UNHCR Time Series Analysis and Asylum Seeker Forecasting

1.1 Prepare Dataset for Analysis

1.2 Import Data

Install packages and load libraries, set working directory

```
setwd("/Users/tessaschneider/Desktop/Final Data Analysis")

df <- read.csv("unhcr_popstats_refugee-status.csv", skip = 2, stringsAsFactors = F,
               na.string=c("", "*"))
df2 <- read.csv("unhcr_popstats_export_persons_of_concern_all_data.csv", skip = 3,
                stringsAsFactors = F, na.string=c("", "*"))
```

1.3 Tidy Data

Convert Total.Population to numeric before merging the datasets (after merging, data before 2000 drops out)

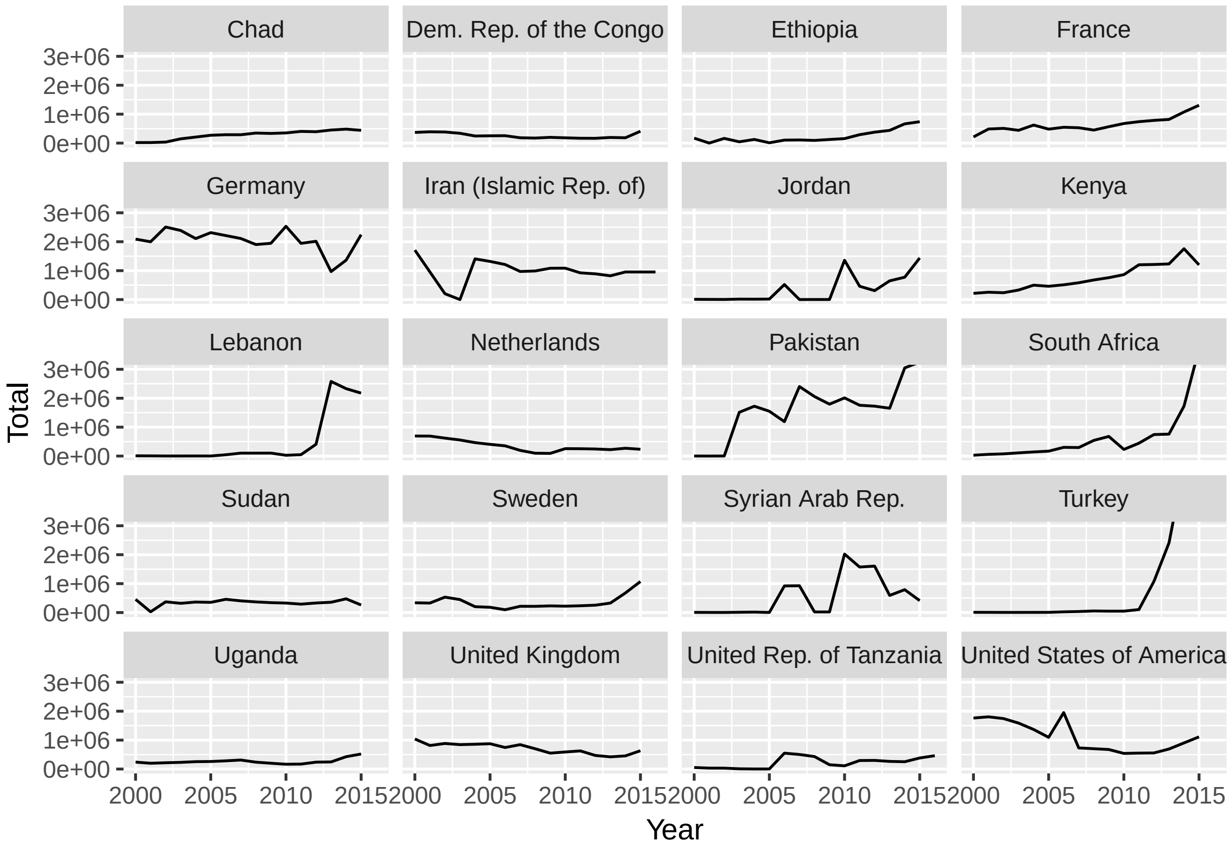
```
df2$Total.Population[df2$Total.Population=="*"] <- "2.5"
df2$Total.Population <- as.numeric(df2$Total.Population)

merged_data <- merge(df2, df, by = c("Year", "Country...territory.of.asylum.residence",
                                      "Origin"))

table(merged_data$Year)
```

```
##
##  2000  2001  2002  2003  2004  2005  2006  2007  2008  2009  2010  2011
##  4301  4683  4988  5473  5668  5778  5806  6064  6120  6207  7080  7209
##  2012  2013  2014  2015  2016
##  7537  8332  9191 10071  9495
```

2.1 Exploring Data: Top Destination Countries

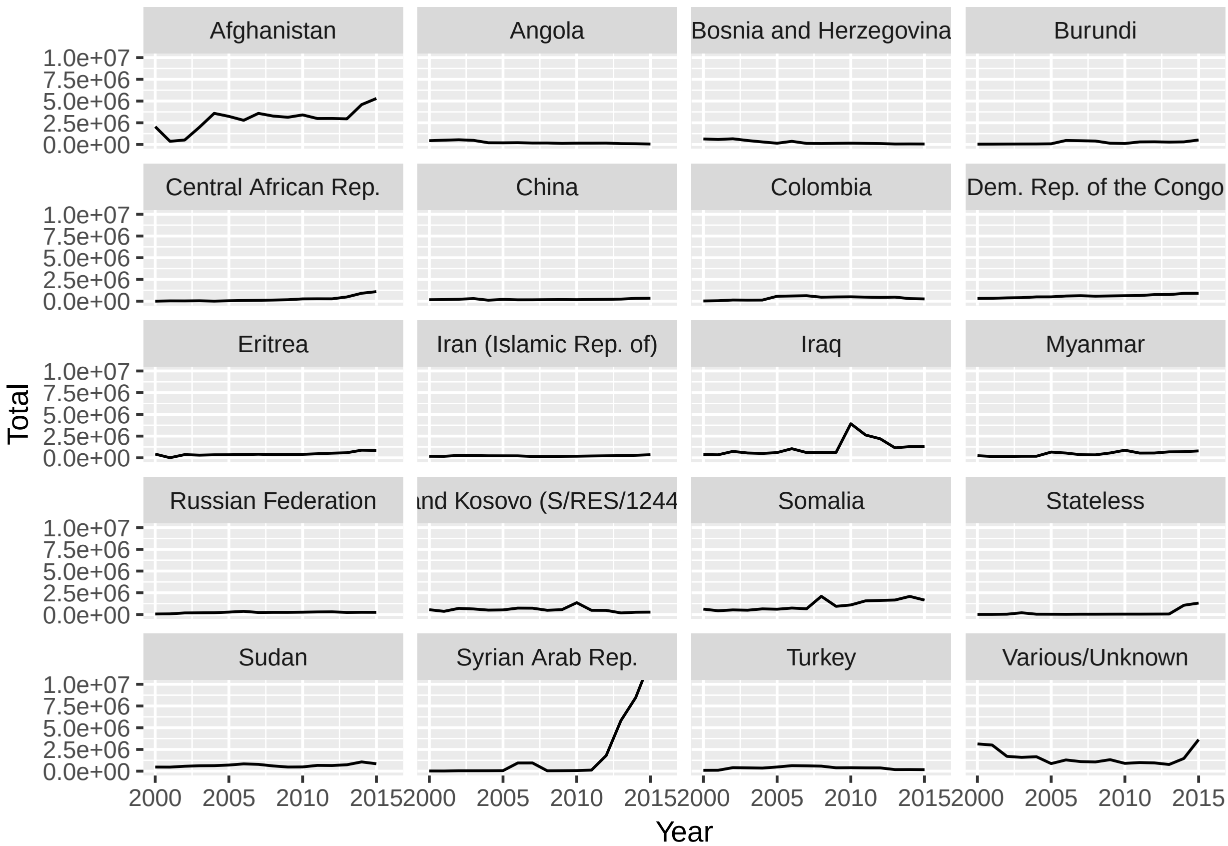
```
destination_country_total <- merged_data %>%
  group_by(Country...territory.of.asylum.residence, Year) %>%
  summarise(Total = sum(Total.Population))

top_destcountries <- destination_country_total %>%
  group_by(Country...territory.of.asylum.residence) %>%
  summarise(Total = sum(Total, na.rm = TRUE)) %>%
  top_n(20)

top_destcountries2 <- as.character(top_destcountries$Country...territory.of.asylum.residence)

destination_country_total %>%
  filter(Country...territory.of.asylum.residence %in% top_destcountries2) %>%
```

```
  ggplot(mapping = aes(x = Year, y = Total)) +
  geom_line() + coord_cartesian(ylim = c(0, 3e6)) +
  facet_wrap( ~ Country...territory.of.asylum.residence, ncol=4)
```



