Anne Arundel County Population Estimate Forecast

Rick Fisher

September 4, 2018

Load needed library files

```
library(forecast)
library(ggplot2)
library(readxl)
```

Import population data from Excel

```
aaPopR <- read_excel("C:/RStats/Forecast/Data/aaPop.xlsx")
```

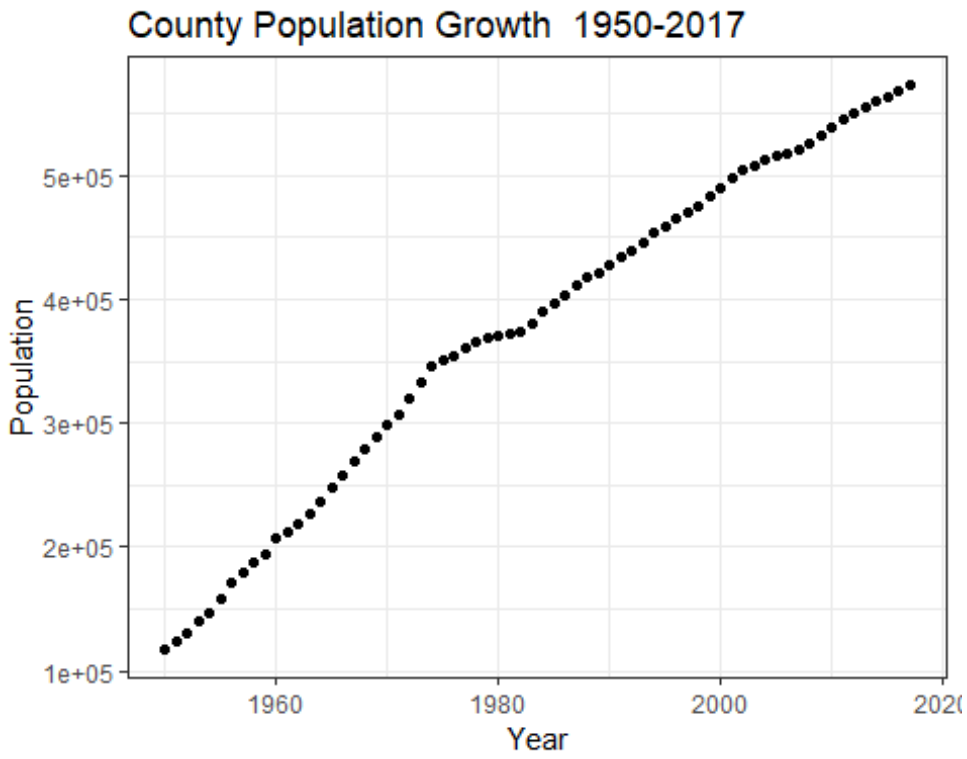
Preview first few columns of data to verify it was loaded into memory properly

```
head(aaPopR)
```

```
## # A tibble: 6 x 2
##    YEAR    POP
##   <dbl>  <dbl>
## 1  1950 117392
## 2  1951 123511
## 3  1952 130385
## 4  1953 140586
## 5  1954 147333
## 6  1955 157516
```

Plot the data

```
aaPopPlot <- ggplot(aaPopR, aes(x = YEAR, y = POP)) +
  geom_point() +
  ggtitle("County Population Growth  1950-2017") +
  scale_x_continuous(name = "Year") +
  scale_y_continuous(name = "Population") +
  theme_bw()
print (aaPopPlot)
```



Run a simple regression model plotting population vs year

```
aaPopReg <- lm(POP ~ YEAR, aaPopR)
```

