La población carcelaria en Colombia 1991-2017

Sergio Solano

Febrero de 2016

## Data

El INPEC publica mensualemente la serie población carcelaria, desde 1991 hasta el mese anterior a la publicación. Esta serie se encuentra separada por situación jurídica (condenados, sindicados) y por genero.

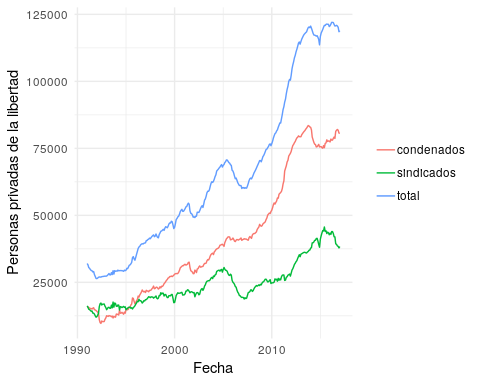
## Objetivo

El proceso a través del cual las personas pasan de una situación jurídica a otra es conocido, y sin embargo no observado, pues no se publican las series de tiempo que reflejan esta transición (cantidad de personas sentenciadas por mes, cantidad de personas liberadas mes, duración de la condena).

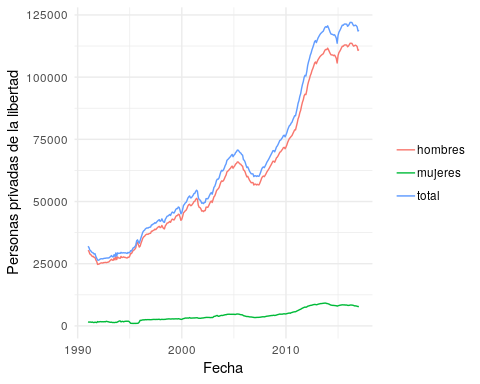
Este ejercicio de demografía, enmarcado en el estudio de poblaciones pequeñas, presenta la oportunidad de comparar la efectividad de diferentes métodos de proyección.

# Análisis exploratorio

La población carcelaria total entre 1991 y 2017 se ha cuadruplicado, al pasar de 32.036 a 128.125 internos. Aunque la mayoría son hombres la población carcelaria feminina en el mismo periodo ha crecido a una tasa mayor, pues se ha multiplicado por cinco, de 1633 en 1991 a 7800 en 2017.

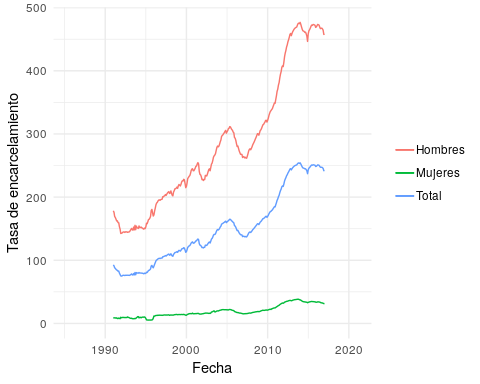


## Saving 5 x 4 in image

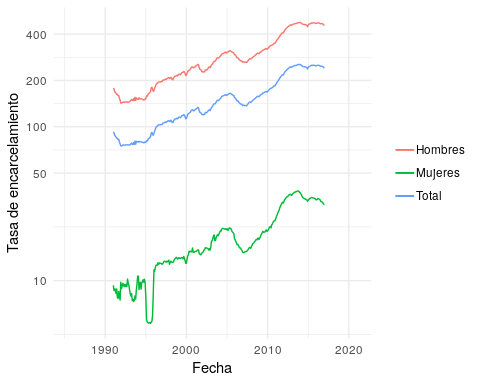


## Saving 5 x 4 in image

El incremento en la población carcelaria podría tomarse como un efecto del crecimiento de la población colombiana. Para validar este supuesto calculamos la tasa de encarcelamiento, que mide la cantidad de personas encarceladas por cada cien mil habitantes. Este indicador pasó de 92 personas por cada cien mil habitantes en enero de 1991 a 242 en enero de 2016. Tal incremento se puede ver tanto en hombres como en mujeres.

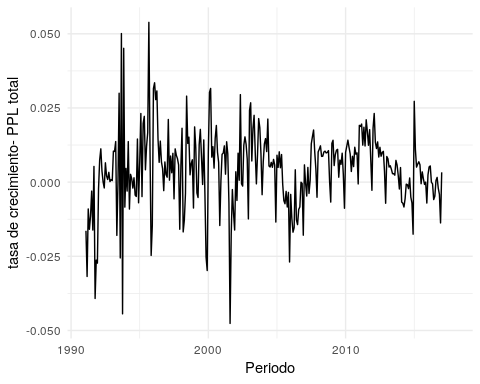


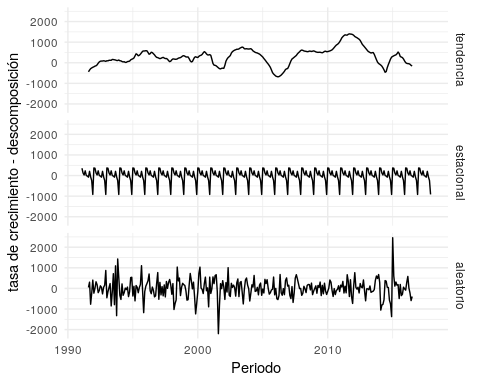
La tasa de encarcelamiento ha crecido de forma exponencial, tanto en hombres como en mujeres, y se puede ver en la gráfica siguiente.

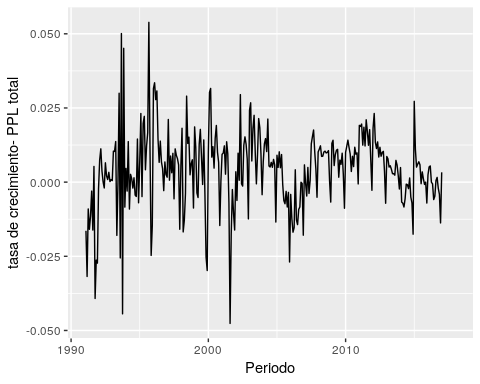


## Crecimiento de la Población Privada de la Libertad (PPL)

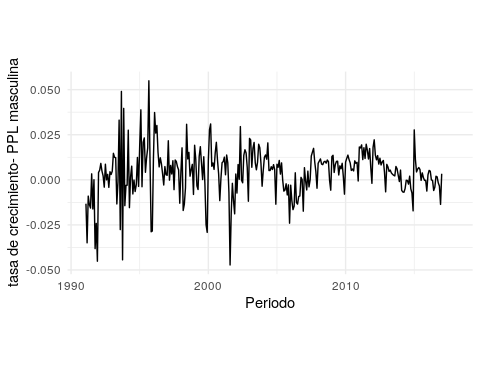
Una primera aproximación al análisis de la población carcelaria, se podría realizar al separar los componentes estacionales, de tendencia y aleatorios de la serie de tiempo. No obstante, es posible inferir que la variabilidad de la serie no es constante. Por esta razón resultaría pertinente analizar la tasa de crecimiento de la población de mes a mes. Una técnica comunmente usada es trabajar con la difencia de los logaritmos de la población, que para variaciones cercanas a cero, se aproxima a la tasa de crecimiento.

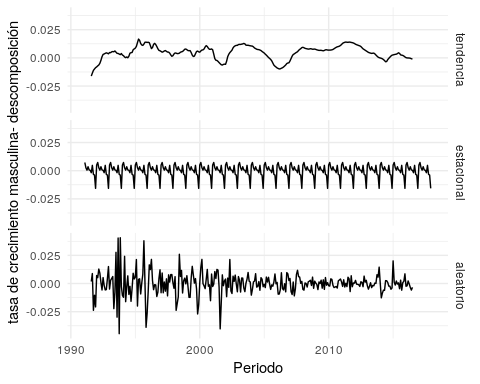




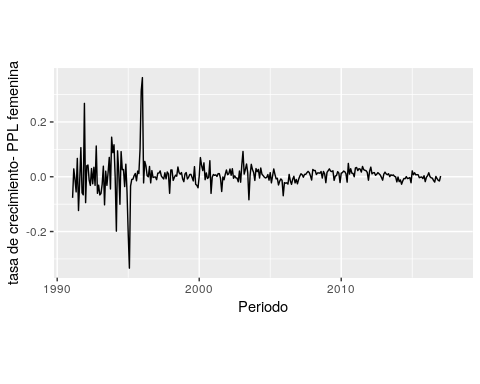


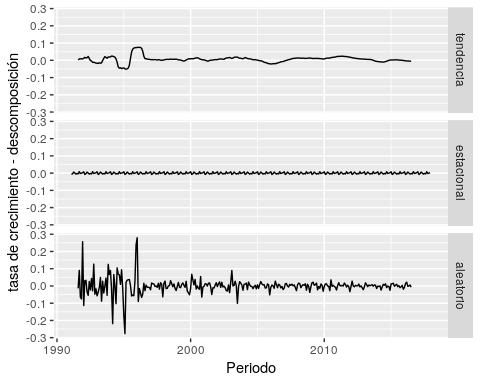
Crecimiento de la población privada de la libertad masculida





Crecimiento de la población privada de la libertad femenina

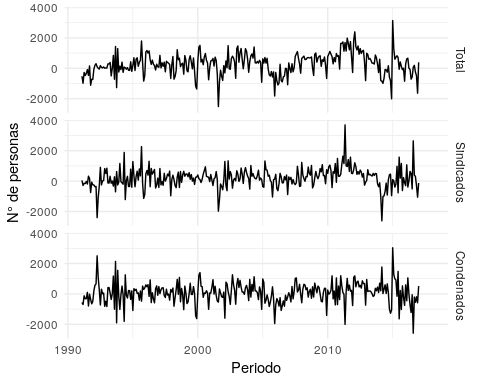


 Como se evidencia que componente aleatorio no tiene varianza constante se genera componente lineal, el análisis pertinente es por sindicados y condenados, pues es el que se considera en el resto de la tesis.