2.2 Exploring Data: Top Origin Countries

```
origin_country_total <- merged_data %>%
  group_by(Origin, Year) %>%
  summarise(Total = sum(Total.Population))

top_origcountries <- origin_country_total %>%
  group_by(Origin) %>%
  summarise(Total = sum(Total, na.rm = TRUE)) %>%
  top_n(20)

top_origcountries2 <- as.character(top_origcountries$Origin)

origin_country_total %>%
  filter(Origin %in% top_origcountries2) %>%
  ggplot(mapping = aes(x = Year, y = Total)) +
  geom_line() + coord_cartesian(ylim = c(0, 1e7)) +
  facet_wrap( ~ Origin, ncol=4)
```



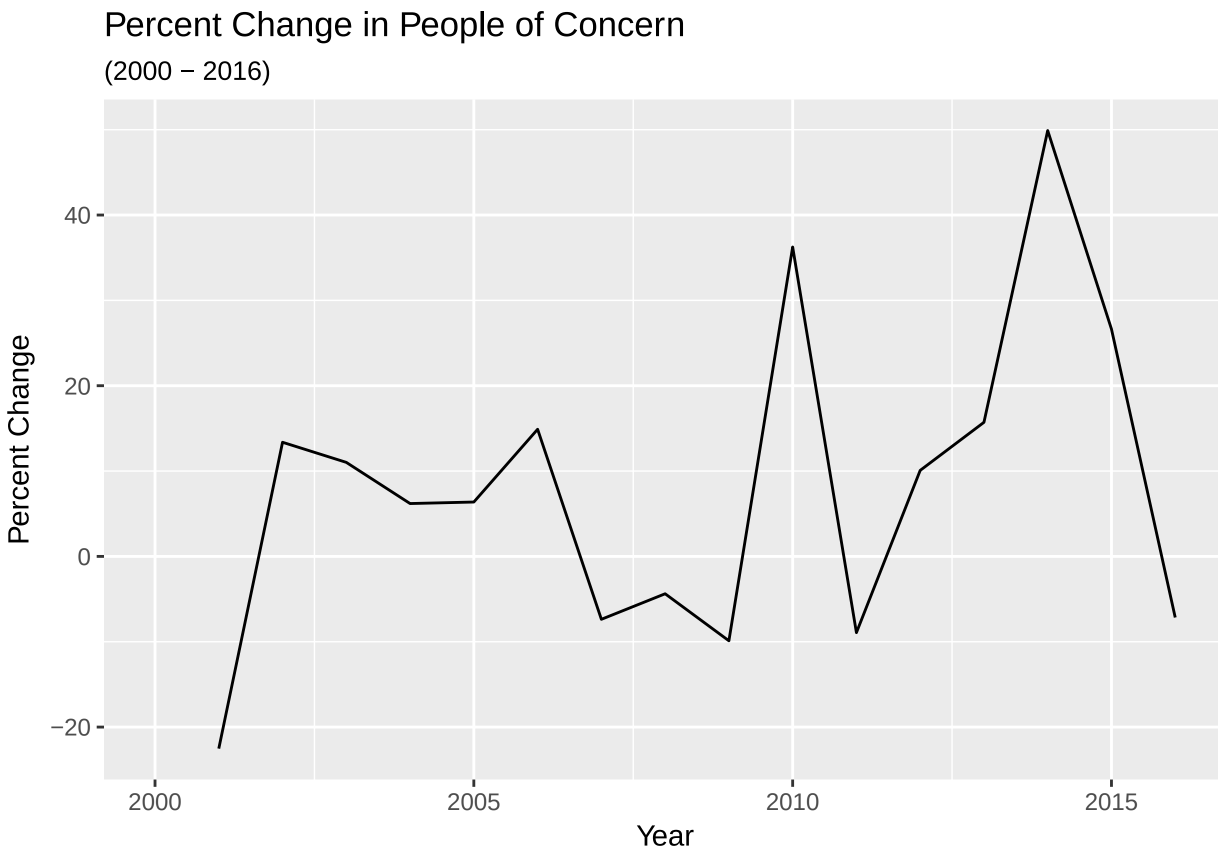

```r
table(merged_data$Year)
```

```
##
##  2000  2001  2002  2003  2004  2005  2006  2007  2008  2009  2010  2011
##  4301  4683  4988  5473  5668  5778  5806  6064  6120  6207  7080  7209
##  2012  2013  2014  2015  2016
##  7537  8332  9191 10071  9495
```

2.3 Exploring Data: Percent Change in Total Population

By "People of Concern"", subset for only PoC category counts by year change value from character to integer

```r
Year_Pop <- aggregate(merged_data$`Total.Population`, by=list(Year = merged_data$Year),
                      FUN=sum, na.rm = TRUE)

Year_Pop$rate <- NA

Year_Pop$rate[which(Year_Pop$Year>2000)] = 100*(diff(Year_Pop$x)/Year_Pop[-nrow(Year_Pop),]$x)

ggplot(Year_Pop, aes(x= Year, y= rate)) + geom_line() +
  labs(title="Percent Change in People of Concern",
       subtitle="(2000 - 2016)",
       x="Year",
       y="Percent Change")
```



