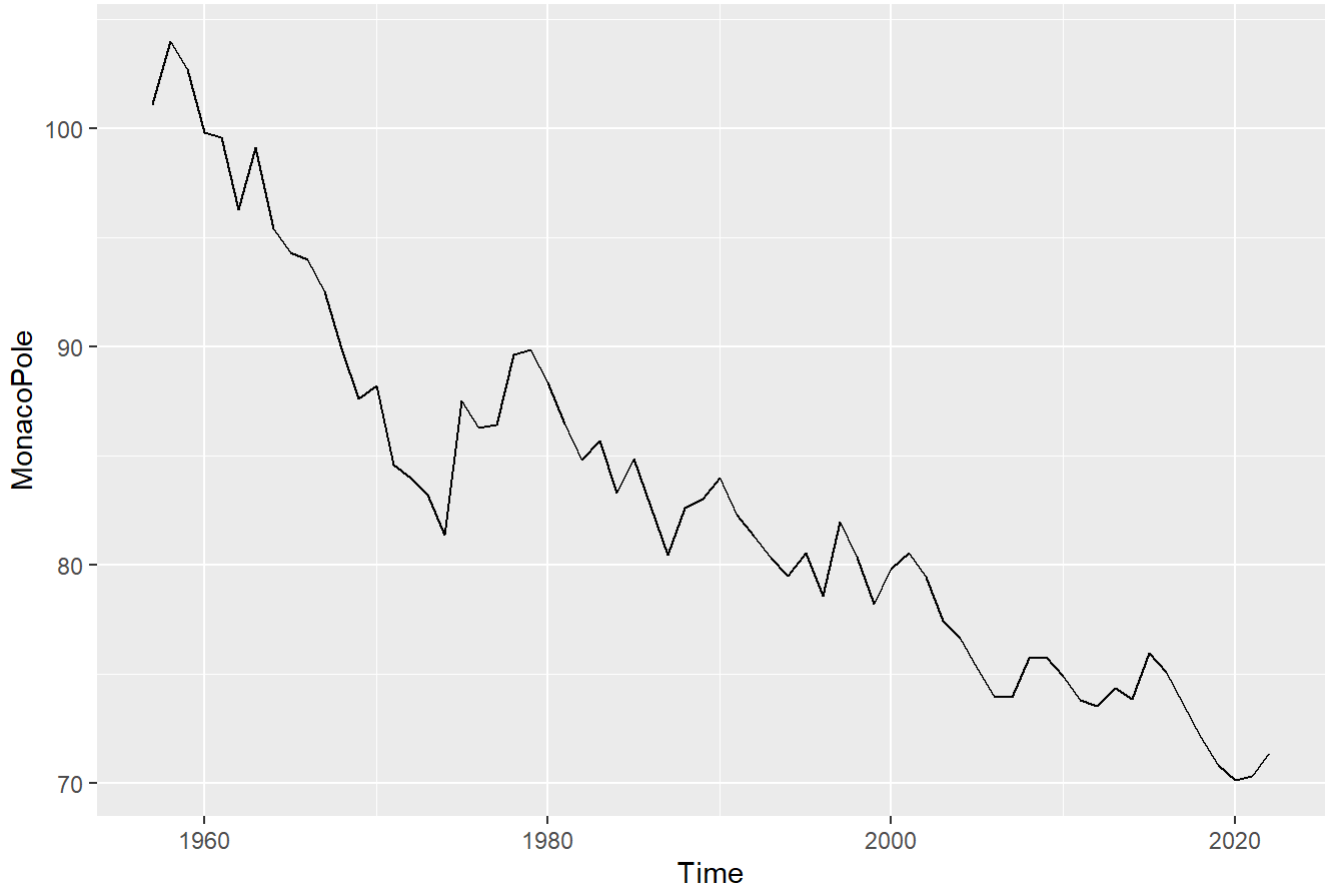
f1 average speed

James Stoner

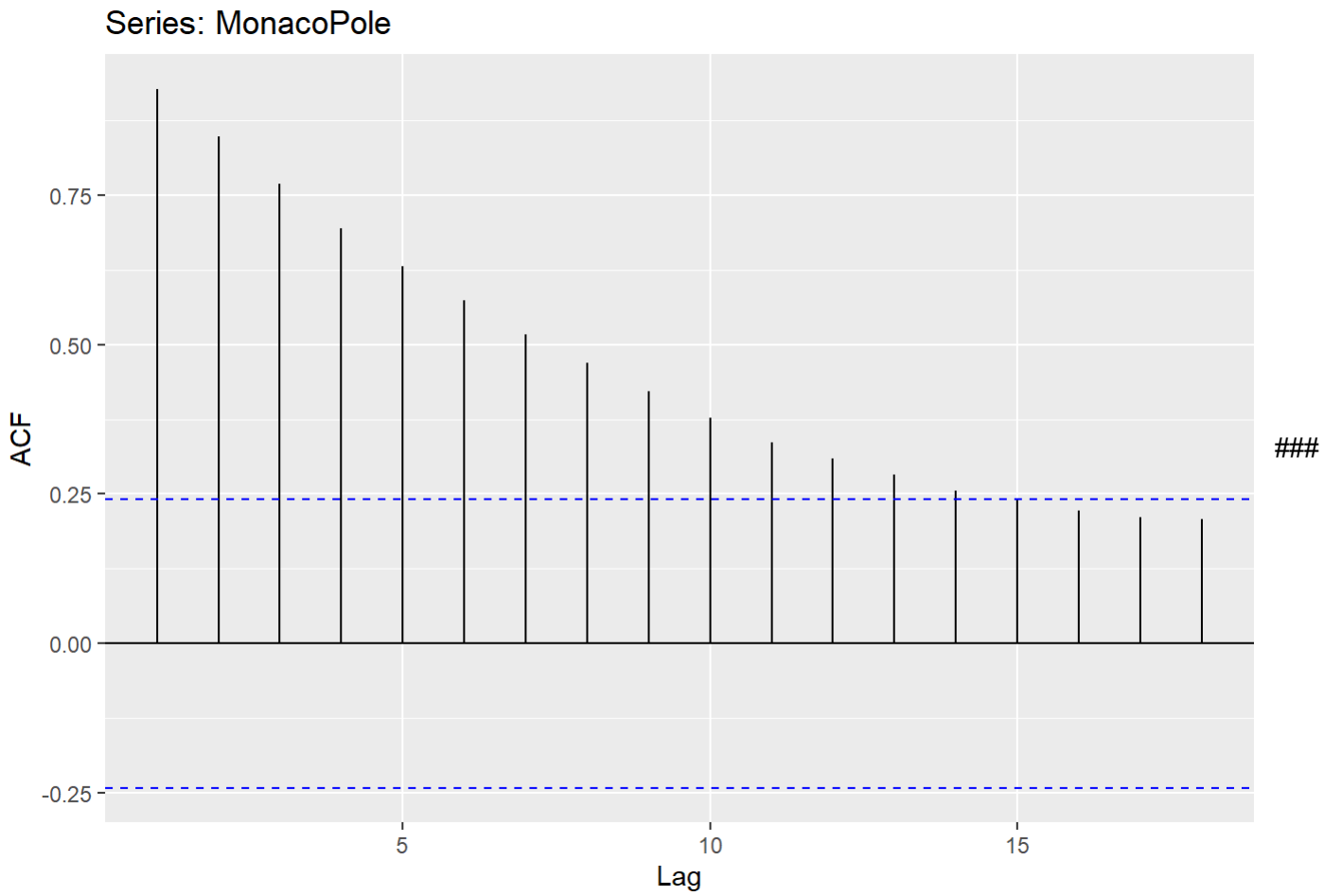
2022-11-22

Plots of Data

```
autoplot(MonacoPole)
```

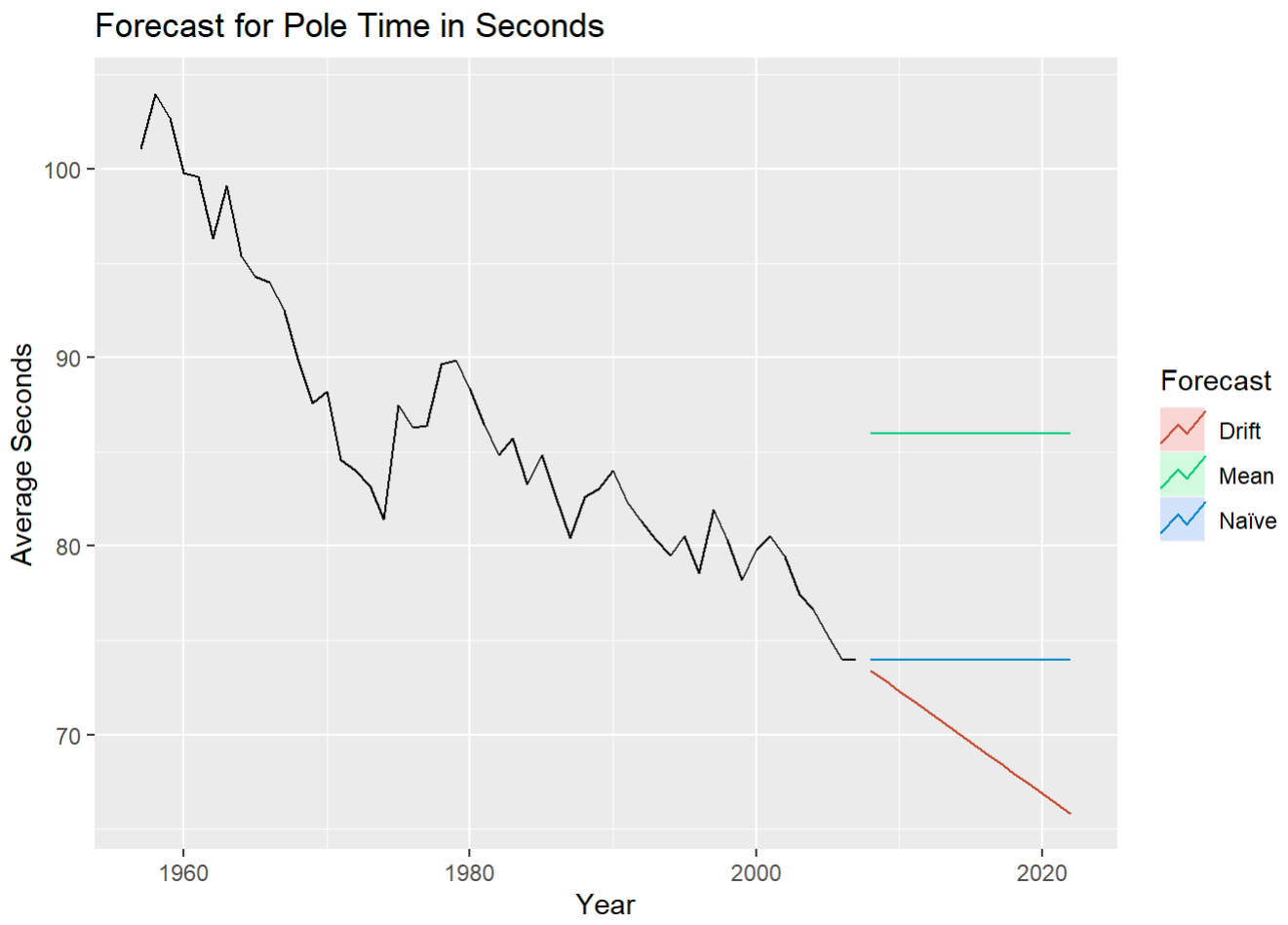



```
ggAcf(MonacoPole)
```



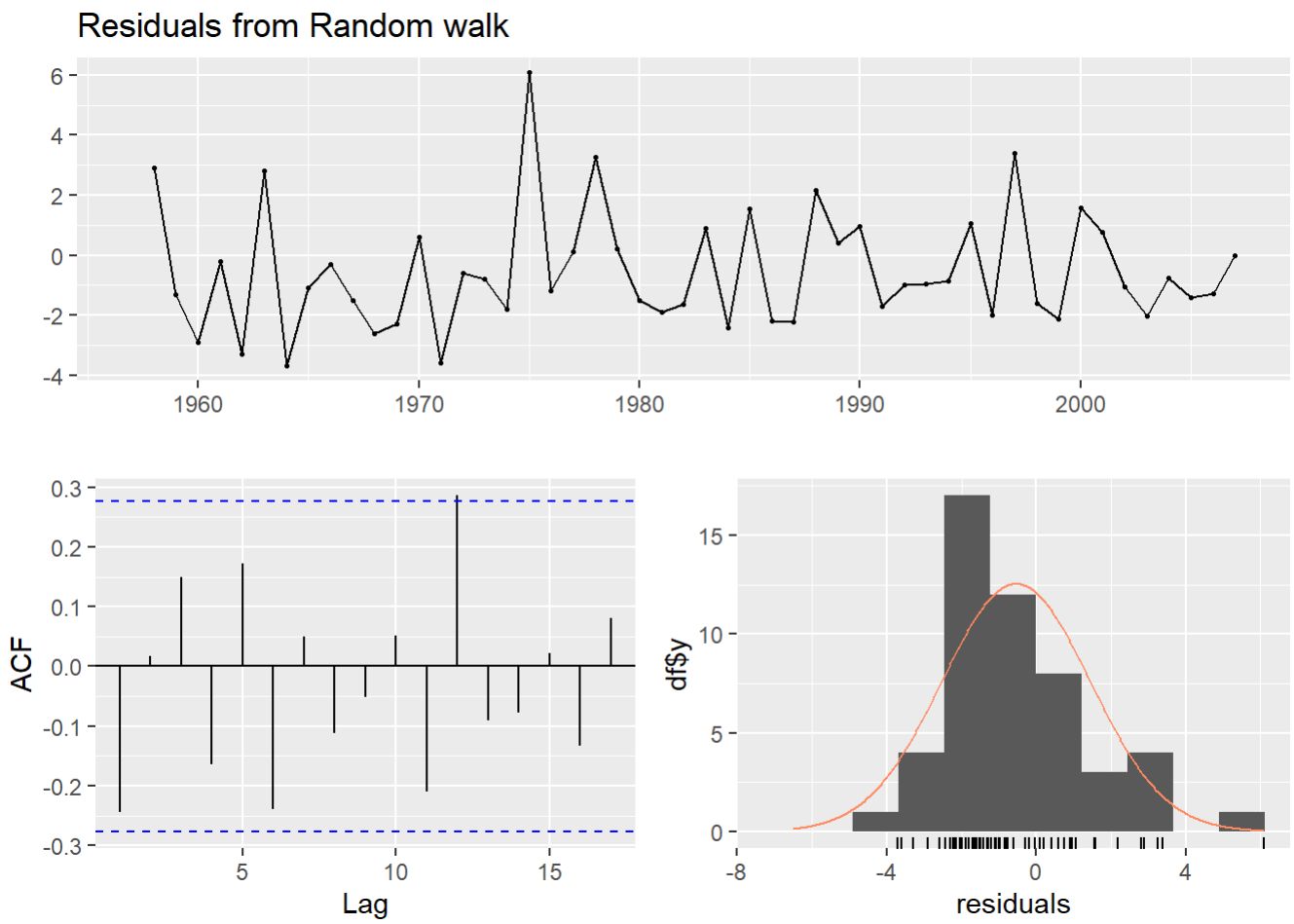
Simple Forecasting Methods

```
MonacoPoleTrain <- window(MonacoPole, end = 2007)
autoplot(MonacoPoleTrain) +
 autolayer(meanf(MonacoPoleTrain, h=15),
    series="Mean", PI=FALSE) +
  autolayer(naive(MonacoPoleTrain, h=15),
    series="Naïve", PI=FALSE) +
  autolayer(rwf(MonacoPoleTrain, drift = TRUE, h=15),