# Población sindicada a serie de tiempo por situación  
ppl\_sitjur %>% spread(key = "categoria", value = valor) -> ppl\_situacion   
# Serie de tiempo de población total  
TPPtimeseries <- ts(ppl\_situacion$total, frequency=12, start=c(1991,1))  
# Serie de tiempo de población sindicada   
SPPtimeseries <- ts(ppl\_situacion$condenados, frequency=12, start=c(1991,1))  
# Serie de tiempo de población condenada  
CPPtimeseries <- ts(ppl\_situacion$sindicados, frequency=12, start=c(1991,1))  
  
# Ajustar a zoo, para poder hacer gráfica  
Total <- as.zoo(diff(TPPtimeseries))  
Sindicados <- as.zoo(diff(SPPtimeseries))  
Condenados <- as.zoo(diff(CPPtimeseries))  
  
# Gráficar crecimient de las tres  
autoplot(merge(Total, Sindicados, Condenados), geom = "line") + ylab("N° de personas") + xlab("Periodo") +theme\_minimal() -> graf\_var  
graf\_var

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.

## Warning: Removed 11 rows containing missing values (geom\_path).



ggsave("variacion\_intermensual.png")

## Saving 5 x 4 in image  
## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.

## Warning: Removed 11 rows containing missing values (geom\_path).

# Descomponer variación intermensual  
DeltaTPP <- decompose(Total)

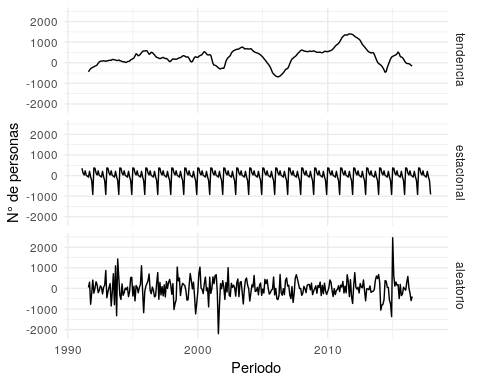
## Warning in decompose(Total): Métodos incompatibles ("Ops.zoo", "Ops.ts")  
## para "-"

## Warning in structure(list(x = x, seasonal = seasonal, trend = trend, random  
## = if (type == : Métodos incompatibles ("Ops.zoo", "Ops.ts") para "-"

# Asignar  
tendencia <- as.zoo(DeltaTPP$trend)  
aleatorio <- as.zoo(DeltaTPP$random)  
estacional <- as.zoo(DeltaTPP$seasonal)  
  
# Generar gráfica   
autoplot(merge(tendencia,estacional, aleatorio), geom = "line") + ylab("N° de personas") + xlab("Periodo") + theme\_minimal() -> graf\_tendencia\_tpp  
graf\_tendencia\_tpp

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.

## Warning: Removed 23 rows containing missing values (geom\_path).



ggsave("variacion\_mensual\_total\_desc.png")

## Saving 5 x 4 in image  
## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.

## Warning: Removed 23 rows containing missing values (geom\_path).

# Descomponer variación intermensual  
DeltaSPP <- decompose(Sindicados)

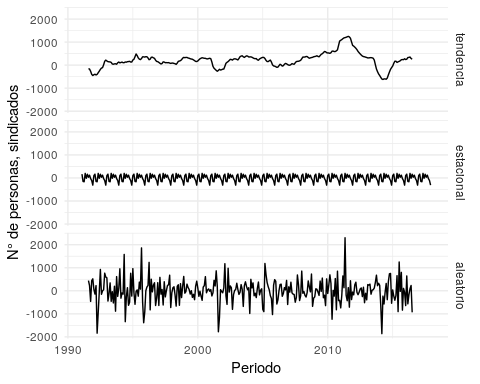
## Warning in decompose(Sindicados): Métodos incompatibles ("Ops.zoo",  
## "Ops.ts") para "-"

## Warning in structure(list(x = x, seasonal = seasonal, trend = trend, random  
## = if (type == : Métodos incompatibles ("Ops.zoo", "Ops.ts") para "-"

# Asignar  
tendencia <- as.zoo(DeltaSPP$trend)  
aleatorio <- as.zoo(DeltaSPP$random)  
estacional <- as.zoo(DeltaSPP$seasonal)  
  
# Generar gráfica  
autoplot(merge(tendencia,estacional, aleatorio), geom = "line") + ylab("N° de personas, sindicados") + xlab("Periodo") + theme\_minimal() -> graf\_tendencia\_spp  
  
# gráfica  
graf\_tendencia\_spp

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.

## Warning: Removed 23 rows containing missing values (geom\_path).



ggsave("variacion\_mensual\_sindicados\_desc.png")

## Saving 5 x 4 in image  
## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.

## Warning: Removed 23 rows containing missing values (geom\_path).

# Descomponer variación intermensual  
DeltaCPP <- decompose(Condenados)

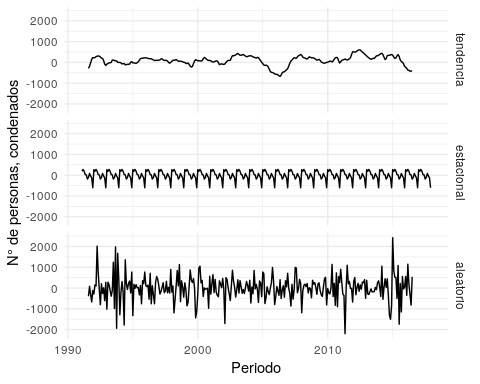
## Warning in decompose(Condenados): Métodos incompatibles ("Ops.zoo",  
## "Ops.ts") para "-"

## Warning in structure(list(x = x, seasonal = seasonal, trend = trend, random  
## = if (type == : Métodos incompatibles ("Ops.zoo", "Ops.ts") para "-"

# Asignar  
tendencia <- as.zoo(DeltaCPP$trend)  
aleatorio <- as.zoo(DeltaCPP$random)  
estacional <- as.zoo(DeltaCPP$seasonal)  
  
# Generar gráfica  
autoplot(merge(tendencia,estacional, aleatorio), geom = "line") + ylab("N° de personas, condenados") + xlab("Periodo") +theme\_minimal()-> graf\_tendencia\_cpp  
  
# gráfica  
graf\_tendencia\_cpp

## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.

## Warning: Removed 23 rows containing missing values (geom\_path).



ggsave("variacion\_mensual\_condenados\_desc.png")

## Saving 5 x 4 in image  
## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.

## Warning: Removed 23 rows containing missing values (geom\_path).

# Procesos ARIMA

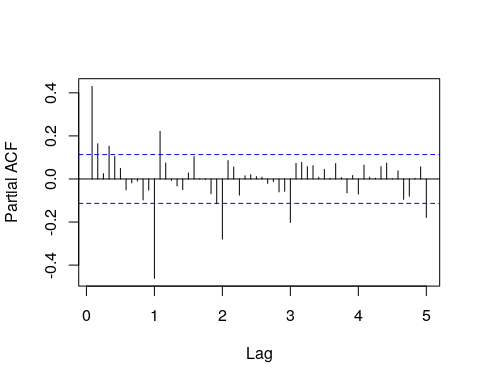
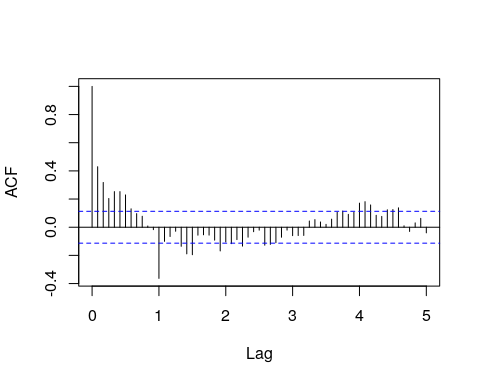
Los procesos ARMA son procesos aleatorios de la forma

Aunque el termino no tiene necesariamente una distribución normal, por el resto de documento se asumirá una distribución normal con media y varianza , a menos que se especifique lo contrario.

