Chile Labor Force-Real World Time Series Analysis

Oguzhan Gurbuz

2023-01-30

#The labor force participation rate can be defined as “an estimate of an economy’s active workforce” (Investopedia). The formula is the number of people ages 16 and older who are employed or actively seeking employment, divided by the total non-institutionalized, civilian working-age population. #It is important to understand this rate since it yields a strong clue about the country’s economy.

#Labor Force Participation Rate = (Number Employed+Number Seeking Work) × 100 / Civilian Non-Institutional Population

#In this time series analysis, we’re going to analyse the labor force participation rate trend of a developing country in the Global South, Chile, having a population of 19,5 million according to the World Bank. The main goal of this analysis is to see the current pattern of the country’s labour force partication rate and forecast the next 6 years.

#The data I collected from the World Bank has a time period of 30 years between 1990 and 2019. We’re going to forecast the value for the next 6 years between 2020 and 2025. This will be possible by using three forecast methods, namely, the Box-Jenkins or ARIMA (Autoregressive Integrated Moving Average), Holt’s linear trend and Holt-Winter damped methods.

```
#Let's install the necessary packages
#install.packages("fpp")
#install.packages("forecast")
#install.packages("ggplot2")

#Let's load the necessary packages.
library(readxl)
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(ggplot2)
library(fpp)
```

```
## Loading required package: fma
```

```
## Loading required package: expsmooth
```

```
## Loading required package: lmtest
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: tseries
```

#PREPARING AND EXPLORING THE DATA

```
#First, we need to load the dataset and assign to "chile_data".
chile_data <- read_excel("Chile Labor Force Participation Rate.xlsx")
```

```
#What's the class of the Chile data set.
class(chile_data) #It's a data frame.
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

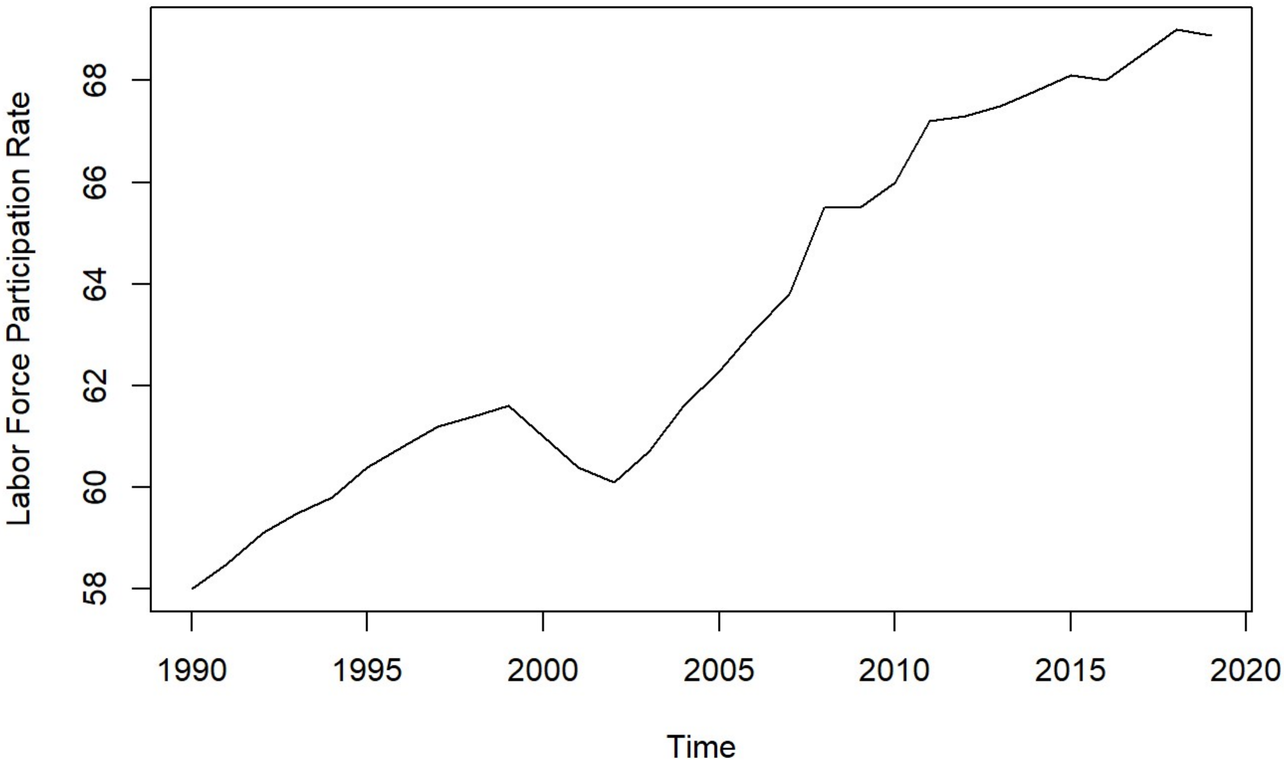
```
#Let's add a name to top of the data
names(chile_data) <- c("Labor Force Participation Rate")
chile_ts <- ts(chile_data, start = 1990,end = 2019,frequency = 1)

#Let's see a few observations from the dataset.
head(chile_ts)
```

```
## Time Series:
## Start = 1990
## End = 1995
## Frequency = 1
##      Labor Force Participation Rate
## [1,]                           58.0
## [2,]                           58.5
## [3,]                           59.1
## [4,]                           59.5
## [5,]                           59.8
## [6,]                           60.4
```

```
#Summary of the data
summary(chile_data)
```

```
##  Labor Force Participation Rate
##  Min.   :58.00
##  1st Qu.:60.48
##  Median :61.95
##  Mean   :63.42
##  3rd Qu.:67.28
##  Max.   :69.00
```

