**TIME SERIES ANALYSIS PROJECT**

**Ruairi Kenealy, C20359883**

Contents

[Executive Summary 1](#_Toc101022226)

[Exploratory Data Analysis 3](#_Toc101022227)

[Data Source 3](#_Toc101022228)

[*R Packages used* 5](#_Toc101022229)

[Plotting the Time series 5](#_Toc101022230)

[Missing Values 7](#_Toc101022231)

[Holdout 7](#_Toc101022232)

[Is the Data Stationary? 9](#_Toc101022233)

[Transform using differencing 10](#_Toc101022234)

[Decomposition, fitting a moving average and LOESS 13](#_Toc101022235)

[LOESS 14](#_Toc101022236)

[Detrending the Data 15](#_Toc101022237)

[Model selection 17](#_Toc101022238)

[Exponential smoothing (ETS) 17](#_Toc101022239)

[ARIMA 19](#_Toc101022240)

[Model comparison 22](#_Toc101022241)

[Forecasting 23](#_Toc101022242)

[Conclusion 25](#_Toc101022243)

Part 1: Nonseasonal Time Series Analysis:

# Executive Summary

I choose to study the trends in unemployment in Ireland because I know I’ll be looking for a job in Ireland when I leave TUD. It’s helpful to know what the overall trend is and what to expect in future years.

This study uses live register data from Ireland from 1967 to 2021. I downloaded the data from the Irish CSO. However, that data was monthly data and I could see some seasonality because the live register tends to decrease during the summer months. To remove that seasonality, I took the average monthly number of people on the live register to produce 54 data points – one for each year from 1967 to 2021. The financial crash of 2008 hit Ireland very hard so that caused the live register to increase way above the trend for about 4-5 years and then subsequently dropped dramatically. I carved out a holdout of values from 2018-2021 which could be used to compare with the forecast on data from 1967-2017. All analysis on the data is based on the observations from 1967-2017.

I checked if the stationarity of the dataset using ACF and ADF. The data was non-stationary until I applied differencing with a lag of 1 year, twice. To examine the dataset in more detail, I decomposed the data. I did Seasonal and Trend decomposition using LOESS. I applied a number of different exponential smoothings (ETS) and ARIMA on the differenced dataset. Then I did some analysis of the results and residuals of the different models which were fitted to the data. Finally, I did a forecast and compared it to the holdout. Both showed a decrease in the live register in the years coming up to 2021. So, the forecasting method appears to be a good guide of future moves in the live register.

# Exploratory Data Analysis

## Data Source

See https://data.cso.ie and search for LRM01 dataset which is live register data for Ireland from 1967-today. I downloaded the data in LRM01 from the CSO website and this data contained number of people on the live register for every month from 1967 to March 2022. The dataset has totals for both sexes, male, female, under 25, over 25, unemployment benefit and unemployment allowance. I just used the data value for both sexes, all age groups and all types of benefit.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Statistic** | **Month** | **Age Group** | **Sex** | **Social Welfare Scheme** | **UNIT** | **VALUE** |
| Persons on Live Register | 1967M01 | All ages | Both sexes | All classes | Number | 58670 |
| Persons on Live Register | 1967M02 | All ages | Both sexes | All classes | Number | 57939 |
| Persons on Live Register | 1967M03 | All ages | Both sexes | All classes | Number | 56880 |
| Persons on Live Register | 1967M04 | All ages | Both sexes | All classes | Number | 51373 |
| Persons on Live Register | 1967M05 | All ages | Both sexes | All classes | Number | 47739 |
| Persons on Live Register | 1967M06 | All ages | Both sexes | All classes | Number | 43839 |
| Persons on Live Register | 1967M07 | All ages | Both sexes | All classes | Number | 42394 |
| Persons on Live Register | 1967M08 | All ages | Both sexes | All classes | Number | 43740 |
| Persons on Live Register | 1967M09 | All ages | Both sexes | All classes | Number | 42504 |
| Persons on Live Register | 1967M10 | All ages | Both sexes | All classes | Number | 47300 |
| Persons on Live Register | 1967M11 | All ages | Both sexes | All classes | Number | 50514 |
| Persons on Live Register | 1967M12 | All ages | Both sexes | All classes | Number | 57395 |
| Persons on Live Register | 1968M01 | All ages | Both sexes | All classes | Number | 61313 |
| Persons on Live Register | 1968M02 | All ages | Both sexes | All classes | Number | 60552 |