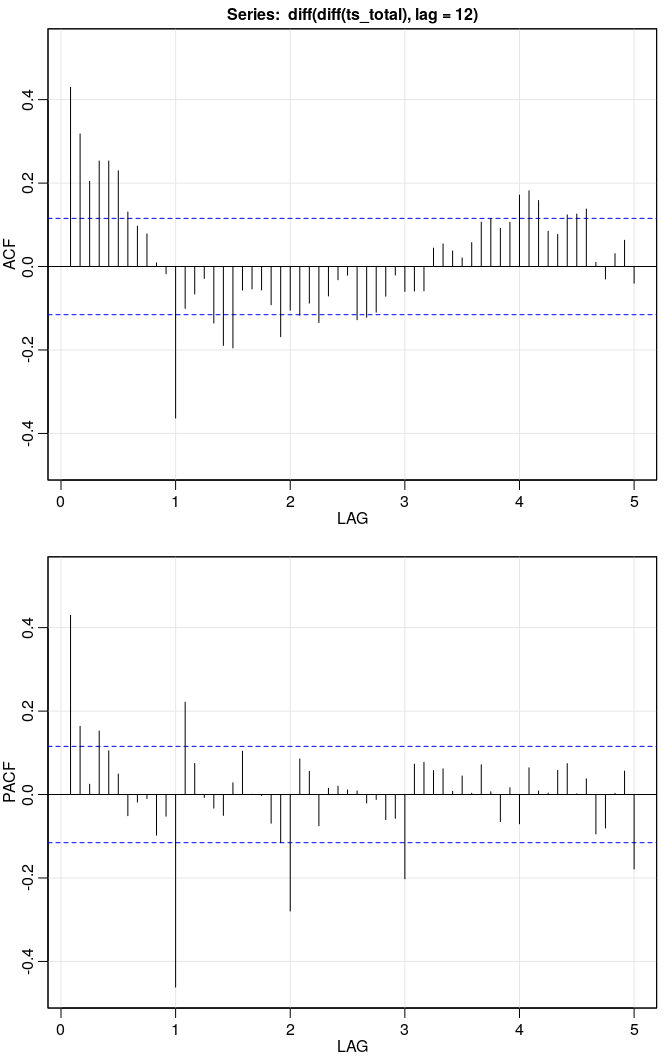
Los procesos ARIMA resultan al considerar una serie de la forma:

Tal que el proceso es un proceso ARMA.

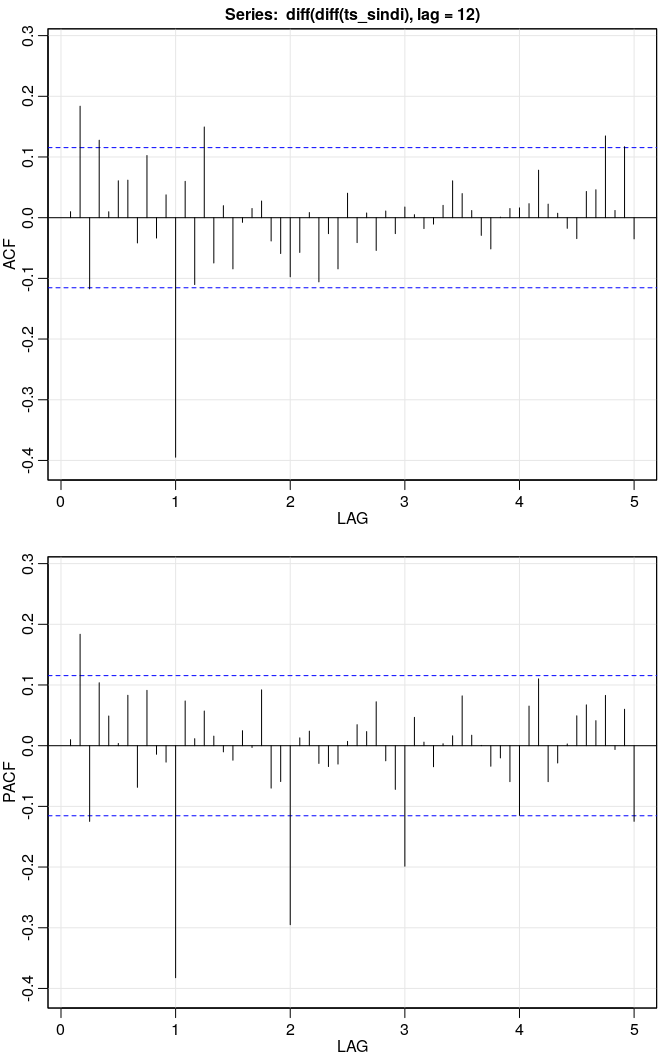
Una primera aproximación a la proyección de poblaciones carcelarias será validar si es posible modelar el proceso como un proceso ARIMA. Con este propósito presentamos las funciones de autocorrelación y autocorrelación parcial de la población total.



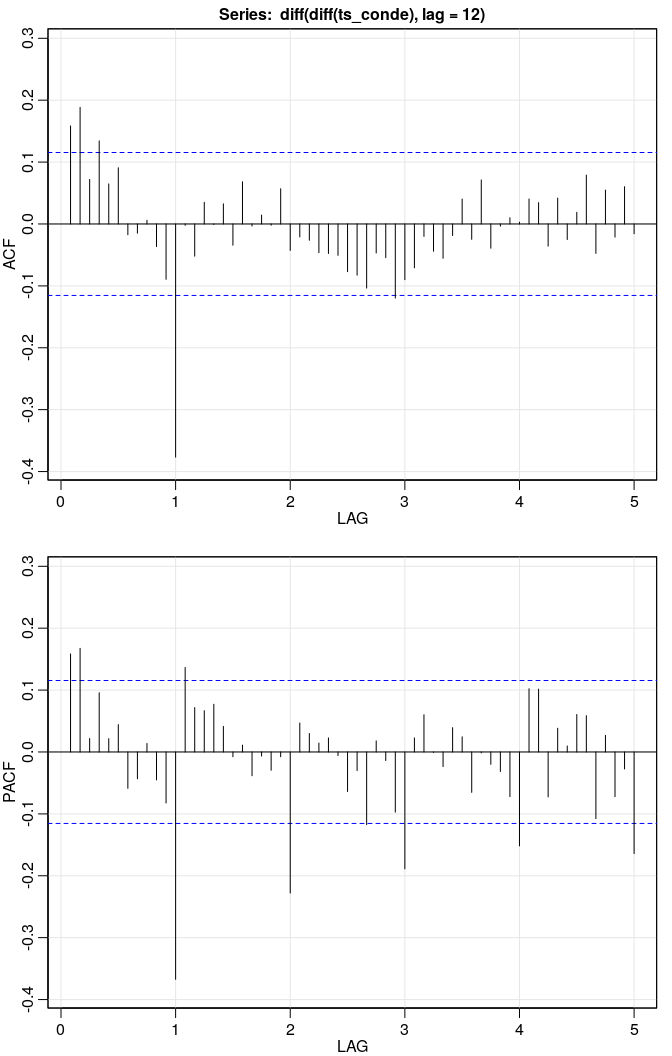
## Crear variables  
# Población carcelaria total  
ppl\_sitjur %>% dplyr::filter(categoria == "total", !(is.na(valor))) -> ppl\_total  
ppl\_total <- ppl\_total$valor  
ts\_total <- ts(ppl\_total, start = 1991, frequency = 12)  
# Sindicados  
ppl\_sitjur %>% dplyr::filter(categoria == "sindicados", !(is.na(valor))) -> ppl\_sindi  
ppl\_sindi <- ppl\_sindi$valor  
ts\_sindi <- ts(ppl\_sindi, start = 1991, frequency = 12)  
# Condenados  
ppl\_sitjur %>% dplyr::filter(categoria == "condenados", !(is.na(valor))) -> ppl\_conde  
ppl\_conde <- ppl\_conde$valor  
ts\_conde <- ts(ppl\_conde, start = 1991, frequency = 12)  
error = 2/sqrt(length(ppl\_conde))  
  
## Población total  
# Calcular acf y pacf  
acf\_total <- as.data.frame(acf2(diff(diff(ts\_total), lag = 12), max.lag = 60))



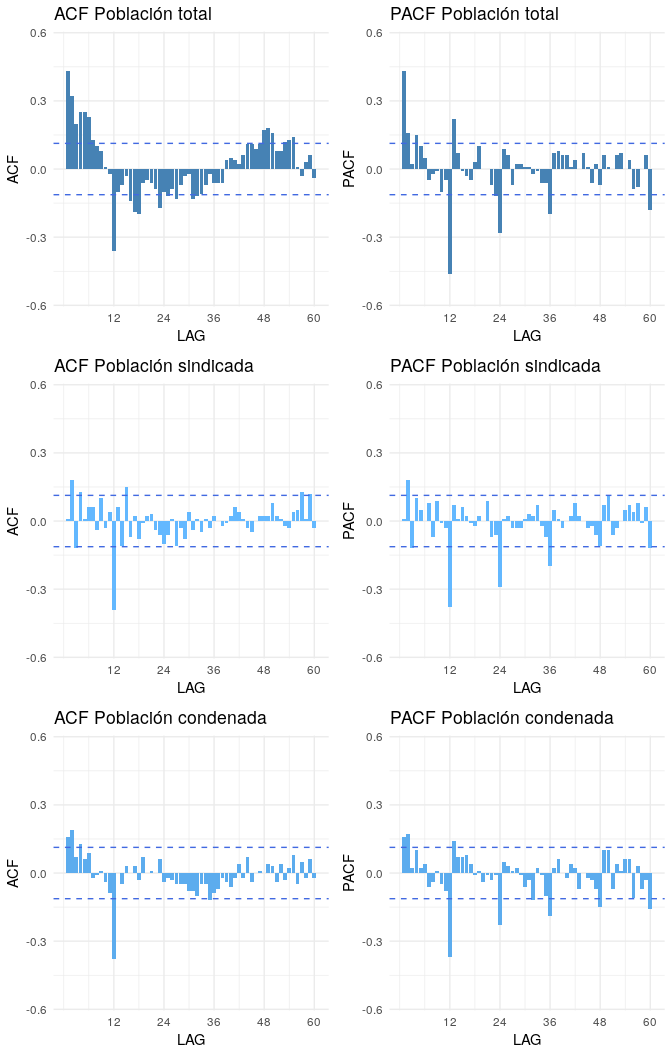
# Gráfica acf total  
acf\_total %>% ggplot() + geom\_bar(aes(x = c(1:60), y = ACF), stat = "identity", fill = "steelblue") + scale\_x\_continuous(breaks = c(12,24,36,48,60)) + xlab("LAG") + ggtitle("ACF Población total") + ylim(c(-0.55,0.55)) + geom\_hline(aes(yintercept = error), colour = "royalblue", linetype="dashed") + geom\_hline(aes(yintercept = -error), colour = "royalblue", linetype="dashed") + theme\_minimal()-> graf\_acf\_total  
# Gráfica pacf total   
acf\_total %>% ggplot() + geom\_bar(aes(x = c(1:60), y = PACF), stat = "identity", fill = "steelblue") + scale\_x\_continuous(breaks = c(12,24,36,48,60)) + xlab("LAG") + ggtitle("PACF Población total") + ylim(c(-0.55,0.55)) + geom\_hline(aes(yintercept = error), colour = "royalblue", linetype="dashed") + geom\_hline(aes(yintercept = -error), colour = "royalblue", linetype="dashed") + theme\_minimal() -> graf\_pacf\_total  
  
