Técnicas Avanzadas de Análisis de Datos

Actividad 1: ¿Es Ben Geen culpable de asesinato o no?

Máster en Gestión y Análisis de Grandes Volúmenes de Datos:

Big Data

23/02/2022

Cristina Varas Menadas

Índice

1. Introducción

En febrero de 2004 el enfermero Benjamin Geen del Horton General Hospital (Banbury, Oxford, UK) fue arrestado, acusado de dañar de manera intencionada a 18 pacientes que se encontraban en la Unidad de Emergencias durante los meses de diciembre de 2003 a febrero de 2004 enviándolos a cuidados intensivos. El enfermero fue sentenciado a cadena perpetua y en la actualidad sigue ingresado en prisión cumpliendo condena.

En el Dataset utilizado se presentan los datos de Paro Cardiaco, Paro Respiratorio y Paro Hipoglucémico, así como el número de admisiones a los servicios de Urgencias desde el mes de noviembre de 1999 hasta el mes de mayo de 2011 de dicho hospital.

En esta actividad se han analizado los datos proporcionados para buscar evidencias de su inocencia o culpabilidad utilizando técnicas estadísticas y el algoritmo ARIMA.

2. Materiales y métodos

El material de datos utilizado consiste en un csv modificado que añade al original tres nuevas columnas:

- "mes_año": contiene la concatenación del mes y del año para después trabajar más fácilmente con el tipo de datos de python "datetime".
- "n_transferidos": contiene el número total de pacientes transferidos de urgencias a UCI (suma de causas por paro respiratorio, paro cardíaco y paro hipoglucémico).
- "ratio": el ratio de trasladados a UCI por número de ingresos en urgencias.

En la actividad se han utilizado los siguientes métodos:

- **Diagrama de cajas**

  Se trata de un gráfico utilizado para representar una variable cuantitativa, permitiendo visualizar, a través de los cuartiles, cómo es la distribución, su grado de asimetría, los valores extremos, la posición de la mediana, etc. Se compone de:

  - Un rectángulo (caja) delimitado por el primer y tercer cuartil (Q1 y Q3). Dentro de la caja una línea indica dónde se encuentra la mediana (segundo cuartil Q2).
  - Dos brazos, uno que empieza en el primer cuartil y acaba en el mínimo, y otro que empieza en el tercer cuartil y acaba en el máximo.
  - Los datos atípicos (o valores extremos) son los valores distintos que no cumplen ciertos requisitos de heterogeneidad de los datos.

Utilizando este método, si los valores de la suma de ingresos (que está en torno a 4) son valores atípicos en comparación al resto de valores de la distribución, se podría considerar culpable al enfermero.

- **Prueba de contraste de hipótesis**

  Una hipótesis estadística es una asunción relativa a una o varias poblaciones, que puede ser cierta o no. Las hipótesis estadísticas se pueden contrastar con la información extraída de las muestras y tanto si se aceptan como si se rechazan se puede cometer un error.

  La hipótesis formulada con intención de rechazarla se llama hipótesis nula y se representa por H_0. Rechazar H_0 implica aceptar una hipótesis alternativa (H_1).

  Lo que se busca con este método estadístico es comprobar si realmente el número de pacientes que ingresan de urgencias a UCI se distribuye uniformemente a lo largo del tiempo. Si se distribuye de manera uniforme, se puede calcular después en qué posición se encuentran los valores atípicos para calcular la probabilidad de ocurrencia de esos valores atípicos en el intervalo elegido.

  Por ejemplo: un valor es atípico si está por encima del percentil 99 o si está por debajo del percentil 1. Es decir, en este caso de ejemplo, el 98% de la distribución son valores esperables, y el 2% restante son valores atípicos.

- **Entrenamiento de un modelo de predicción de regresión lineal: ARIMA.**

  La regresión lineal consiste en generar un modelo de regresión (ecuación de una recta) que permita explicar la relación lineal que existe entre dos variables.

  Aplicando un modelo de autoregresión (ya que nuestro problema es de series temporales), es posible predecir el próximo valor esperado en el tiempo. Aplicando un modelo entrenado sobre los datos esperados correctos , si sale que la diferencia que hay entre el valor observado y el valor esperado predicho por el modelo es muy grande, implica que el valor es atípico.

  Por lo tanto, en concreto ARIMA, es un modelo de regresión de series temporales. Permite descomponer las series temporales en tendencias, estacionalidad y ruido. Es decir, se puede hacer un análisis de series temporales y, como se tienen datos mensuales, se puede decir que la estacionalidad es de 12 meses y por lo tanto que, por ejemplo, el mes 1 del siguiente año debería parecerse al mes 1 del año anterior. Y dando un periodo de 12 meses de estacionalidad, se puede concluir que cada 12 observaciones se están repitiendo los mismos datos por estar en el periodo de un año.