::::::::::::

::::::::::::

::::::::::::

::::::::::::

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Persons on Live Register | 2021M06 | All ages | Both sexes | All classes | Number | 175281 |
| Persons on Live Register | 2021M07 | All ages | Both sexes | All classes | Number | 184213 |
| Persons on Live Register | 2021M08 | All ages | Both sexes | All classes | Number | 179761 |
| Persons on Live Register | 2021M09 | All ages | Both sexes | All classes | Number | 162898 |
| Persons on Live Register | 2021M10 | All ages | Both sexes | All classes | Number | 165671 |
| Persons on Live Register | 2021M11 | All ages | Both sexes | All classes | Number | 164626 |
| Persons on Live Register | 2021M12 | All ages | Both sexes | All classes | Number | 163856 |
| Persons on Live Register | 2022M01 | All ages | Both sexes | All classes | Number | 162578 |
| Persons on Live Register | 2022M02 | All ages | Both sexes | All classes | Number | 163248 |
| Persons on Live Register | 2022M03 | All ages | Both sexes | All classes | Number | 178996 |

Since this data could be seasonal and also to reduce the amount of data, I got the monthly mean for each year. I used @AVG in Excel to produce one value for every year giving 54 data points.

This is the data used:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Statistic | Month | Age Group | Sex | Social Welfare Scheme | UNIT | VALUE | MONTHLY\_MEAN |
| Persons on Live Register | 1967M12 | All ages | Both sexes | All classes | Number | 57395 | 50024 |
| Persons on Live Register | 1968M12 | All ages | Both sexes | All classes | Number | 54789 | 53064 |
| Persons on Live Register | 1969M11 | All ages | Both sexes | All classes | Number | 50952 | 51128.75 |
| Persons on Live Register | 1970M12 | All ages | Both sexes | All classes | Number | 62202 | 58779 |
| Persons on Live Register | 1971M12 | All ages | Both sexes | All classes | Number | 71108 | 57237.5 |

:::::::::::

:::::::::::

:::::::::::

:::::::::::

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Persons on Live Register | 2016M12 | All ages | Both sexes | All classes | Number | 276502 | 302660.8 |
| Persons on Live Register | 2017M12 | All ages | Both sexes | All classes | Number | 236268 | 258580.1 |
| Persons on Live Register | 2018M12 | All ages | Both sexes | All classes | Number | 199669 | 220064.6 |
| Persons on Live Register | 2019M12 | All ages | Both sexes | All classes | Number | 181996 | 191528.5 |
| Persons on Live Register | 2020M12 | All ages | Both sexes | All classes | Number | 189860 | 208486.8 |
| Persons on Live Register | 2021M12 | All ages | Both sexes | All classes | Number | 163856 | 175359.6 |

The values are taken from column 8 which is labelled MONTHLY\_MEAN and divided by 1,000 to give results in 1000’s.

Code: values = live\_register[8]/1000

Headers in the Data

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 Statistic  2 Month  3 Age Group  4 Sex  5 Social Welfare Scheme  6 Unit  7 VALUE  8 MONTHLY\_MEAN |  |  |  |  |  |  |  |

Labels on Graphs

Y axis: Thousands of People

X axis: Year

Software used

R Studio version: 2022.02.1 Build 461

R.version(): R version 4.1.3 (2022-03-10)"

Excel version: Microsoft 365

## R Packages used

library(readr)

library(forecast)

library(tseries)

## Plotting the Time series

# Reading in the data

setwd("C:/Users/ruair/Downloads/Forecasting\_Time\_series")

live\_register = read.csv("live\_register\_ireland\_annual.csv",header=T)

values = live\_register[8]/1000

# Initial plot

values = ts(values, start=1967, frequency=1)

ts.plot(values, main="Live Register Annual", ylab="Number of People (1000’s)", type="l")

Chart, line chart

Description automatically generated

There was an increase in unemployment in Ireland during the 1980’s which is reflected int the graph above. The financial crash of 2008 hit Ireland very hard so that caused the live register to increase way above the trend for about 4-5 years and subsequently dropped dramatically. If those two temporary surges are ignored, there is a general upward trend. Since this data is not based on per capita, the increased trend is more than likely due to the increase in population in Ireland from 1967-2021.