## Población sindicada  
# Calcular acf y pacf  
acf\_sindi <- as.data.frame(acf2(diff(diff(ts\_sindi), lag = 12), max.lag = 60))



# Gráfica acf sindicados  
acf\_sindi %>% ggplot() + geom\_bar(aes(x = c(1:60), y = ACF), stat = "identity", fill = "steelblue1") + scale\_x\_continuous(breaks = c(12,24,36,48,60)) + xlab("LAG") + ggtitle("ACF Población sindicada") + ylim(c(-0.55,0.55)) + geom\_hline(aes(yintercept = error), colour = "royalblue", linetype="dashed") + geom\_hline(aes(yintercept = -error), colour = "royalblue", linetype="dashed") + theme\_minimal() -> graf\_acf\_sindi  
# Gráfica pacf sindicados   
acf\_sindi %>% ggplot() + geom\_bar(aes(x = c(1:60), y = PACF), stat = "identity", fill = "steelblue1") + scale\_x\_continuous(breaks = c(12,24,36,48,60)) + xlab("LAG") + ggtitle("PACF Población sindicada") + ylim(c(-0.55,0.55)) + geom\_hline(aes(yintercept = error), colour = "royalblue", linetype="dashed") + geom\_hline(aes(yintercept = -error), colour = "royalblue", linetype="dashed") + theme\_minimal() -> graf\_pacf\_sindi  
  
## Población condenada  
# Calcular acf y pacf  
acf\_conde <- as.data.frame(acf2(diff(diff(ts\_conde), lag = 12), max.lag = 60))



# Gráfica acf condenados  
acf\_conde %>% ggplot() + geom\_bar(aes(x = c(1:60), y = ACF), stat = "identity", fill = "steelblue2") + scale\_x\_continuous(breaks = c(12,24,36,48,60)) + xlab("LAG") + ggtitle("ACF Población condenada") + ylim(c(-0.55,0.55)) + geom\_hline(aes(yintercept = error), colour = "royalblue", linetype="dashed") + geom\_hline(aes(yintercept = -error), colour = "royalblue", linetype="dashed") + theme\_minimal() -> graf\_acf\_conde  
  
# Gráfica pacf condenados  
acf\_conde %>% ggplot() + geom\_bar(aes(x = c(1:60), y = PACF), stat = "identity", fill = "steelblue2") + scale\_x\_continuous(breaks = c(12,24,36,48,60)) + xlab("LAG") + ggtitle("PACF Población condenada") + ylim(c(-0.55,0.55)) + geom\_hline(aes(yintercept = error), colour = "royalblue", linetype="dashed") + geom\_hline(aes(yintercept = -error), colour = "royalblue", linetype="dashed") + theme\_minimal() -> graf\_pacf\_conde  
  
#png(file = 'ACF\_var\_pob.png', height = 750, width = 500, res = 85)  
ACF\_var\_pob <- grid.arrange(graf\_acf\_total, graf\_pacf\_total,graf\_acf\_sindi, graf\_pacf\_sindi,graf\_acf\_conde, graf\_pacf\_conde, ncol = 2)



ACF\_var\_pob

## TableGrob (3 x 2) "arrange": 6 grobs  
## z cells name grob  
## 1 1 (1-1,1-1) arrange gtable[layout]  
## 2 2 (1-1,2-2) arrange gtable[layout]  
## 3 3 (2-2,1-1) arrange gtable[layout]  
## 4 4 (2-2,2-2) arrange gtable[layout]  
## 5 5 (3-3,1-1) arrange gtable[layout]  
## 6 6 (3-3,2-2) arrange gtable[layout]

#dev.copy(png,'ACF\_var\_pob')  
#dev.off()

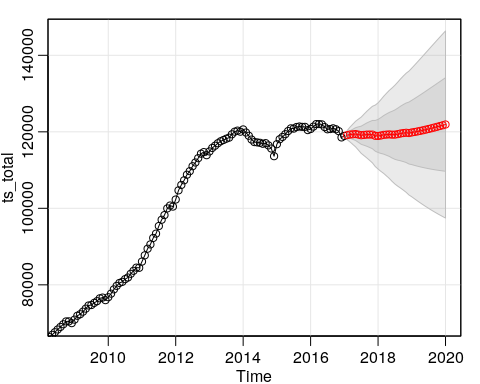
Las funciones de autocorrelación y correlación cruzada, de $Y\_t - Y\_{t - 1} $ muestran:

* La función de autocorrelación decae, y la función de autocorrelación parcial también decae, sugiriendo un ARMA.
* La función de autocorrelación se encuetra sobre las dos desviaciones estándar, sugiriendo que son significativas.
* La función de autocorrelación tiene un único pico en el periodo doce, la función de autocorrelación parcial decae lentamente, sugiriendo un MA(1)

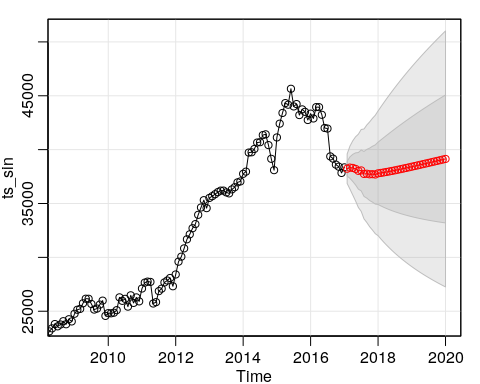
En estas condiciones podemos ajustar un ARIMA (1,1,1,0,0,1)

Todos los parámetros del modelo tienen un p\_value que sugiere que son significativos, excepto por la constante. Sin embargo, la autocorrelación del periodo dos sugiere que podemos estimar un ARIMA (1,1,1,0,0,2)

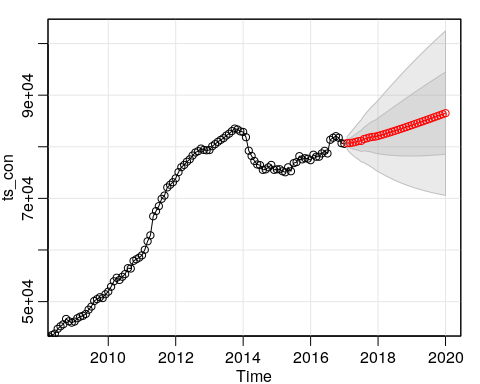
El nuevo término SA2 resulta siginicativo y presenta criterios de información de valor más pequeño, lo que implica que el modelo ARIMA (1,1,1,0,0,2) resulta más apropiado que el modelo ARIMA (1,1,1,0,0,1). Proyecciones de la población carcelaria del modelo ARIMA (1,1,1,0,0,2) se presentan a continuación:



Se presenta proyección de la población sindicada:



Se presenta proyección de la población condenada:



## Limitaciones de los modelos ARIMA en proyección de poblaciones carcelarias

Aunque es posible seguir analizando modelos ARIMA hasta encontrar el que presente mejores criterios de información, detendremos el analisis en este punto, para centrarnos en las limitaciones del enfoque.

1. La serie de población carcelarias incluye, por lo menos, dos series de tiempo con un comportamiento diferente: La población condenada y la población sindicada. Estos procesos, son en principio procesos autoregresivos, donde además se podría intuir el error no es independiente (como en la población total).
2. Los shocks en el sistema no son necesariamente estables en el tiempo. Las variaciones en los parámetros del modelo dependen de variables exógenas, como la cantidad de personas que ingresan al sistema y la dureza de las penas.
3. Los intervalos de confianza crecen rápidamente pues el modelo se ve afectado fuertemente por cambios en el nivel.
4. Puesto que el modelo supone unos parámetros fijos no da cuenta de la influencia de variables exógenas en el crecimiento de la población carcelaria.