```
#We can plot the data and see how it increased in time.
plot(chile_ts)
```



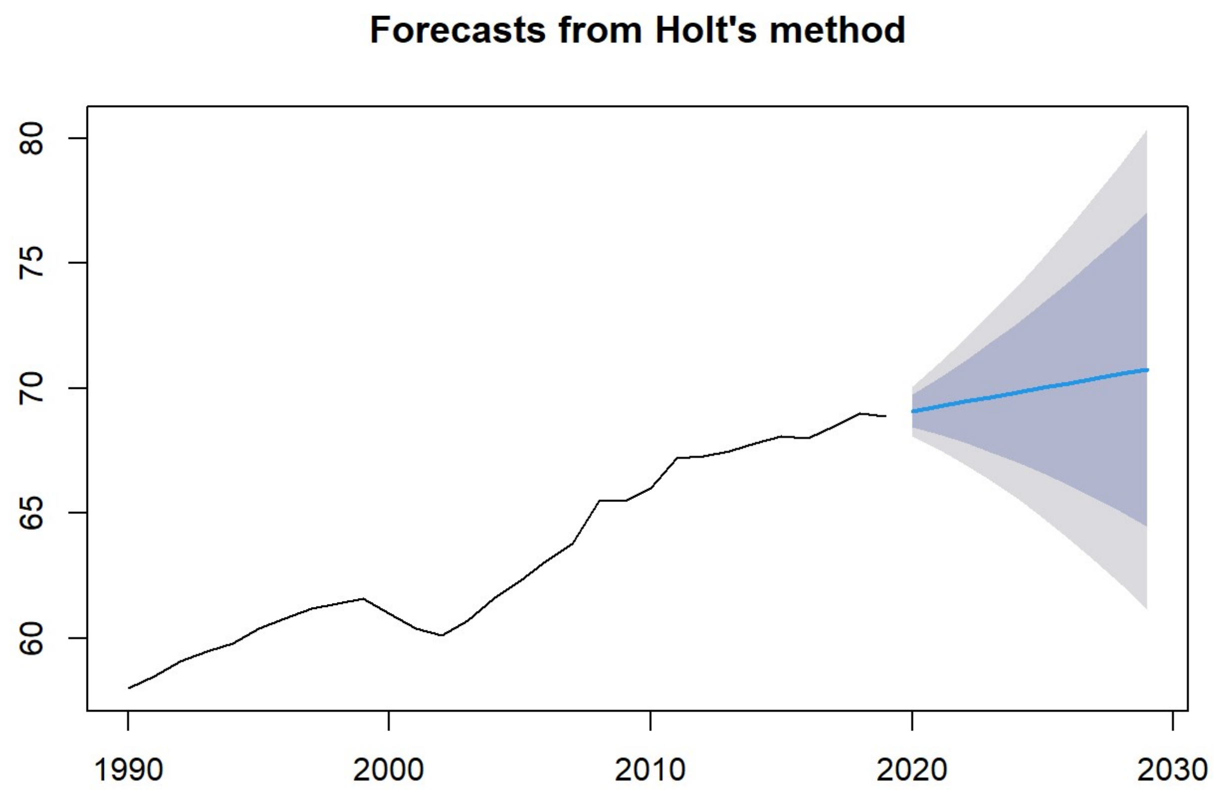

```
#Based on the plot, we can say the labor force participation rate data set seems trending,
but apparently, there is no seasonality. That means the Holt function for a linear trend m
odel will be a good one to forecast.
```

#Holt Linear and Holt-Winter's Models

```
#Let's set up a linear trend model for the Chile labor participation time series. We can u
se the holt() function, being a component of the forecast package. The argument h is used
to specify the forecast length. For the first observation, let's go with 10 (h = 10). In o
ther words, that is a forecast of 10 years into the future.

# Exponential smoothing with holt-trend

holt.linear = holt(chile_ts, h = 10)

summary(holt.linear)
```

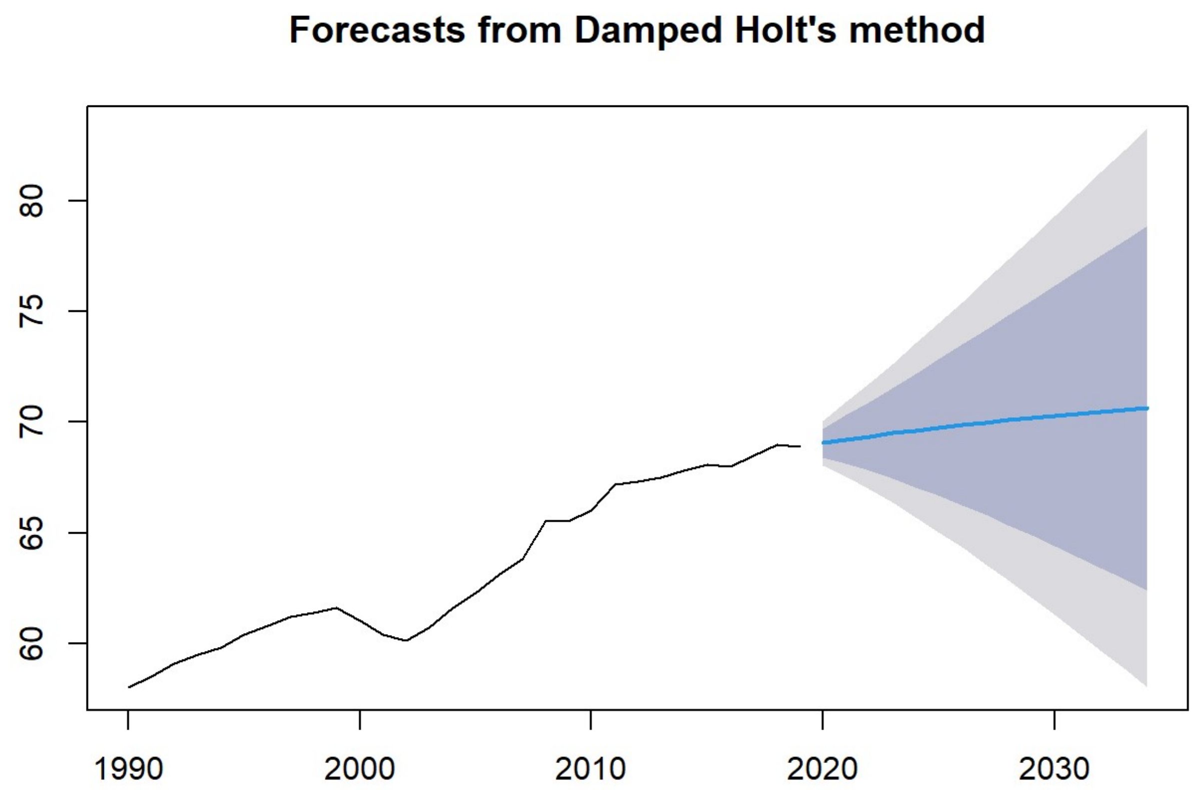
```
##
## Forecast method: Holt's method
##
## Model Information:
## Holt's method
##
## Call:
##  holt(y = chile_ts, h = 10)
##
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 0.4021
##
##   Initial states:
##     l = 57.5226
##     b = 0.4805
##
##   sigma:  0.5106
##
##      AIC     AICc      BIC
## 67.41627 69.91627 74.42226
##
## Error measures:
##                       ME      RMSE       MAE         MPE      MAPE      MASE
## Training set -0.02438653 0.4753705 0.3501402 -0.03513517 0.5468293 0.7100745
##                    ACF1
## Training set 0.05602751
##
## Forecasts:
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2020       69.08635 68.43195 69.74075 68.08553 70.08717
## 2021       69.27265 68.14573 70.39958 67.54917 70.99614
## 2022       69.45895 67.82686 71.09105 66.96288 71.95503
## 2023       69.64526 67.46626 71.82425 66.31277 72.97774
## 2024       69.83156 67.06366 72.59946 65.59842 74.06469
## 2025       70.01786 66.62051 73.41521 64.82207 75.21365
## 2026       70.20416 66.13863 74.26970 63.98646 76.42186
## 2027       70.39046 65.61976 75.16116 63.09431 77.68662
## 2028       70.57677 65.06556 76.08797 62.14810 79.00543
## 2029       70.76307 64.47748 77.04865 61.15010 80.37604
```

```
#When we look at the values from the summary, we get the values for the alpha and beta par
ameters, alpha is 0.999 and beta is 0.402. This shows the trend (the slope of the time ser
ious plot) is fairly constant throughout. Here, alpha and beta needs further investigatio
n:

#Alpha(α): The level component smoothing parameter is Alpha specifying the weight attribut
ed to the most recent data in the prediction (the year of 2019). A value close to 1 highli
ghts the most recent observation, whereas a value close to 0 highlights past observations.
In this case, alpha is 0.9999, suggesting that the model gives the most recent observation
virtually full weight.