Values

>Values

Time Series:

Start = 1967

End = 2021

Frequency = 1

[1] 220.000 225.000 225.000 225.000 226.287 234.525 235.000 230.150 234.000 249.000 230.750 235.000 230.485 230.000

[15] 235.000 231.950 239.000 240.000 245.000 240.000 248.000 240.000 239.000 240.000 238.000 226.000 231.600 235.000

[29] 240.000 245.000 250.000 250.000 250.000 245.000 248.000 245.000 240.000 238.900 243.232 240.000 243.000 257.500

[43] 256.000 251.500 253.000 255.000 254.510 265.000 244.750 250.000 220.000 225.000 225.000 225.000 226.287

## Missing Values

Check to see if there are any gaps in the data.

complete <- TRUE

for(c in complete.cases(values)) {

if(!c){

complete == FALSE

}

}

if(complete){

print("No missing values were found")

} else {

print ("Missing values found")

}

*Result: No missing values were found*

## Holdout

I have data up to 2021 but I decided to analyse the data up to 2016 and carve out a holdout from 2017-2021. This lets me check the accuracy of the model by comparing observed data to forecast data.

#set up a holdout set of observations for comparison with the forecast

values.holdout <- window(values, start=2017,end=2021)

values <- window(values, start=1967,end=2016)

#plot the shortened series and the holdout

ts.plot(cbind(values, values.holdout), main="Live Register Monthly Mean",

ylab="No. of People (1000's)", type="l", col=c("red", "blue"), lty=c(1, 2))

Chart, line chart

Description automatically generated

Observation Values Red, Holdout Blue

# Is the Data Stationary?

If we want to fit the ARIMA model for a forecast, we need to have a stationary dataset. If the data are not stationary then the dataset will have to be transformed using differencing. There are two unitroot tests to check if a time series is stationary. They are the **Augmented Dickey-Fuller** (ADF) test and the **Kwiatkowski–Phillips–Schmidt–Shin** (KPSS). Both are hypothesis tests used to estimate whether a stochastic process is stationary.

>adf.test(values)

Output:

Augmented Dickey-Fuller Test

data: values

Dickey-Fuller = -2.5597, Lag order = 3, p-value = 0.349

alternative hypothesis: stationary

The null hypothesis of the ADF test is non-stationarity so the p-value must be below 0.05 to reject the Null hypothesis. In this case, we accept the null hypothesis of non-stationarity because the p value is greater than 0.05, therefore we assume the data to be non-stationary. This is not surprising considering there is an upward trend due to population increase in Ireland from 1967-2021

The KPSS test has two components Trend and Level. The Null hypothesis here is Stationarity so the p-value must be below 0.5 if we are to reject that Null hypothesis.

kpss.test(values, null="Trend")

Output:

KPSS Test for Trend Stationarity

data: values

KPSS Trend = 0.094448, Truncation lag parameter = 3, p-value = 0.1

kpss.test(values, null="Level")

Output:

KPSS Test for Level Stationarity

data: values

KPSS Level = 0.77923, Truncation lag parameter = 3, p-value = 0.01

We cannot reject the Null hypothesis for Trend Stationarity because the p-value is 0.1. But we must reject the Null hypothesis for Level Stationarity and assume this data is Nonstationary. This concurs with the results from the ADF test above. So we need to use the diff() function to transform the dataset with differencing.

## Transform using differencing

Differencing is one way to make a non-stationary time series stationary. It computes the differences between consecutive observations. Differencing can help stabilise the mean of a time series by removing changes in the level of a time series, and therefore eliminating (or reducing) trend and seasonality. In the R code below, we first derive a dataset of the differences between the current year and the previous year. Then we compute the difference with the previous year to the previous year. The plot has a line graph of all three.

values\_diff1 = diff(values, lag = 1)

values\_diff2 = diff(values\_diff1)

tdiff <- cbind(values, values\_diff1, values\_diff2)

plot(tdiff, main="Differencing")

Chart, line chart, histogram

Description automatically generated

Original Values, 1st diff and 2nd diff 1

Then we need to do a stationary analysis of both differences to see which will give us a stationary dataset. We need to do as few differences as possible so that the dataset is not distorted. It’s not recommended to go beyond two differences.

#Get ADF and ACF for 1st Difference

par(mfrow=c(2,1))

acf(values\_diff1)

adf.test(values\_diff1)

Table

Description automatically generated with medium confidence

Output from Augmented Dickey-Fuller Test

data: values\_diff1

Dickey-Fuller = -3.0488, Lag order = 3, p-value = 0.1544

alternative hypothesis: stationary

The p-value of ADF for first order differencing is 0.1544. It's above 0.05 so we accept the null hypothesis of non-stationary.