# MODELOS VAR:

ts\_sincon <- cbind (diff(ts\_sin), diff(ts\_con))  
VARselect(ts\_sincon, lag.max=9, type="const")

## $selection  
## AIC(n) HQ(n) SC(n) FPE(n)   
## 4 1 1 4   
##   
## $criteria  
## 1 2 3 4 5  
## AIC(n) 2.568714e+01 2.567870e+01 2.567962e+01 2.564005e+01 2.566054e+01  
## HQ(n) 2.571656e+01 2.572774e+01 2.574827e+01 2.572831e+01 2.576841e+01  
## SC(n) 2.576068e+01 2.580127e+01 2.585121e+01 2.586067e+01 2.593018e+01  
## FPE(n) 1.431474e+11 1.419456e+11 1.420773e+11 1.365678e+11 1.393981e+11  
## 6 7 8 9  
## AIC(n) 2.566920e+01 2.567951e+01 2.568274e+01 2.568267e+01  
## HQ(n) 2.579669e+01 2.582662e+01 2.584946e+01 2.586900e+01  
## SC(n) 2.598787e+01 2.604721e+01 2.609946e+01 2.614842e+01  
## FPE(n) 1.406175e+11 1.420829e+11 1.425524e+11 1.425561e+11

ts\_sincon\_mod <- VAR(ts\_sincon,p = 1)  
forecast\_var <- predict(ts\_sincon\_mod, n.ahead = 60)  
ts\_sincon\_mod

##   
## VAR Estimation Results:  
## =======================   
##   
## Estimated coefficients for equation diff.ts\_sin.:   
## =================================================   
## Call:  
## diff.ts\_sin. = diff.ts\_sin..l1 + diff.ts\_con..l1 + const   
##   
## diff.ts\_sin..l1 diff.ts\_con..l1 const   
## 0.1103404 0.1373275 36.6481979   
##   
##   
## Estimated coefficients for equation diff.ts\_con.:   
## =================================================   
## Call:  
## diff.ts\_con. = diff.ts\_sin..l1 + diff.ts\_con..l1 + const   
##   
## diff.ts\_sin..l1 diff.ts\_con..l1 const   
## 0.2591926 0.3470605 117.6919129

forecast\_var

## $diff.ts\_sin.  
## fcst lower upper CI  
## [1,] 76.66558 -1281.091 1434.422 1357.757  
## [2,] 73.76214 -1293.174 1440.698 1366.936  
## [3,] 73.62323 -1295.151 1442.397 1368.774  
## [4,] 73.56753 -1295.580 1442.715 1369.148  
## [5,] 73.54243 -1295.682 1442.766 1369.224  
## [6,] 73.53110 -1295.708 1442.771 1369.240  
## [7,] 73.52599 -1295.717 1442.769 1369.243  
## [8,] 73.52368 -1295.720 1442.767 1369.243  
## [9,] 73.52264 -1295.721 1442.766 1369.244  
## [10,] 73.52217 -1295.721 1442.766 1369.244  
## [11,] 73.52195 -1295.722 1442.766 1369.244  
## [12,] 73.52186 -1295.722 1442.765 1369.244  
## [13,] 73.52182 -1295.722 1442.765 1369.244  
## [14,] 73.52180 -1295.722 1442.765 1369.244  
## [15,] 73.52179 -1295.722 1442.765 1369.244  
## [16,] 73.52178 -1295.722 1442.765 1369.244  
## [17,] 73.52178 -1295.722 1442.765 1369.244  
## [18,] 73.52178 -1295.722 1442.765 1369.244  
## [19,] 73.52178 -1295.722 1442.765 1369.244  
## [20,] 73.52178 -1295.722 1442.765 1369.244  
## [21,] 73.52178 -1295.722 1442.765 1369.244  
## [22,] 73.52178 -1295.722 1442.765 1369.244  
## [23,] 73.52178 -1295.722 1442.765 1369.244  
## [24,] 73.52178 -1295.722 1442.765 1369.244  
## [25,] 73.52178 -1295.722 1442.765 1369.244  
## [26,] 73.52178 -1295.722 1442.765 1369.244  
## [27,] 73.52178 -1295.722 1442.765 1369.244  
## [28,] 73.52178 -1295.722 1442.765 1369.244  
## [29,] 73.52178 -1295.722 1442.765 1369.244  
## [30,] 73.52178 -1295.722 1442.765 1369.244  
## [31,] 73.52178 -1295.722 1442.765 1369.244  
## [32,] 73.52178 -1295.722 1442.765 1369.244  
## [33,] 73.52178 -1295.722 1442.765 1369.244  
## [34,] 73.52178 -1295.722 1442.765 1369.244  
## [35,] 73.52178 -1295.722 1442.765 1369.244  
## [36,] 73.52178 -1295.722 1442.765 1369.244  
## [37,] 73.52178 -1295.722 1442.765 1369.244  
## [38,] 73.52178 -1295.722 1442.765 1369.244  
## [39,] 73.52178 -1295.722 1442.765 1369.244  
## [40,] 73.52178 -1295.722 1442.765 1369.244  
## [41,] 73.52178 -1295.722 1442.765 1369.244  
## [42,] 73.52178 -1295.722 1442.765 1369.244  
## [43,] 73.52178 -1295.722 1442.765 1369.244  
## [44,] 73.52178 -1295.722 1442.765 1369.244  
## [45,] 73.52178 -1295.722 1442.765 1369.244  
## [46,] 73.52178 -1295.722 1442.765 1369.244  
## [47,] 73.52178 -1295.722 1442.765 1369.244  
## [48,] 73.52178 -1295.722 1442.765 1369.244  
## [49,] 73.52178 -1295.722 1442.765 1369.244  
## [50,] 73.52178 -1295.722 1442.765 1369.244  
## [51,] 73.52178 -1295.722 1442.765 1369.244  
## [52,] 73.52178 -1295.722 1442.765 1369.244  
## [53,] 73.52178 -1295.722 1442.765 1369.244  
## [54,] 73.52178 -1295.722 1442.765 1369.244  
## [55,] 73.52178 -1295.722 1442.765 1369.244  
## [56,] 73.52178 -1295.722 1442.765 1369.244  
## [57,] 73.52178 -1295.722 1442.765 1369.244  
## [58,] 73.52178 -1295.722 1442.765 1369.244  
## [59,] 73.52178 -1295.722 1442.765 1369.244  
## [60,] 73.52178 -1295.722 1442.765 1369.244  
##   
## $diff.ts\_con.  
## fcst lower upper CI  
## [1,] 208.6590 -1007.066 1424.384 1215.725  
## [2,] 209.9804 -1066.589 1486.549 1276.569  
## [3,] 209.6864 -1078.969 1498.342 1288.655  
## [4,] 209.5484 -1081.556 1500.653 1291.104  
## [5,] 209.4860 -1082.117 1501.089 1291.603  
## [6,] 209.4579 -1082.247 1501.162 1291.704  
## [7,] 209.4452 -1082.280 1501.170 1291.725  
## [8,] 209.4395 -1082.290 1501.169 1291.729  
## [9,] 209.4369 -1082.293 1501.167 1291.730  
## [10,] 209.4357 -1082.295 1501.166 1291.730  
## [11,] 209.4352 -1082.295 1501.166 1291.730  
## [12,] 209.4349 -1082.296 1501.165 1291.730  
## [13,] 209.4348 -1082.296 1501.165 1291.730  
## [14,] 209.4348 -1082.296 1501.165 1291.730  
## [15,] 209.4348 -1082.296 1501.165 1291.730  
## [16,] 209.4348 -1082.296 1501.165 1291.730  
## [17,] 209.4347 -1082.296 1501.165 1291.730  
## [18,] 209.4347 -1082.296 1501.165 1291.730  
## [19,] 209.4347 -1082.296 1501.165 1291.730  
## [20,] 209.4347 -1082.296 1501.165 1291.730  
## [21,] 209.4347 -1082.296 1501.165 1291.730  
## [22,] 209.4347 -1082.296 1501.165 1291.730  
## [23,] 209.4347 -1082.296 1501.165 1291.730  
## [24,] 209.4347 -1082.296 1501.165 1291.730  
## [25,] 209.4347 -1082.296 1501.165 1291.730  
## [26,] 209.4347 -1082.296 1501.165 1291.730  
## [27,] 209.4347 -1082.296 1501.165 1291.730  
## [28,] 209.4347 -1082.296 1501.165 1291.730  
## [29,] 209.4347 -1082.296 1501.165 1291.730  
## [30,] 209.4347 -1082.296 1501.165 1291.730  
## [31,] 209.4347 -1082.296 1501.165 1291.730  
## [32,] 209.4347 -1082.296 1501.165 1291.730  
## [33,] 209.4347 -1082.296 1501.165 1291.730  
## [34,] 209.4347 -1082.296 1501.165 1291.730  
## [35,] 209.4347 -1082.296 1501.165 1291.730  
## [36,] 209.4347 -1082.296 1501.165 1291.730  
## [37,] 209.4347 -1082.296 1501.165 1291.730  
## [38,] 209.4347 -1082.296 1501.165 1291.730  
## [39,] 209.4347 -1082.296 1501.165 1291.730  
## [40,] 209.4347 -1082.296 1501.165 1291.730  
## [41,] 209.4347 -1082.296 1501.165 1291.730  
## [42,] 209.4347 -1082.296 1501.165 1291.730  
## [43,] 209.4347 -1082.296 1501.165 1291.730  
## [44,] 209.4347 -1082.296 1501.165 1291.730  
## [45,] 209.4347 -1082.296 1501.165 1291.730  
## [46,] 209.4347 -1082.296 1501.165 1291.730  
## [47,] 209.4347 -1082.296 1501.165 1291.730  
## [48,] 209.4347 -1082.296 1501.165 1291.730  
## [49,] 209.4347 -1082.296 1501.165 1291.730  
## [50,] 209.4347 -1082.296 1501.165 1291.730  
## [51,] 209.4347 -1082.296 1501.165 1291.730  
## [52,] 209.4347 -1082.296 1501.165 1291.730  
## [53,] 209.4347 -1082.296 1501.165 1291.730  
## [54,] 209.4347 -1082.296 1501.165 1291.730  
## [55,] 209.4347 -1082.296 1501.165 1291.730  
## [56,] 209.4347 -1082.296 1501.165 1291.730  
## [57,] 209.4347 -1082.296 1501.165 1291.730  
## [58,] 209.4347 -1082.296 1501.165 1291.730  
## [59,] 209.4347 -1082.296 1501.165 1291.730  
## [60,] 209.4347 -1082.296 1501.165 1291.730

# Cite packages   
if(nchar(system.file(package="astsa"))) citation("astsa")

##   
## To cite package 'astsa' in publications use:  
##   
## David Stoffer (2017). astsa: Applied Statistical Time Series  
## Analysis. R package version 1.8.  
## https://CRAN.R-project.org/package=astsa  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Manual{,  
## title = {astsa: Applied Statistical Time Series Analysis},  
## author = {David Stoffer},  
## year = {2017},  
## note = {R package version 1.8},  
## url = {https://CRAN.R-project.org/package=astsa},  
## }  
##   
## ATTENTION: This citation information has been auto-generated from  
## the package DESCRIPTION file and may need manual editing, see  
## 'help("citation")'.