#The Beta (β) smoothing parameter for the slope component changes the weight assigned to t
he most recent forecast error in the prediction. A value close to 1 emphasizes the most re
cent prediction mistake, whereas a value close to 0 emphasizes previous forecast errors. I
n this case, beta is 0.4021, showing the model prioritizes earlier forecast errors above t
he most recent forecast error.

#Let's see the plot of the Holt linear model to visualize the pattern.
plot(holt.linear)
```



```
#The line in blue is the forecasted ten years 2020 to 2029. And the light and dark shaded
zones are the confidence intervals with the 80 and 95% confidence. We can also see that th
e slope of the forecasted period is nearly the same as with the period between 2010 and 20
20 and between 1990 and 2000. This simply means the model takes a trend slope of at least
the last 10 years.

#Of course, the level value is added to this slope to get the corresponding forecasted val
ue. But let's go back to the summary output. Under initial states, we get the values for t
he slope and the level value. These values are used together with the smoothing parameters
in the initial level and slope equations. To calculate the first forecasted observation, w
e get information criteria for the model as well as arrow measures, and we get the forecas
t points and the confidence intervals which are used in the plot.

#One of the problems in the data is the trend. It is not realistic and the trend cannot co
ntinue indefinitely. I mean, for a country's labor force participation rate, it's not poss
ible to converge to the level of 100%. There will always be people unwilling to participat
e in the labor market. That means the curve will likely flatten out in the range of 90%.

#We can do this with the Holt function using the damped argument. A damped Holt linear tre
nd model assumes that the trend cannot be constant forever. Using this model, we suggest g
rowth needs to come to an end at some point. The curve needs to flatten out. Here the damp
ing parameter phi(φ) comes into play. Let's first have a look at the formula:
# Forecasting formula = yt+h|t = l + hbt
# Damped forecasting formula = yt+h|t = l + (φ + φ^2 + φ^3 + ... + φ^h)bt
# 0 < Damping parameter phi(φ) < 1

#If phi(φ)  is one, it is the same as a standard Holt linear model, whereas when phi is cl
ose to 0, the curve gets flat fairly soon. In practice, the parameter phi(φ) is set somewh
ere between 0.8 and 0.98. Using a phi(φ) at around 0.8, we make sure that short run foreca
sts still contain the trend, whereas the longer forecast legs are at a flat a curve, we ca
n easily adjust our whole model with a damping parameter.

#Let's also have a look at the values, l and b. The initial level (l = 57.5226) and initia
l slope (b = 0.4805) are the starting values that the Holt's method uses to generate its f
orecast. The accuracy of the forecast is influenced by the initial level and slope. These
values are estimated based on the historical data and are used to calculate the first fore
casted value. The level refers to the baseline value or average value of the series, while
the slope represents the rate of change over time. The level and slope values determine th
e overall shape and direction of the forecast. If the level is high, the forecasted values
will tend to be higher, while a low level will result in lower forecasted values. The slop
e determines the rate of change of the series over time. A positive slope means that the s
eries is increasing, while a negative slope means that it is decreasing.
```

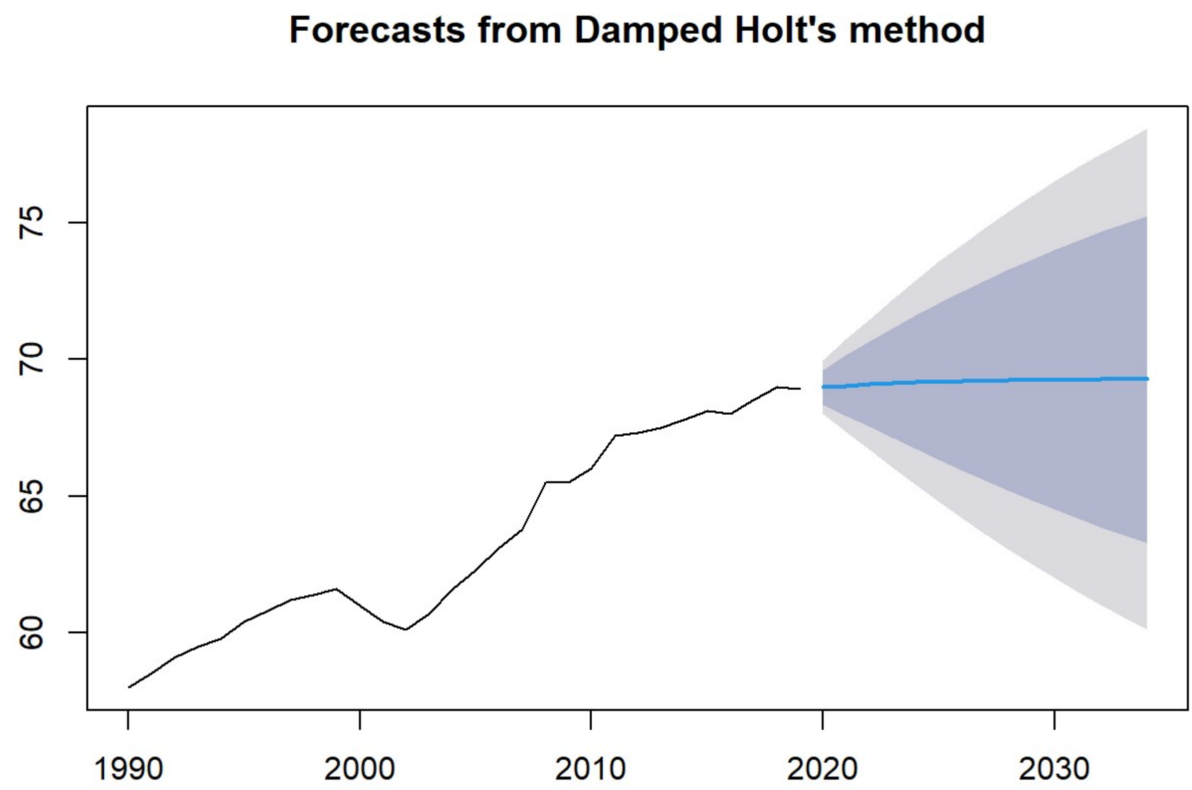
```
#To show such a Holt model with a damping parameter, I will directly plot the Holt model w
ith h=15. So that the flat or damped curve gets optimally visualized. The code itself is t
he same as with a standard Holt model. However, you need to set damped to "TRUE" or "T" th
at we are just going to select the best suitable parameter phi(φ) for you.