We check the ACF and ADF of the 2nd difference using this code

par(mfrow=c(2,1))

acf(values\_diff2)

adf.test(values\_diff2)

Output of Augmented Dickey-Fuller Test

data: values\_diff2

Dickey-Fuller = -4.3427, Lag order = 3, p-value = 0.01

alternative hypothesis: stationary

Warning message:

In adf.test(values\_diff2) : p-value smaller than printed p-value

Graphical user interface, application, table

Description automatically generated

ACF of 2nd Difference 1

The 2nd difference appears to be stationary so the ARIMA can be fitted to it.

# Decomposition, fitting a moving average and LOESS

Time series data can exhibit a variety of patterns, and it is often helpful to split a time series into several components, each representing an underlying pattern category. Since this is non-seasonal data, we want to decompose the data into trend and white noise. We do this to get a better insight into the time series and improve the accuracy of the forecasts.

Also, it is could be helpful to first transform or adjust the series in order to make the decomposition as simple as possible. For example, the live register data could be transformed into per capita data by getting a dataset for the population over the same time period.

time points for fitting trends

time.pts = c(1:length(values))

time.pts = c(time.pts - min(time.pts))/max(time.pts)

moving average

ma.values <- ma(values, order=4, centre = FALSE)

mav.fit = ksmooth(time.pts, values, kernel = "box", bandwidth = 1.1)

values.fit.mav = ts(mav.fit$y,start=1967,frequency=1)

ts.plot(values,ylab="number of people", main="Observed Values vs MA vs Kernel smoothing")

lines(ma.values,lwd=2, lty=4, col="violet")

lines(values.fit.mav, col="red")

legend(x="topleft", c("Observed", "MA", "K smoothing"), col=c("gray10","violet", "red"), lty=c(1, 4))

Chart, line chart

Description automatically generated

## LOESS

loess.fit = loess(as.matrix(values)~time.pts, degree=2)

values.fit.loess = ts(fitted(loess.fit),start=1967)

plot(values.fit.loess)

Chart, line chart

Description automatically generated

ts.plot(values,ylab="number of people", main="Observed Values vs MA vs LOESS")

lines(ma.values,lwd=2, lty=4, col="violet")

lines(values.fit.loess, col="red")

legend(x="topleft", c("Observed", "MA", "LOESS"), col=c("gray10","violet", "red"), lty=c(1, 4))

Chart, line chart, histogram

Description automatically generated

## Detrending the Data

There are two different methods of detrending depending on whether a time series is additive or multiplicative. For an additive decomposition, this is done by subtracting the trend estimates from the series. For a multiplicative decomposition, this is done by dividing the series by the trend values. Using MA and LOESS at 0.5 span

Chart, line chart

Description automatically generated plot(cbind (ma.values, smoothed.values.loess50), main = "Trend Component")

# Model selection

## Exponential smoothing (ETS)

Forecasts produced using exponential smoothing methods are weighted averages of past observations, with the weights decaying exponentially as the observations get older. So the more recent the observation the higher the associated weight. This framework generates reliable forecasts quickly and for a wide range of time series. Because the time series is additive and doesn’t have any seasonality, we could use simple exponential smoothing for short term forecasts. However, Holt’s smoothing would probably be a better selection for this time series.

Apply the Holt Winters filtering using this code

HoltWinters(values\_diff2, beta=FALSE, gamma=FALSE)

plot(HoltWinters(values\_diff2, beta=FALSE, gamma=FALSE))

Output:

HoltWinters(x = values\_diff2, beta = FALSE, gamma = FALSE)

Smoothing parameters:

alpha: 6.610696e-05

beta : FALSE

gamma: FALSE

Coefficients:

[,1]

a -4.962373

Chart, line chart

Description automatically generated

Repeat the Holt Winters filtering but with different smoothing parameters. Beta is 0

values.hsmooth <- HoltWinters(values, gamma=FALSE)

values.hsmooth

plot(values.hsmooth)

Output

Call:

HoltWinters(x = values, gamma = FALSE)

Smoothing parameters:

alpha: 1

beta : 0

gamma: FALSE

Coefficients:

[,1]

a 302.6608

b 3.0400

Chart, line chart, histogram

Description automatically generated

## 

## ARIMA

Since we determined earlier that we needed to diff the values twice in order to get a stationary time series, we used values\_diff2 for the ARIMA model

