Part II III IV - Maryland Poverty Level

Lebo Sango

2024-04-24

Data Cleaning

```
maryland_raw = read.csv("/Users/lebsan/Documents/STAT 5084 - Time Series/County Level Project/Universe_I

maryland =maryland_raw %>%
  mutate( SAIPE = as.numeric(SAIPE),SNAP = as.numeric(SNAP),
          IRS_exempt_State = as.numeric(IRS_exempt_State),
          Poverty_Universe = as.numeric(Poverty_Universe)) %>%
  select(c(year,County, SAIPE, SNAP, IRS_exempt_State, Poverty_Universe)) %>%
  as_tsibble(index =year, key = County )

maryland %>% head(5)
```

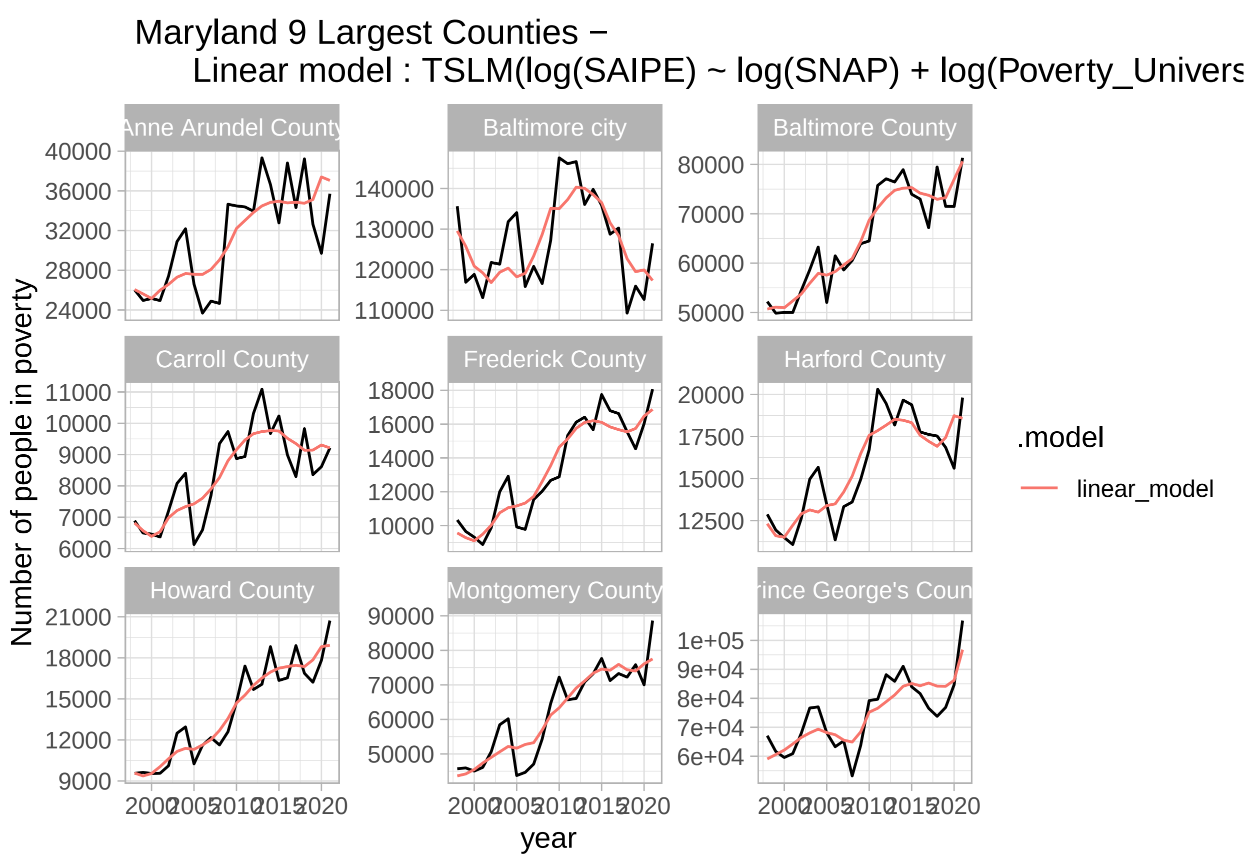
```
## # A tsibble: 5 x 6 [1Y]
## # Key:       County [1]
##    year County          SAIPE  SNAP IRS_exempt_State Poverty_Universe
##   <int> <chr>           <dbl> <dbl>            <dbl>            <dbl>
## 1  1998 Allegany County 10473  6650           472945            69532
## 2  1999 Allegany County  9270  6294           468976            69404
## 3  2000 Allegany County  9445  5922           465555            68408
## 4  2001 Allegany County  8954  6365           475208            68151
## 5  2002 Allegany County  9418  6864           487317            67632
```