3. Resultados

3.1. Diagrama de cajas

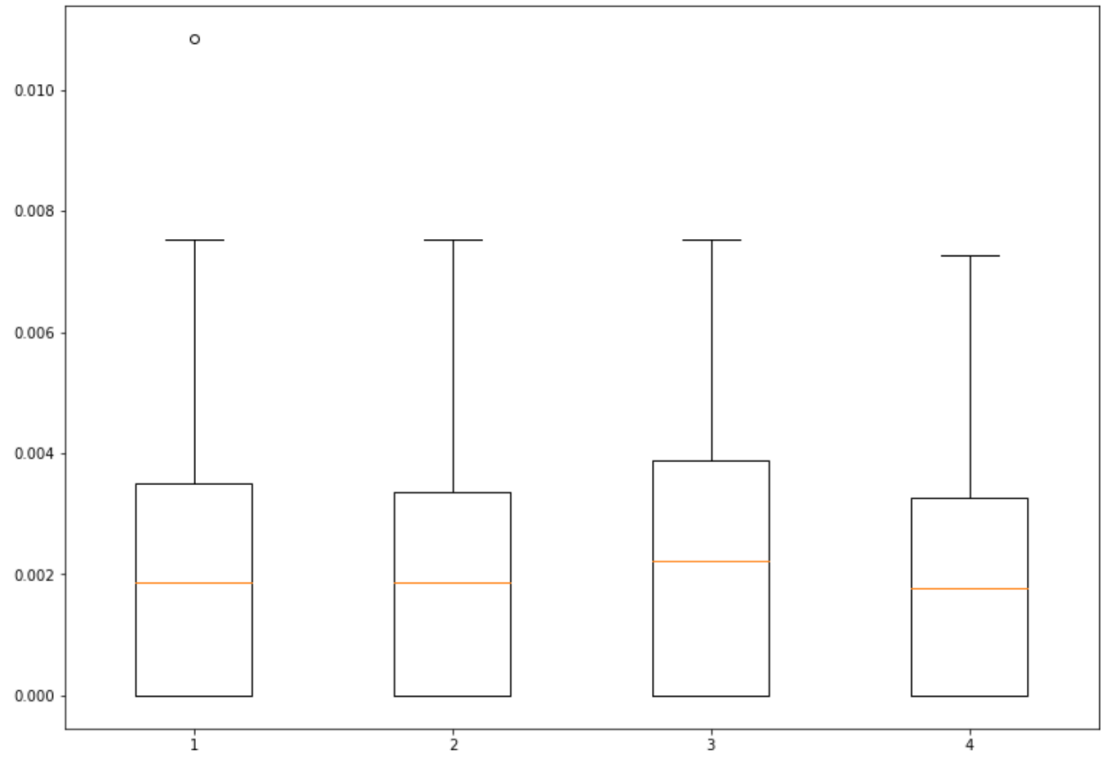
En este apartado se han analizado los datos de la columna “ratio” en diferentes periodos utilizando diagrama de cajas.

En la siguiente figura podemos ver de izquierda a derecha los diagramas de cajas correspondientes a los siguientes datos:

- Ratio de todo el dataset (con los meses en los que trabajó Ben incluidos)
- Ratio del dataset menos los meses en los que trabajó Ben.
- Ratio de los meses ANTES de que trabajara Ben (hasta noviembre de 2003)
- Ratio de los meses DESPUÉS de que trabajara Ben (desde marzo de 2004)

```
fig = plt.figure(figsize=(10, 7))

data = [datos_ratio_conBen, datos_ratio_sinBen, datos_ratio_hasta_noviembre2003, datos_ratio_desde_marzo2014]
ax = fig.add_axes([0, 0, 1, 1])
plt.boxplot(data)
plt.show()
```



Podemos observar claramente que en el ratio que contempla los meses en los que trabajó Ben, hay valores atípicos representados por el punto superior de la gráfica. En cambio, en el resto de diagramas de cajas, no hay valores atípicos, la mediana es aproximadamente la misma y el cuerpo y los brazos del diagrama son muy parecidos.

Siguiendo este análisis de la estadística clásica, en la que se demuestra que los meses que trabajó influyeron en la aparición de valores atípicos, se puede considerar que Ben es culpable.

Pero estos datos pueden no ser suficientes para culpabilizar a una persona. Es necesario analizar la uniformidad de la distribución de los datos y la estacionalidad para comprobar que este suceso no se repite, por ejemplo, durante los primeros meses del año.

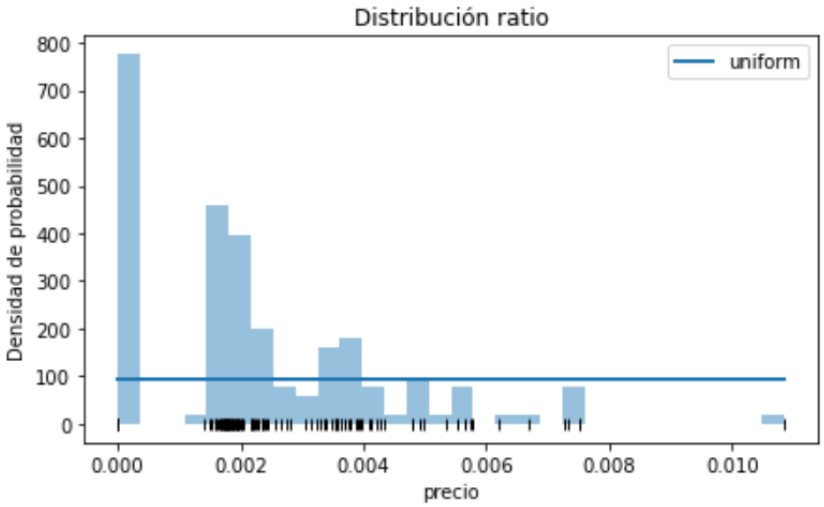
3.2. Comprobación de distribución uniforme mediante contraste de Hipótesis

3.2.1. Análisis de la distribución uniforme

A continuación se ha desarrollado un análisis sobre la uniformidad de la distribución de los datos en el tiempo.

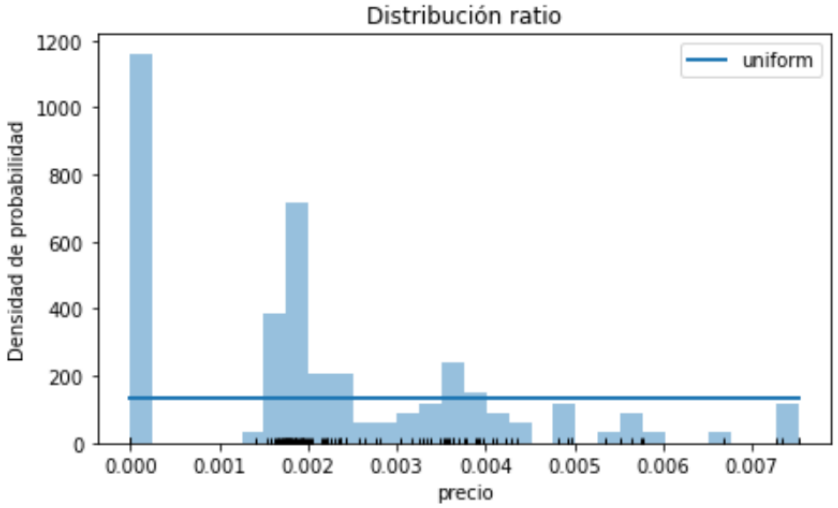
En esta primera imágen, en la que se han analizado todos los meses del dataset (incluidos los meses de trabajo de Ben), se puede observar claramente que los datos no siguen una distribución uniforme. Lo más normal es encontrar ratios bajos, y la densidad de probabilidad del ratio va disminuyendo conforme aumenta el ratio. De nuevo se observa el caso extremo más allá del 0,010 de ratio, que coincide con uno de los meses de trabajo de Ben.

```
---------------------
Resultados del ajuste
---------------------
Distribución:   uniform
Dominio:        [0.0, 1.0]
Parámetros:     {'loc': 0.0, 'scale': 0.01084991}
Log likelihood: 628.7801906605716
AIC:            -1253.5603813211433
BIC:            -1247.691433454882
```