    series="Drift", PI=FALSE) +
  ggtitle("Forecast for Pole Time in Seconds") +
  xlab("Year") + ylab("Average Seconds") +
  guides(colour=guide_legend(title="Forecast"))
```



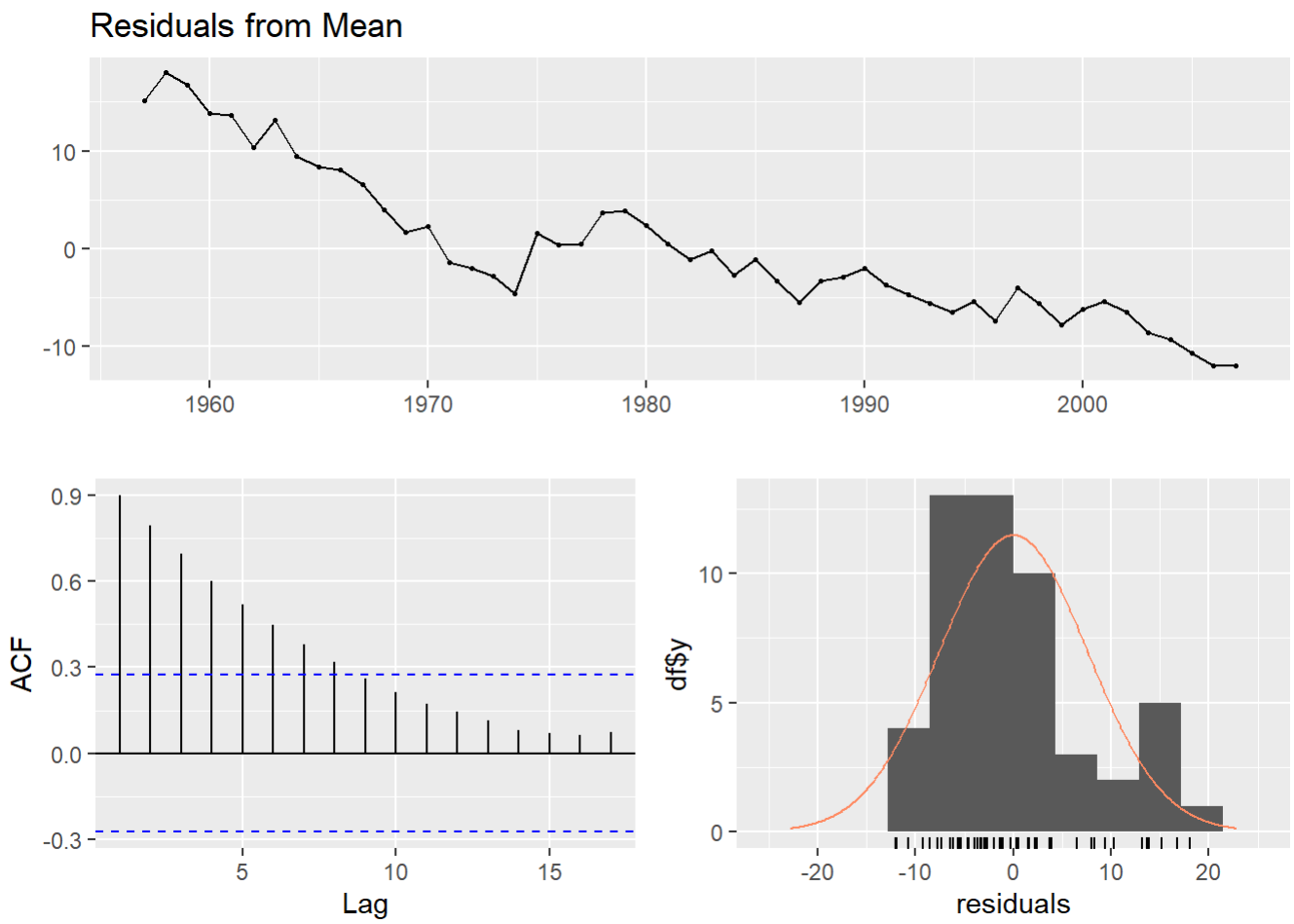
###Checking best benchmark method

```
checkresiduals(rwf(MonacoPoleTrain))
```



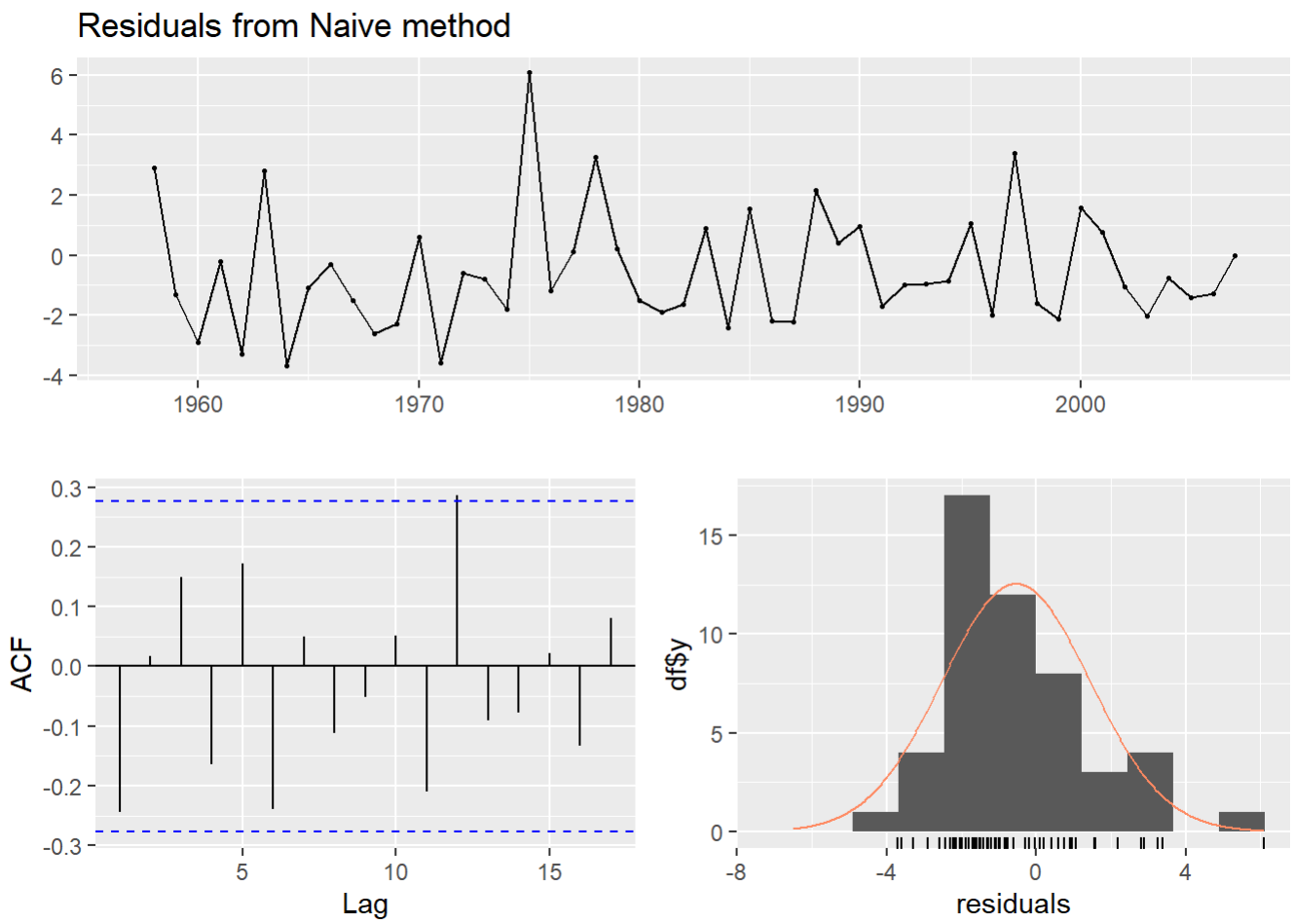

```
##
##  Ljung-Box test
##
## data:  Residuals from Random walk
## Q* = 12.302, df = 10, p-value = 0.2653
##
## Model df: 0.   Total lags used: 10
```

```
checkresiduals(meanf(MonacoPoleTrain))
```