# Documento tesis

### Total 111002  
png(file = 'Arima\_total\_111002.png', height = 750, width = 500, res = 85)  
arima\_total\_1 <- sarima(ts\_total,1,1,1,0,0,2, S = 12, details = FALSE)  
dev.off()

## png   
## 2

### Total 111001  
png(file = 'Arima\_total\_111001.png', height = 750, width = 500, res = 85)  
arima\_total\_2 <- sarima(ts\_total,1,1,1,0,0,1, S = 12, details = FALSE)  
dev.off()

## png   
## 2

auto.arima(ts\_total)

## Series: ts\_total   
## ARIMA(3,2,1)(0,0,2)[12]   
##   
## Coefficients:  
## ar1 ar2 ar3 ma1 sma1 sma2  
## 0.0520 -0.0864 -0.1553 -0.6756 0.2357 0.1908  
## s.e. 0.1034 0.0748 0.0712 0.0931 0.0575 0.0491  
##   
## sigma^2 estimated as 363346: log likelihood=-2430.23  
## AIC=4874.46 AICc=4874.83 BIC=4900.64

### Total Auto Arima  
png(file = 'Arima\_total\_321002.png', height = 750, width = 500, res = 85)  
arima\_total\_3 <- sarima(ts\_total,3,2,1,0,0,2, S = 12, details = FALSE)  
dev.off()

## png   
## 2

auto.arima(ts\_total, ic = "aic")

## Series: ts\_total   
## ARIMA(3,2,1)(0,0,2)[12]   
##   
## Coefficients:  
## ar1 ar2 ar3 ma1 sma1 sma2  
## 0.0520 -0.0864 -0.1553 -0.6756 0.2357 0.1908  
## s.e. 0.1034 0.0748 0.0712 0.0931 0.0575 0.0491  
##   
## sigma^2 estimated as 363346: log likelihood=-2430.23  
## AIC=4874.46 AICc=4874.83 BIC=4900.64

### Total 121002  
png(file = 'Arima\_total\_121002.png', height = 750, width = 500, res = 85)  
arima\_total\_4 <- sarima(ts\_total,1,2,1,0,0,2, S = 12, details = FALSE)  
dev.off()

## png   
## 2

Indicadores <- matrix(c(arima\_total\_1$AIC,arima\_total\_1$BIC,arima\_total\_2$AIC,arima\_total\_2$BIC,arima\_total\_3$AIC,arima\_total\_3$BIC,arima\_total\_4$AIC,arima\_total\_4$BIC),ncol = 2)  
  
AIC\_total <- AIC(arima\_total\_1$fit,arima\_total\_2$fit, arima\_total\_3$fit,arima\_total\_4$fit)

## Warning in AIC.default(arima\_total\_1$fit, arima\_total\_2$fit,  
## arima\_total\_3$fit, : models are not all fitted to the same number of  
## observations

BIC\_total <- BIC(arima\_total\_1$fit,arima\_total\_2$fit, arima\_total\_3$fit,arima\_total\_4$fit)

## Warning in BIC.default(arima\_total\_1$fit, arima\_total\_2$fit,  
## arima\_total\_3$fit, : models are not all fitted to the same number of  
## observations

Indicadores\_df <- cbind(c("(1,1,1,0,0,2)","(1,1,1,0,0,1)","(3,2,1,0,0,2)","(1,2,1,0,0,2)"),round(AIC\_total$AIC,2), round(BIC\_total$BIC,2))  
  
colnames(Indicadores\_df) <- c("ORDEN","AIC","BIC")  
#xtable(Indicadores\_df)  
#xtable(arima\_total\_4$ttable)  
#xtable(arima\_total\_4$ttable, label = "parámetros\_121002", caption = "Parámetros del modelo (1,2,1,0,0,2)")

### sindicado 111001  
png(file = 'Arima\_sindi\_11101.png', height = 750, width = 500, res = 85)  
arima\_sindi\_1 <- sarima(ts\_sindi,1,1,1,0,0,1, S = 12, details = FALSE)  
dev.off()

## png   
## 2

### sindicado 211001  
png(file = 'Arima\_sindi\_211001.png', height = 750, width = 500, res = 85)  
arima\_sindi\_2 <- sarima(ts\_sindi,2,1,1,0,0,1, S = 12, details = FALSE)  
dev.off()

## png   
## 2

auto.arima(ts\_sindi, ic = "aic")

## Series: ts\_sindi   
## ARIMA(0,1,2)(0,0,1)[12]   
##   
## Coefficients:  
## ma1 ma2 sma1  
## 0.0611 0.160 0.1547  
## s.e. 0.0572 0.053 0.0571  
##   
## sigma^2 estimated as 467153: log likelihood=-2477.86  
## AIC=4963.72 AICc=4963.85 BIC=4978.7

### sindicado Auto Arima  
png(file = 'Arima\_sindi\_012001.png', height = 750, width = 500, res = 85)  
arima\_sindi\_3 <- sarima(ts\_sindi,0,1,2,0,0,1, S = 12, details = FALSE)  
dev.off()

## png   
## 2

### Total 121002  
png(file = 'Arima\_sindi\_011000.png', height = 750, width = 500, res = 85)  
arima\_sindi\_4 <- sarima(ts\_sindi,0,1,1,0,0,0, S = 12, details = FALSE)  
dev.off()

## png   
## 2

AIC\_sindi <- AIC(arima\_sindi\_1$fit,arima\_sindi\_2$fit, arima\_sindi\_3$fit,arima\_sindi\_4$fit)  
BIC\_sindi <- BIC(arima\_sindi\_1$fit,arima\_sindi\_2$fit, arima\_sindi\_3$fit,arima\_sindi\_4$fit)  
  
Indicadores\_df\_sindi <- cbind(c("(1,1,1,0,0,1)","(2,1,1,0,0,1)","(0,1,2,0,0,1)","(0,1,1,0,0,0)"),round(AIC\_sindi$AIC,2), round(BIC\_sindi$BIC,2))  
  
colnames(Indicadores\_df\_sindi) <- c("ORDEN","AIC","BIC")  
xtable(Indicadores\_df\_sindi)

## % latex table generated in R 3.5.1 by xtable 1.8-3 package  
## % Fri Nov 9 21:14:04 2018  
## \begin{table}[ht]  
## \centering  
## \begin{tabular}{rlll}  
## \hline  
## & ORDEN & AIC & BIC \\   
## \hline  
## 1 & (1,1,1,0,0,1) & 4968.8 & 4987.52 \\   
## 2 & (2,1,1,0,0,1) & 4964.1 & 4986.56 \\   
## 3 & (0,1,2,0,0,1) & 4964.26 & 4982.98 \\   
## 4 & (0,1,1,0,0,0) & 4974.16 & 4985.39 \\   
## \hline  
## \end{tabular}  
## \end{table}

#xtable(arima\_total\_4$ttable)  
xtable(arima\_sindi\_3$ttable, label = "parámetros\_sindi\_011001", caption = "Parámetros del modelo (0,1,1,0,0,1)")

## % latex table generated in R 3.5.1 by xtable 1.8-3 package  
## % Fri Nov 9 21:14:04 2018  
## \begin{table}[ht]  
## \centering  
## \begin{tabular}{rrrrr}  
## \hline  
## & Estimate & SE & t.value & p.value \\   
## \hline  
## ma1 & 0.06 & 0.06 & 0.99 & 0.32 \\   
## ma2 & 0.16 & 0.05 & 2.91 & 0.00 \\   
## sma1 & 0.15 & 0.06 & 2.57 & 0.01 \\   
## constant & 64.94 & 53.21 & 1.22 & 0.22 \\   
## \hline  
## \end{tabular}  
## \caption{Parámetros del modelo (0,1,1,0,0,1)}   
## \label{parámetros\_sindi\_011001}  
## \end{table}

## Población condenada

### condenada 111001  
png(file = 'Arima\_conde\_111001.png', height = 750, width = 500, res = 85)  
arima\_conde\_1 <- sarima(ts\_conde,1,1,1,0,0,1, S = 12, details = FALSE)  
dev.off()