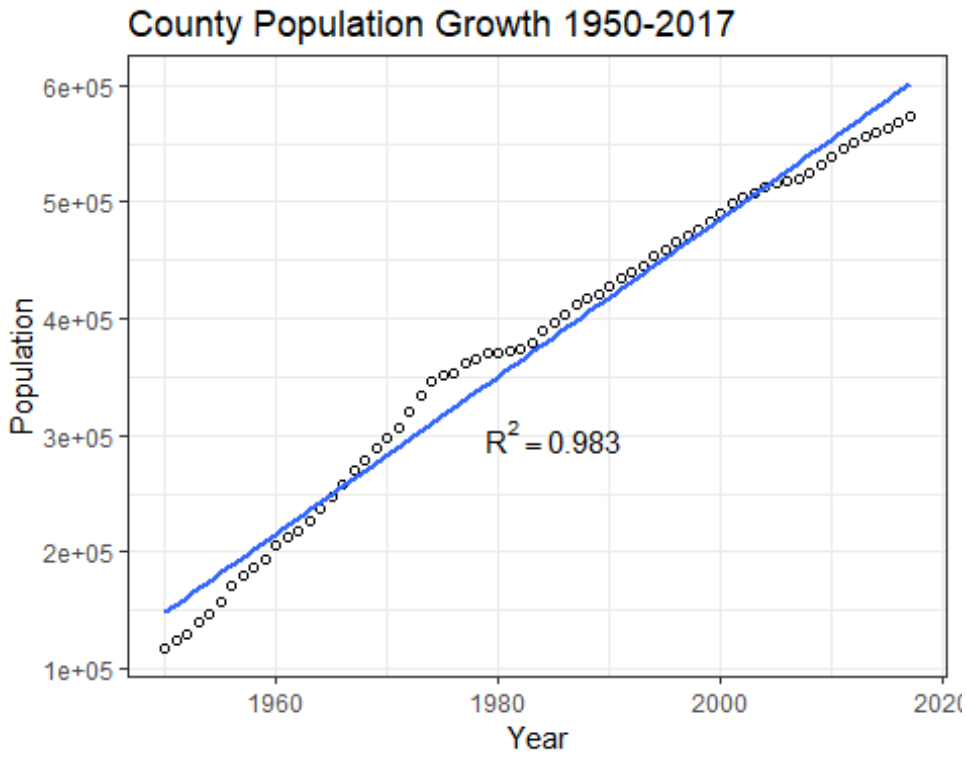
View regression model summary

```
summary(aaPopReg)
```

```
##
## Call:
## lm(formula = POP ~ YEAR, data = aaPopR)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -31352 -14705   3410  11793  35635
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.304e+07  2.179e+05  -59.84   <2e-16 ***
## YEAR         6.763e+03  1.098e+02   61.57   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17780 on 66 degrees of freedom
## Multiple R-squared:  0.9829, Adjusted R-squared:  0.9826
## F-statistic:  3790 on 1 and 66 DF,  p-value: < 2.2e-16
```

Visualize the regression model

```
aaPopPlot <- ggplot(aaPopR, aes(x = YEAR, y = POP)) +
    geom_point(shape = 1) +
    geom_smooth(method = 'lm', se = FALSE) +
    ggtitle("County Population Growth 1950-2017") +
    scale_x_continuous(name = "Year") +
    scale_y_continuous(name = "Population") +
    annotate("text", x=1985, y=300000, label = "R^2 == 0.983", parse=T) +
    theme_bw()
  print(aaPopPlot)
```