Se ha querido hacer esta misma prueba con todos los meses en los que Ben no trabajó, obtenido los siguientes resultados:

```
--------------------
Resultados del ajuste
--------------------
Distribución:   uniform
Dominio:        [0.0, 1.0]
Parámetros:     {'loc': 0.0, 'scale': 0.007518797}
Log likelihood: 655.3067830477148
AIC:            -1306.6135660954296
BIC:            -1300.8178864955278
```



De nuevo los datos no siguen una distribución uniforme y vemos que la densidad de probabilidad de que no ocurra ningún traslado a UCI aumenta y desaparece la posibilidad de tener un ratio de traslado en torno al 0,010.

Si consideramos que no hay cambio en la uniformidad de los datos, Ben no es claro culpable, pues no interfiere en la uniformidad de los datos. Sigue siendo muy parecido.

Sin embargo, viendo un pequeño detalle en la frecuencia del ratio de traslados a urgencias, Ben tiene una ligera influencia sobre el aumento del ratio de traslados de urgencias a UCI. Vemos que la probabilidad de que no se produzca ningún traslado de urgencias a la UCI baja de casi 1200 puntos a algo menos de 800 puntos cuando se tienen en cuenta los meses en los que trabaja. Por lo que siguiendo este apunte, se podría considerar a Ben culpable.

3.2.2. Análisis de normalidad

Mediante el análisis de normalidad se pretende analizar si los datos disponibles podrían proceder de una población con una distribución normal. Mediante el test de normalidad se puede analizar si se puede rechazar la hipótesis nula de que hay normalidad en los datos.

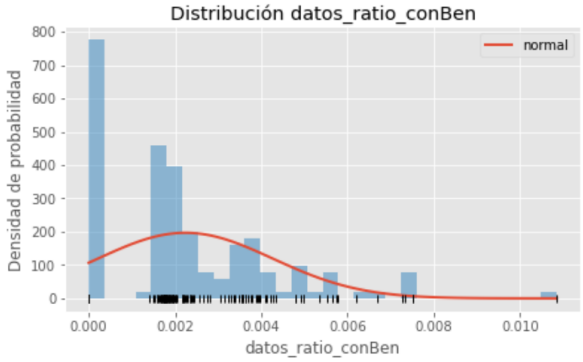
Este análisis de datos se ha hecho sobre dos bases de datos:

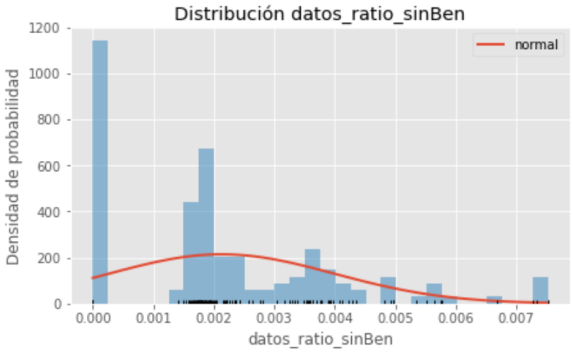
- Datos teniendo en cuenta los meses en los que trabajó Ben
- Datos teniendo NO teniendo en cuenta los meses en los que trabajó Ben

Además se han utilizado varias técnicas de análisis de normalidad:

- **Métodos gráficos**

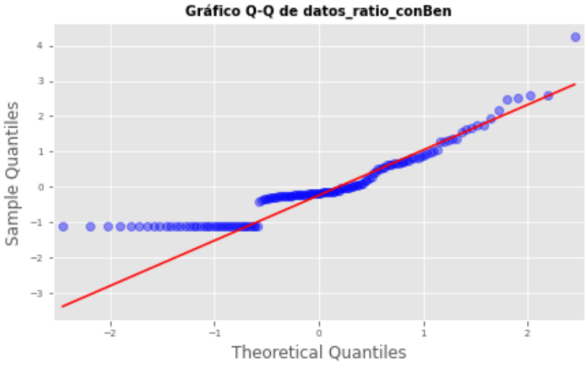
  Histograma + curva normal teórica

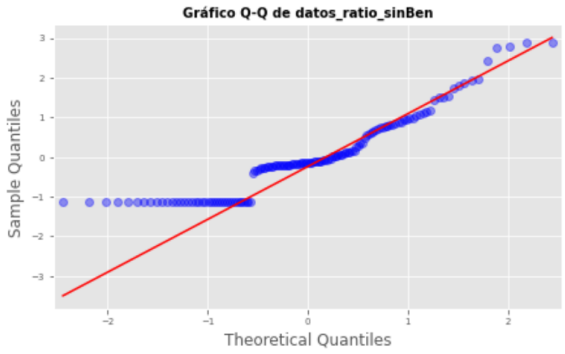




  Viendo estos gráficos se puede concluir a simple vista que la gráfica no sigue una curva de normalidad.

  Gráfico Q-Q





  Analizando el gráfico Q-Q, se puede observar que los datos no se ajustan a la recta, por lo que la distribución no es normal.