Linear models

```
## # A tibble: 7 x 4
##   Linear_models    AIC    CV    BIC
##   <chr>          <dbl> <dbl>  <dbl>
## 1 model5        -2547. 0.277 -2462.
## 2 model7        -2559. 0.286 -2474.
## 3 model4        -2665. 0.236 -2552.
## 4 model2        -2808. 0.184 -2695.
## 5 model1        -2861. 0.173 -2719.
## 6 model3        -2823. 0.174 -2738.
## 7 model6        -2878. 0.163 -2764.
```

Lower cross validation and BIC model 6 is the best model. SNAP and Poverty Universe displayed a strong correlation coefficient with SAIPE. In addition, model 1 including all the dependent variable had very close precision crieteria compared to model6. I decided to proceed with model 6 because two of the precision criteria were the smallest. This distinction can be caused by the low correlation coeffient of 0.118 betwen SAIPE and IRS_exempt_State which hinder the model.

```r
# Plot of the fitted predictions of the nine biggest counties with the best linear model.
maryland %>%
  filter(County %in% c("Montgomery County", "Prince George's County"
                      , "Baltimore County", "Anne Arundel County",
                      "Baltimore city", "Howard County",
                      "Frederick County", "Harford County",
                      "Carroll County" ) ) %>%
  model(linear_model = TSLM(log(SAIPE) ~ log(SNAP) + log(Poverty_Universe))) %>%
  augment() %>% ggplot(aes(x=year))+
  geom_line(aes(y=SAIPE)) +
  geom_line(aes(y=.fitted, color=.model)) +
  facet_wrap(.~County, scales = "free_y")+
  labs(title = " Maryland 9 Largest Counties -
       Linear model : TSLM(log(SAIPE) ~ log(SNAP) + log(Poverty_Universe))",
       y=" Number of people in poverty")+
  theme_light()
```




```r
# Residual plot of the nine largest counties
MD_resid = maryland %>% filter(County %in% c("Montgomery County", "Prince George's County"
                      , "Baltimore County", "Anne Arundel County",
                      "Baltimore city", "Howard County",
                      "Frederick County", "Harford County",
                      "Carroll County" ) ) %>%
```

```r
  model(model6 = TSLM(log(SAIPE) ~ log(SNAP) + log(Poverty_Universe))) %>%
  augment()
MD_resid %>%  autoplot(.innov)
```


```r
# LjungBox test on every county of Maryland state

MD_resid2 = maryland %>%
  model(model6 = TSLM(log(SAIPE) ~ log(SNAP) + log(Poverty_Universe))) %>%
  augment()

MD_resid2 %>% select(County ,.model,.innov) %>% group_by(County) %>%
  features(.innov, ljung_box) %>% filter(lb_pvalue <= 0.05)
```

```
## # A tibble: 4 x 4
##   County                 .model lb_stat lb_pvalue
##   <chr>                  <chr>    <dbl>     <dbl>
## 1 Cecil County           model6    8.87   0.00291
## 2 Dorchester County      model6    5.53   0.0187
## 3 Prince George's County model6    5.32   0.0211
## 4 Talbot County          model6    4.59   0.0322
```

The only counties that do have white noise are Prince George's county, Talbot County, Dorchester County and Cecil county while the rest does not exhibits autocorrelation. Overall the model does better at capturing the trend but fails to capture cyclicalities. Furthermore, I expect to employ more sophisticated models that can capture the cyclicalities and fluctuations of SAIPE.