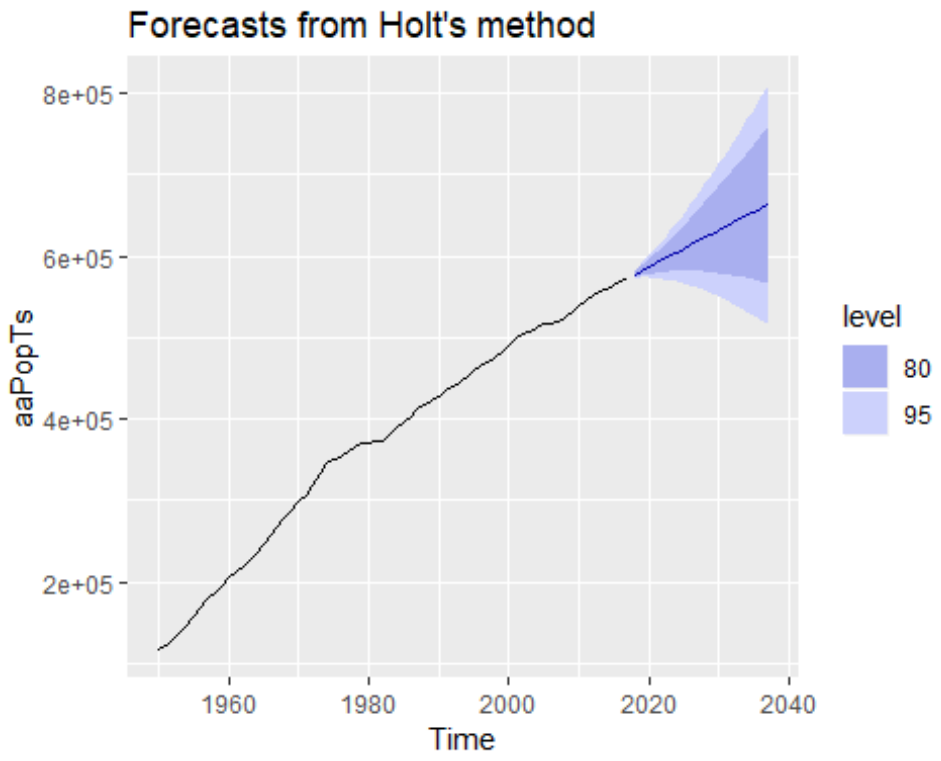


Create a time series object

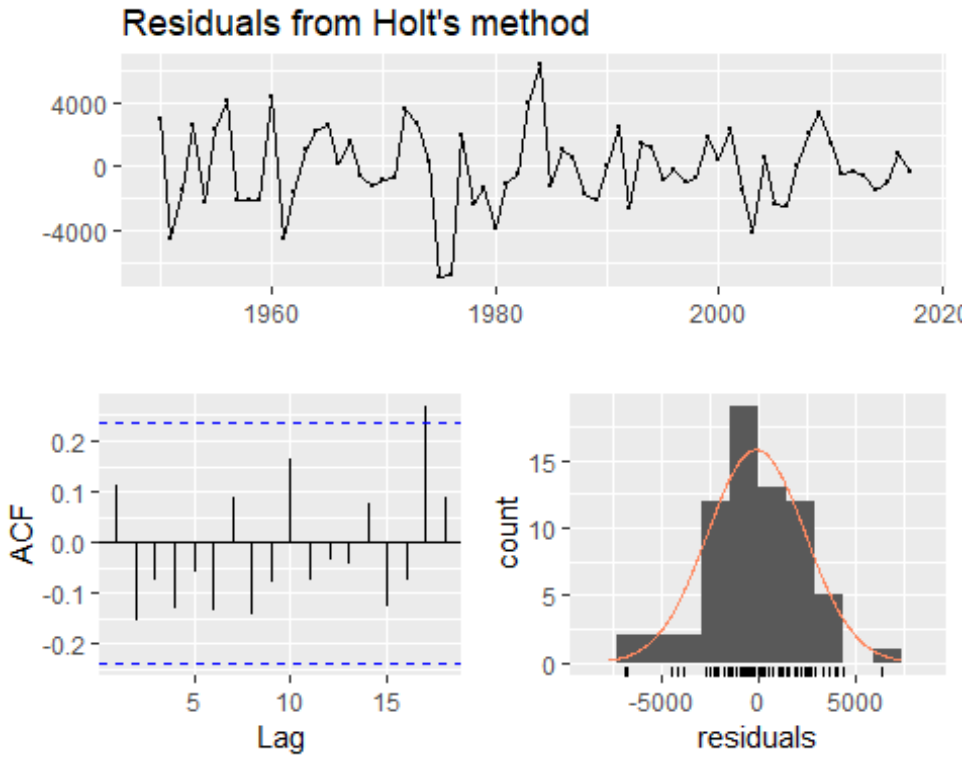
```
aaPopTs <- ts(aaPopR[2], start = c(1950,1), frequency = 1)
```