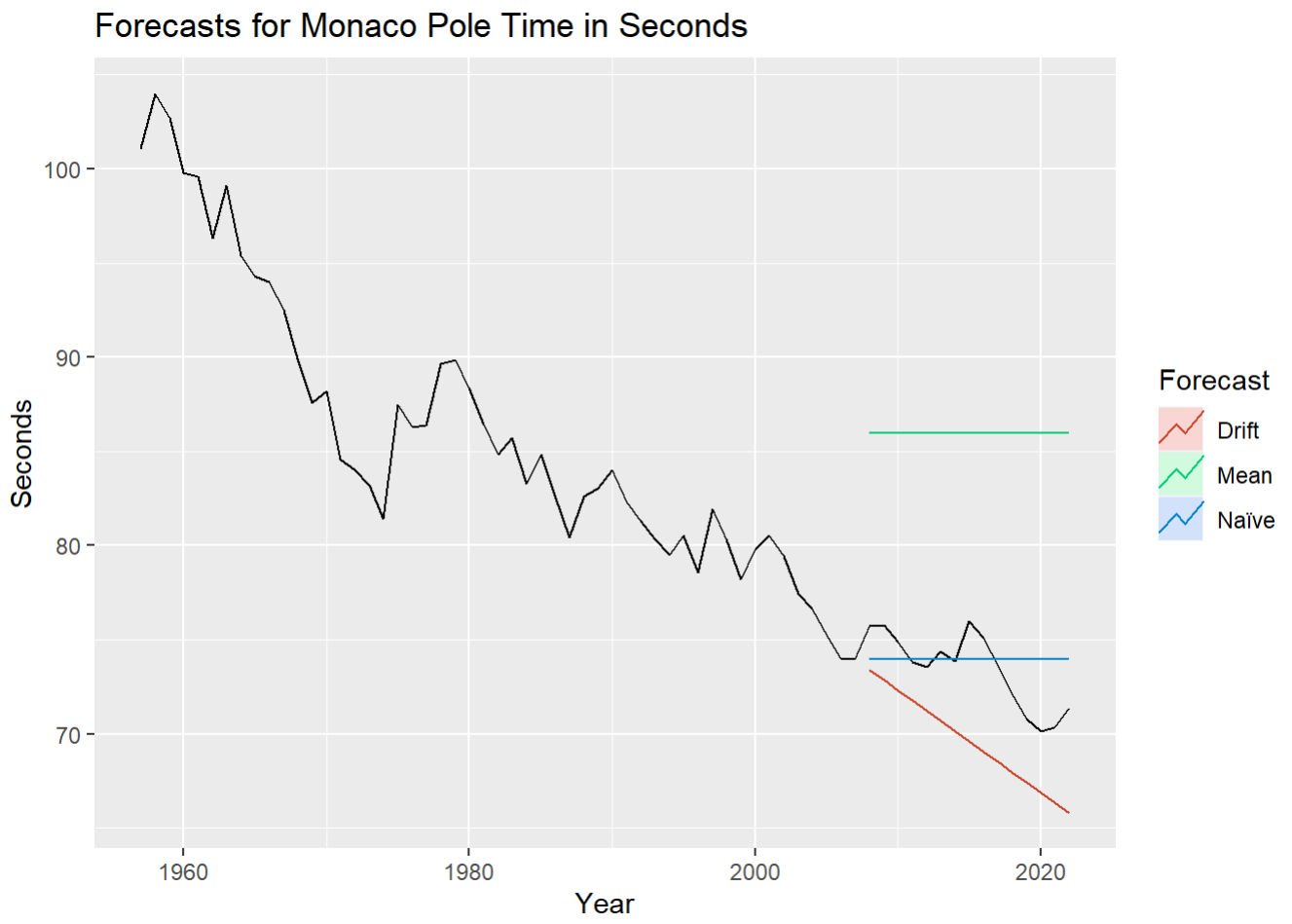

```
##
##  Ljung-Box test
##
## data:  Residuals from Mean
## Q* = 177.25, df = 9, p-value < 2.2e-16
##
## Model df: 1.   Total lags used: 10
```

```
checkresiduals(naive(MonacoPoleTrain))
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Naive method
## Q* = 12.302, df = 10, p-value = 0.2653
##
## Model df: 0.   Total lags used: 10
```

```
MonacoPoleTrain <- window(MonacoPole, end = 2007)
MonacoPoleTrain_average <- meanf(MonacoPoleTrain,h=15)
MonacoPoleTrain_drift <- rwf(MonacoPoleTrain, drift=TRUE, h=15)
MonacoPoleTrain_naive <- naive(MonacoPoleTrain,h=15)
autoplot(MonacoPole) +
  autolayer(MonacoPoleTrain_average, series="Mean", PI=FALSE) +
  autolayer(MonacoPoleTrain_drift, series="Drift", PI=FALSE) +
  autolayer(MonacoPoleTrain_naive, series="Naïve", PI=FALSE) +
  xlab("Year") + ylab("Seconds") +
  ggtitle("Forecasts for Monaco Pole Time in Seconds") +
  guides(colour=guide_legend(title="Forecast"))
```




```
MonacoPoleTest <- window(MonacoPole, start = 2008)
accuracy(MonacoPoleTrain_average, MonacoPoleTest)
```

```
##                        ME      RMSE       MAE         MPE      MAPE     MASE
## Training set  2.507084e-15  7.537127  6.034669  -0.7341814  6.921058 3.565072
## Test set     -1.251530e+01 12.664760 12.515302 -17.1235657 17.123566 7.393604
##                    ACF1 Theil's U
## Training set 0.8996292        NA
## Test set     0.7755574  12.16494
```

```
accuracy(MonacoPoleTrain_drift, MonacoPoleTest)
```

```
##                        ME     RMSE      MAE         MPE     MAPE      MASE
## Training set 8.526827e-16 1.968141 1.541538 -0.0145785 1.779954 0.9106867
## Test set     3.822680e+00 4.049913 3.822680  5.2082799 5.208280 2.2583062
##                    ACF1 Theil's U
## Training set -0.2441621        NA
## Test set      0.6169437  3.909908
```

```
accuracy(MonacoPoleTrain_naive, MonacoPoleTest)
```

```
##                     ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.54276 2.041609 1.692720 -0.6526238 1.963291 1.0000000
## Test set     -0.51940 2.008271 1.600867 -0.7783225 2.209787 0.9457363
##                    ACF1 Theil's U
## Training set -0.2441621        NA
## Test set      0.7755574  1.956846
```

###Checking best benchmark method using Cross-Validation

```
MonacoPoleCV <- tsCV(MonacoPole, rwf, drift=TRUE, h=15)
sqrt(mean(MonacoPoleCV^2, na.rm=TRUE))
```

```
## [1] 7.6508
```

```
MonacoPoleAverageCV <- tsCV(MonacoPole, meanf, h=15)
sqrt(mean(MonacoPoleAverageCV^2, na.rm=TRUE))
```

```
## [1] 11.11983
```

```
MonacoPoleNaiveCV <- tsCV(MonacoPole, naive, h=15)
sqrt(mean(MonacoPoleNaiveCV^2, na.rm=TRUE))
```

```
## [1] 5.516656
```

Advanced Forecasting Methods SES

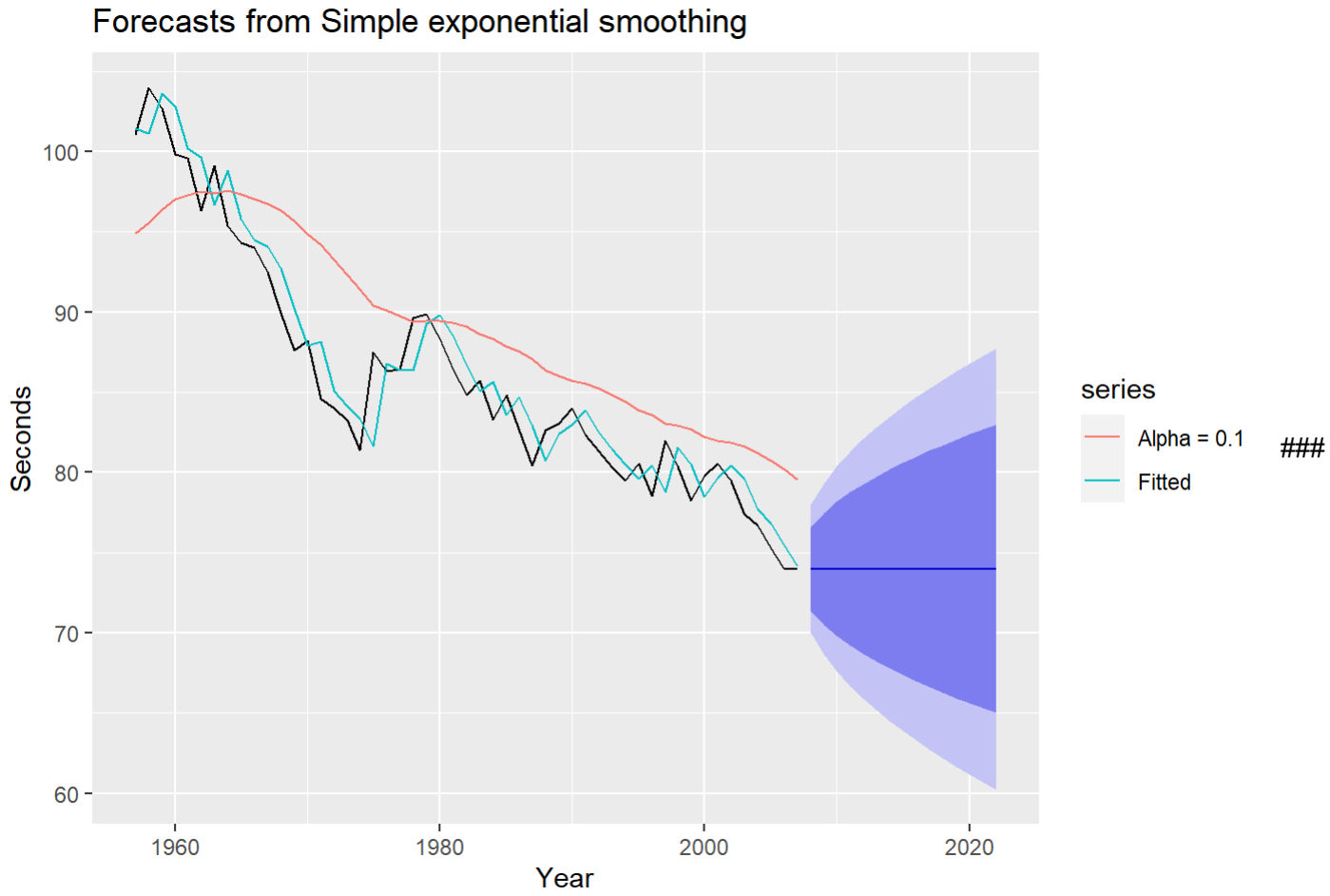
```
SESpole <- ses(MonacoPoleTrain, h = 15)
summary(SESpole)
```

```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = MonacoPoleTrain, h = 15)
##
##   Smoothing parameters:
##     alpha = 0.8774
##
##   Initial states:
##     l = 101.4312
##
##   sigma:  2.0432
##
##      AIC     AICc      BIC
## 277.3639 277.8745 283.1593
##
## Error measures:
##                      ME     RMSE      MAE        MPE    MAPE      MASE
## Training set -0.613322 2.002742 1.675943 -0.7348174 1.94532 0.9900888
##                     ACF1
## Training set -0.1231987
##
## Forecasts:
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2008       73.98681 71.36834 76.60529 69.98220 77.99142
## 2009       73.98681 70.50334 77.47029 68.65930 79.31433
## 2010       73.98681 69.81395 78.15967 67.60497 80.36865
## 2011       73.98681 69.22331 78.75032 66.70166 81.27196
## 2012       73.98681 68.69823 79.27540 65.89862 82.07501
## 2013       73.98681 68.22076 79.75286 65.16840 82.80523
## 2014       73.98681 67.77992 80.19371 64.49419 83.47944
## 2015       73.98681 67.36838 80.60525 63.86479 84.10884
## 2016       73.98681 66.98097 80.99266 63.27230 84.70133
## 2017       73.98681 66.61388 81.35974 62.71089 85.26273
## 2018       73.98681 66.26423 81.70939 62.17614 85.79748
## 2019       73.98681 65.92974 82.04389 61.66458 86.30905
## 2020       73.98681 65.60859 82.36504 61.17343 86.80020
## 2021       73.98681 65.29930 82.67432 60.70041 87.27321
## 2022       73.98681 65.00066 82.97297 60.24367 87.72995
```

```
SES1pole <- ses(MonacoPoleTrain, alpha = 0.1, h = 15)

SESpole %>%
  accuracy() %>%
  round(2)
```

```
##                 ME RMSE  MAE   MPE MAPE MASE  ACF1
## Training set -0.61    2 1.68 -0.73 1.95 0.99 -0.12
```

```
SESpole %>%

  autoplot() +

  autolayer(fitted(SESpole), series = "Fitted") +

  autolayer(fitted(SES1pole), series = "Alpha = 0.1") +

  ylab("Seconds") +

  xlab("Year")
```