- **Métodos analíticos de asimetría y curtosis**

  Los estadísticos de asimetría (Skewness) y curtosis pueden emplearse para detectar desviaciones de la normalidad. Un valor de curtosis y/o coeficiente de asimetría entre -1 y 1, es generalmente considerada una ligera desviación de la normalidad. Entre -2 y 2 hay una evidente desviación de la normal pero no extrema.

```
print('Kursotis:', stats.kurtosis(datos_ratio_conBen))
print('Skewness:', stats.skew(datos_ratio_conBen))
```

```
Kursotis: 1.7036019339339
Skewness: 1.1006101564059843
```

```
print('Kursotis:', stats.kurtosis(datos_ratio_sinBen))
print('Skewness:', stats.skew(datos_ratio_sinBen))
```

```
Kursotis: 0.3649609805780605
Skewness: 0.8103229690459524
```

Los datos que contienen los meses en los que Ben trabajó, muestran una evidente desviación de la normal porque se encuentran entre -2 y 2, aunque no es extrema. En cambio, en los datos que no contienen esos meses, los datos están entre -1 y 1 y por lo tanto la desviación de la normal es más pequeña. Esto podría dar a entender que Ben provoca que haya más asimetría en los datos, pero no es demasiado pronunciado, pues cuando él no trabaja, también se produce una desviación.

- **Contraste de hipótesis:** tests de Shapiro-Wilk y D'Agostino's K-squared.

  Los test Shapiro-Wilk y D'Agostino's K-squared test son test de hipótesis que consideran como hipótesis nula que los datos proceden de una distribución normal.

  Añadir que quizá el test de Shapiro-Wil tiene una elevada sensibilidad a pequeñas desviaciones de la normal y puede no ser el más apropiado en este caso, pues se dispone de más de 50 datos.

  El p-value de estos test indica la probabilidad de obtener unos datos como los observados si realmente procediesen de una población con una distribución normal con la misma media y desviación que estos. Por lo tanto, si el p-value es menor que un determinado valor (típicamente 0.05), entonces se considera que hay evidencias suficientes para rechazar la normalidad.

```
# Shapiro-Wilk test
# ==============================================================================
shapiro_test = stats.shapiro(datos_ratio_conBen)
shapiro_test
```

```
ShapiroResult(statistic=0.8843843340873718, pvalue=5.26779375675801e-09)
```

```
# Shapiro-Wilk test
# ==============================================================================
shapiro_test = stats.shapiro(datos_ratio_sinBen)
shapiro_test
```

```
ShapiroResult(statistic=0.894536554813385, pvalue=2.3010009897461714e-08)
```

Los resultados de Shapiro Wilks muestran un p-valor muy bajo, y dado que es menor que 0,05, se considera que hay evidencias suficientes para rechazar la normalidad.

```
# D'Agostino's K-squared test
# ============================================================================
k2, p_value = stats.normaltest(datos_ratio_conBen)
print(f"Estadístico = {k2}, p-value = {p_value}")
```

```
Estadístico = 29.957171197666824, p-value = 3.12523678689713e-07
```

```
# D'Agostino's K-squared test
# ============================================================================
k2, p_value = stats.normaltest(datos_ratio_sinBen)
print(f"Estadístico = {k2}, p-value = {p_value}")
```

```
Estadístico = 14.315468673297305, p-value = 0.0007788170931479071
```

Los resultados del test de D'Agostino's K-squared test igualmente son muy bajos, de nuevo inferiores a 0,5 y por lo tanto hay evidencias suficientes para rechazar la normalidad.

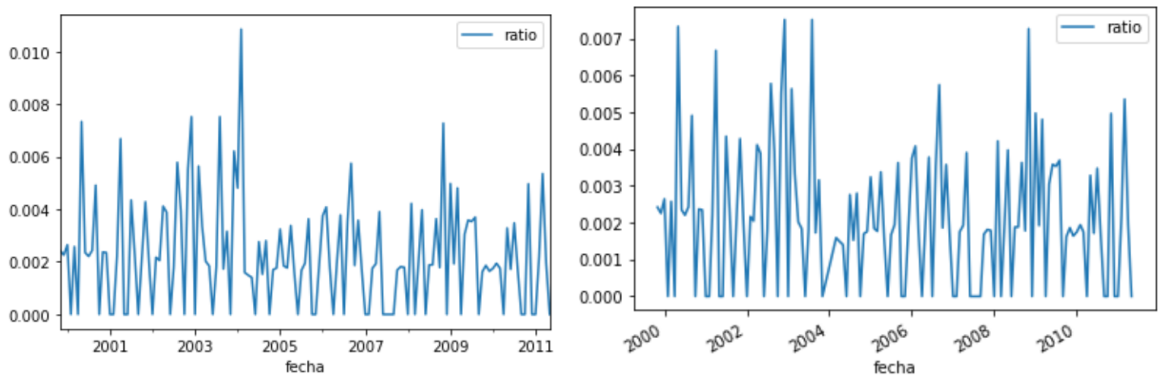
Como conclusión de las diferentes pruebas de normalidad realizadas, se puede determinar que los datos no siguen una distribución normal en ninguno de los escenarios (teniendo en cuenta los meses en los que trabajó Ben y los que no).

Por lo tanto no hay evidencias suficientes para determinar que Ben es culpable, pues no hay un cambio de normalidad evidente entre los dos datasets analizados. Siguiendo esta prueba, se considera que Ben podría ser inocente.

3.3. Entrenamiento de un modelo de predicción de regresión lineal: ARIMA

Como se explicaba en el punto 2, ARIMA es un algoritmo de regresión de series temporales, como primer paso se debe analizar su estacionalidad.

Primero se han representado gráficos de líneas para observar las tendencias y los picos de ratio más elevado. El primer gráfico representa todos los ratios de todos los meses y el segundo gráfico excluye los meses en los que Ben trabajó. Estas representaciones pueden proporcionar pistas para buscar patrones repetidos a lo largo del tiempo.



Vemos de nuevo que la diferencia más grande se encuentra en los meses en los que Ben trabajó.