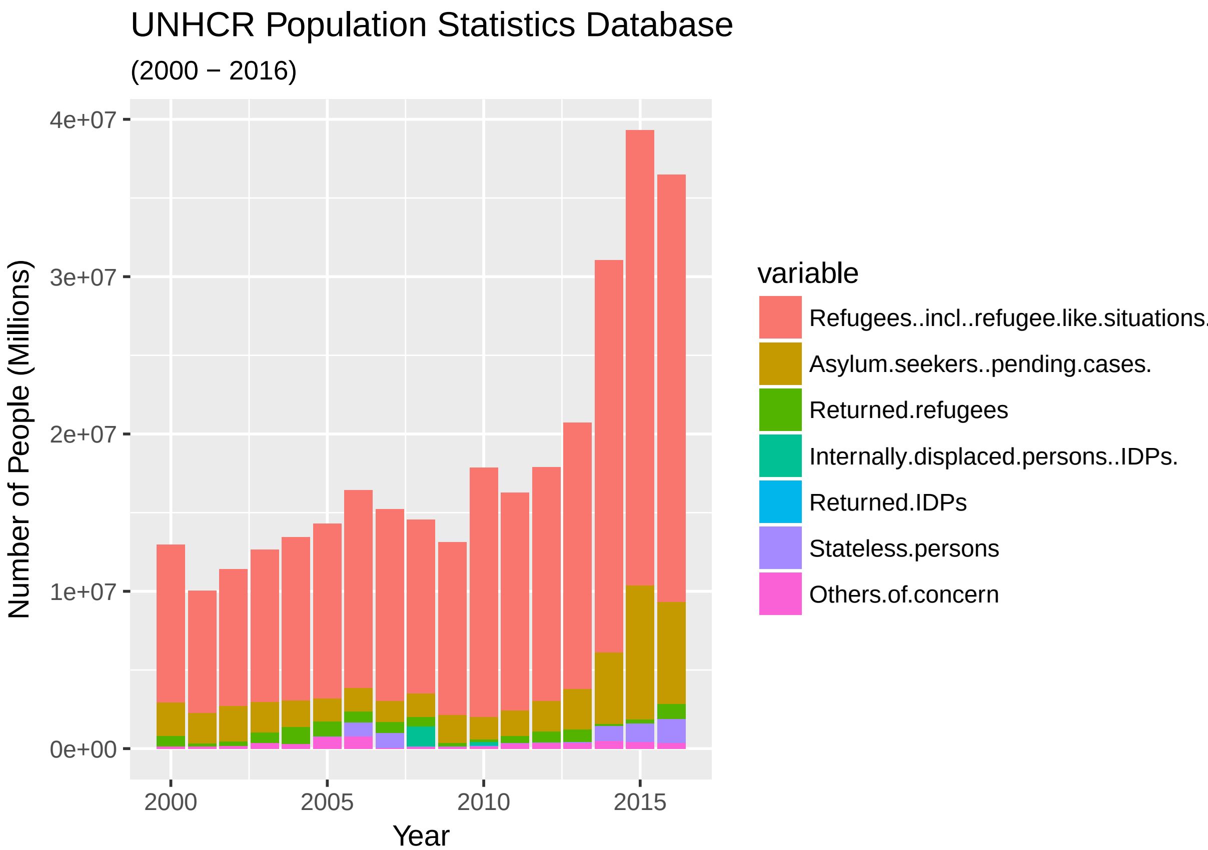

```r
PoC_count <- merged_data[c(1,4:10)]

PoC_count <- melt(PoC_count, id=c("Year"))

str(PoC_count)
```

```
## 'data.frame':    798021 obs. of  3 variables:
##  $ Year    : int  2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 ...
##  $ variable: Factor w/ 7 levels "Refugees..incl..refugee.like.situations.",..: 1 1 1 1 1 1 1 1 1 1 .
##  $ value   : int  NA NA 9 507 2 5 NA 1 5 20 ...
```

```r
PoC_count$value <- as.integer(PoC_count$value)
```

Starting from 2013 the number of refugees has increased dramatically and with it pending cases for asylum seekers have also increased

```r
ggplot(PoC_count,aes(Year,value, na.rm = TRUE)) +
  geom_bar(aes(fill=variable),stat="identity") +
  labs(title="UNHCR Population Statistics Database",
       subtitle="(2000 - 2016)",
       x="Year",
       y="Number of People (Millions)")
```



3.1 Time Series Analysis: Preparation

- y is PoC in Germany
- x is PoC in all countries in database
- t is Years (2000-2016)

All variables used in the model must be declared as time series

```
Germany_PoC <- merged_data %>% group_by(Country...territory.of.asylum.residence, Year) %>%
  filter('Germany' %in% Country...territory.of.asylum.residence) %>%
  summarise(Total = sum(Total.Population, na.rm = TRUE))

Germany_data <- merge(Germany_PoC, Year_Pop, by = "Year")

Germany_data$Year <- ts(Germany_data$Year)
Germany_data$Total <- ts(Germany_data$Total)
Germany_data$x <- ts(Germany_data$x)
```

3.2 Time Series Analysis: Test for Time Series Problems

Test for Persistence or Dependence

Row is <1 so it meets the stability condition for weak dependency

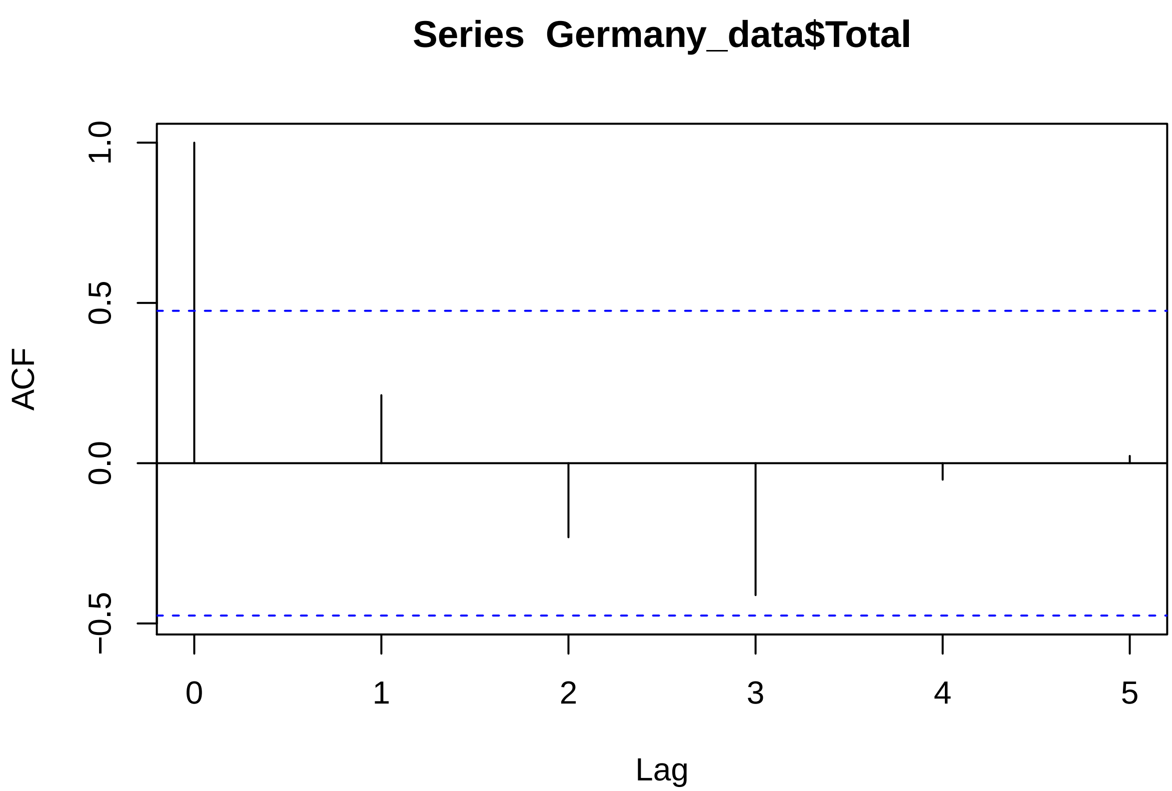
```
summary(dynlm(Total ~ L(Total, 1), data = Germany_data))
```

```
##
## Time series regression with "ts" data:
## Start = 2, End = 17
##
## Call:
## dynlm(formula = Total ~ L(Total, 1), data = Germany_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1167358  -221340  -102116   196356  1555201
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.183e+06  7.834e+05   1.510    0.153
## L(Total, 1) 4.726e-01  3.772e-01   1.253    0.231
##
## Residual standard error: 583500 on 14 degrees of freedom
## Multiple R-squared:  0.1008, Adjusted R-squared:  0.03661
## F-statistic:  1.57 on 1 and 14 DF,  p-value: 0.2307
```

Test for Persistence or Dependence

Germany's Total persons of concern annual data shows that the correlation of lags of the Total Population variable drops to zero after 1 lag with statistical insignificant correlation after 1 lag, therefore it is not persistent