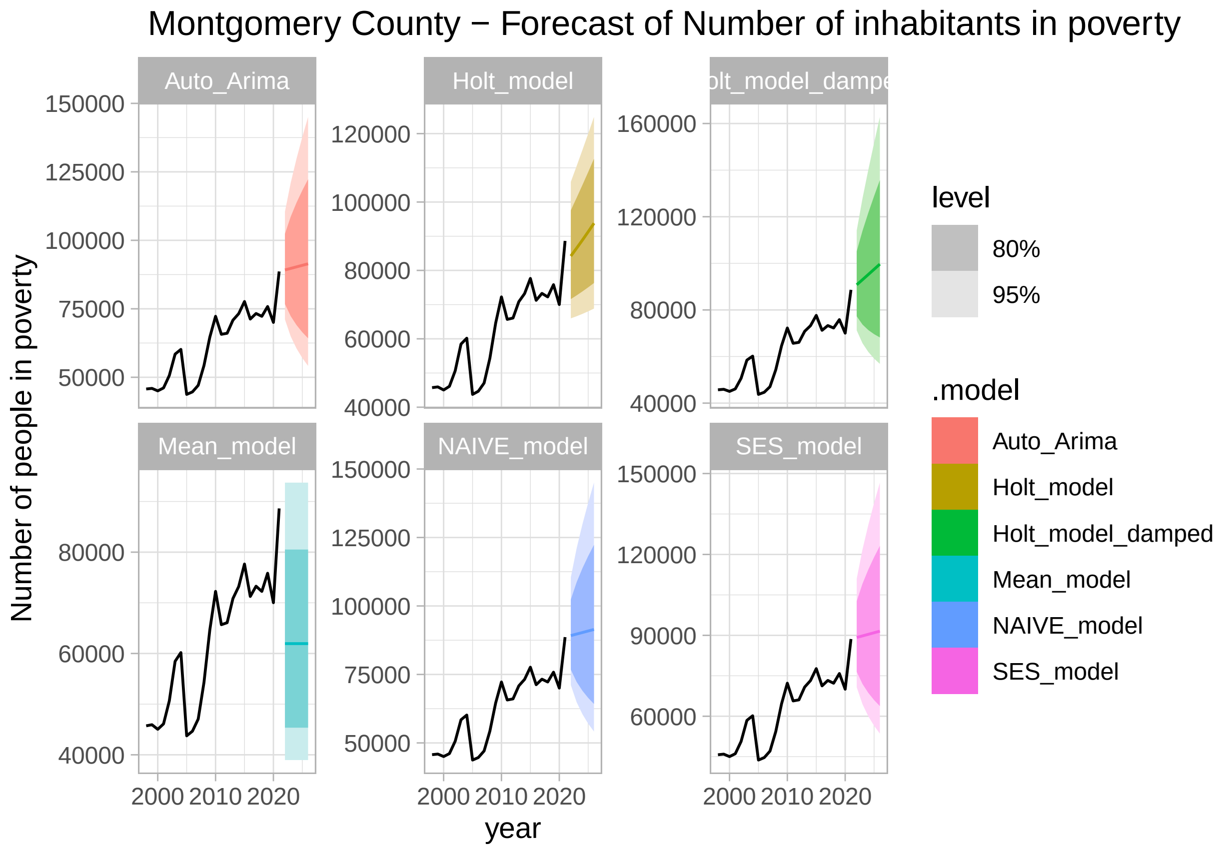
Part 3 - Stochastic Models

Single County Forecasts

```
stochastic_model = maryland %>% filter(County %in% "Montgomery County") %>%
  model(
                NAIVE_model = NAIVE(log(SAIPE)),

                Mean_model = MEAN(log(SAIPE)),

                SES_model = ETS( log(SAIPE) ~ error("A")+trend("N")+
                                   season("N")),

                Holt_model = ETS(log(SAIPE) ~ error("A")+trend("A")+
                                  season("N")),

                Holt_model_damped = ETS(log(SAIPE) ~ error("A")+trend("Ad")+
                                         season("N")),

                Auto_Arima = ARIMA(log(SAIPE)))
```

Plotting the number in poverty data along with a five-year forecast

```
stochastic_model %>% forecast(h="5 years") %>% autoplot(maryland)+
  facet_wrap(~.model, scales = "free_y")+
  theme_light()+
  labs(title = " Montgomery County - Forecast of Number of inhabitants in poverty",
       y = "Number of people in poverty")
```



The best model for this county is the Auto Arima with a low root mean square error and mean average percentage error.