A continuación se ha ejecutado el test de estacionalidad de la librería *pmdarima* sobre todos los ratios de todos los meses del dataset y se ha obtenido lo siguiente:

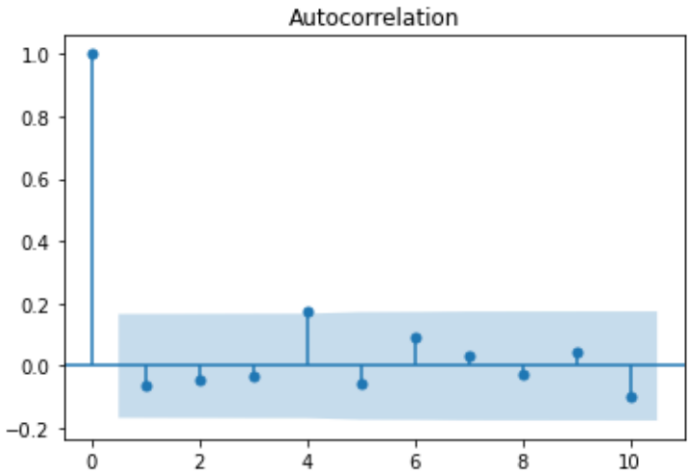
```
from pmdarima.arima.stationarity import ADFTest

# Probar si se debe diferenciar el nivel de significancia alpha = 0,05
adf_test = ADFTest(alpha=0.05)
adf_test.should_diff(train_df)
```

```
(0.01, False)
```

Con estos datos podemos concluir que los datos no son estacionarios. Por lo tanto necesitaremos el concepto “Integrated (I)”, denotado por “d” en las series temporales para hacer los datos estacionarios cuando se construya el modelo ARIMA.

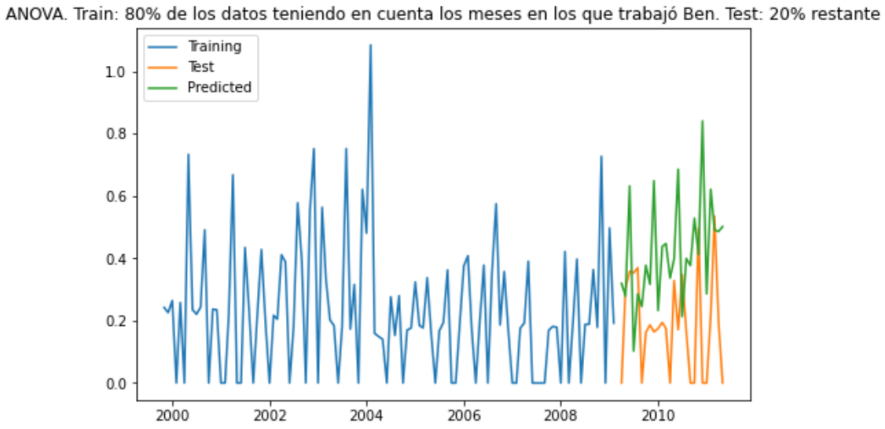
Además se ha analizado la autocorrelación, la cual es una función de similitud entre dos observaciones a la diferencia de tiempo entre ella y sirve como herramienta matemática para encontrar patrones repetitivos.



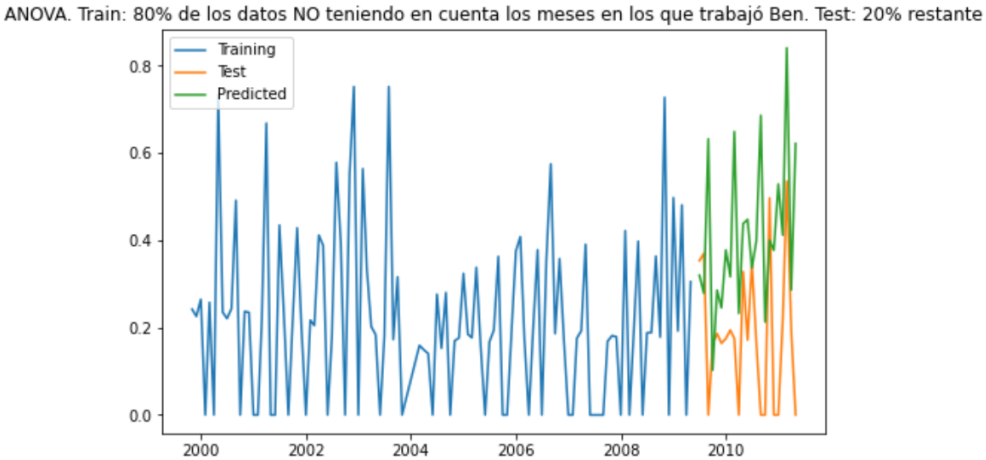
Podemos observar una correlación significativa en el primer desfase, seguido de correlaciones que no son significativas. Esto indica un término autorregresivo en los datos. Dado que el número de correlaciones significativas indican el orden del término autorregresivo, en este caso, el término autorregresivo sería 1.

Continuando con el análisis se han entrenado diferentes modelos ARIMA con diferentes datasets de entrenamiento. Con esto se pretende observar si lo predicho por ARIMA con cada uno de los modelos sobre el dataset de test se corresponde con lo esperado. Nota*: para mejorar el pronóstico del algoritmo ARIMA y dado que los datos de los ratios son tan pequeños que se diferencian en apenas milésimas, se han multiplicado x 100 todos los ratios. De esta manera el algoritmo observa mayor distancia entre los datos y ofrecerá mayor precisión en sus predicciones. A partir de ahora se hablará de porcentaje de ratio.

- **Prueba 1**: Dataset de training: 80% de los datos teniendo en cuenta los meses en los que Ben trabajó. Test: 20% restante.