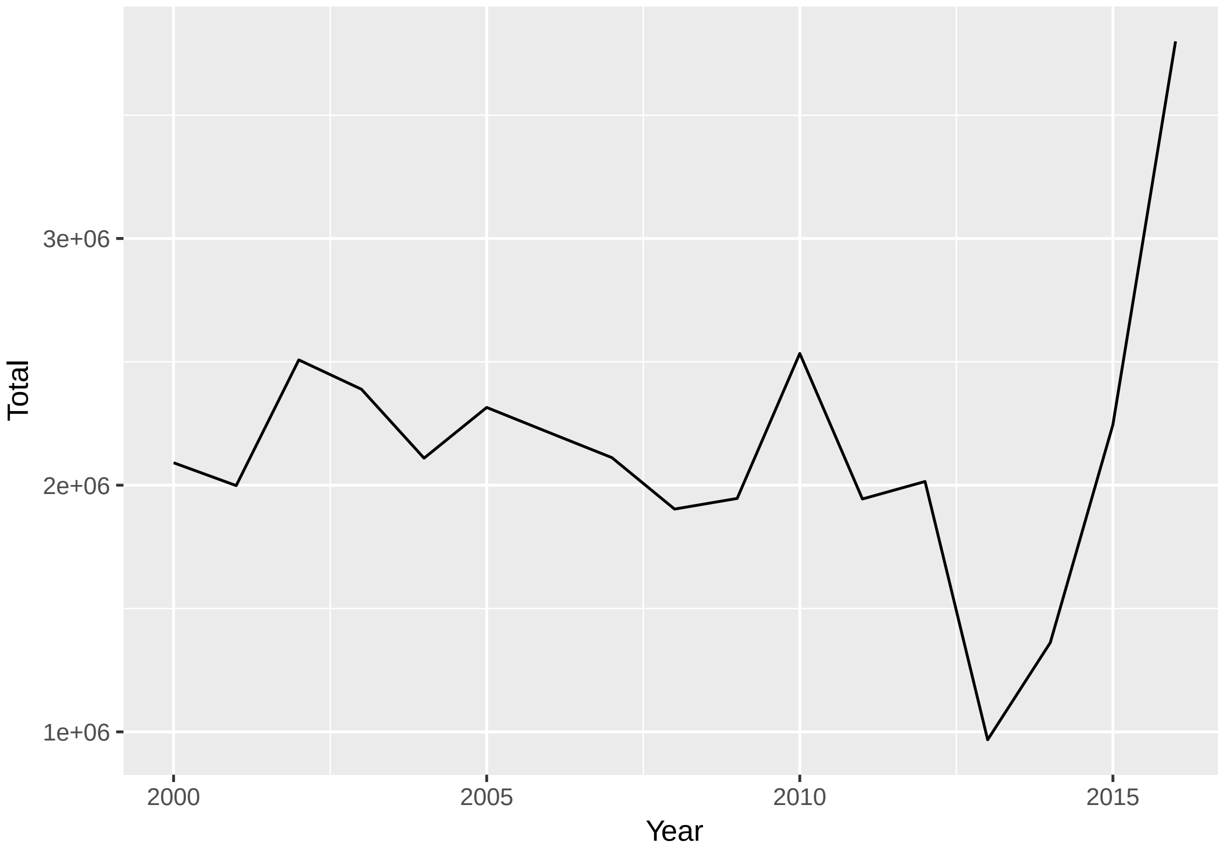
```r
acf(Germany_data$Total, na.action = na.pass, lag.max = 5)
```



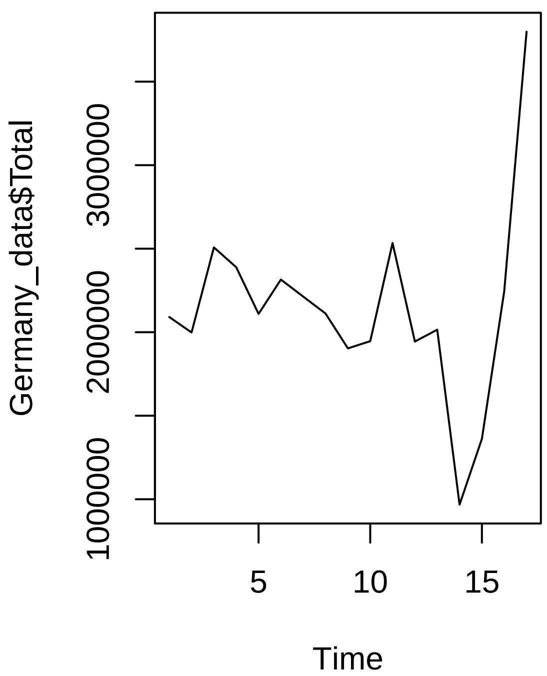
Tests for Stationarity

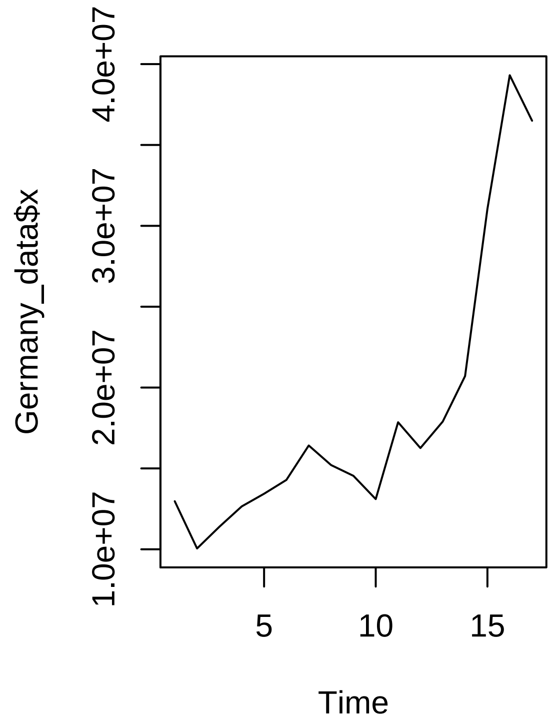
Germany Total PoC annual is trending after 2012 Stochastic trend (increases and decreases inconsistently) in the Germany Total plot Deterministic trend (increases and decreases consistently) in the Germany x plot

```
ggplot(data=Germany_data,
       mapping = aes(x = Year, y = Total)) + geom_line()
```




```
par(mfrow = c(1,2))
plot(Germany_data$Total)
plot(Germany_data$x)
```





Tests for Stationarity - Unit Root Test - Dickey Fuller Test

(p value <.05 then there is no unit root)

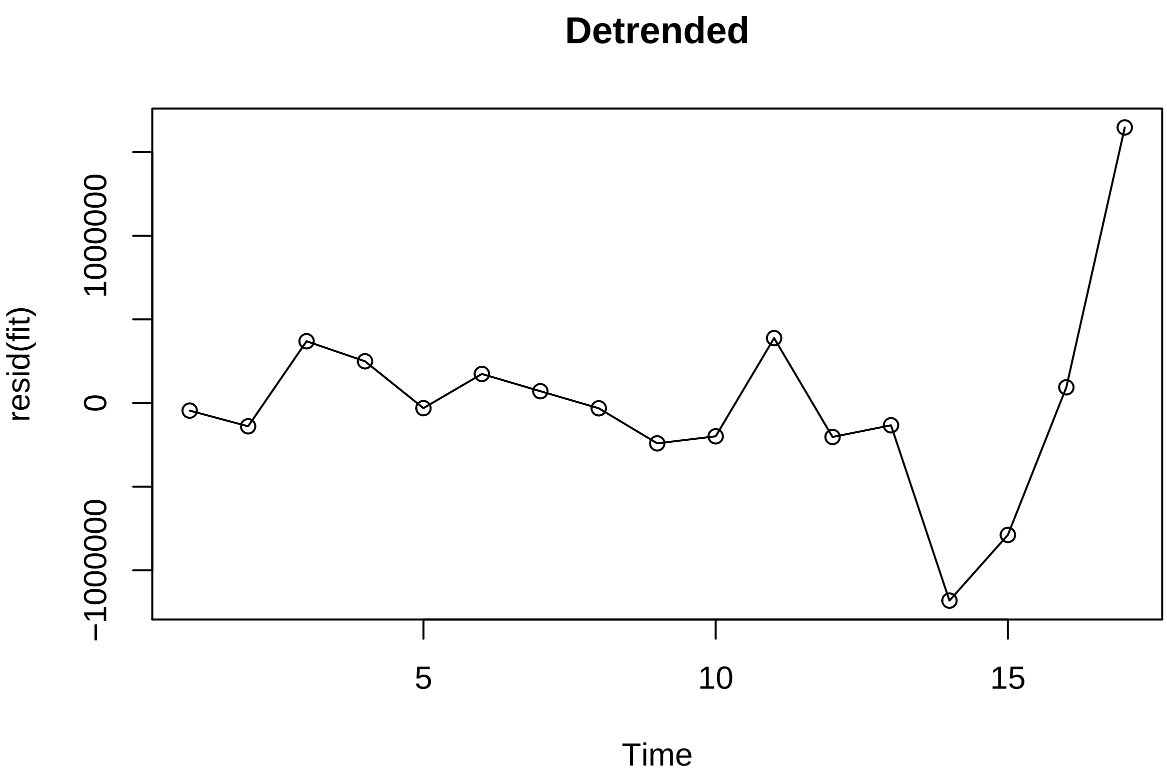
```
adf.test(Germany_data$Total)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  Germany_data$Total
## Dickey-Fuller = -4.0015, Lag order = 2, p-value = 0.0232
## alternative hypothesis: stationary
```