# Phi(φ) auto generated
plot(holt(chile_ts, h = 15, damped = T))
```




```
# To see the generated value for phi(φ)
summary(holt(chile_ts, h = 15, damped = T))
```

```
##
## Forecast method: Damped Holt's method
##
## Model Information:
## Damped Holt's method
##
## Call:
##  holt(y = chile_ts, h = 15, damped = T)
##
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 0.3769
##     phi   = 0.9507
##
##   Initial states:
##     l = 57.3834
##     b = 0.3774
##
##   sigma:  0.512
##
##      AIC     AICc      BIC
## 68.39650 72.04868 76.80369
##
## Error measures:
##                      ME      RMSE       MAE        MPE      MAPE     MASE
## Training set 0.03037541 0.4673593 0.3463455 0.05159627 0.5428329 0.702379
##                    ACF1
## Training set 0.05511592
##
## Forecasts:
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2020       69.06198 68.40587 69.71809 68.05855 70.06542
## 2021       69.21593 68.10933 70.32253 67.52353 70.90833
## 2022       69.36228 67.79165 70.93290 66.96022 71.76434
## 2023       69.50141 67.44523 71.55758 66.35676 72.64605
## 2024       69.63367 67.07132 72.19602 65.71489 73.55244
## 2025       69.75940 66.67276 72.84605 65.03879 74.48002
## 2026       69.87893 66.25255 73.50531 64.33286 75.42500
## 2027       69.99256 65.81350 74.17162 63.60124 76.38388
## 2028       70.10059 65.35814 74.84304 62.84763 77.35354
## 2029       70.20328 64.88869 75.51787 62.07532 78.33124
## 2030       70.30091 64.40716 76.19465 61.28720 79.31461
## 2031       70.39372 63.91530 76.87213 60.48584 80.30159
## 2032       70.48194 63.41468 77.54921 59.67350 81.29039
## 2033       70.56582 62.90668 78.22496 58.85218 82.27946
## 2034       70.64555 62.39253 78.89857 58.02365 83.26746
```

```
# Here the phi = 0.9507: It is close to 1, meaning the damping component is highly persist
ent. This implies that the trend component may change slowly over time. Let's change it ma
nually to 0.8.

# Manual setting of phi
plot(holt(chile_ts, h = 15, damped = T, phi = 0.8))
```



#Simple Exponential Smoothing Method (SES)

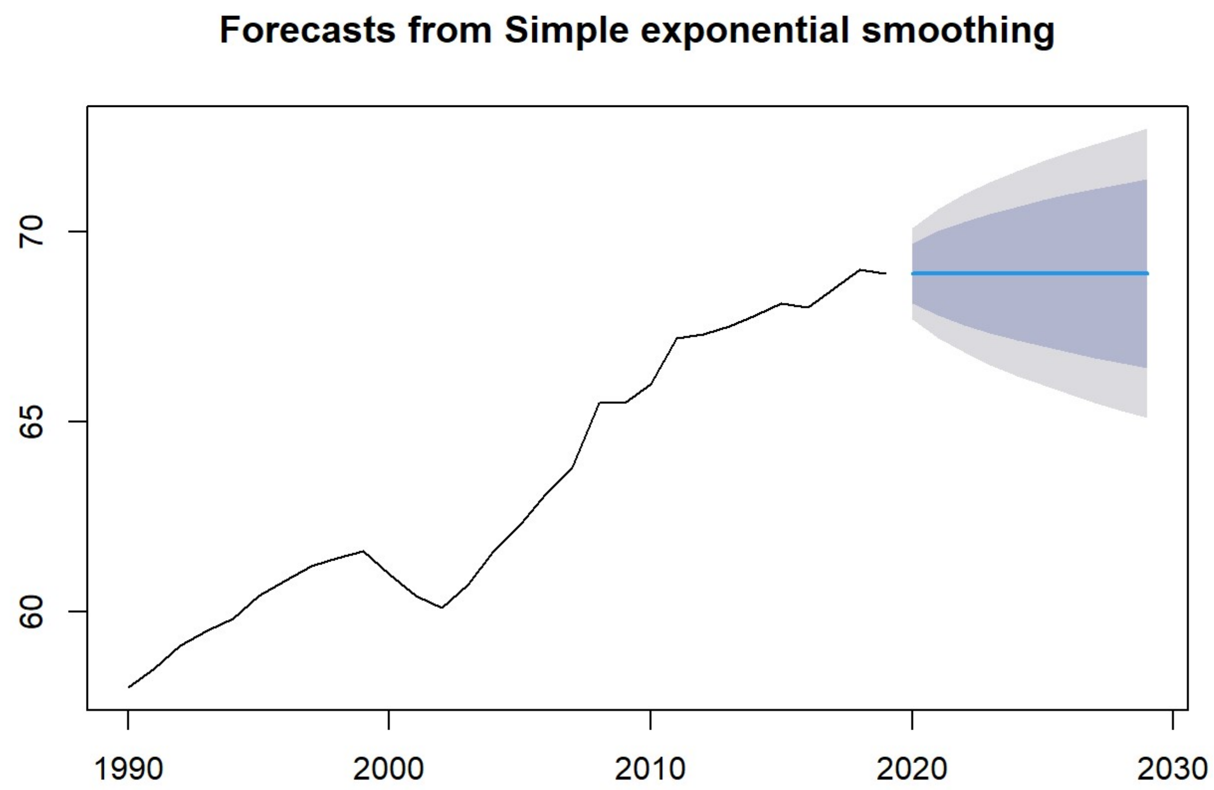
```
#We can apply the SES method if it's non-stationary. Let's apply a ADF test and see the st
ationarity:
adf.test(chile_ts) #Because the p-value is greater than 0.05, we cannot reject the alterna
tive hypothesis and the time series seems to be non-stationary.
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  chile_ts
## Dickey-Fuller = -2.7818, Lag order = 3, p-value = 0.2723
## alternative hypothesis: stationary
```

