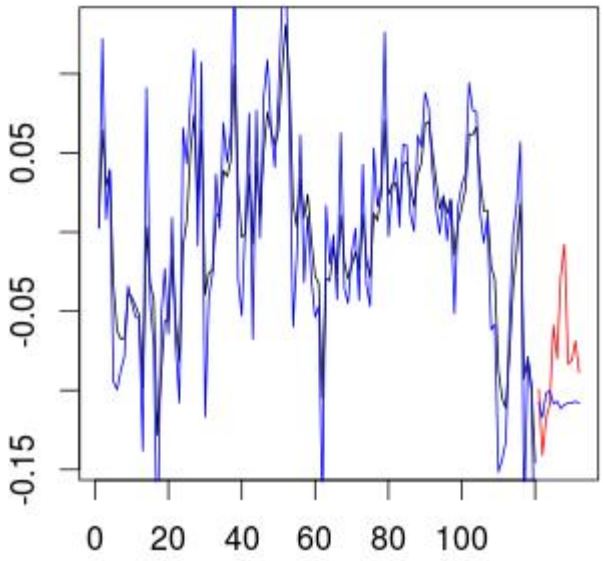
- Vamos a asumir un modelo AR(4). Como hemos diferenciado 1 instante de tiempo, podríamos incluir la diferencia dentro del modelo ajustando un ARIMA(4, 1, 0).
- También calculamos error de ajuste y test (para comparar con otros modelos, si deseamos probar con más).

```
170 # Ajustamos el modelo
171 modelo.H1<- arima(serieTr.SinTendEst.H1, order=c(4, 1, 0))
172 valoresAjustados.H1<- serieTr.SinTendEst.H1+modelo.H1$residuals;
173
174 # Calculamos las predicciones
175 Predicciones.H1<- predict(modelo.H1, n.ahead = NPred);
176 valoresPredichos.H1<- Predicciones.H1$pred; # Cogemos las predicciones
177
178 # Calculamos el error cuadrático acumulado del ajuste, en ajuste y en test
179 errorTr.H1<- sum((modelo.H1$residuals)^2);
180 errorTs.H1<- sum((valoresPredichos.H1-serieTs.SinTendEst.H1)^2);
```

Análisis de la serie

- Ilustramos los resultados a continuación:

```
187 # Mostramos las gráficas del ajuste y predicción en test
188 plot.ts(serieTr.SinTendEst.H1, xlim=c(1, tiempoTs[length(tiempoTs)]))
189 lines(valoresAjustados.H1, col="blue")
190 lines(tiempoTs, serieTs.SinTendEst.H1, col="red")
191 lines(tiempoTs, valoresPredichos.H1, col="blue")
```



Manuel Pegalajar Cuéllar (manupc@decsai.ugr.es)
Dpto. Ciencias de la Computación e I.A., Universidad de Granada

Análisis de la serie

- Finalmente, validamos el modelo:
  - Test de Box-Pierce para aleatoriedad de residuos (lo pasa).
  - Tests de Jarque Bera y Shapiro-Wilk para normalidad de residuos (los pasa)
  - Mostramos histograma y función de densidad para confirmación gráfica.

```
200 # Tests para la selección del modelo y su validación
201 boxtestM1<- Box.test(modelo.H1$residuals) # Test de aleatoriedad de Box-Pierce
202
203 # Test de normalidad de Jarque Bera
204 JB.H1<- jarque.bera.test(modelo.H1$residuals);
205
206 # Test de normalidad de Shapiro-Wilk
207 SW.H1<- shapiro.test(modelo.H1$residuals);
208
209 hist(modelo.H1$residuals, col="blue", prob=T,ylim=c(0,20),xlim=c(-0.2,0.2))
210 lines(density(modelo.H1$residuals))
```

Análisis de la serie

- Una vez que hemos validado todo el modelo, volvemos a seguir los pasos iniciales, sin dividir la serie en ajuste y test, para hacer la predicción de los meses de 1960.

```
226 serieEntera<- serie.log; # Cogemos toda la serie
227 tiempo<- 1:length(serieEntera)
228 parametros <- lm (serieEntera ~ tiempo  ) # Ajustamos modelo de tendencia
229 TendEstimada<-parametros$coefficients[1]+tiempo*parametros$coefficients[2]
230 serieSinTend<- serieEntera-TendEstimada;
231 aux<-ts(serieEntera, frequency = 12);
232 aux<-decompose(aux)$seasonal;
233 estacionalidad<-as.numeric(aux[1:12]);
234 aux<-rep(estacionalidad, length(serieSinTend)/length(estacionalidad));
235 serieSinTendEst<- serieSinTend-aux;
236 modelo<- arima(serieSinTendEst, order=c(4, 1, 0))
237 valoresAjustados<- serieSinTendEst+modelo$residuals;
238 Predicciones<- predict(modelo, n.ahead = NPred);
239 valoresPredichos<- Predicciones$pred; # Cogemos las predicciones
```