Detrend: When there is a Deterministic Trend

Regress y, x1 and x2 on trend term(Year) and intercept, save residuals for y, x1 and x2, and then regress y residual on x1 residual and x2 residual The regression with residuals shows an increase in the correlation, but it is still not statistically significant Even after detrending there is still no statistically significant coefficient

```
fit = lm(Germany_data$Total ~ Germany_data$Year, na.action = NULL)
plot(resid(fit), type="o", main="Detrended")
```



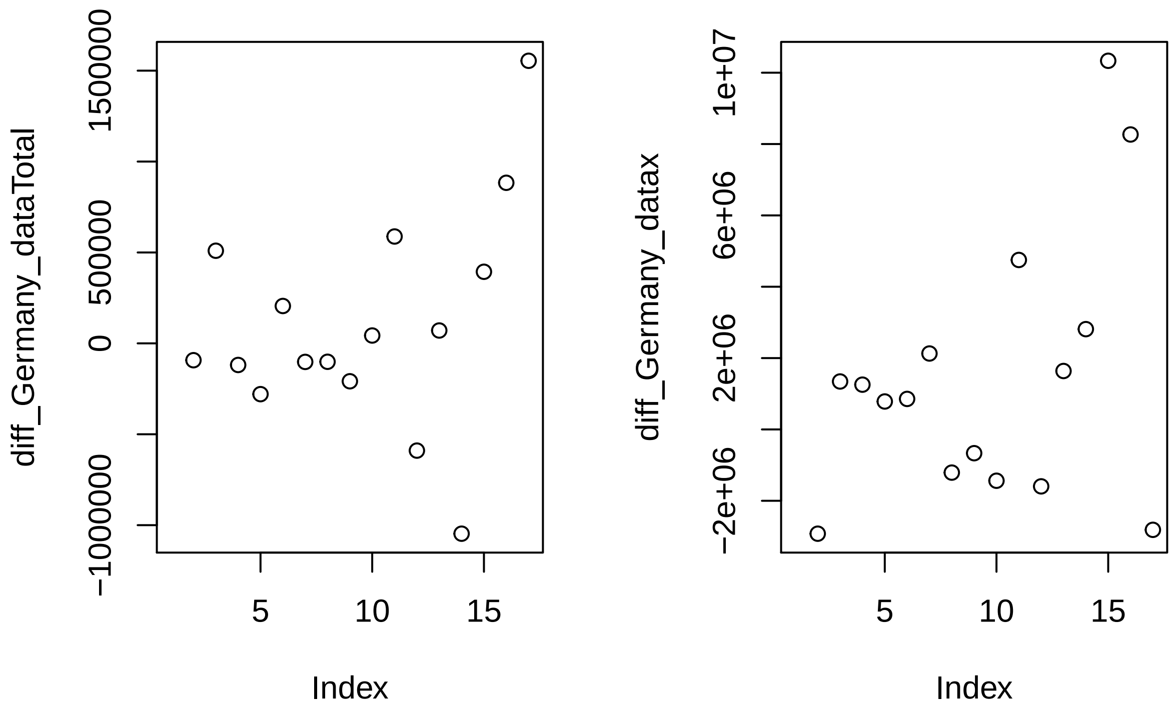

```
fit1 <- lm(Germany_data$Total ~ Germany_data$Year)
res_Germany_dataTotal <- residuals(fit1)

fit2 <- lm(Germany_data$x ~ Germany_data$Year)
res_Germany_datax <- residuals(fit2)

summary(m3 <- dynlm(res_Germany_dataTotal ~ res_Germany_datax))
```

```
##
## Time series regression with "numeric" data:
## Start = 1, End = 17
##
## Call:
## dynlm(formula = res_Germany_dataTotal ~ res_Germany_datax)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -974291 -205771   50893  170461 1342713
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -2.175e-12  1.326e+05   0.000    1.000
## res_Germany_datax  4.506e-02  2.726e-02   1.653    0.119
##
## Residual standard error: 546900 on 15 degrees of freedom
## Multiple R-squared:  0.1541, Adjusted R-squared:  0.09771
## F-statistic: 2.733 on 1 and 15 DF,  p-value: 0.1191
```

Detrend: When there is a Stochastic Trend

First differencing then plotting shows that the trend was removed in this case

```
diff_Germany_dataTotal <- c(NA, diff(Germany_data$Total))
diff_Germany_datax <- c(NA, diff(Germany_data$x))

par(mfrow = c(1,2))
plot(diff_Germany_dataTotal)
plot(diff_Germany_datax)
```



3.3 Run OLS regression

This time series regression resulted in no statistically significant correlation between the selected variables Since there is monthly data on asylum seekers, perhaps it is possible to predict future numbers of asylum seekers in Germany through a forecasting model (there are clear limitations in only looking at one variable, so these predictions cannot be interpreted as exact predictions)

```
summary(m1 <- dynlm(Germany_data$Total ~ Germany_data$x, Germany_data$Year))
```

```
##
## Time series regression with "ts" data:
## Start = 1, End = 17
##
## Call:
## dynlm(formula = Germany_data$Total ~ Germany_data$x, data = Germany_data$Year)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1211044  -166431    17327   234881  1377285
##
## Coefficients:
```

```
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.860e+06  3.367e+05   5.526 5.82e-05 ***
## Germany_data$x 1.539e-02  1.659e-02   0.928    0.368
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 578300 on 15 degrees of freedom
## Multiple R-squared:  0.05424,    Adjusted R-squared:  -0.008807
## F-statistic: 0.8603 on 1 and 15 DF,  p-value: 0.3683
```

```
summary(m2 <- dynlm(diff_Germany_dataTotal ~ diff_Germany_datax))
```

```
##
## Time series regression with "numeric" data:
## Start = 1, End = 16
##
## Call:
## dynlm(formula = diff_Germany_dataTotal ~ diff_Germany_datax)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1193689  -234226   -54765   183611  1575194
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        6.296e+04  1.654e+05   0.381    0.709
## diff_Germany_datax 2.980e-02  4.267e-02   0.698    0.496
##
## Residual standard error: 612300 on 14 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.03366,    Adjusted R-squared:  -0.03536
## F-statistic: 0.4877 on 1 and 14 DF,  p-value: 0.4964
```