```
#SES Method
ses.model = ses(chile_ts, h = 10)
summary(ses.model)
```

```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = chile_ts, h = 10)
##
##   Smoothing parameters:
##     alpha = 0.9999
##
##   Initial states:
##     l = 58.0002
##
##   sigma:  0.6139
##
##      AIC     AICc      BIC
## 76.68724 77.61032 80.89083
##
## Error measures:
##                     ME      RMSE       MAE       MPE      MAPE      MASE
## Training set 0.3633638 0.5930491 0.4767109 0.5697154 0.7542966 0.9667563
##                   ACF1
## Training set 0.2977159
##
## Forecasts:
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2020       68.90001 68.11331 69.68671 67.69686 70.10316
## 2021       68.90001 67.78751 70.01251 67.19858 70.60144
## 2022       68.90001 67.53750 70.26252 66.81623 70.98379
## 2023       68.90001 67.32673 70.47329 66.49389 71.30613
## 2024       68.90001 67.14104 70.65898 66.20990 71.59012
## 2025       68.90001 66.97316 70.82686 65.95315 71.84687
## 2026       68.90001 66.81878 70.98124 65.71704 72.08298
## 2027       68.90001 66.67509 71.12493 65.49728 72.30274
## 2028       68.90001 66.54012 71.25990 65.29088 72.50914
## 2029       68.90001 66.41247 71.38755 65.09565 72.70437
```

```
plot(ses.model)
```

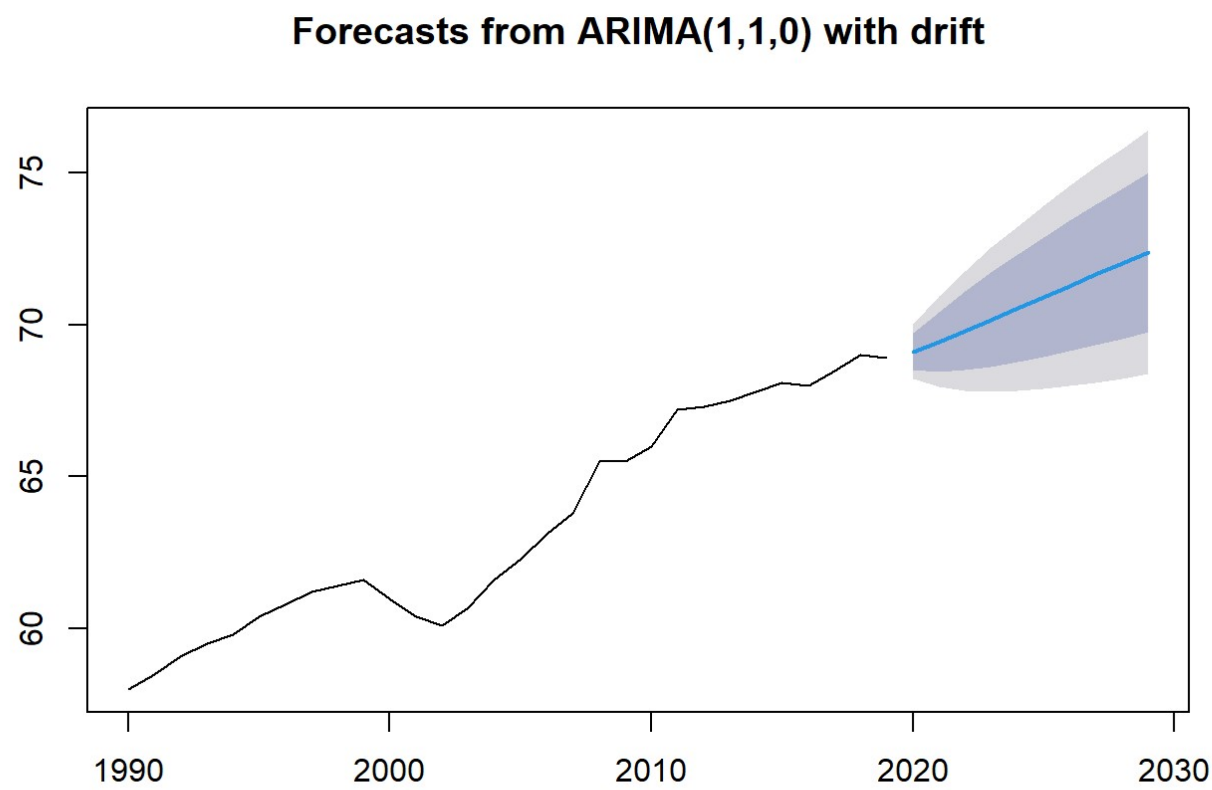


#ARIMA Model

```
#ARIMA or a form of the Box-Jenkins model is a standard modelling system for time series.
ARIMA models are flexible and very general. The random walks, exponential smoothing or aut
o regressive models can all be explained with that system of ARIMA.

#The most important thing to keep in mind is the trend of the data set. There is a trend i
n the time series meaning the autocorrelation is present. This implies that an observation
at an earlier time point influences the later observations. The values tend to increase or
decrease over time. Since this is a yearly data set without season, a standard ARIMA model
is suitable to create a model.

# Auto Generated ARIMA
arima.chile = auto.arima(chile_ts)

summary(arima.chile)
```

```
## Series: chile_ts
## ARIMA(1,1,0) with drift
##
## Coefficients:
##          ar1   drift
##       0.3114  0.3705
## s.e.  0.1758  0.1189
##
## sigma^2 = 0.2152:  log likelihood = -17.88
## AIC=41.77   AICc=42.73   BIC=45.87
##
## Training set error measures:
##                        ME      RMSE       MAE           MPE      MAPE     MASE
## Training set 0.0003628838 0.4401274 0.3230306 -0.0001701745 0.5057719 0.655097
##                       ACF1
## Training set -0.0004546612
```