Análisis de la serie

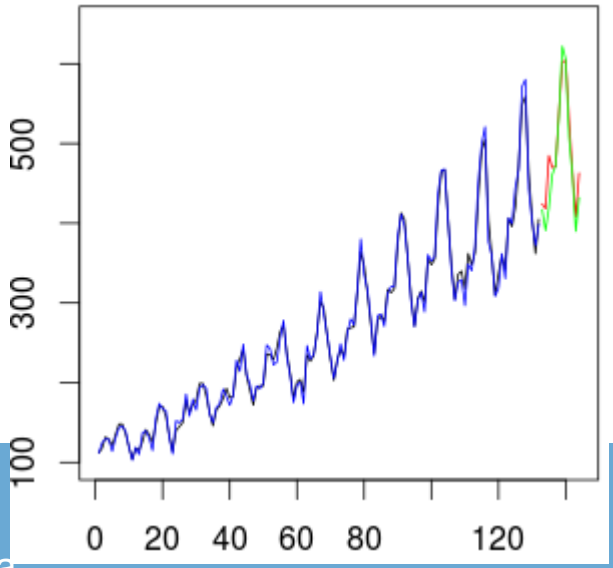
- Por último, deshacemos los cambios realizamos para calcular las predicciones reales.

```
242 # Por último, deshacemos cambios
243 valoresAjustados<- valoresAjustados+aux; # Estacionalidad
244 valoresPredichos<- valoresPredichos+estacionalidad;
245
246 valoresAjustados<- valoresAjustados+TendEstimada; # Tendencia
247 tiempoPred<- (tiempo[length(tiempo)]+(1:NPred));
248 TendEstimadaPred<-parametros$coefficients[1]+tiempoPred*parametros$coefficients[2]
249 valoresPredichos<- valoresPredichos+TendEstimadaPred;
250
251 valoresAjustados<- exp(valoresAjustados); # Transformación log de los datos
252 valoresPredichos<- exp(valoresPredichos);
253
254 plot.ts(serie, xlim=c(1, max(tiempoPred)), ylim=c(100, 650))
255 lines(valoresAjustados, col="blue")
256 lines(valoresPredichos, col="red")
```

Análisis de la serie

- Si tuviésemos disponibles los datos correctos de la predicción, podríamos compararlos con los resultados para validar el modelo.

```
258 # Cargamos los valores reales de predicción y los mostramos
259 predReales<-scan("pasajeros_1960.predict")
260 lines(tiempoPred, predReales, col="green")
261
262 # Calculamos el error de predicción
263 ErrorMedio<- sum( abs(predReales-valoresPredichos) )
```



Manuel Pegalajar Cuéllar (manupc@decsai.ugr.es)
Dpto. Ciencias de la Computación e I.A., Universidad de Granada

Índice

- Introducción de la serie temporal a estudiar
- Análisis de la serie
- **Ejercicio propuesto y condiciones de entrega**
- Evaluación

Ejercicio propuesto y condiciones de entrega

- **Trabajo guiado a realizar:**
  - El alumno deberá implementar el modelo hecho en clase y, dentro del código fuente, comentar cada paso indicando porqué se debe hacer ese paso y qué efectos, parámetros y salida tiene cada comando utilizado.
  - calcular las predicciones de la serie.

Ejercicio propuesto y condiciones de entrega

- **Condiciones de entrega:**
  - El método del alumno deberá entregarse en un fichero cuyo nombre sea “EjercicioGuiado_DNI.R”. Ejemplo: “EjercicioGuiado_65739294.R”.
  - El fichero implementará una función que devuelva los valores predichos para la serie.
  - Las 3 primeras líneas del fichero serán:
    - Línea 1: Nombre, apellidos y DNI del alumno
    - Línea 2: E-mail del alumno
    - Línea 3: Texto “Ejercicio guiado. Curso 20XX-20XX”, sustituyendo XX por el curso académico actual.

Ejercicio propuesto y condiciones de entrega

- El fichero de texto de entrega deberá estar **completamente comentado, describiendo cada paso que se realice**, justificando:
  - 1. Análisis inicial de la serie: Qué se observa visualmente (tendencia o no, estacionalidad o no), justificando el análisis con datos objetivos (procedentes del análisis visual preliminar de la serie y sus componentes).
  - 2. Justificar si hay necesidad de preprocesar los datos iniciales, e indicar qué transformación se realiza y porqué.
  - 3. Justificar, en caso de haber tendencia o estacionalidad, cuál de las dos se debe eliminar antes.
  - 4. En el caso de existir tendencia, justificar qué modelo de tendencia se utiliza para eliminarla (filtros, aproximación funcional, diferenciación).

Ejercicio propuesto y condiciones de entrega

- 5. En el caso de existir estacionalidad, justificar qué modelo se utiliza para eliminarla.
- 6. Explicación del procedimiento seguido para comprobar y conseguir la estacionaridad, en base a los ADF, ACF, PACF.
- 7. Justificar la selección del modelo de predicción.
- 8. Explicar cómo se valida el modelo ajustado, describiendo qué es cada test, para qué se utiliza y qué resultados puede proporcionar.

Ejercicio propuesto y condiciones de entrega

- 9. Describir, en el caso de existir varios modelos de predicción, qué criterio se ha escogido para seleccionar el mejor de ellos (AIC, MSE, etc.), justificando la elección del criterio.
- 10. Describir los pasos necesarios para conseguir la predicción real de los valores de la serie.

- El fichero **deberá remitirse al profesor de prácticas a través de PRADO, en la entrega para ejercicio guiado de Series Temporales antes de las 23.59h del día 23 de Marzo**:

Manuel Pegalajar Cuéllar (manupc@decsai.ugr.es)
Dpto. Ciencias de la Computación e I.A., Universidad de Granada

Índice

- Introducción de la serie temporal a estudiar
- Análisis de la serie
- Ejercicio propuesto y condiciones de entrega
- **Evaluación**

Evaluación

- El ejercicio se evaluará entre 0 y 10. **Contribución a la calificación final: 1 punto.**
- **Cada uno de los ítems comentados anteriormente se valora entre 0 y 1 punto.**
- Por cada ítem, se valorará la claridad e idoneidad de la justificación de las decisiones tomadas en cada parte:
  - 0=Mala justificación, mala idoneidad, mal código
  - 10= Buena justificación, buena idoneidad, buen código.
- Las puntuaciones intermedias entre 0 y 10 se calcularán gradualmente, considerando de mayor a menor importancia: Justificación, Idoneidad, código.