## png   
## 2

### condenado 211001  
png(file = 'Arima\_conde\_211001.png', height = 750, width = 500, res = 85)  
arima\_conde\_2 <- sarima(ts\_conde,2,1,1,0,0,1, S = 12, details = FALSE)  
dev.off()

## png   
## 2

auto.arima(ts\_conde, ic = "aic")

## Series: ts\_conde   
## ARIMA(1,1,1)(0,0,1)[12] with drift   
##   
## Coefficients:  
## ar1 ma1 sma1 drift  
## 0.8823 -0.7293 0.0935 198.6161  
## s.e. 0.0641 0.0935 0.0604 86.2840  
##   
## sigma^2 estimated as 384640: log likelihood=-2447.04  
## AIC=4904.08 AICc=4904.27 BIC=4922.79

### condenado Auto Arima  
png(file = 'Arima\_conde\_012001.png', height = 750, width = 500, res = 85)  
arima\_conde\_3 <- sarima(ts\_conde,0,1,2,0,0,1, S = 12, details = FALSE)  
dev.off()

## png   
## 2

### Total 121002  
png(file = 'Arima\_conde\_011000.png', height = 750, width = 500, res = 85)  
arima\_conde\_4 <- sarima(ts\_conde,0,1,1,0,0,0, S = 12, details = FALSE)  
dev.off()

## png   
## 2

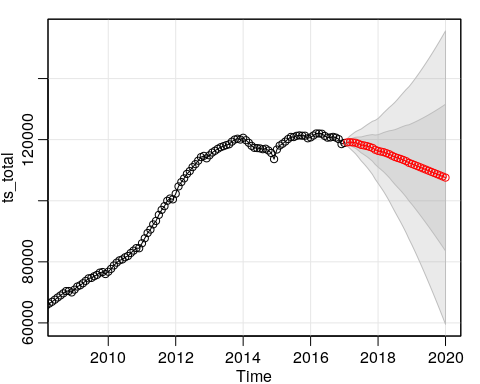
AIC\_conde <- AIC(arima\_conde\_1$fit,arima\_conde\_2$fit, arima\_conde\_3$fit,arima\_conde\_4$fit)  
BIC\_conde <- BIC(arima\_conde\_1$fit,arima\_conde\_2$fit, arima\_conde\_3$fit,arima\_conde\_4$fit)  
  
Indicadores\_df\_conde <- cbind(c("(1,1,1,0,0,1)","(2,1,1,0,0,1)","(0,1,1,0,0,1)","(0,1,1,0,0,0)"),round(AIC\_conde$AIC,2), round(BIC\_conde$BIC,2))  
  
colnames(Indicadores\_df\_conde) <- c("ORDEN","AIC","BIC")  
xtable(Indicadores\_df\_conde)

## % latex table generated in R 3.5.1 by xtable 1.8-3 package  
## % Fri Nov 9 21:14:04 2018  
## \begin{table}[ht]  
## \centering  
## \begin{tabular}{rlll}  
## \hline  
## & ORDEN & AIC & BIC \\   
## \hline  
## 1 & (1,1,1,0,0,1) & 4904.08 & 4922.79 \\   
## 2 & (2,1,1,0,0,1) & 4906 & 4928.46 \\   
## 3 & (0,1,1,0,0,1) & 4914.84 & 4933.55 \\   
## 4 & (0,1,1,0,0,0) & 4925.62 & 4936.85 \\   
## \hline  
## \end{tabular}  
## \end{table}

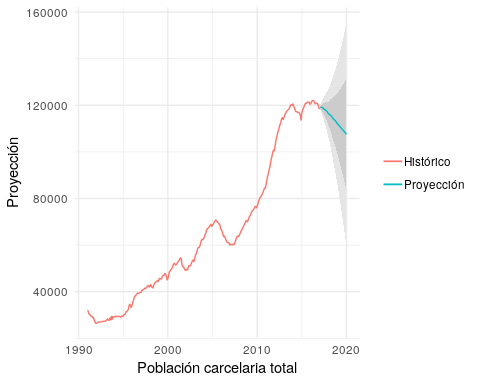
#xtable(arima\_total\_4$ttable)  
xtable(arima\_conde\_1$ttable, label = "parámetros\_conde\_111001", caption = "Parámetros del modelo (1,1,1,0,0,1)")

## % latex table generated in R 3.5.1 by xtable 1.8-3 package  
## % Fri Nov 9 21:14:04 2018  
## \begin{table}[ht]  
## \centering  
## \begin{tabular}{rrrrr}  
## \hline  
## & Estimate & SE & t.value & p.value \\   
## \hline  
## ar1 & 0.88 & 0.06 & 13.75 & 0.00 \\   
## ma1 & -0.73 & 0.09 & -7.80 & 0.00 \\   
## sma1 & 0.09 & 0.06 & 1.55 & 0.12 \\   
## constant & 198.62 & 86.28 & 2.30 & 0.02 \\   
## \hline  
## \end{tabular}  
## \caption{Parámetros del modelo (1,1,1,0,0,1)}   
## \label{parámetros\_conde\_111001}  
## \end{table}

# Proyección  
forecast <- sarima.for(ts\_total,n.ahead = 36, 1,2,1,0,0,2, S = 12)



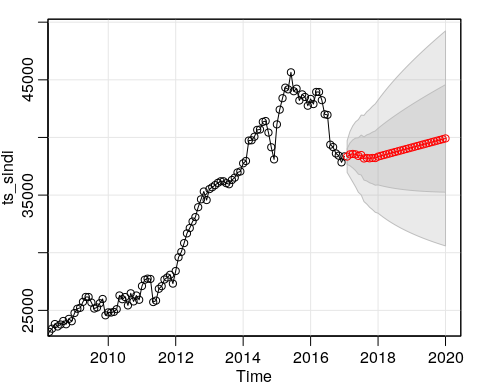
# data frame  
forecast\_frame <- data.frame(Proyección=as.matrix(forecast$pred),Error=as.matrix(forecast$se), date=as.Date(as.yearmon(time(forecast$pred)))) %>% mutate(lim\_sup = Proyección+Error, lim\_inf = Proyección - Error)  
#data histórica  
hist\_frame <- data.frame(Historico=as.matrix(ts\_total), date=as.Date(as.yearmon(time(ts\_total))))  
# ggplot  
ggplot() + geom\_ribbon(data = forecast\_frame, aes(x = date, ymin = Proyección - 2\*Error, ymax = Proyección+2\*Error, fill = "2 sd" ), fill = "grey90") + geom\_ribbon(data = forecast\_frame, aes(x = date, ymin = Proyección -Error, ymax = Proyección+Error,fill = "1 sd" ), fill = "grey80") + geom\_line(data = forecast\_frame, aes(x = date, y = Proyección, colour = "Proyección")) + geom\_line(data = hist\_frame, aes(x = date, y = Historico, colour = "Histórico")) + xlab("Población carcelaria total") + theme\_minimal() + theme(legend.title=element\_blank()) -> graf\_pry\_total\_4  
graf\_pry\_total\_4



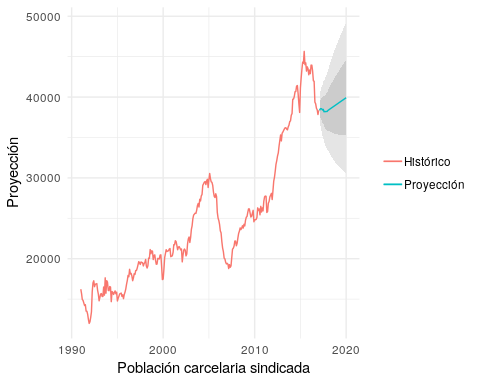
ggsave("graf\_pry\_total\_4.png")

## Saving 5 x 4 in image

# Proyección  
forecast <- sarima.for(ts\_sindi,n.ahead = 36, 0,1,1,0,0,1, S = 12)



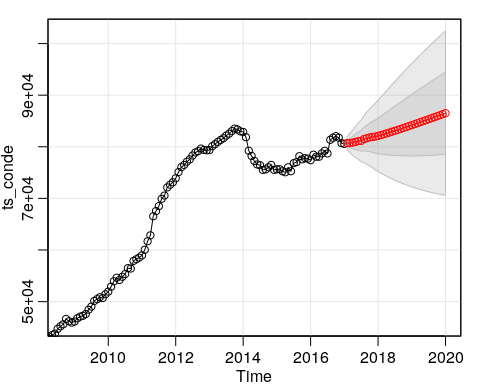
# data frame  
forecast\_frame <- data.frame(Proyección=as.matrix(forecast$pred),Error=as.matrix(forecast$se), date=as.Date(as.yearmon(time(forecast$pred)))) %>% mutate(lim\_sup = Proyección+Error, lim\_inf = Proyección - Error)  
#data histórica  
hist\_frame <- data.frame(Historico=as.matrix(ts\_sindi), date=as.Date(as.yearmon(time(ts\_sindi))))  
# ggplot  
ggplot() + geom\_ribbon(data = forecast\_frame, aes(x = date, ymin = Proyección - 2\*Error, ymax = Proyección+2\*Error, fill = "2 sd" ), fill = "grey90") + geom\_ribbon(data = forecast\_frame, aes(x = date, ymin = Proyección -Error, ymax = Proyección+Error,fill = "1 sd" ), fill = "grey80") + geom\_line(data = forecast\_frame, aes(x = date, y = Proyección, colour = "Proyección")) + geom\_line(data = hist\_frame, aes(x = date, y = Historico, colour = "Histórico")) + xlab("Población carcelaria sindicada") + theme\_minimal() + theme(legend.title=element\_blank()) -> graf\_pry\_total\_3  
graf\_pry\_total\_3