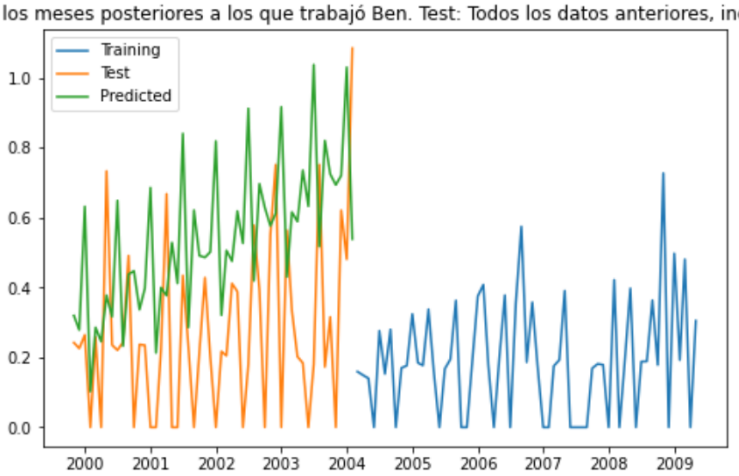


- **Prueba 2**: Train: 80% de los datos teniendo NO teniendo en cuenta los meses en los que trabajó Ben. Test: 20% restante.



- **Prueba 3**: Train: datos a partir de marzo 2004, es decir cuando Ben ya no trabaja. Test: Datos anteriores al mes en el que Ben deja de trabajar (se incluyen los meses en los que Ben trabaja para observar si pueden darse anomalías).

ANOVA. Train: Datos de los meses posteriores a los que trabajó Ben. Test: Todos los datos anteriores, incluidos los que trabajó Ben



Analizando la calidad del modelo, se puede ver que aunque predice una tendencia similar a los valores de porcentaje de ratio reales, no es demasiado preciso. Se puede observar en detalle el entrenamiento del modelo y un resumen de los resultados en el Anexo.

Comparando los dos primeros gráficos, se puede observar que ambos muestran una predicción muy parecida. Por lo tanto no hay una diferencia clara entre ambas predicciones, a pesar de que los datos de entrenamiento de la segunda prueba no contemplan los meses en los que se producen datos anómalos.

La misma sensación surge al observar el gráfico de la tercera prueba. El modelo se ha entrenado con los datos a partir del mes en el que Ben ya no trabajó y se ha probado con los datos del dataset anteriores a este mes. En el tramo utilizado para train tenemos certeza de que ya no hay influencia del enfermero. Aunque, sí que es cierto que se dan situaciones parecidas a cuando trabajaba. Lo que observamos en la predicción es que se dan situaciones anómalas prácticamente a lo largo de todo el tramo de test.

Por lo tanto, dado que no se dan únicamente situaciones anómalas en el intervalo en el que Ben trabajó, no se podría determinar que Ben es culpable. El modelo no predice una situación anómala entrenando exceptualmente con los meses en los que trabajó Ben.

4. Conclusiones

Tras aplicar 3 tipos de análisis diferentes, vemos que pueden obtenerse conclusiones diferentes.

Por un lado, los análisis estadísticos clásicos como es el **boxplot** determinan claramente que Ben podría ser **CULPABLE** debido a la evidencia de casos anómalos cuando Ben trabajó en el hospital.

Sin embargo, analizando los datos en mayor profundidad con las **pruebas de distribución uniforme y de normalidad**, no se presentan cambios evidentes realizando estas pruebas sobre los datos teniendo en cuenta y no teniendo en cuenta los meses en los que Ben trabajó. Solo en caso de presentar resultados diferentes con evidencias en la uniformidad y la normalidad de los datos cuando Ben no trabaja, se podría considerar a Ben culpable. Por lo tanto, con estos estadísticos, consideramos a Ben **INOCENTE**.

Se recogen las mismas conclusiones al aplicar un modelo de regresión lineal para series temporales como es **ANOVA**. No se detectan anomalías en sus predicciones en ninguna de las pruebas realizadas para la temporada en la que trabajó Ben y por lo tanto se puede considerar **INOCENTE** ante los datos analizados.

Como resumen de las conclusiones, ante los datos analizados se puede decir que es más probable que Ben sea INOCENTE, aunque para hacer un juicio más acertado, se deberían tener en cuenta otras variables externas como la existencia de enfermedades infecciosas o mortales, la eficiencia y la calidad del trabajo de sus compañeros o la cantidad de personal y médicos cualificados durante los meses en los que Ben trabajó.

5. Referencias

https://towardsdatascience.com/time-series-forecasting-using-auto-arima-in-python-bb83e49210cd

https://www.cienciadedatos.net/documentos/pystats01-ajuste-distribuciones-python.html

https://programmerclick.com/article/7684422434/

https://www.cienciadedatos.net/documentos/pystats06-analisis-normalidad-python.html