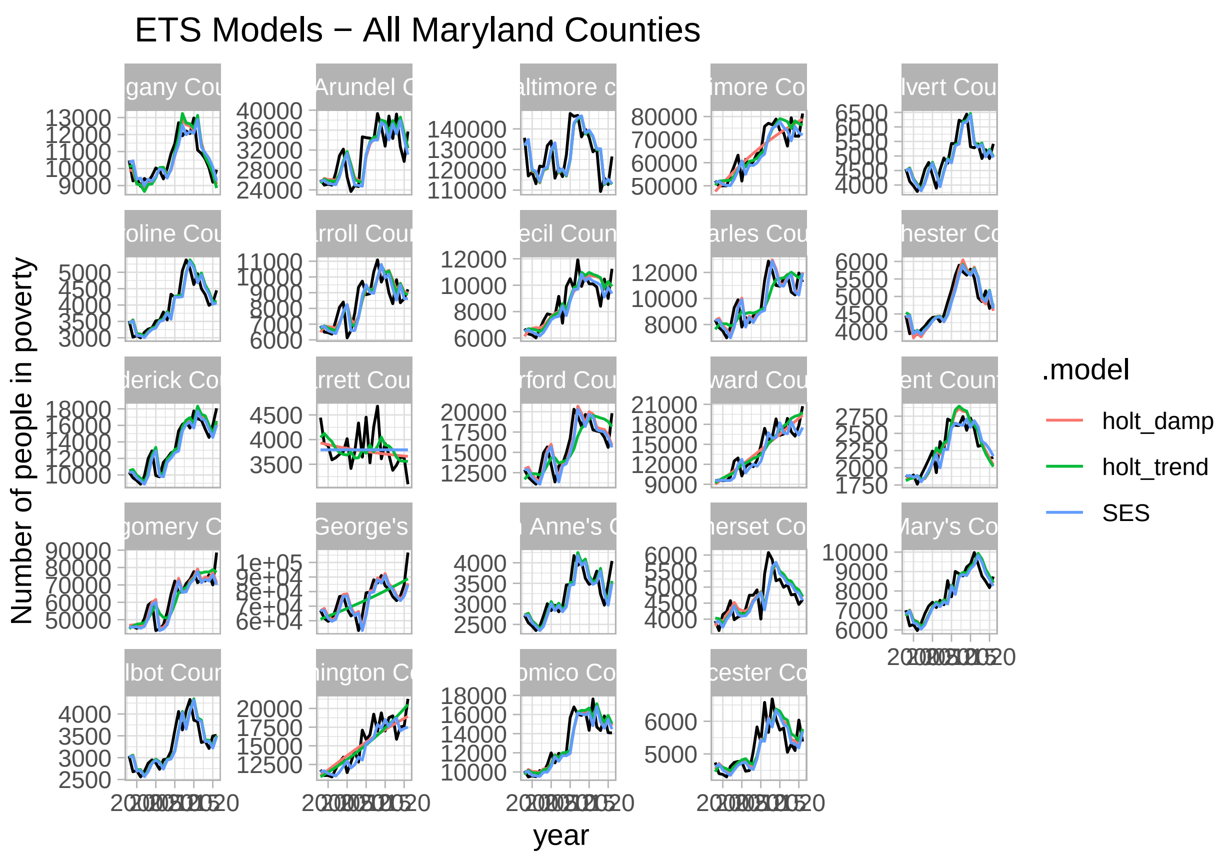
```
# The auto arima is the model that exhibits the smallest RMSE accross Maryland counties.
stochastic_model %>% accuracy() %>%
  group_by( stochatic_models = .model, Maryland_County = County) %>%
  summarise(RMSE = sum(RMSE), MAPE = sum(MAPE)) %>%
  arrange(min(RMSE))
```

```
## `summarise()` has grouped output by 'stochatic_models'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 6 x 4
## # Groups:   stochatic_models [6]
##   stochatic_models  Maryland_County     RMSE  MAPE
##   <chr>             <chr>              <dbl> <dbl>
## 1 Auto_Arima        Montgomery County  6804.  7.88
## 2 Holt_model        Montgomery County  6561.  8.48
## 3 Holt_model_damped Montgomery County  6640.  7.42
## 4 Mean_model        Montgomery County 12979. 19.7
## 5 NAIVE_model       Montgomery County  6950.  8.18
## 6 SES_model         Montgomery County  6804.  7.84
```