4.1 Forecasting Number of Future Asylum Seekers in Germany: Preparation

As before, we convert values to numeric, create an object that sums all origin countries to Germany by month, declare variables as time series variables

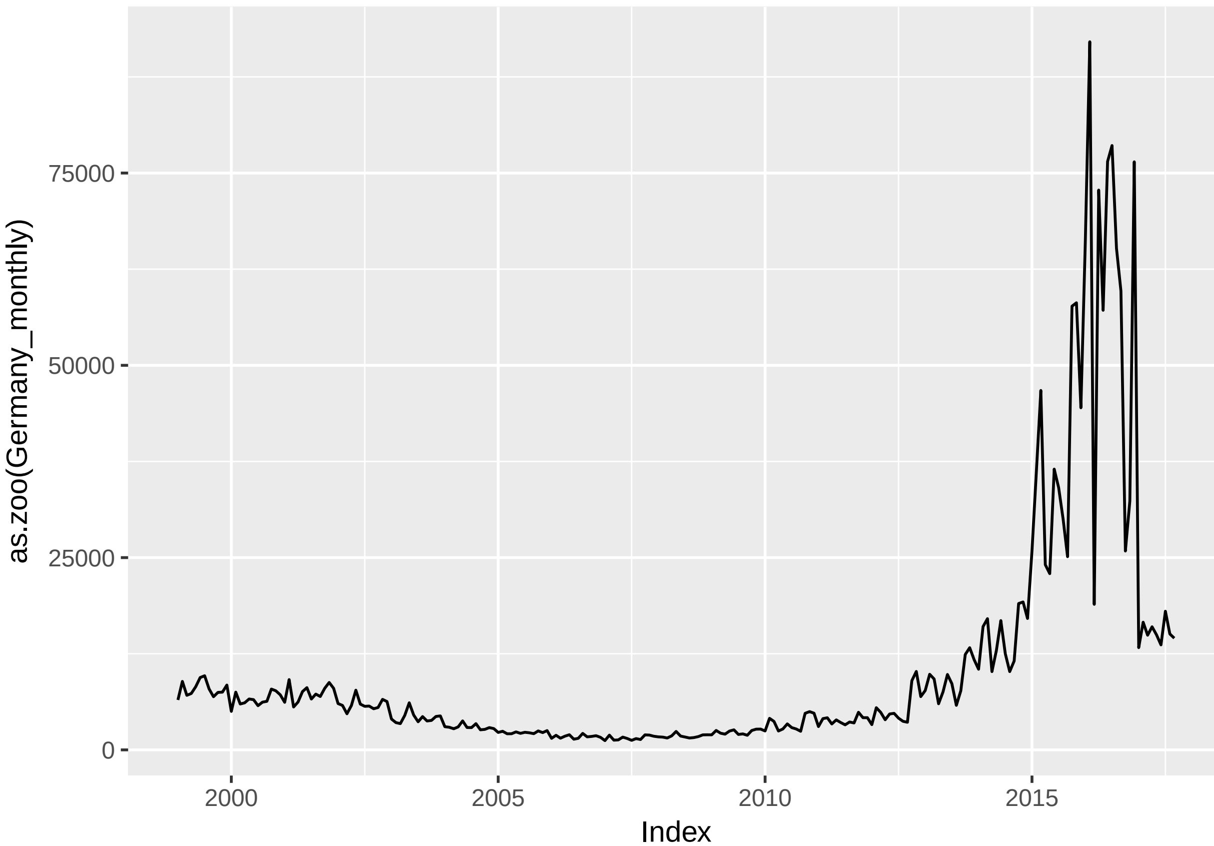
```
df3 <- read.csv("unhcr_popstats_export_asylum_seekers_monthly_2017_12_04_203715.csv",
                skip = 2, stringsAsFactors = F)

df3$Value[df3$Value=="*"] <- "0"
df3$Value <- as.numeric(df3$Value)

Germany_monthlyasylum_total <- df3 %>%
  group_by(Country...territory.of.asylum.residence, Year, Month) %>%
  summarise(Total = sum(Value))

Germany_monthly <- ts(Germany_monthlyasylum_total$Total,
                      start = c(1999, 1), frequency = 12)
```

4.2 Forecasting Number of Future Asylum Seekers in Germany: Test for Time Series Problems

Stationarity Test

Plot and observe trends

```
autoplot(as.zoo(Germany_monthly), geom = "line")
```



Persistence Test 1

After dynlm, row is <1 so it meets the stability condition for weak dependency)

```
summary(dynlm(Germany_monthly ~ L(Germany_monthly, 1)))
```

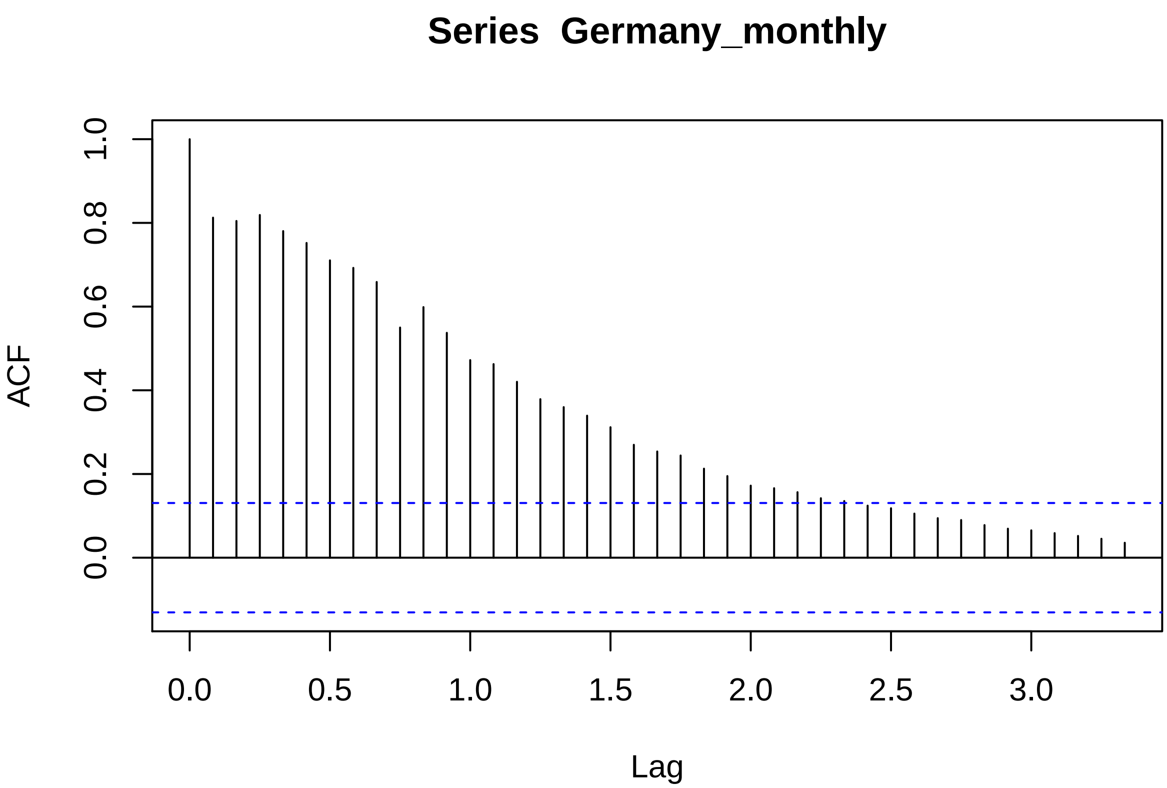
```
##
## Time series regression with "ts" data:
## Start = 1999(2), End = 2017(9)
##
## Call:
## dynlm(formula = Germany_monthly ~ L(Germany_monthly, 1))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -57837  -1677  -1197     32  55483
##
## Coefficients:
```

```
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1919.7596   722.1649   2.658  0.00842 **
## L(Germany_monthly, 1)    0.8129     0.0391  20.792  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9061 on 222 degrees of freedom
## Multiple R-squared:  0.6607, Adjusted R-squared:  0.6592
## F-statistic: 432.3 on 1 and 222 DF,  p-value: < 2.2e-16
```