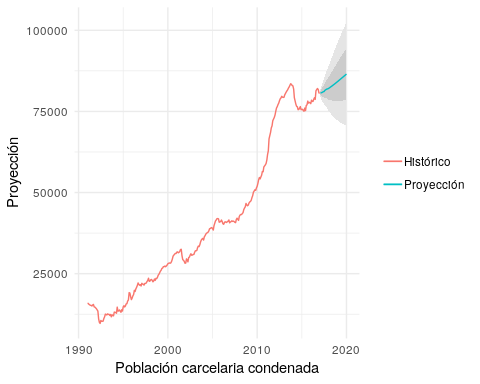
ggsave("graf\_pry\_sindi\_3.png")

## Saving 5 x 4 in image

# Proyección  
forecast <- sarima.for(ts\_conde,n.ahead = 36, 1,1,1,0,0,1, S = 12)



# data frame  
forecast\_frame <- data.frame(Proyección=as.matrix(forecast$pred),Error=as.matrix(forecast$se), date=as.Date(as.yearmon(time(forecast$pred)))) %>% mutate(lim\_sup = Proyección+Error, lim\_inf = Proyección - Error)  
#data histórica  
hist\_frame <- data.frame(Historico=as.matrix(ts\_conde), date=as.Date(as.yearmon(time(ts\_conde))))  
# ggplot  
ggplot() + geom\_ribbon(data = forecast\_frame, aes(x = date, ymin = Proyección - 2\*Error, ymax = Proyección+2\*Error, fill = "2 sd" ), fill = "grey90") + geom\_ribbon(data = forecast\_frame, aes(x = date, ymin = Proyección -Error, ymax = Proyección+Error,fill = "1 sd" ), fill = "grey80") + geom\_line(data = forecast\_frame, aes(x = date, y = Proyección, colour = "Proyección")) + geom\_line(data = hist\_frame, aes(x = date, y = Historico, colour = "Histórico")) + xlab("Población carcelaria condenada") + theme\_minimal()+ theme(legend.title=element\_blank()) -> graf\_pry\_conde\_1  
graf\_pry\_conde\_1



ggsave("graf\_pry\_conde\_1.png")

## Saving 5 x 4 in image

# Método de ratio Correlation.

# url <- "https://www.dropbox.com/s/94oblxtlc8anlk1/PRY\_POB\_NAL\_DANE\_EDAD.csv?dl=1"  
# destfile <- "POB\_NAL\_DAN\_EDAD.csv"  
# curl\_download(url, destfile)

# pob\_nal\_dane\_edad = read.csv("POB\_NAL\_DAN\_EDAD.csv",header = TRUE,sep = ",",stringsAsFactors=FALSE)  
#   
# pob\_nal\_dane\_edad %<>% gather(Rango\_edad, Poblacion, -Año, -Grupos.de.edad) %>% spread(Grupos.de.edad, Poblacion)  
#   
# pob\_nal\_dane\_edad %<>% dplyr::filter(Año>1990)  
#   
# pob\_nal\_dane\_edad$Rango\_edad<- gsub("X","",pob\_nal\_dane\_edad$Rango\_edad)  
# pob\_nal\_dane\_edad$Rango\_edad<- gsub("\\.","-",pob\_nal\_dane\_edad$Rango\_edad)  
# pob\_nal\_dane\_edad$Rango\_edad<- gsub("0-4","00-04",pob\_nal\_dane\_edad$Rango\_edad)  
# pob\_nal\_dane\_edad$Rango\_edad<- gsub("400-044","40-44",pob\_nal\_dane\_edad$Rango\_edad)  
# pob\_nal\_dane\_edad$Rango\_edad<- gsub("5-9","05-09",pob\_nal\_dane\_edad$Rango\_edad)  
#   
# pob\_nal\_dane\_edad %>% ggplot(aes(x=Año, y = Total/1000000, fill = Rango\_edad)) + geom\_area(stat="identity") + theme\_minimal() + ylab("Proyecciones de población nacional (MM)") + scale\_fill\_manual(values = colorRampPalette(brewer.pal(12, "Paired"))(17)) + theme(legend.position="none") -> pob\_total\_edad\_stack  
#   
# pob\_nal\_dane\_edad %>% ggplot(aes(x=Año, y = Total/1000000, colour = Rango\_edad)) + geom\_line() + theme\_minimal() + ylab("Proyecciones de población nacional (MM)") + scale\_colour\_manual(values = colorRampPalette(brewer.pal(12, "Paired"))(17)) + theme(legend.position="none") -> pob\_total\_edad  
#   
# ### Crear columna de participación   
#   
# pob\_nal\_dane\_edad %<>% group\_by(Año) %>% summarise(Totales\_año = sum(Total)) %>% right\_join(pob\_nal\_dane\_edad, by = NULL) %>% mutate (Participacion= Total/Totales\_año\*100)   
#   
# pob\_nal\_dane\_edad %>% ggplot(aes(x=Año, y = Participacion, fill = Rango\_edad)) + geom\_area(stat="identity") + theme\_minimal() + ylab("Proyecciones de población nacional (%)") + scale\_fill\_manual(values = colorRampPalette(brewer.pal(12, "Paired"))(17)) -> pob\_total\_edad\_stack\_part  
#   
# pob\_nal\_dane\_edad %>% ggplot(aes(x=Año, y = Participacion, colour = Rango\_edad)) + geom\_line() + theme\_minimal() + ylab("Proyecciones de población nacional (%)") + scale\_colour\_manual(values = colorRampPalette(brewer.pal(12, "Paired"))(17)) -> pob\_total\_edad\_part  
#   
# # Generar gráfica  
# #png(file = 'POB\_NAL\_EDAD.png', height = 850, width = 600, res = 85)  
# POB\_NAL\_EDAD <- grid.arrange(pob\_total\_edad\_stack, pob\_total\_edad\_stack\_part,pob\_total\_edad, pob\_total\_edad\_part, ncol = 2)  
# POB\_NAL\_EDAD  
# #dev.off()

# # Cargar datos   
# pob\_nal\_dane\_edad = read.csv("POB\_NAL\_DAN\_EDAD.csv",header = TRUE,sep = ",",stringsAsFactors=FALSE)  
#   
# # cambiar estructura donde el genero es columna y los gruposde edad es fila  
# pob\_nal\_dane\_edad %<>% gather(Rango\_edad, Poblacion, -Año, -Grupos.de.edad) %>% spread(Grupos.de.edad, Poblacion)  
#   
# pob\_nal\_dane\_edad %<>% dplyr::filter(Año>1990)  
#   
# # Corregir titulos de rango etario  
# pob\_nal\_dane\_edad$Rango\_edad<- gsub("X","",pob\_nal\_dane\_edad$Rango\_edad)  
# pob\_nal\_dane\_edad$Rango\_edad<- gsub("\\.","-",pob\_nal\_dane\_edad$Rango\_edad)  
# pob\_nal\_dane\_edad$Rango\_edad<- gsub("0-4","00-04",pob\_nal\_dane\_edad$Rango\_edad)  
# pob\_nal\_dane\_edad$Rango\_edad<- gsub("400-044","40-44",pob\_nal\_dane\_edad$Rango\_edad)  
# pob\_nal\_dane\_edad$Rango\_edad<- gsub("5-9","05-09",pob\_nal\_dane\_edad$Rango\_edad)  
#   
# # Población DANE por edad  
# pob\_nal\_dane\_edad %>% dplyr::select(Año, Rango\_edad, Total) %>% tidyr::spread(Rango\_edad, Total) -> pob\_dane\_edad  
#   
# # Acumular rangos etarios que se trabajan acumulados  
# Menor\_14 <-apply(pob\_dane\_edad[,2:4],1,sum)  
# Mayor\_60 <- apply(pob\_dane\_edad[,16:18],1,sum)  
#   
# # Crear arreglo de regresores  
# Regresores <- cbind(Menor\_14,pob\_dane\_edad[,5:15],Mayor\_60)  
#   
# # Calcular población total  
# Pob\_total <- apply(Regresores,1,sum)  
#   
# # Población total en mayo de cada año  
# ppl\_sitjur %>% dplyr::filter(categoria =="total", mes == 5, !is.na(valor)) %>% dplyr::select(valor, anio) -> pob\_inpec\_total  
#   
# # Calcular tasa de encarcelamiento  
# registros <- dim(pob\_inpec\_total)[1]  
# tasa\_encar <- pob\_inpec\_total[,1]/Pob\_total[1:registros]  
#   
#   
#   
# # Guardar modelo de regresión  
# tasa\_regresores <- cbind(tasa\_encar = pob\_inpec\_total[,1],Regresores[1:registros,])   
#   
# # Salvar  
# write.table(tasa\_regresores,"tasa\_regresores.csv",sep=",",row.names = FALSE)

# tasa\_regresores <- read.table("tasa\_regresores.csv", sep = ",", header = TRUE)  
#   
# ddist <- datadist(tasa\_regresores)  
# options(datadst = "ddist")  
#   
# reg <- lm(tasa\_encar ~ -1 + X15.19 + X20.24 + X20.24 + X25.29 + X30.34 + X35.39 + X40.44 + X45.49 + X50.54 + X55.59 + Mayor\_60, data = tasa\_regresores)  
# summary(reg)  
#   
# step(reg,direction="backward")