Advanced Forecasting Methods Holts Linear Trend Method and Damped methods

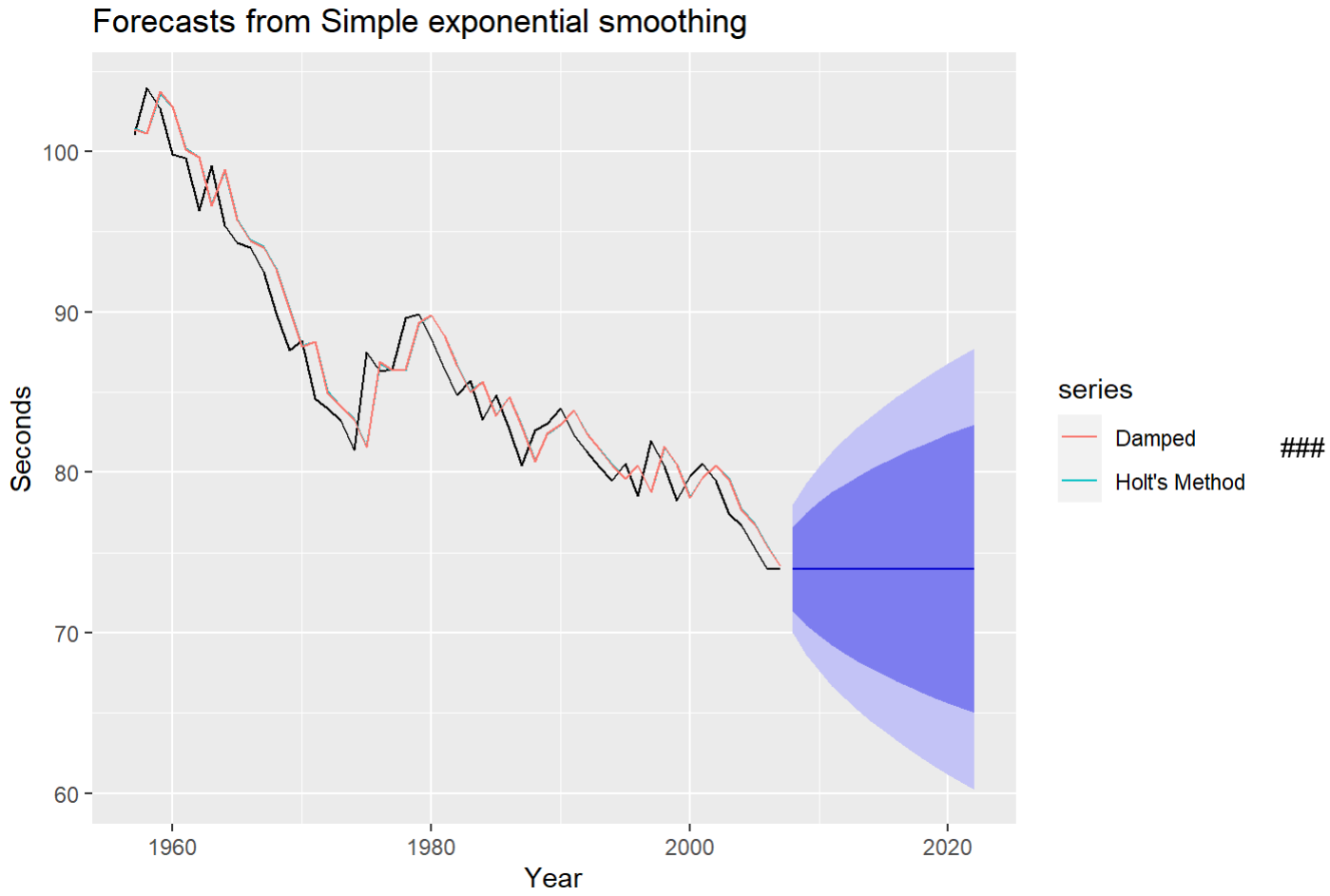
```
holtpole <- ses(MonacoPoleTrain, h = 15)
summary(holtpole)
```

```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = MonacoPoleTrain, h = 15)
##
##   Smoothing parameters:
##     alpha = 0.8774
##
##   Initial states:
##     l = 101.4312
##
##   sigma:  2.0432
##
##      AIC     AICc      BIC
## 277.3639 277.8745 283.1593
##
## Error measures:
##                      ME     RMSE      MAE        MPE    MAPE      MASE
## Training set -0.613322 2.002742 1.675943 -0.7348174 1.94532 0.9900888
##                     ACF1
## Training set -0.1231987
##
## Forecasts:
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2008       73.98681 71.36834 76.60529 69.98220 77.99142
## 2009       73.98681 70.50334 77.47029 68.65930 79.31433
## 2010       73.98681 69.81395 78.15967 67.60497 80.36865
## 2011       73.98681 69.22331 78.75032 66.70166 81.27196
## 2012       73.98681 68.69823 79.27540 65.89862 82.07501
## 2013       73.98681 68.22076 79.75286 65.16840 82.80523
## 2014       73.98681 67.77992 80.19371 64.49419 83.47944
## 2015       73.98681 67.36838 80.60525 63.86479 84.10884
## 2016       73.98681 66.98097 80.99266 63.27230 84.70133
## 2017       73.98681 66.61388 81.35974 62.71089 85.26273
## 2018       73.98681 66.26423 81.70939 62.17614 85.79748
## 2019       73.98681 65.92974 82.04389 61.66458 86.30905
## 2020       73.98681 65.60859 82.36504 61.17343 86.80020
## 2021       73.98681 65.29930 82.67432 60.70041 87.27321
## 2022       73.98681 65.00066 82.97297 60.24367 87.72995
```

```
holt1pole <- ses(MonacoPoleTrain, damped = TRUE, alpha = 0.9, h = 15)

holtpole %>%
  accuracy() %>%
  round(2)
```

```
##                 ME RMSE  MAE   MPE MAPE MASE  ACF1
## Training set -0.61    2 1.68 -0.73 1.95 0.99 -0.12
```

```
holtpole %>%

  autoplot() +

  autolayer(fitted(holtpole), series = "Holt's Method") +

  autolayer(fitted(holt1pole), series = "Damped") +

  ylab("Seconds") +

  xlab("Year")
```