```
plot(forecast(arima.chile, h = 10))
```




```
# Exact calculation of ARIMA parameters
auto.arima(chile_ts, stepwise=F, approximation=F)
```

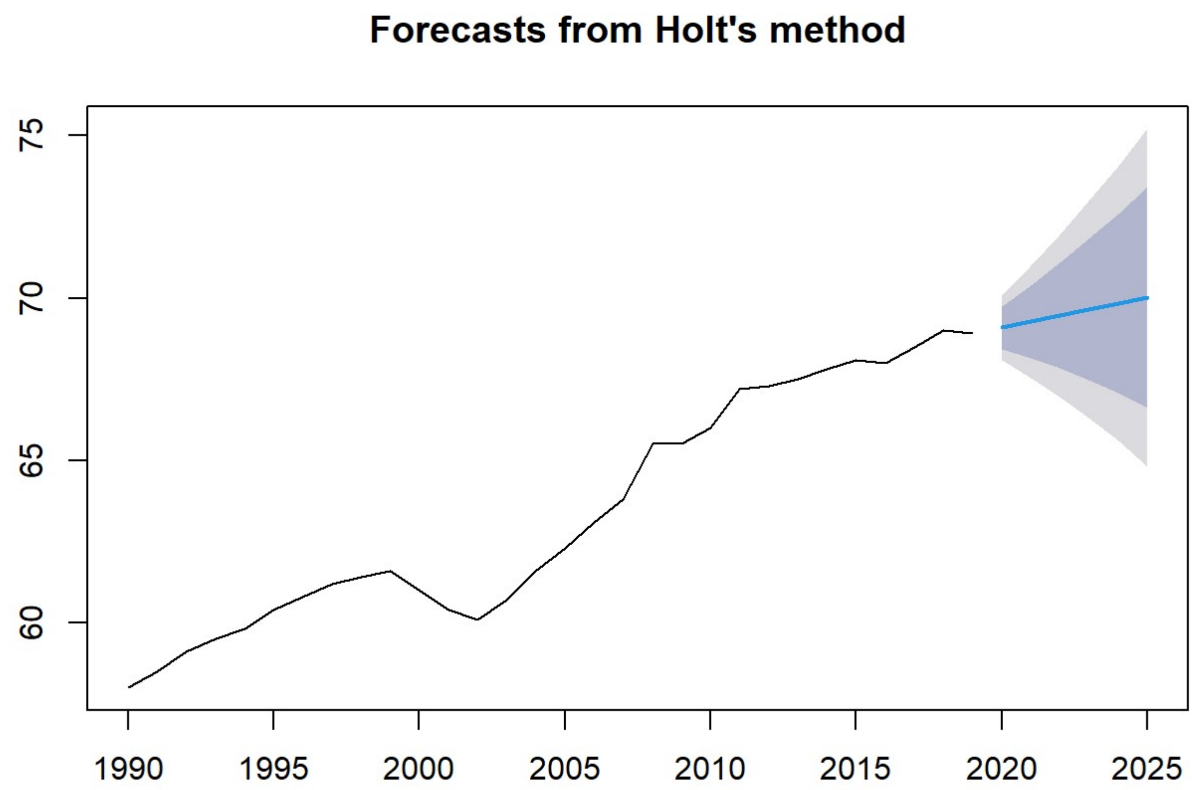
```
## Series: chile_ts
## ARIMA(1,1,0) with drift
##
## Coefficients:
##          ar1   drift
##       0.3114  0.3705
## s.e.  0.1758  0.1189
##
## sigma^2 = 0.2152:  log likelihood = -17.88
## AIC=41.77   AICc=42.73   BIC=45.87
```

```
#The first coefficient, ar1, represents the AutoRegressive (AR) term, and it measures the
influence of past values on the current value of the time series. In this case, the estima
ted value of ar1 is 0.3114, which means that past values of the time series have a positiv
e impact on the current value. The standard error of the ar1 coefficient, 0.1758, indicate
s the uncertainty in the estimated value.

#The second coefficient, drift, represents the linear trend in the time series data. The e
stimated value of drift is 0.3705, which means that the labor force participation rate in
Chile is increasing at a rate of 0.3705 units per time step on average. The standard error
of the drift coefficient, 0.1189, indicates the uncertainty in the estimated value of the
trend.
```

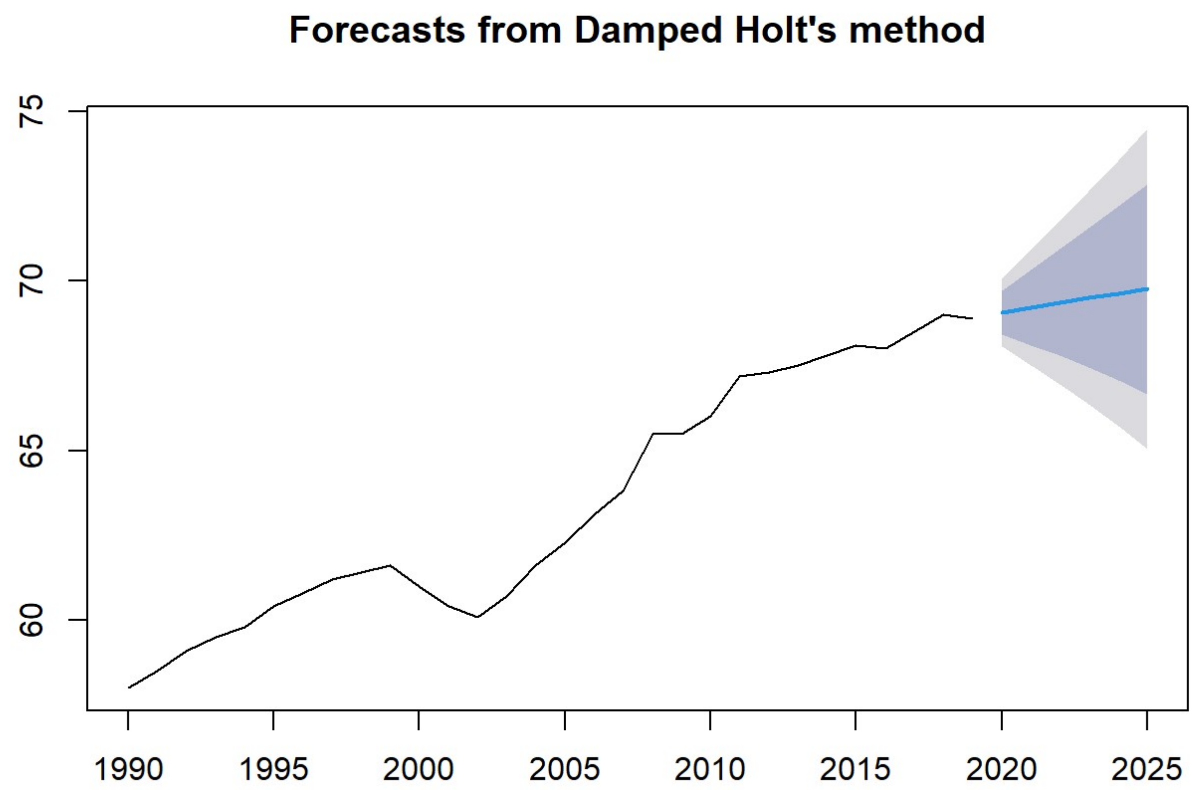
#FORECASTING AND DATA VISUALIZATION