Exponential Smoothing Models

```r
ES_maryland = maryland  %>% model(

                  SES = ETS(log(SAIPE)~error("A") + trend("N") +season("N")),

                  holt_trend = ETS(log(SAIPE)~error("A")+trend("A")+season("N")),

                  holt_damp = ETS(log(SAIPE)~error("A")+trend("Ad")+season("N")))

# Fitting the ETS models into every country of the state of Maryland
ES_maryland %>%
  augment() %>% ggplot(aes(x=year))+
  geom_line(aes(y=SAIPE)) +
  geom_line(aes(y=.fitted, color=.model)) +
  facet_wrap(~County, scales = "free_y") +
  labs(title = " ETS Models - All Maryland Counties",
       y = " Number of people in poverty")+
  theme_light()
```



Best Performing ETS Model

I selected the Holt_damped model because per the results below , it shows the smallest RSME and MAPE values against the other models. One particularity of the holt damped model is its addtive trend feature which implies that long run forecast as h approaches infinity, the damping parameter will be constant , while in the short- forecast will be trended.

```r
ES_maryland %>% accuracy() %>%
  group_by(Exponential_smoothing_model = .model) %>%
  summarise(RMSE = sum(RMSE),
            MAPE = sum(MAPE)) %>%
  arrange(min(RMSE))
```

```
## # A tibble: 3 x 3
##   Exponential_smoothing_model   RMSE  MAPE
##   <chr>                        <dbl> <dbl>
## 1 SES                         48970.  175.
## 2 holt_damp                   46967.  174.
## 3 holt_trend                  48702.  180.
```

ARIMA Models

The most commonly selected model is the ARIMA(0,1,0) evaluated at difference, the ARIMA(1,0,0) with a mean constant, and ARIMA(0,1,1) with a drift.

```r
# The most selected ARIMA model is the model evaluated at difference
maryland %>% model(ARIMA(log(SAIPE)))
```

```
## # A mable: 24 x 2
## # Key:     County [24]
##    County                  `ARIMA(log(SAIPE))`
##    <chr>                               <model>
##  1 Allegany County       <ARIMA(1,0,0) w/ mean>
##  2 Anne Arundel County           <ARIMA(0,1,0)>
##  3 Baltimore County     <ARIMA(0,1,1) w/ drift>
##  4 Baltimore city        <ARIMA(1,0,0) w/ mean>
##  5 Calvert County                <ARIMA(0,1,0)>
##  6 Caroline County               <ARIMA(0,1,0)>
##  7 Carroll County                <ARIMA(0,1,0)>
##  8 Cecil County                  <ARIMA(1,1,0)>
##  9 Charles County                <ARIMA(0,1,0)>
## 10 Dorchester County             <ARIMA(0,1,0)>
## # i 14 more rows
```