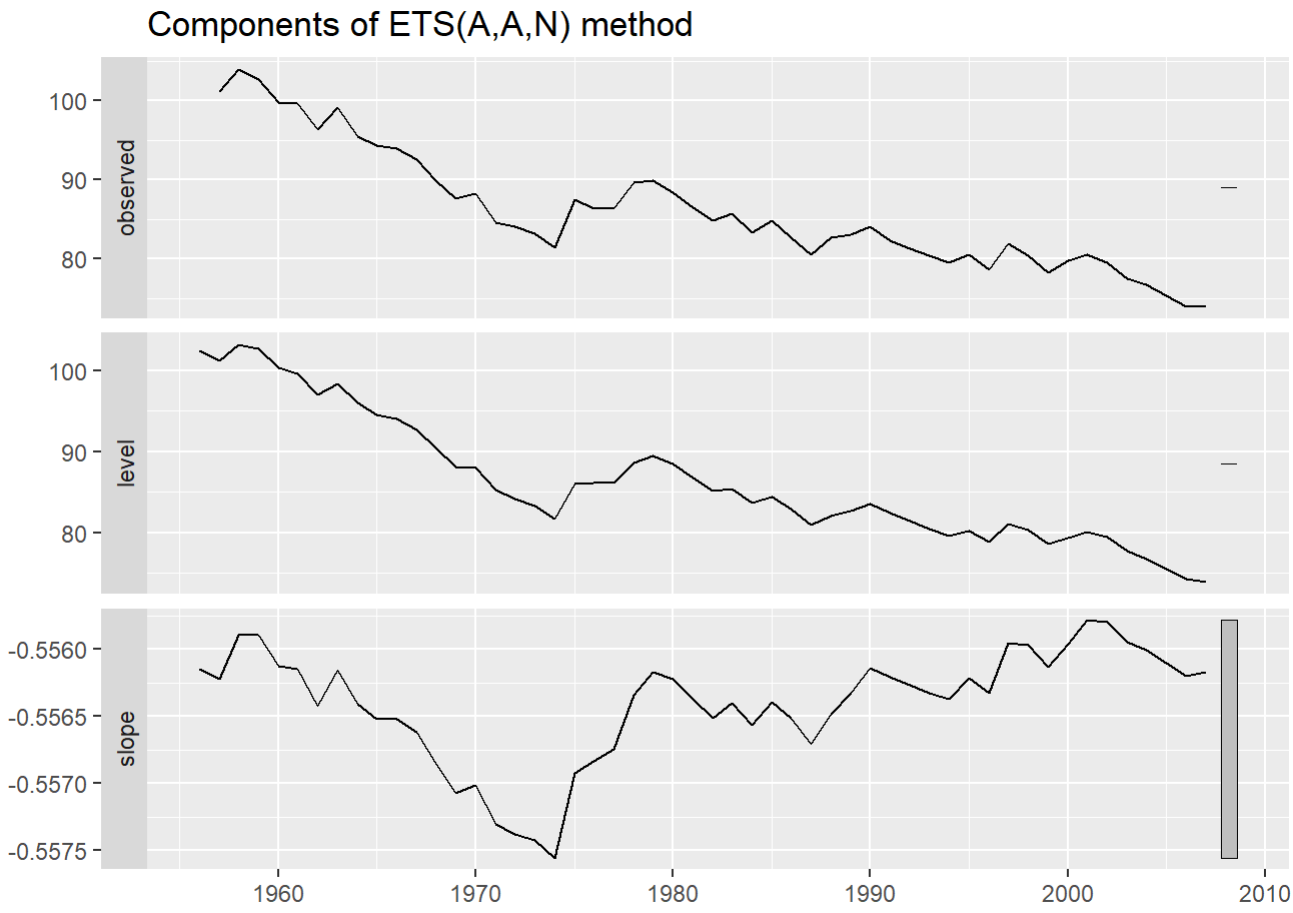


Advanced Forecasting Methods ETS

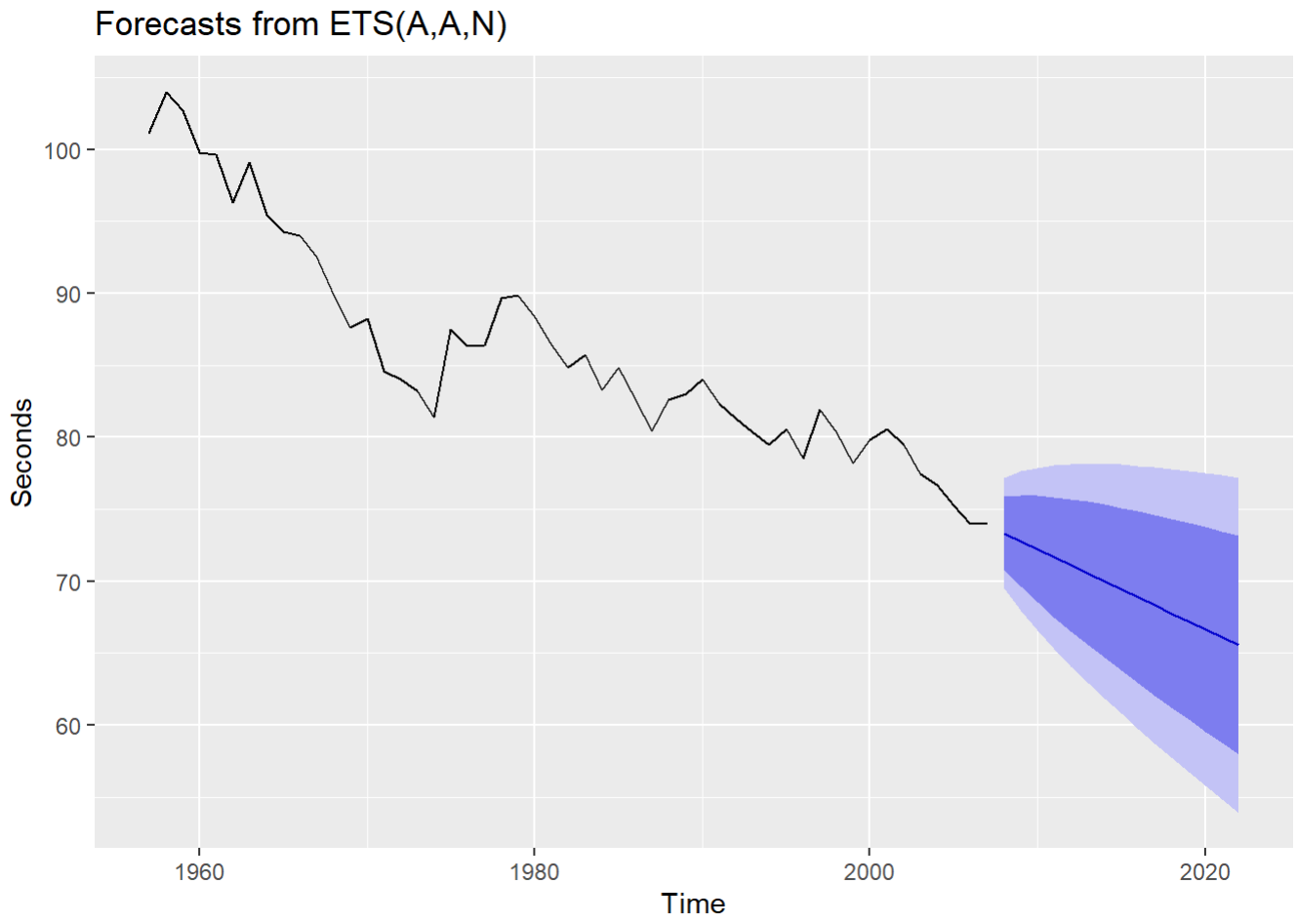
```
MonacoPoleETS <- ets(MonacoPoleTrain)
summary(MonacoPoleETS)
```

```
## ETS(A,A,N)
##
## Call:
##  ets(y = MonacoPoleTrain)
##
##   Smoothing parameters:
##     alpha = 0.7589
##     beta  = 1e-04
##
##   Initial states:
##     l = 102.4359
##     b = -0.5561
##
##   sigma:  1.9673
##
##      AIC     AICc      BIC
## 275.3759 276.7092 285.0350
##
## Training set error measures:
##                       ME     RMSE      MAE          MPE     MAPE      MASE
## Training set -0.004390536 1.888554 1.491227 -0.01681682 1.723302 0.8809648
##                      ACF1
## Training set -0.02113771
```

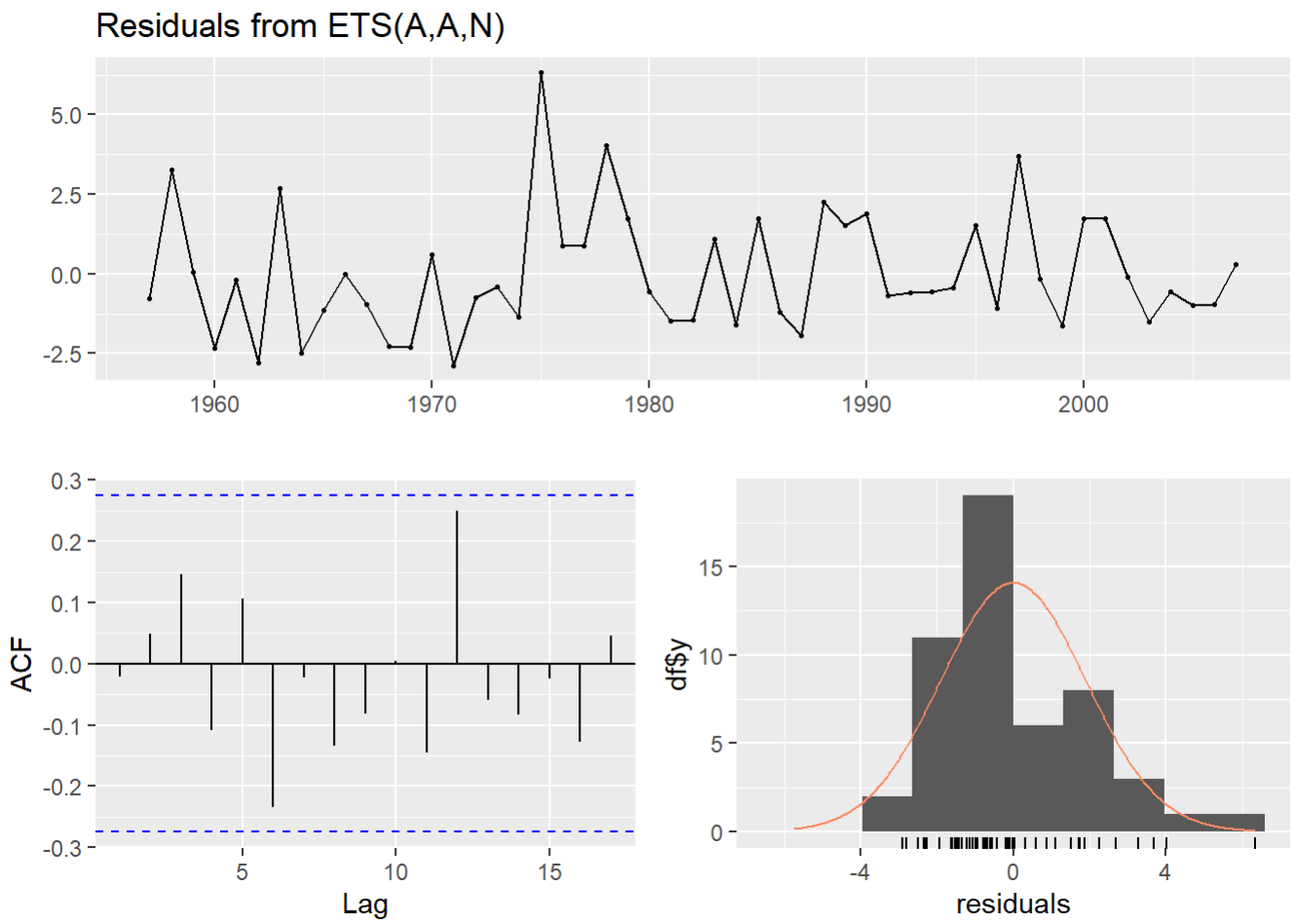
```
autoplot(MonacoPoleETS)
```




```
MonacoPoleETS %>% forecast(h=15) %>%
  autoplot() +
  ylab("Seconds")
```