Persistence Test 2

After acf, Germany monthly's correlation of lags drops to zero after 2.5 lags therefore it is not persistent

```
acf(Germany_monthly, na.action = na.pass, lag.max = 40)
```



Persistence Test 3

After Dickey Fuller Test for Unit Root, p value is <.05 then there is no unit root)

```
adf.test(Germany_monthly)
```

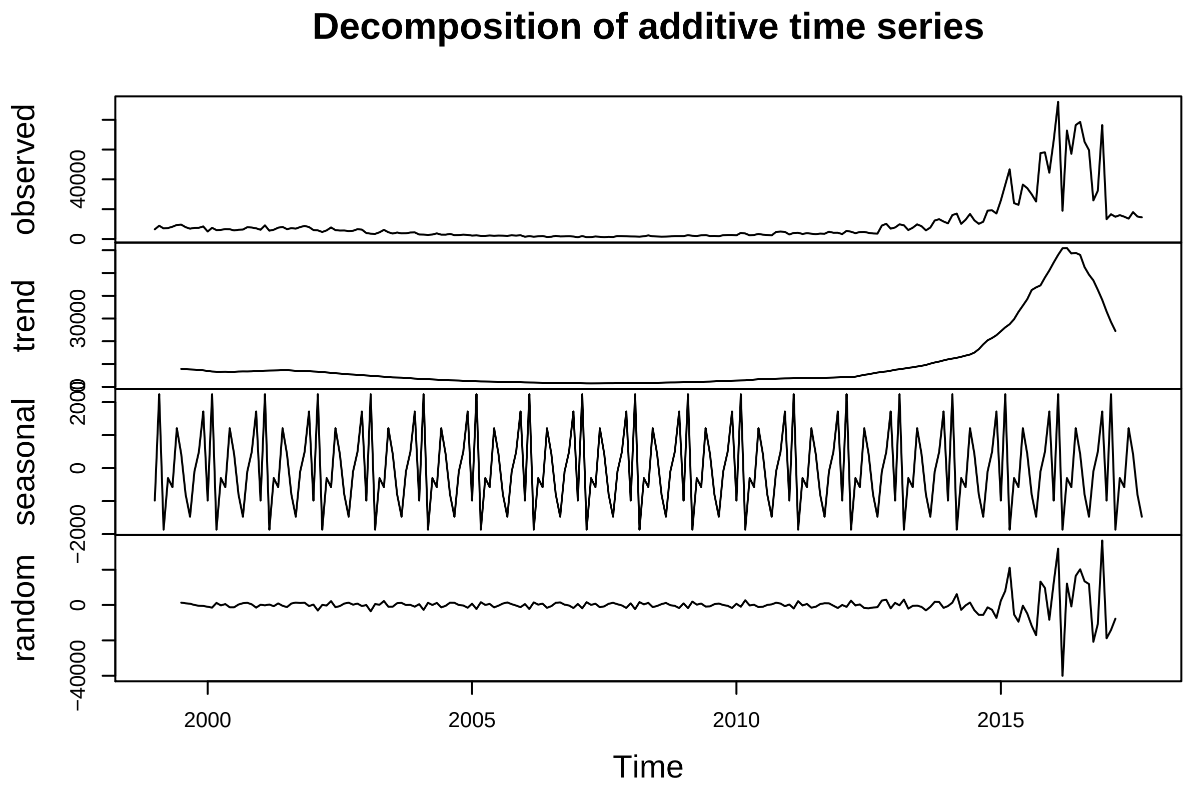
```
##
##  Augmented Dickey-Fuller Test
##
## data:  Germany_monthly
## Dickey-Fuller = -2.4282, Lag order = 6, p-value = 0.3961
## alternative hypothesis: stationary
```

4.3 Forecasting Number of Future Asylum Seekers in Germany

Decompose

Then we can decompose the additives of time series. This returns estimates of the seasonal component, trend component and irregular components or "random"

```
plot(decompose(Germany_monthly))
```



4.4 Forecasting Number of Future Asylum Seekers in Germany

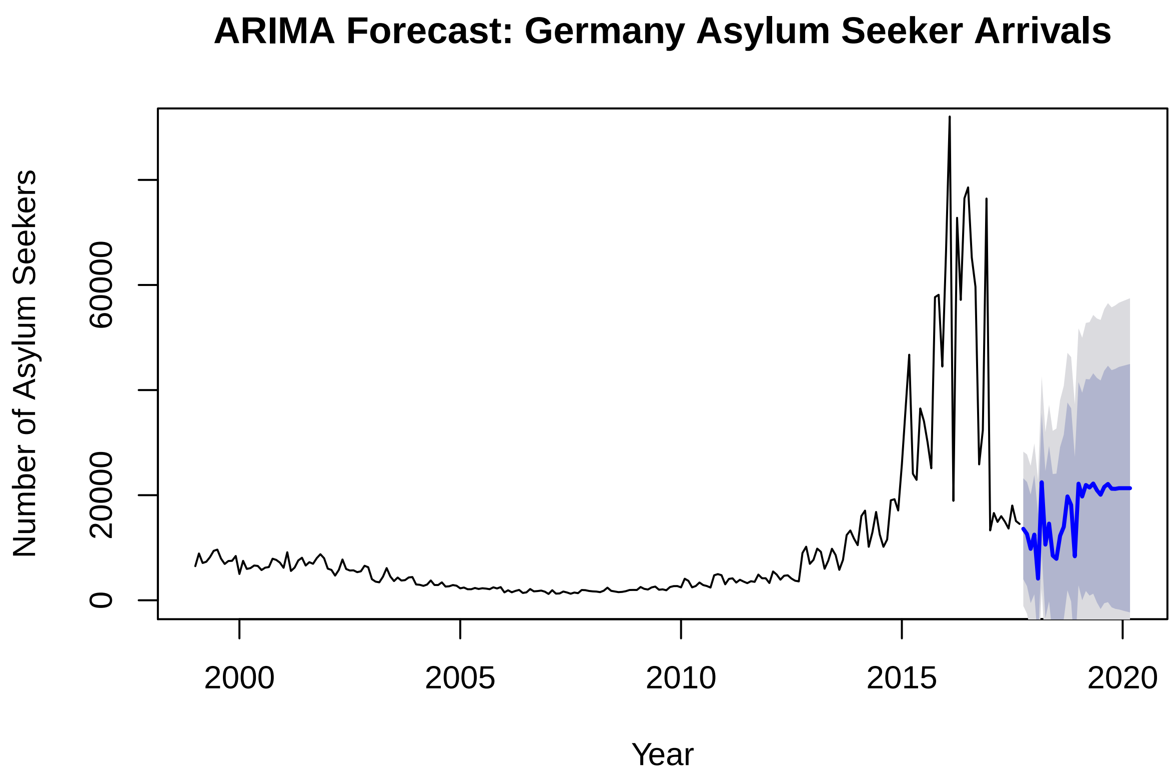
Seasonal Changes

To look more closely at the seasonal changes in the number of asylum seekers we use the "stl" function Germany has had a positive net flow of asylum seekers in June, July, November, December and the highest typically in February between 2000 and 2015

4.5 Forecasting Number of Future Asylum Seekers in Germany

The ARIMA forecasting method shows possible future changes in the number of asylum seekers in Germany in the next years The wide confidence intervals show the uncertainty in forecasting with the dark grey representing 95 percent confidence and the light grey representing 80 percent confidence

```
plot(forecast(auto.arima(Germany_monthly), 30),
     main = "ARIMA Forecast: Germany Asylum Seeker Arrivals",
     ylab = "Number of Asylum Seekers",
     xlab = "Year", ylim=c(0, 90000))
```