Best Performing Arima model

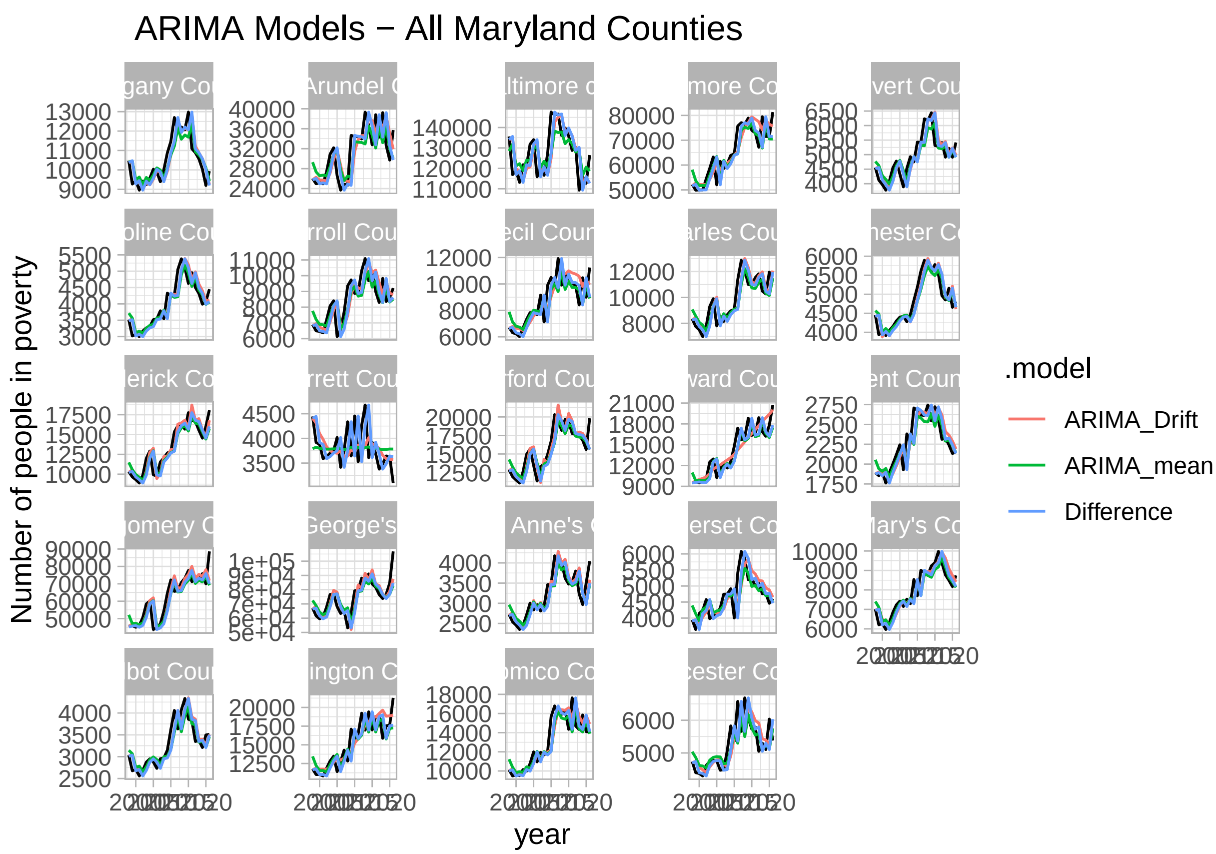
The best ARIMA model is the model with the drift with the smallest RMSE and MAPE. Moreover, the model forecast follows a straight line. That is , the forecast indicates the number of people in poverty is increasing as the trned is sloping upward, so the constant is non-zero and d is 1.

```r
maryland %>% model(Difference = ARIMA(log(SAIPE) ~ pdq(0,1,0)),
                   ARIMA_Drift = ARIMA(log(SAIPE) ~ 1 + pdq(0,1,1)),
                   ARIMA_mean = ARIMA(log(SAIPE) ~ 1 + pdq(1,0,0))) %>%
  accuracy() %>%

  group_by(Arima_models = .model) %>%
```

```r
  summarise(RMSE = sum(RMSE),
            MAPE = sum(MAPE)) %>%

  arrange(min(RMSE))
```

```
## # A tibble: 3 x 3
##   Arima_models   RMSE  MAPE
##   <chr>         <dbl> <dbl>
## 1 ARIMA_Drift  47456.  174.
## 2 ARIMA_mean   48611.  178.
## 3 Difference   50380.  182.
```

```r
# Fitting the ARIMA models to every county

maryland %>% model(Difference = ARIMA(log(SAIPE) ~ pdq(0,1,0)),

                   ARIMA_Drift = ARIMA(log(SAIPE) ~ 1 + pdq(0,1,1)),

                   ARIMA_mean = ARIMA(log(SAIPE) ~ 1 + pdq(1,0,0))) %>%

  augment() %>% ggplot(aes(x=year))+

  geom_line(aes(y=SAIPE)) +

  geom_line(aes(y=.fitted, color=.model)) +

  facet_wrap(~County, scales = "free_y") +

  labs(title = " ARIMA Models - All Maryland Counties",
       y = " Number of people in poverty")+

  theme_light()
```



Cross validation

The best model is the ARIMA evaluated at difference or ARIMA(0,1,0)