```
#As I declared before in the first section, the final model forecasts the next 6 years.
# Creating final models
holt.linear.final = holt(chile_ts, h = 6)
holt.damped.final = holt(chile_ts, h = 6, damped = T)
ses.model.final = ses(chile_ts, h = 6)
arima.chile.final = forecast(auto.arima(chile_ts), h = 6)
```

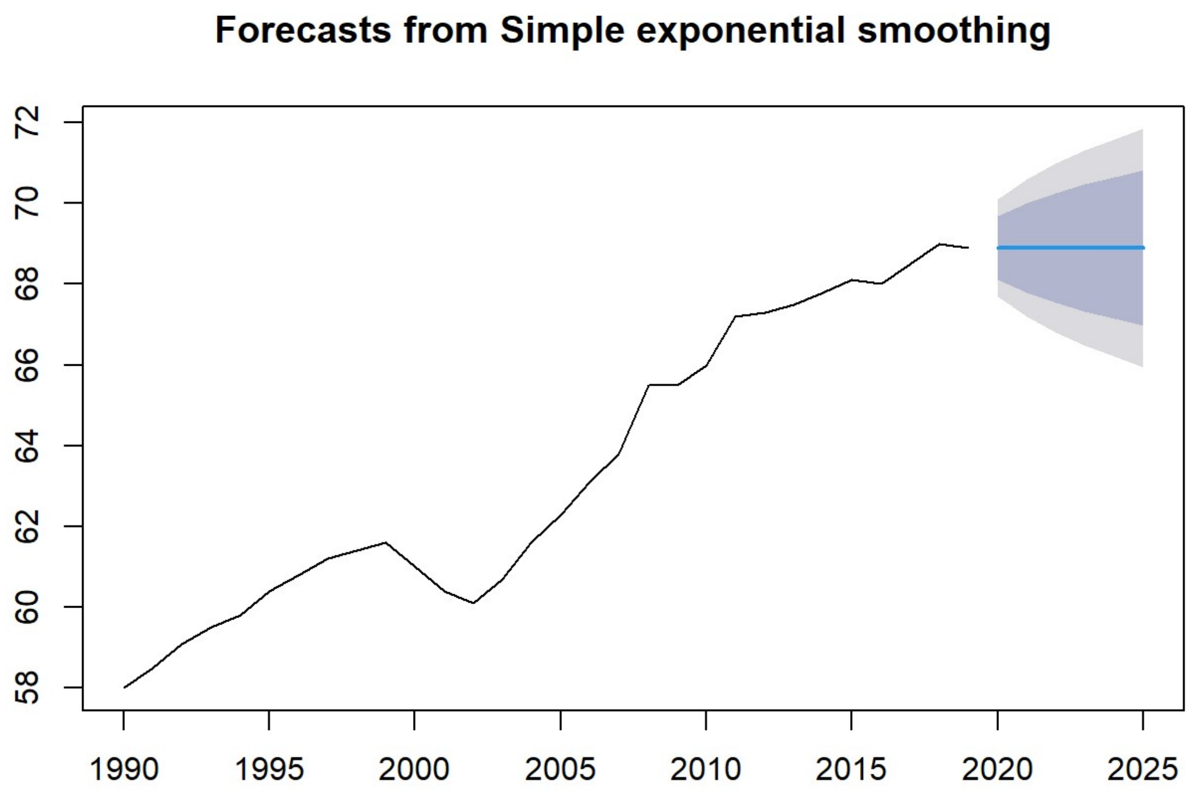
```
#Plotting the forecasts of the three models we set
plot(holt.linear.final)
```



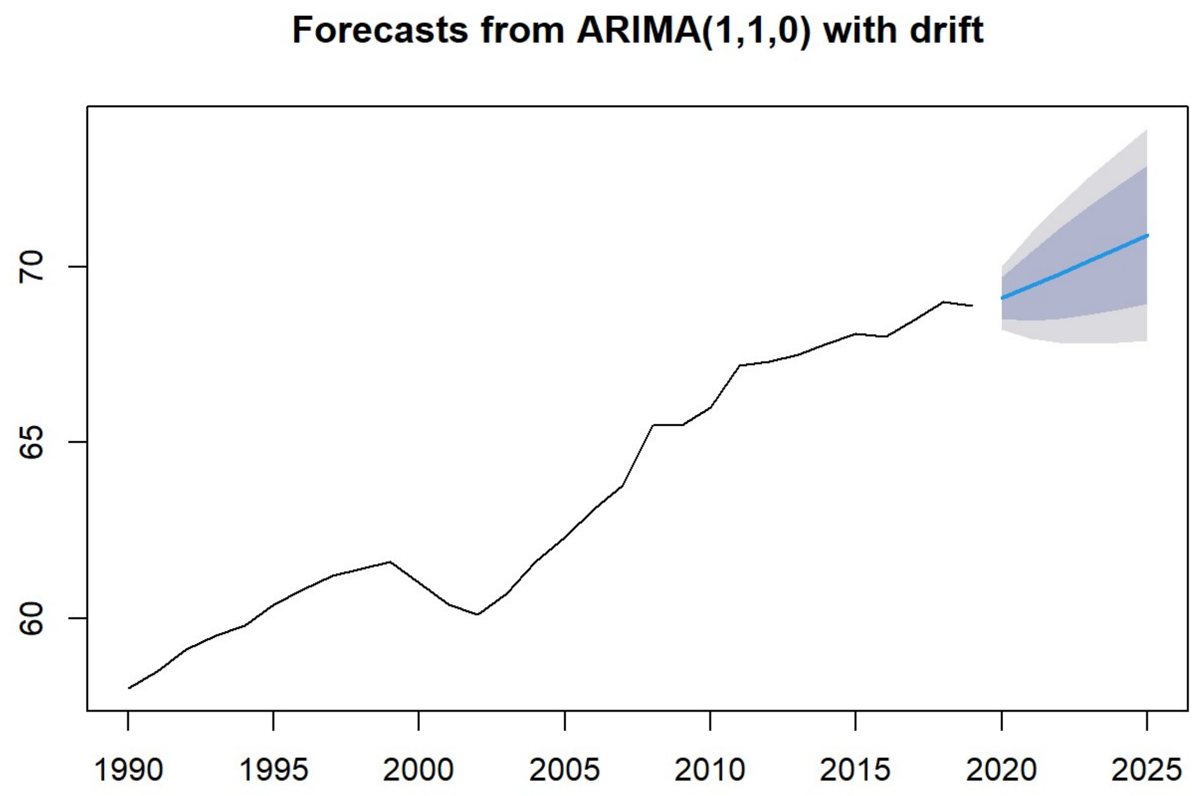

```
plot(holt.damped.final)
```



```
plot(ses.model.final)
```




```
plot(arima.chile.final)
```




```
#Checking the forecasting results of the Holt trend model
print(holt.linear.final)
```

```
##      Point Forecast     Lo 80    Hi 80    Lo 95    Hi 95
## 2020       69.08635 68.43195 69.74075 68.08553 70.08717
## 2021       69.27265 68.14573 70.39958 67.54917 70.99614
## 2022       69.45895 67.82686 71.09105 66.96288 71.95503
## 2023       69.64526 67.46626 71.82425 66.31277 72.97774
## 2024       69.83156 67.06366 72.59946 65.59842 74.06469
## 2025       70.01786 66.62051 73.41521 64.82207 75.21365
```

```
#Checking the forecasting results of the Holt damped model
print(holt.damped.final)
```