ARIMA Forecast Values

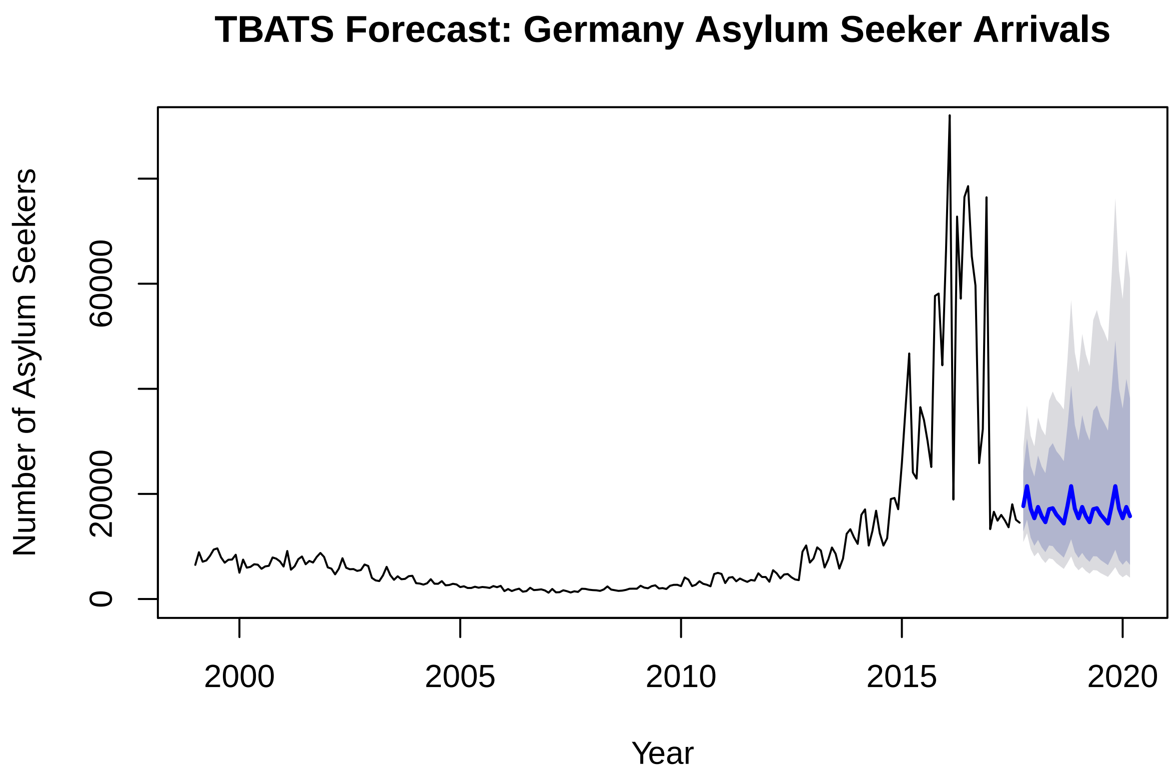
```r
forecast(auto.arima(Germany_monthly), 24)
```

```
##          Point Forecast        Lo 80    Hi 80      Lo 95    Hi 95
## Oct 2017      13600.045   4001.42590 23198.66  -1079.776 28279.87
## Nov 2017      12602.114   2690.66757 22513.56  -2556.135 27760.36
## Dec 2017       9783.970   -539.22427 20107.16  -6003.993 25571.93
## Jan 2018      12492.452   1138.14968 23846.75  -4872.455 29857.36
## Feb 2018       4125.247  -8174.02286 16424.52 -14684.863 22935.36
## Mar 2018      22453.966   9277.32413 35630.61   2302.031 42605.90
## Apr 2018      10594.123  -3405.01062 24593.26 -10815.704 32003.95
## May 2018      14587.126   -188.78676 29363.04  -8010.682 37184.93
## Jun 2018       8506.528  -7007.31837 24020.38 -15219.853 32232.91
## Jul 2018       7885.977  -8332.26208 24104.22 -16917.679 32689.63
## Aug 2018      12268.575  -4624.71179 29161.86 -13567.478 38104.63
## Sep 2018      13974.220  -3568.15671 31516.60 -12854.530 40802.97
## Oct 2018      19766.918   1900.25116 37633.58  -7557.791 47091.63
## Nov 2018      18144.497   -251.95305 36540.95  -9990.446 46279.44
## Dec 2018       8374.367 -10523.66592 27272.40 -20527.680 37276.41
## Jan 2019      22188.573   2857.62113 41519.53  -7375.567 51752.71
## Feb 2019      19746.164     -8.22251 39500.55 -10465.563 49957.89
## Mar 2019      21940.352   1771.41815 42109.29  -8905.370 52786.07
## Apr 2019      21460.855    885.72496 42035.98 -10006.091 52927.80
## May 2019      22219.733   1246.27226 43193.19  -9856.407 54295.87
## Jun 2019      20951.172   -413.19458 42315.54 -11722.807 53625.15
## Jul 2019      20087.011  -1661.23559 41835.26 -13174.062 53348.08
## Aug 2019      21572.478   -552.98949 43697.95 -12265.504 55410.46
```

```
## Sep 2019       22135.917   -360.44712 44632.28 -12269.303 56541.14
```

The TBATS forecasting method shows another possible future change in the number of asylum seekers in Germany in the next years

```r
plot(forecast(tbats(Germany_monthly), 30),
     main = "TBATS Forecast: Germany Asylum Seeker Arrivals",
     ylab = "Number of Asylum Seekers",
     xlab = "Year", ylim=c(0, 90000))
```



TBATS Forecast Values

```r
forecast(tbats(Germany_monthly), 24)
```

```
##          Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
## Oct 2017       17677.72 12843.324 24331.86 10844.966 28815.39
## Nov 2017       21482.21 15101.356 30559.19 12531.104 36827.17
## Dec 2017       17205.55 11693.651 25315.53  9531.590 31057.88
## Jan 2018       15371.73 10134.801 23314.71  8129.215 29066.77
## Feb 2018       17522.07 11242.393 27309.38  8888.643 34541.02
## Mar 2018       15753.26  9844.956 25207.34  7676.085 32329.65
## Apr 2018       14612.84  8911.857 23960.77  6859.269 31130.86
## May 2018       17094.15 10186.737 28685.32  7745.093 37728.38
## Jun 2018       17283.33 10070.144 29663.29  7565.720 39482.50
## Jul 2018       16052.74  9158.985 28135.25  6805.177 37866.81
## Aug 2018       15243.82  8521.070 27270.52  6262.908 37103.21
## Sep 2018       14373.66  7877.165 26227.99  5729.283 36060.75
## Oct 2018       17677.72  9519.380 32827.97  6859.701 45556.21
## Nov 2018       21482.21 11365.829 40602.86  8114.143 56874.17
## Dec 2018       17205.55  8929.264 33152.89  6309.918 46915.19
```

```
## Jan 2019       15371.73  7832.675 30167.21  5481.564 43106.30
## Feb 2019       17522.07  8776.609 34981.94  6086.666 50441.87
## Mar 2019       15753.26  7756.225 31995.61  5330.319 46557.28
## Apr 2019       14612.84  7077.428 30171.27  4821.688 44286.35
## May 2019       17094.15  8148.019 35862.68  5504.322 53087.34
## Jun 2019       17283.33  8108.219 36840.83  5431.517 54996.34
## Jul 2019       16052.74  7417.964 34738.70  4929.540 52274.73
## Aug 2019       15243.82  6939.160 33487.34  4574.840 50793.91
## Sep 2019       14373.66  6447.303 32044.75  4217.526 48986.59
```