ETS

```
# Building Training Sets
ES_maryland_stretch = maryland %>% stretch_tsibble(.init = 10)

ES_maryland_training = ES_maryland_stretch  %>% model(

                   SES = ETS(log(SAIPE)~error("A") + trend("N") +season("N")),

                   holt_trend = ETS(log(SAIPE)~error("A")+trend("A")+season("N")),

                   holt_damp = ETS(log(SAIPE)~error("A")+trend("Ad")+season("N")))
```

```
# Accuracy Check
ES_maryland_training %>% accuracy() %>%
  group_by(ETS_Models = .model) %>%
  summarise( RMSE = sum(RMSE)) %>%
  arrange(RMSE)
```

```
## # A tibble: 3 x 2
##   ETS_Models    RMSE
##   <chr>        <dbl>
## 1 holt_damp  630247.
## 2 holt_trend 646244.
## 3 SES        668244.
```

```
# Cross validation between training and test data
ES_maryland_training %>%
forecast(h="5 years") %>%
  accuracy(maryland) %>%
  group_by(ETS_Models = .model) %>%
  summarise(RMSE = sum(RMSE)) %>%
  arrange(RMSE)
```

```
## # A tibble: 3 x 2
##   ETS_Models   RMSE
##   <chr>       <dbl>
## 1 SES        85069.
## 2 holt_damp  88696.
## 3 holt_trend 96787.
```

ARIMA

```
ARIMA_maryland_strech = maryland %>% stretch_tsibble(.init = 10)

ARIMA_maryland_training = ARIMA_maryland_strech %>% model(Difference = ARIMA(log(SAIPE) ~ pdq(0,1,0)),
                   ARIMA_Drift = ARIMA(log(SAIPE) ~ 1 + pdq(0,1,1)),
                   ARIMA_mean = ARIMA(log(SAIPE) ~ 1 + pdq(1,0,0)))
```

```
# Accuracy Check

ARIMA_maryland_training %>% accuracy() %>%
  group_by(ARIMA_Models = .model) %>%
  summarise( RMSE = sum(RMSE)) %>%
  arrange(RMSE)
```

```
## # A tibble: 3 x 2
##   ARIMA_Models    RMSE
##   <chr>          <dbl>
## 1 ARIMA_Drift  625396.
## 2 Difference   692686.
## 3 ARIMA_mean      NaN
```

```
# Cross validation between training and test data
ARIMA_maryland_training %>%
forecast(h="5 years") %>%
  accuracy(maryland) %>%
  group_by(ARIMA_Models = .model) %>%
  summarise(RMSE = sum(RMSE)) %>%
  arrange(RMSE)
```

```
## # A tibble: 3 x 2
##   ARIMA_Models   RMSE
##   <chr>         <dbl>
## 1 Difference   77983.
## 2 ARIMA_mean   89593.
## 3 ARIMA_Drift  91901.
```

Forecasts

The 5 counties with the largest increase in poverty level in the next 5 years are Somerset, Baltimore city, Allegany, Dorechester, and Washington counties.

```r
# Forecasting poverty
poverty_forecast = maryland %>%
  model(Arima_Diff = ARIMA(log(SAIPE) ~ pdq(0,1,0))) %>%
  forecast(h="5 years")

# Extracting current population which is in 2021 for every county
current_population = maryland %>%  filter(year == 2021) %>%
  select(County, Poverty_Universe)


# Join current populatin in 2021 with forecast data
merge(current_population,poverty_forecast, by = c("County")) %>%
  mutate(Poverty_Percent_change = .mean/Poverty_Universe * 100) %>%
  group_by(County) %>% summarise(Poverty_level_Prct = max(Poverty_Percent_change)) %>%
  arrange(desc(Poverty_level_Prct))
```

```
## # A tibble: 24 x 2
##    County                 Poverty_level_Prct
##    <chr>                               <dbl>
##  1 Somerset County                      24.2
##  2 Baltimore city                       23.2
##  3 Allegany County                      16.6
##  4 Dorchester County                    15.1
##  5 Washington County                    15.1
##  6 Wicomico County                      14.5
##  7 Caroline County                      13.7
##  8 Kent County                          12.2
##  9 Prince George's County               11.8
## 10 Cecil County                         11.5
## # i 14 more rows
```