Anexo

Modelo Arima

3.1. Dataset entrenamiento 1: 80% de los datos teniendo en cuenta los meses de Ben

```
train_df_1 = train_df[:112]
arima_model = pmdarima.auto_arima(train_df_1, start_p=0, d=1, start_q=0,
                          max_p=5, max_d=5, max_q=5, start_P=0,
                          D=1, start_Q=0, max_P=5, max_D=5,
                          max_Q=5, m=12, seasonal=True,
                          error_action='warn', trace=True,
                          supress_warnings=True, stepwise=True,
                          random_state=20, n_fits=50)
```

```
Performing stepwise search to minimize aic
 ARIMA(0,1,0)(0,1,0)[12]             : AIC=108.416, Time=0.03 sec
 ARIMA(1,1,0)(1,1,0)[12]             : AIC=55.745, Time=0.14 sec
 ARIMA(0,1,1)(0,1,1)[12]             : AIC=inf, Time=0.13 sec
 ARIMA(1,1,0)(0,1,0)[12]             : AIC=78.821, Time=0.02 sec
 ARIMA(1,1,0)(2,1,0)[12]             : AIC=56.416, Time=0.12 sec
 ARIMA(1,1,0)(1,1,1)[12]             : AIC=inf, Time=0.35 sec
 ARIMA(1,1,0)(0,1,1)[12]             : AIC=inf, Time=0.23 sec
 ARIMA(1,1,0)(2,1,1)[12]             : AIC=inf, Time=0.95 sec
 ARIMA(0,1,0)(1,1,0)[12]             : AIC=87.601, Time=0.05 sec
 ARIMA(2,1,0)(1,1,0)[12]             : AIC=46.132, Time=0.07 sec
 ARIMA(2,1,0)(0,1,0)[12]             : AIC=67.400, Time=0.03 sec
 ARIMA(2,1,0)(2,1,0)[12]             : AIC=47.797, Time=0.27 sec
 ARIMA(2,1,0)(1,1,1)[12]             : AIC=inf, Time=0.43 sec
 ARIMA(2,1,0)(0,1,1)[12]             : AIC=inf, Time=0.29 sec
 ARIMA(2,1,0)(2,1,1)[12]             : AIC=inf, Time=1.12 sec
 ARIMA(3,1,0)(1,1,0)[12]             : AIC=36.586, Time=0.16 sec
 ARIMA(3,1,0)(0,1,0)[12]             : AIC=53.222, Time=0.06 sec
 ARIMA(3,1,0)(2,1,0)[12]             : AIC=37.146, Time=0.31 sec
 ARIMA(3,1,0)(1,1,1)[12]             : AIC=inf, Time=0.41 sec
 ARIMA(3,1,0)(0,1,1)[12]             : AIC=inf, Time=0.34 sec
 ARIMA(3,1,0)(2,1,1)[12]             : AIC=inf, Time=1.29 sec
 ARIMA(4,1,0)(1,1,0)[12]             : AIC=37.369, Time=0.26 sec
 ARIMA(3,1,1)(1,1,0)[12]             : AIC=inf, Time=0.45 sec
 ARIMA(2,1,1)(1,1,0)[12]             : AIC=inf, Time=0.50 sec
 ARIMA(4,1,1)(1,1,0)[12]             : AIC=inf, Time=0.58 sec
 ARIMA(3,1,0)(1,1,0)[12] intercept   : AIC=38.551, Time=0.23 sec

Best model:  ARIMA(3,1,0)(1,1,0)[12]
Total fit time: 8.840 seconds
```

```
arima_model.summary()
```

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	112
Model:	SARIMAX(3, 1, 0)x(1, 1, 0, 12)	**Log Likelihood**	-13.293
Date:	Mon, 14 Feb 2022	**AIC**	36.586
Time:	22:00:26	**BIC**	49.561
Sample:	0	**HQIC**	41.835
	- 112		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.8365	0.088	-9.557	0.000	-1.008	-0.665
ar.L2	-0.5770	0.112	-5.138	0.000	-0.797	-0.357
ar.L3	-0.3437	0.113	-3.055	0.002	-0.564	-0.123
ar.S.L12	-0.4438	0.110	-4.025	0.000	-0.660	-0.228
sigma2	0.0739	0.011	6.734	0.000	0.052	0.095

Ljung-Box (L1) (Q):	0.21	**Jarque-Bera (JB):**	0.17
Prob(Q):	0.65	**Prob(JB):**	0.92
Heteroskedasticity (H):	0.88	**Skew:**	-0.10
Prob(H) (two-sided):	0.71	**Kurtosis:**	2.94

3.2. Dataset entrenamiento 2: 80% de los datos teniendo sin tener en cuenta los meses de Ben

```
datos_hasta_noviembre2003 = train_df[:49]
datos_desde_marzo2004 = train_df[52:115]
train_df_2 = pd.concat([datos_hasta_noviembre2003, datos_desde_marzo2004])
arima_model_2 = pmdarima.auto_arima(train_df_2, start_p=0, d=1, start_q=0,
                          max_p=5, max_d=5, max_q=5, start_P=0,
                          D=1, start_Q=0, max_P=5, max_D=5,
                          max_Q=5, m=12, seasonal=True,
                          error_action='warn', trace=True,
                          supress_warnings=True, stepwise=True,
                          random_state=20, n_fits=50)
```

```
Performing stepwise search to minimize aic
 ARIMA(0,1,0)(0,1,0)[12]             : AIC=99.862, Time=0.03 sec
 ARIMA(1,1,0)(1,1,0)[12]             : AIC=54.641, Time=0.11 sec
 ARIMA(0,1,1)(0,1,1)[12]             : AIC=inf, Time=0.26 sec
 ARIMA(1,1,0)(0,1,0)[12]             : AIC=70.632, Time=0.02 sec
 ARIMA(1,1,0)(2,1,0)[12]             : AIC=47.829, Time=0.16 sec
 ARIMA(1,1,0)(3,1,0)[12]             : AIC=43.441, Time=0.38 sec
 ARIMA(1,1,0)(4,1,0)[12]             : AIC=42.893, Time=0.76 sec
 ARIMA(1,1,0)(5,1,0)[12]             : AIC=40.862, Time=1.45 sec
 ARIMA(1,1,0)(5,1,1)[12]             : AIC=42.853, Time=2.58 sec
 ARIMA(1,1,0)(4,1,1)[12]             : AIC=42.507, Time=1.98 sec
 ARIMA(0,1,0)(5,1,0)[12]             : AIC=68.348, Time=0.94 sec
 ARIMA(2,1,0)(5,1,0)[12]             : AIC=27.122, Time=1.86 sec
 ARIMA(2,1,0)(4,1,0)[12]             : AIC=27.542, Time=0.97 sec
 ARIMA(2,1,0)(5,1,1)[12]             : AIC=29.119, Time=3.31 sec
 ARIMA(2,1,0)(4,1,1)[12]             : AIC=28.032, Time=1.68 sec
 ARIMA(3,1,0)(5,1,0)[12]             : AIC=8.462, Time=2.24 sec
 ARIMA(3,1,0)(4,1,0)[12]             : AIC=9.981, Time=1.49 sec
 ARIMA(3,1,0)(5,1,1)[12]             : AIC=10.353, Time=3.49 sec
 ARIMA(3,1,0)(4,1,1)[12]             : AIC=10.010, Time=1.97 sec
 ARIMA(4,1,0)(5,1,0)[12]             : AIC=8.528, Time=2.12 sec
 ARIMA(3,1,1)(5,1,0)[12]             : AIC=0.915, Time=3.00 sec
 ARIMA(3,1,1)(4,1,0)[12]             : AIC=5.436, Time=1.48 sec
 ARIMA(3,1,1)(5,1,1)[12]             : AIC=2.901, Time=5.47 sec
 ARIMA(3,1,1)(4,1,1)[12]             : AIC=3.945, Time=2.98 sec
 ARIMA(2,1,1)(5,1,0)[12]             : AIC=inf, Time=4.56 sec
 ARIMA(4,1,1)(5,1,0)[12]             : AIC=inf, Time=6.14 sec
 ARIMA(3,1,2)(5,1,0)[12]             : AIC=1.259, Time=6.54 sec
 ARIMA(2,1,2)(5,1,0)[12]             : AIC=inf, Time=5.40 sec
 ARIMA(4,1,2)(5,1,0)[12]             : AIC=2.203, Time=7.28 sec
 ARIMA(3,1,1)(5,1,0)[12] intercept   : AIC=2.907, Time=5.80 sec

Best model:  ARIMA(3,1,1)(5,1,0)[12]
Total fit time: 76.492 seconds
```

```
arima_model_2.summary()
```