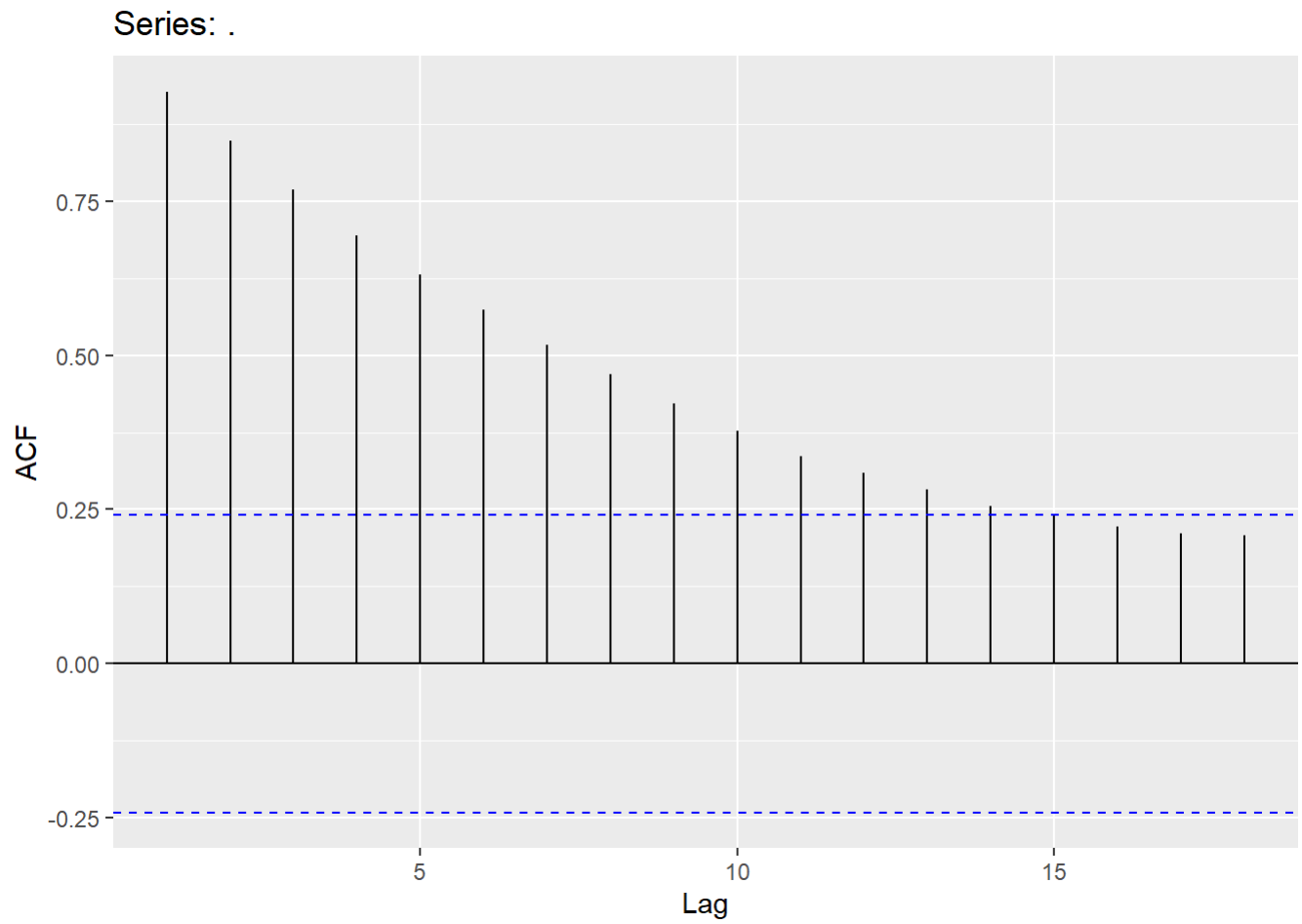

```
checkresiduals(MonacoPoleETS)
```



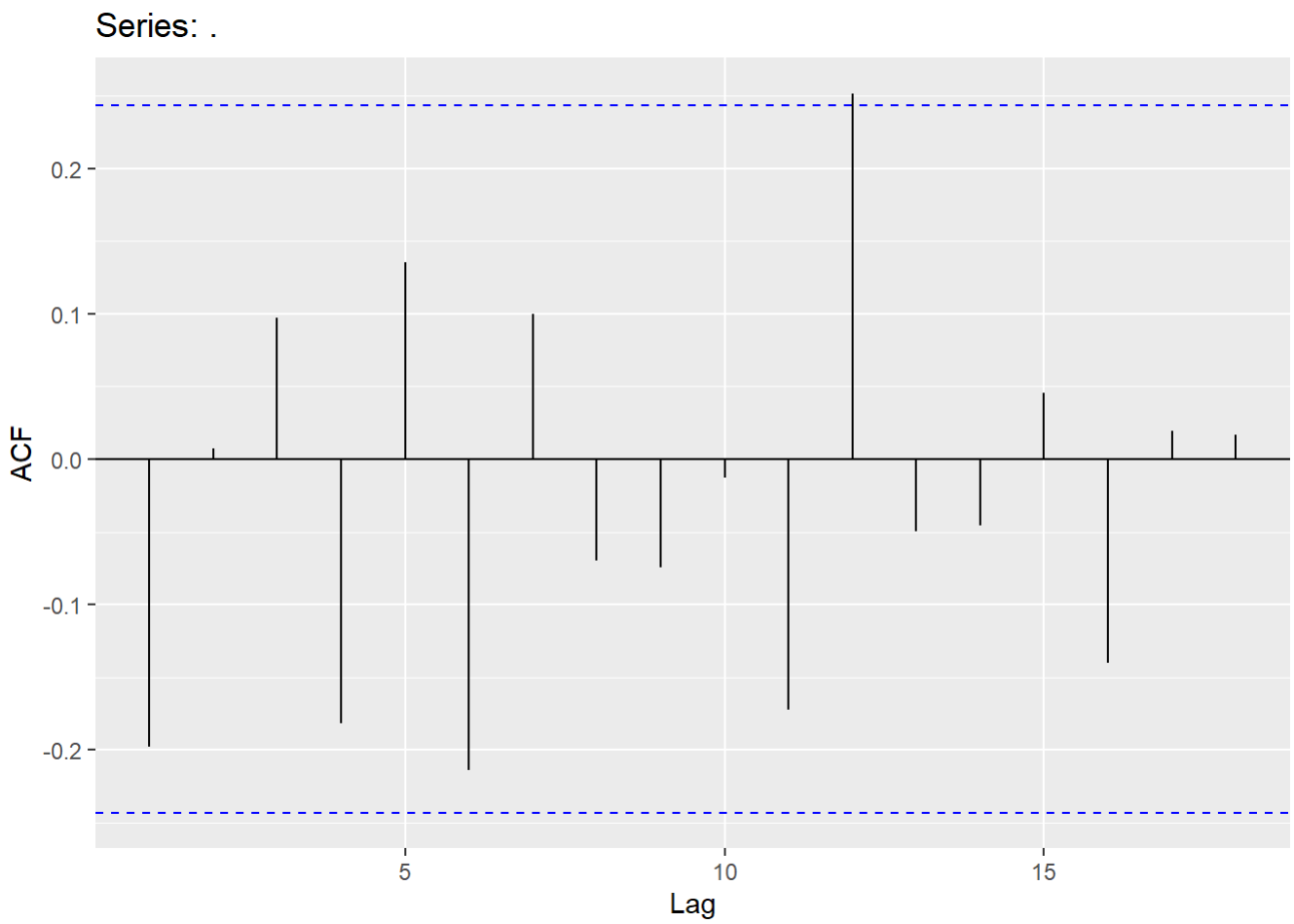

```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,A,N)
## Q* = 7.6256, df = 6, p-value = 0.2668
##
## Model df: 4.   Total lags used: 10
```

Advanced Forecasting Methods ARIMA Models

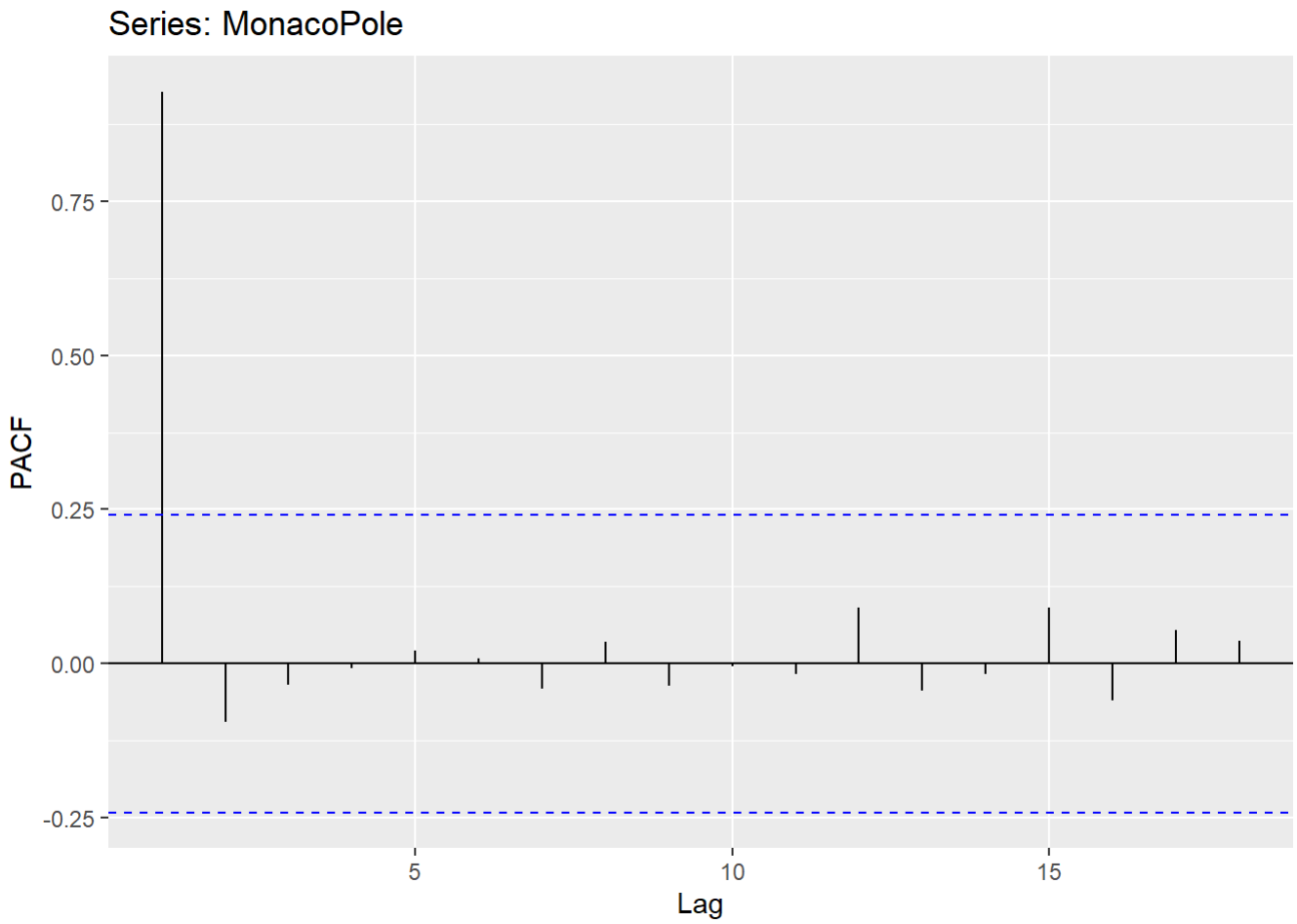
```
MonacoPole %>%
  ggAcf()
```




```
MonacoPole %>%
  diff() %>%
  ggAcf()
```




```
ggPacf(MonacoPole)
```




```
MonacoPole %>% diff() %>% ur.kpss() %>% summary()
```