```
##      Point Forecast     Lo 80    Hi 80    Lo 95    Hi 95
## 2020       69.06198 68.40587 69.71809 68.05855 70.06542
## 2021       69.21593 68.10933 70.32253 67.52353 70.90833
## 2022       69.36228 67.79165 70.93290 66.96022 71.76434
## 2023       69.50141 67.44523 71.55758 66.35676 72.64605
## 2024       69.63367 67.07132 72.19602 65.71489 73.55244
## 2025       69.75940 66.67276 72.84605 65.03879 74.48002
```

```
#Checking the forecasting results of the Simple Exponential Smoothing model
print(ses.model.final)
```

```
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2020       68.90001 68.11331 69.68671 67.69686 70.10316
## 2021       68.90001 67.78751 70.01251 67.19858 70.60144
## 2022       68.90001 67.53750 70.26252 66.81623 70.98379
## 2023       68.90001 67.32673 70.47329 66.49389 71.30613
## 2024       68.90001 67.14104 70.65898 66.20990 71.59012
## 2025       68.90001 66.97316 70.82686 65.95315 71.84687
```

```
#Checking the forecasting results of the ARIMA model
print(arima.chile.final)
```

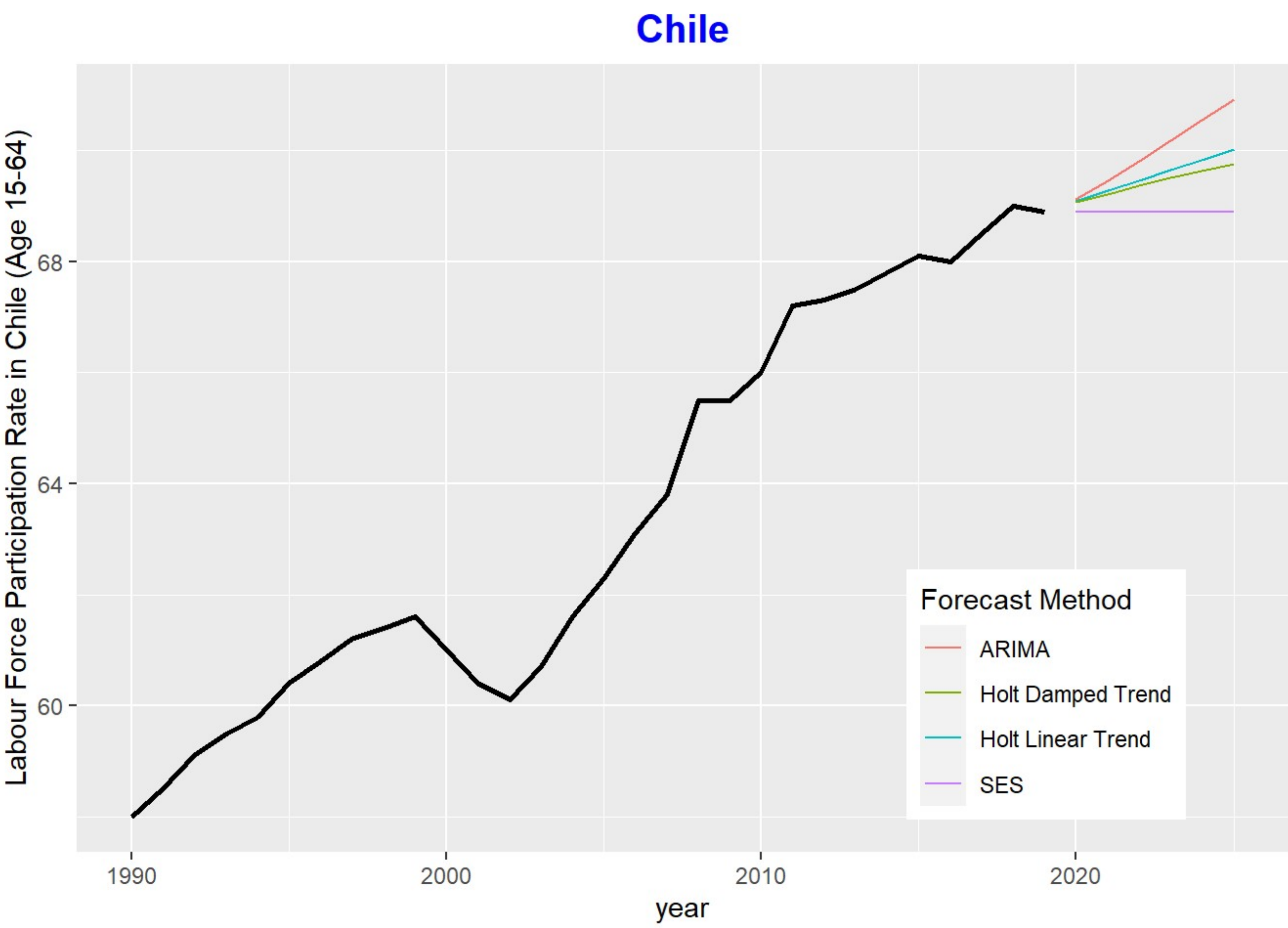
```
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2020       69.12400 68.52944 69.71856 68.21470 70.03330
## 2021       69.44890 68.46836 70.42944 67.94930 70.94851
## 2022       69.80523 68.51579 71.09468 67.83319 71.77727
## 2023       70.17135 68.62400 71.71869 67.80488 72.53781
## 2024       70.54051 68.76978 72.31124 67.83241 73.24861
## 2025       70.91062 68.94092 72.88032 67.89823 73.92301
```

```
autoplot(chile_ts) + geom_line(size=1) +
  forecast::autolayer(holt.linear.final$mean, series = "Holt Linear Trend") + geom_line(si
ze=1) +
  forecast::autolayer(holt.damped.final$mean, series = "Holt Damped Trend") + geom_line(si
ze=1) +
  forecast::autolayer(ses.model.final$mean, series = "SES") + geom_line(size=1) +
  forecast::autolayer(arima.chile.final$mean, series = "ARIMA") + geom_line(size=1) +
  xlab("year") + ylab("Labour Force Participation Rate in Chile (Age 15-64)") +
  guides(colour=guide_legend(title="Forecast Method")) + theme(legend.position = c(0.8, 0.
2)) +
  ggtitle("Chile") + theme(plot.title=element_text(family="Calibri", hjust = 0.5, color =
"blue",
                                                      face="bold", size=15))
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
```

```
## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font family not
## found in Windows font database
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## family not found in Windows font database
```



#Final Comments

#The point forecast values show that the labor force participation rate in Chile will increase to around 70%.

#It can be seen that the predicted values and prediction intervals increase steadily from 2020 to 2025. This may suggest that the time series has a positive trend.

#However, this is a simple time series analysis with one year frequency and with 30 observations. Without additional information on the labor force participation rate and more frequent data, it is difficult to make any further conclusions.