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	112
Model:	SARIMAX(3, 1, 1)x(5, 1, [], 12)	**Log Likelihood**	9.542
Date:	Mon, 14 Feb 2022	**AIC**	0.915
Time:	22:02:04	**BIC**	26.867
Sample:	0	**HQIC**	11.415
	- 112		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.2837	0.116	-2.444	0.015	-0.511	-0.056
ar.L2	-0.2857	0.134	-2.137	0.033	-0.548	-0.024
ar.L3	-0.1984	0.138	-1.441	0.149	-0.468	0.071
ma.L1	-0.7866	0.101	-7.765	0.000	-0.985	-0.588
ar.S.L12	-0.9891	0.125	-7.906	0.000	-1.234	-0.744
ar.S.L24	-0.8955	0.171	-5.230	0.000	-1.231	-0.560
ar.S.L36	-0.7569	0.202	-3.750	0.000	-1.152	-0.361
ar.S.L48	-0.6067	0.159	-3.810	0.000	-0.919	-0.295
ar.S.L60	-0.3826	0.157	-2.443	0.015	-0.689	-0.076
sigma2	0.0374	0.007	5.236	0.000	0.023	0.051

Ljung-Box (L1) (Q):	0.02	**Jarque-Bera (JB):**	2.08
Prob(Q):	0.90	**Prob(JB):**	0.35
Heteroskedasticity (H):	0.83	**Skew:**	0.28
Prob(H) (two-sided):	0.59	**Kurtosis:**	2.57

3.3. Dataset entrenamiento 3: entrenamiento con los datos posteriores a los meses trabajados por Ben

```
train_df_3 = train_df[52:115]
arima_model_2 = pmdarima.auto_arima(train_df_3, start_p=0, d=1, start_q=0,
                        max_p=5, max_d=5, max_q=5, start_P=0,
                        D=1, start_Q=0, max_P=5, max_D=5,
                        max_Q=5, m=12, seasonal=True,
                        error_action='warn', trace=True,
                        supress_warnings=True, stepwise=True,
                        random_state=20, n_fits=50)
```

```
Performing stepwise search to minimize aic
 ARIMA(0,1,0)(0,1,0)[12]             : AIC=38.452, Time=0.02 sec
 ARIMA(1,1,0)(1,1,0)[12]             : AIC=6.313, Time=0.07 sec
 ARIMA(0,1,1)(0,1,1)[12]             : AIC=inf, Time=0.21 sec
 ARIMA(1,1,0)(0,1,0)[12]             : AIC=16.328, Time=0.03 sec
 ARIMA(1,1,0)(2,1,0)[12]             : AIC=6.271, Time=0.19 sec
 ARIMA(1,1,0)(3,1,0)[12]             : AIC=7.895, Time=0.49 sec
 ARIMA(1,1,0)(2,1,1)[12]             : AIC=inf, Time=0.70 sec
 ARIMA(1,1,0)(1,1,1)[12]             : AIC=5.790, Time=0.21 sec
 ARIMA(1,1,0)(0,1,1)[12]             : AIC=3.796, Time=0.08 sec
 ARIMA(1,1,0)(0,1,2)[12]             : AIC=5.791, Time=0.19 sec
 ARIMA(1,1,0)(1,1,2)[12]             : AIC=7.796, Time=0.27 sec
 ARIMA(0,1,0)(0,1,1)[12]             : AIC=inf, Time=0.17 sec
 ARIMA(2,1,0)(0,1,1)[12]             : AIC=3.899, Time=0.08 sec
 ARIMA(1,1,1)(0,1,1)[12]             : AIC=inf, Time=0.19 sec
 ARIMA(2,1,1)(0,1,1)[12]             : AIC=inf, Time=0.23 sec
 ARIMA(1,1,0)(0,1,1)[12] intercept   : AIC=5.779, Time=0.11 sec

Best model:  ARIMA(1,1,0)(0,1,1)[12]
Total fit time: 3.230 seconds
```

```
arima_model_2.summary()
```

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	63
Model:	SARIMAX(1, 1, 0)x(0, 1, [1], 12)	**Log Likelihood**	1.102
Date:	Mon, 14 Feb 2022	**AIC**	3.796
Time:	22:02:31	**BIC**	9.532
Sample:	0	**HQIC**	5.981
	- 63		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.6030	0.115	-5.244	0.000	-0.828	-0.378
ma.S.L12	-0.7181	0.317	-2.262	0.024	-1.340	-0.096
sigma2	0.0470	0.012	3.801	0.000	0.023	0.071

Ljung-Box (L1) (Q):	0.82	**Jarque-Bera (JB):**	0.33
Prob(Q):	0.37	**Prob(JB):**	0.85
Heteroskedasticity (H):	1.22	**Skew:**	0.09
Prob(H) (two-sided):	0.69	**Kurtosis:**	2.64