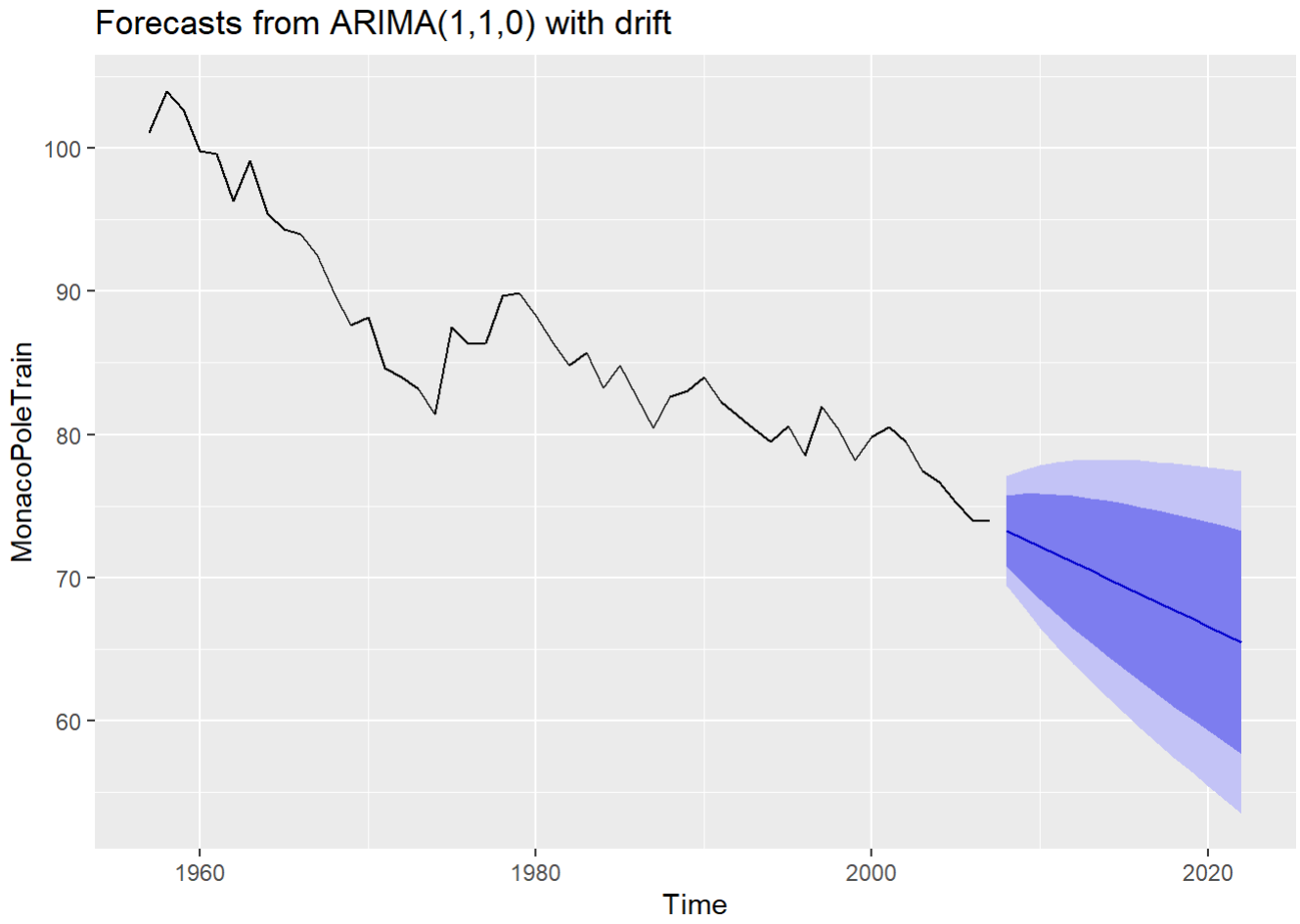
```
##
## ########################
## # KPSS Unit Root Test #
## ########################
##
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 0.1152
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
ndiffs(MonacoPole)
```

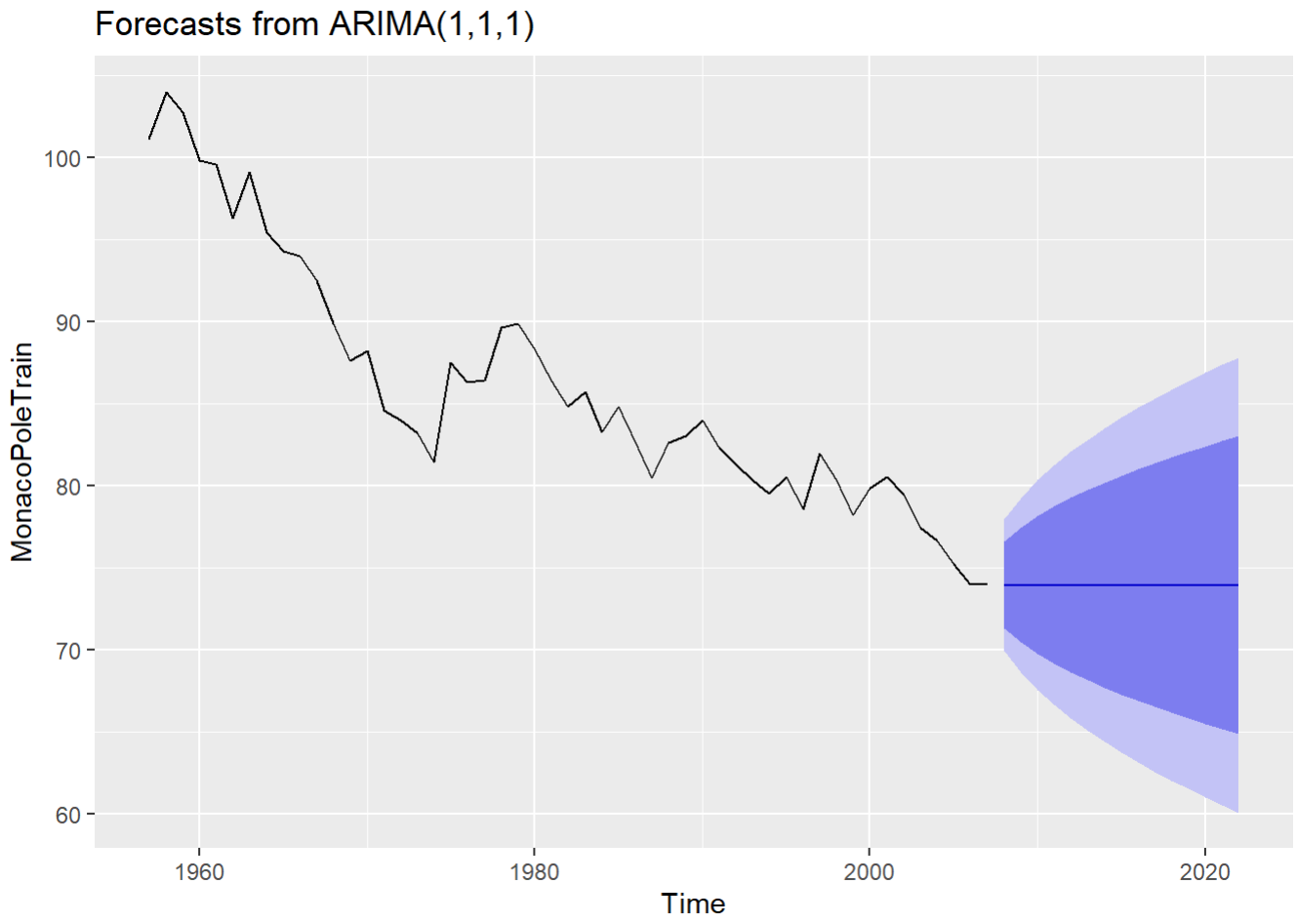
```
## [1] 1
```

```
#########################################################################################################
#####################################################################
```

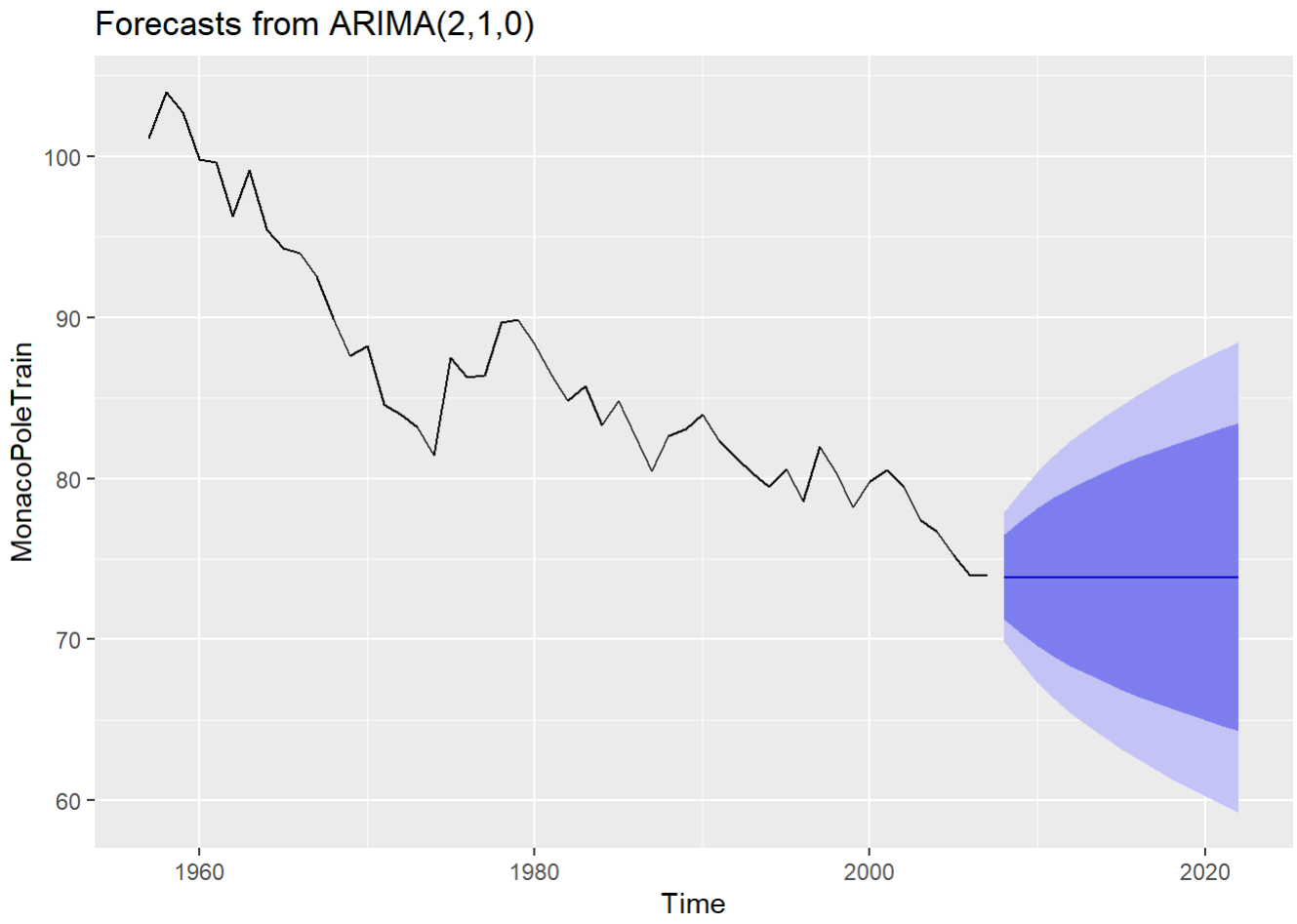
```
fit <- auto.arima(MonacoPoleTrain, seasonal=FALSE)
fit %>% forecast(h=15) %>% autoplot(include=80)
```