Forecast a 20 year trend using Holt's method

```
fcPopHolt <- holt(aaPopTs, h = 20)
autoplot(fcPopHolt)
```




```
checkresiduals(fcPopHolt)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Holt's method
## Q* = 11.099, df = 6, p-value = 0.08537
##
## Model df: 4.   Total lags used: 10
```

```
summary(fcPopHolt)
```

```
##
## Forecast method: Holt's method
##
## Model Information:
## Holt's method
##
## Call:
##  holt(y = aaPopTs, h = 20)
##
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 0.5055
##
##   Initial states:
##     l = 105320.1615
##     b = 9161.702
##
##   sigma:  2571.861
##
##      AIC     AICc      BIC
## 1360.728 1361.696 1371.826
##
## Error measures:
##                       ME     RMSE     MAE         MPE      MAPE     MASE
## Training set -135.3381 2495.072 1961.86 -0.03563866 0.7006412 0.288355
##                    ACF1
## Training set 0.1132046
##
## Forecasts:
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2018       577745.1 574449.1 581041.0 572704.3 582785.8
## 2019       582255.1 576298.5 588211.7 573145.2 591364.9
## 2020       586765.1 577854.1 595676.1 573136.9 600393.2
## 2021       591275.1 579101.8 603448.3 572657.7 609892.5
## 2022       595785.1 580056.8 611513.4 571730.7 619839.5
## 2023       600295.1 580736.7 619853.5 570383.1 630207.1
## 2024       604805.1 581157.9 628452.4 568639.8 640970.5
## 2025       609315.1 581334.5 637295.8 566522.5 652107.8
## 2026       613825.2 581279.0 646371.3 564050.1 663600.2
## 2027       618335.2 581002.2 655668.1 561239.3 675431.0
```

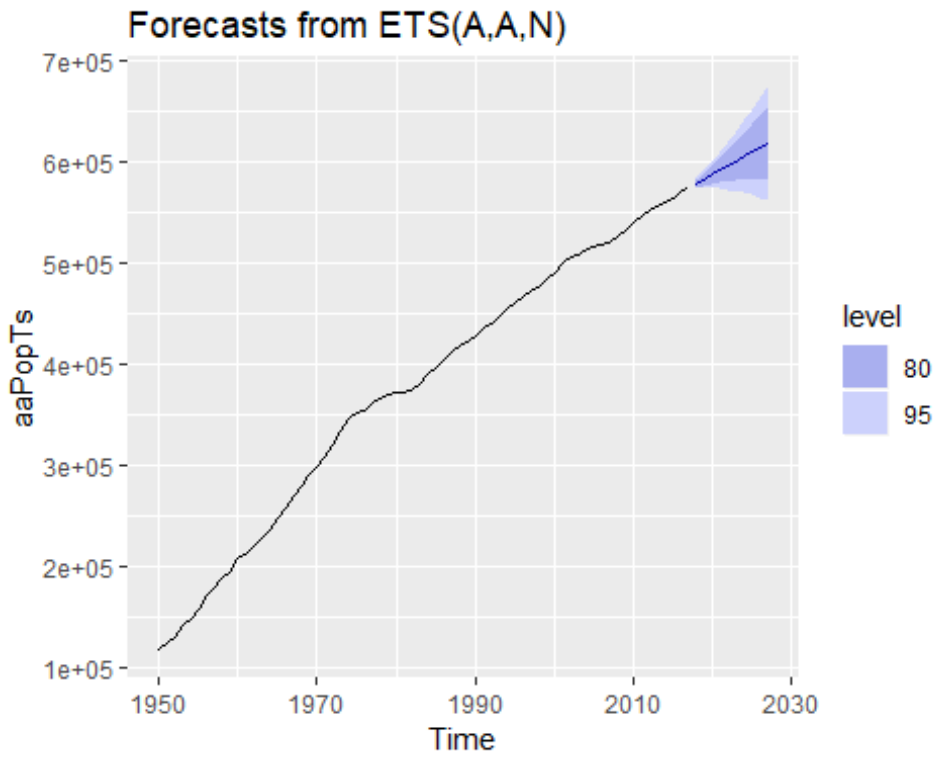
```
## 2028       622845.2 580513.6 665176.8 558104.6 687585.7
## 2029       627355.2 579821.6 674888.8 554658.9 700051.5
## 2030       631865.2 578933.8 684796.6 550913.6 712816.8
## 2031       636375.2 577856.9 694893.5 546879.2 725871.3
## 2032       640885.2 576597.0 705173.4 542564.9 739205.5
## 2033       645395.2 575159.8 715630.7 537979.4 752811.1
## 2034       649905.3 573550.3 726260.2 533130.4 766680.1
## 2035       654415.3 571773.2 737057.4 528025.1 780805.4
## 2036       658925.3 569832.8 748017.8 522670.1 795180.5
## 2037       663435.3 567733.2 759137.4 517071.5 809799.1
```