Code

>arima(values\_diff2)

Output:

Call:

arima(x = values\_diff2)

Coefficients:

intercept

-0.9093

s.e. 4.2601

sigma^2 estimated as 871.1: log likelihood = -230.58, aic = 465.17

We also use the second difference for the ACF and PACF

>acf(values)

Chart

Description automatically generated

>pacf(values)

Chart, box and whisker chart

Description automatically generated

# Model comparison

A great advantage of using the ETS model is that information criteria can be used for model selection. We have Akaike’s Information Criterion (AIC) for the ETS but we can’t really use it to compare ETS and ARIMA because of the differences in the model classes. So we use Time Series Cross Validation (CV)

fets <- function(x, h) {

forecast(ets(x), h = h)

}

farima <- function(x, h) {

forecast(auto.arima(x), h=h)

}

ARIMA CV error:

res.autoarima <- tsCV(values, fautoarima, h=5)

ETS CV error:

res.ets <- tsCV(values, fets, h=5)

mean(res.ets^2, na.rm=TRUE)

Result:

5598.029

mean(res.autoarima^2, na.rm=TRUE)

Result:

NaN

Here the ETS and Arima forecast are being put into variables

fets <- function(x, h) {

forecast(ets(x, model="MAN"), h = h)

}

farima <- function(x, h, order) {

forecast(arima(x, order = c(0,2,2)), h = h)

}

# Compute CV errors for ETS

res.MANets <- tsCV(values, fets, h=1)

# Compute CV errors for ARIMA

res.arima0\_2\_2 <- tsCV(values, farima, h=1)

# Find MSE of each model class

sqrt(mean(res.MANets^2, na.rm=TRUE))

sqrt(mean(res.arima0\_2\_2^2, na.rm=TRUE))

res.MANets

cres.MANets <- tsCV(values, fets, h=1)

>

> # Compute CV errors for ARIMA

> res.arima0\_2\_2 <- tsCV(values, farima, h=1)

No errors found for the ARIMA

> res.arima0\_2\_2

Time Series:

Start = 1967

End = 2016

Frequency = 1

[1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA

[40] NA NA NA NA NA NA NA NA NA NA NA

>

> # Find MSE of each model class

> sqrt(mean(res.MANets^2, na.rm=TRUE))

[1] NaN

>

There are no errors in the Square Root check

> sqrt(mean(res.arima0\_2\_2^2, na.rm=TRUE))

[1] NaN

>

Also no errors in the MANets check

> res.MANets

Time Series:

Start = 1967

End = 2016

Frequency = 1

[1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA

[40] NA NA NA NA NA NA NA NA NA NA NA

So our checks indicate that there are no errors.

# Forecasting

Here we check how well the model has performed on the data using diagnostic tools and accuracy measures that allow one model to be compared against another.

values.fit.MANets <- ets(values, model="MAN")

coef(values.fit.MANets)

summary(values.fit.MANets)

Result:

ETS(M,A,N)

Call:

ets(y = values, model = "MAN")

Smoothing parameters:

alpha = 0.9999

beta = 0.9999

Initial states:

l = 48.52

b = -0.4182

sigma: 0.1429

AIC AICc BIC

578.4758 579.7003 588.5125

Training set error measures:

ME = -0.5945962 RMSE = 29.10223

MAE = 17.32728 MPE = 0.5968386 MAPE = 9.152305

MASE = 0.8254568 ACF1 = -0.04320899

forecast.fit.MANets <- forecast(values.fit.MANets, h=3, bootstrap = TRUE)

plot(forecast.fit.MANets)

Chart, line chart

Description automatically generated

# Conclusion

I carved out a holdout of values from 2018-2021 and compared it to the forecast on data from 1967-2017. The final plot above compares the forecast and compared to the holdout. Both showed a decrease in the live register in the years coming up to 2021. So, the forecasting method appears to be a good guide of future moves in the live register.

It would be a good idea to correlate the trend in the live register population with the overall working age population in Ireland from 1967-2022. Ireland’s population has been increasing a lot since the 1960’s. A per capita graph would be useful for exploratory analysis.

Also, Covid-19 more than likely caused an aberration in the data so it would be good to revisit this study around 2025 to get an idea of the long-term trend.

My overall conclusion is that the future is bright for employment in Ireland because the live register appears to be decreasing.