Validate the use of Holt's forecast method by comparing results to ETS.

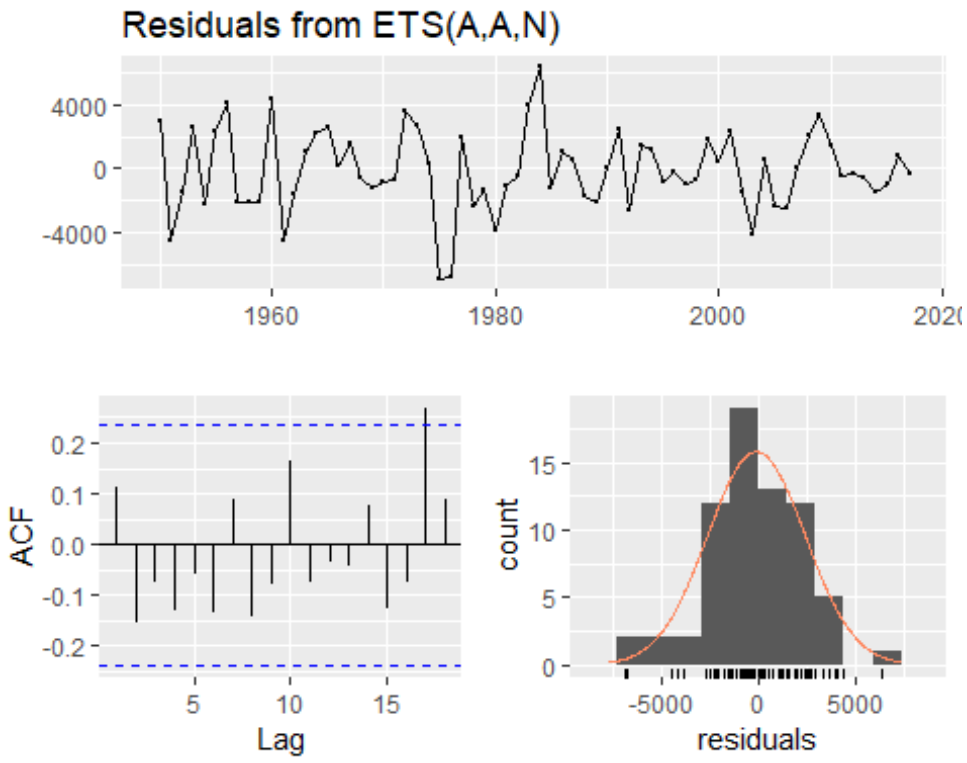
```
fitaaPop <- ets(aaPopTs)
summary(fitaaPop)
```

```
## ETS(A,A,N)
##
## Call:
##  ets(y = aaPopTs)
##
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 0.5054
##
##   Initial states:
##     l = 105320.1597
##     b = 9161.7018
##
##   sigma:  2571.861
##
##      AIC     AICc      BIC
## 1360.728 1361.696 1371.826
##
## Training set error measures:
##                      ME     RMSE      MAE          MPE      MAPE     MASE
## Training set -135.3577 2495.072 1961.853 -0.03564391 0.7006366 0.288354
##                    ACF1
## Training set 0.1132628
```

```
autoplot(forecast(fitaaPop))
```




```
checkresiduals(fitaaPop)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,A,N)
## Q* = 11.099, df = 6, p-value = 0.08537
##
## Model df: 4.   Total lags used: 10
```