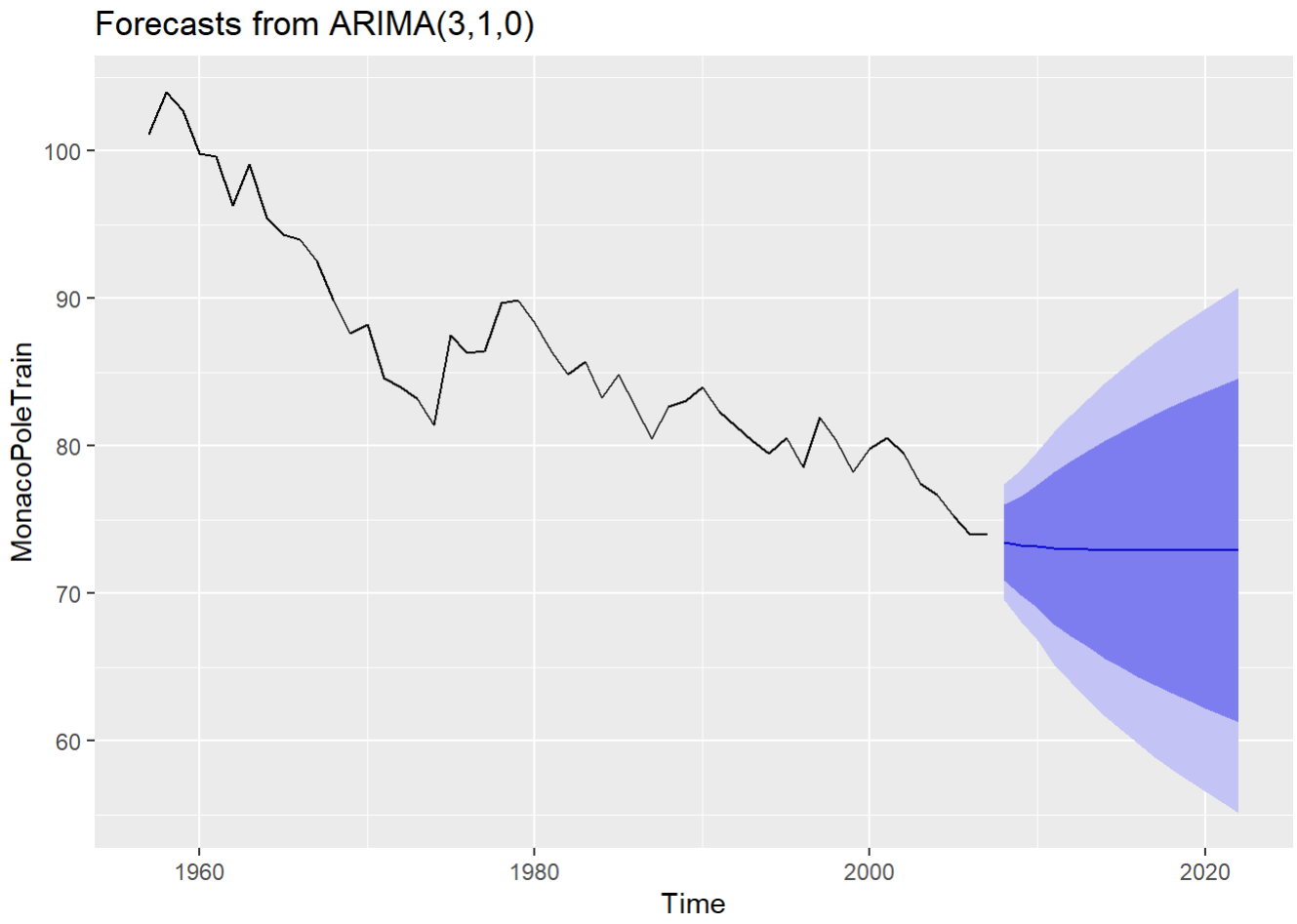

```
fit2 <- Arima(MonacoPoleTrain, order=c(1,1,1))
fit2 %>% forecast(h=15) %>% autoplot(include=80)
```



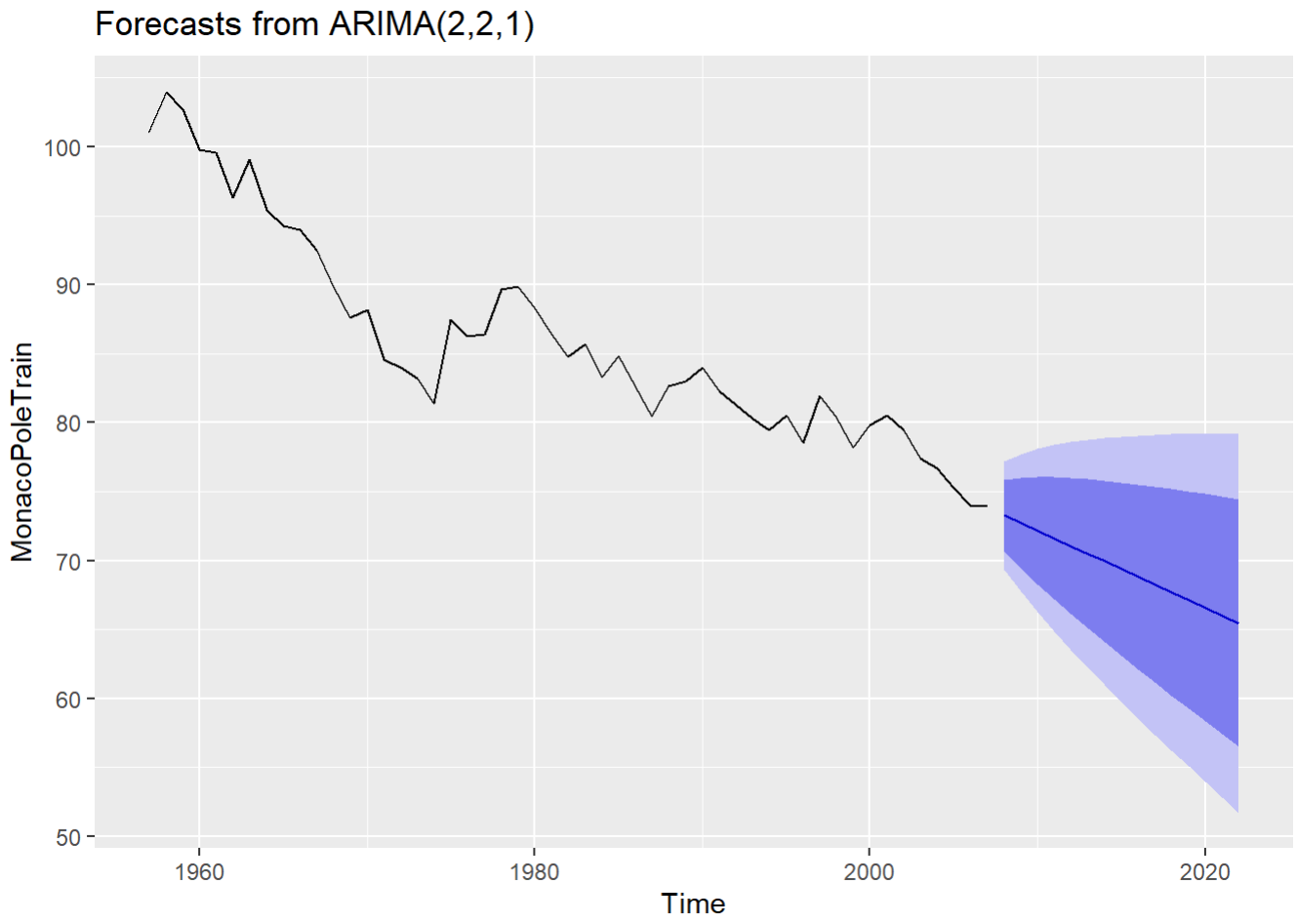

```
fit3 <- Arima(MonacoPoleTrain, order=c(2,1,0))
fit3 %>% forecast(h=15) %>% autoplot(include=80)
```



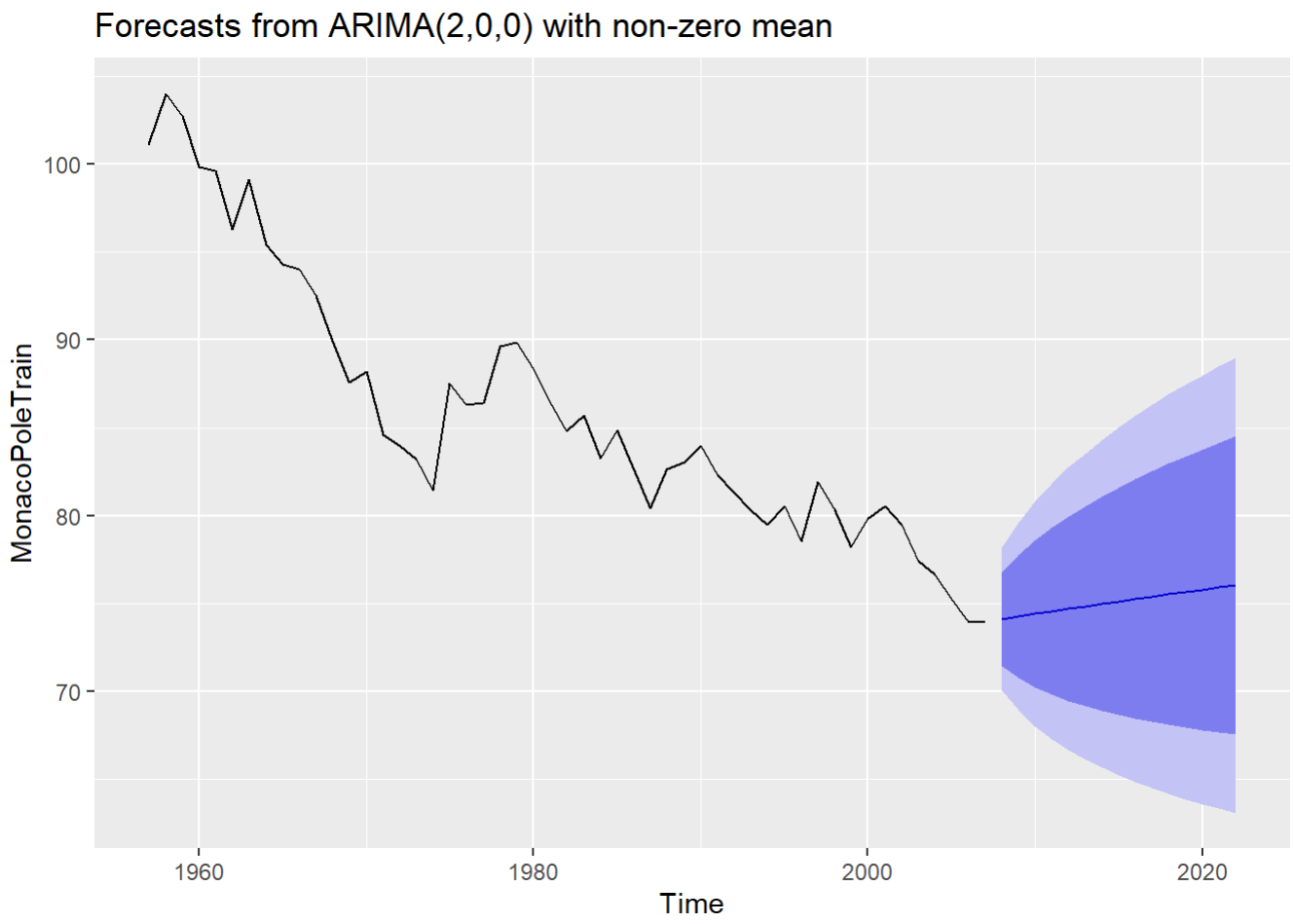

```
fit4 <- Arima(MonacoPoleTrain, order=c(3,1,0))
fit4 %>% forecast(h=15) %>% autoplot(include=80)
```



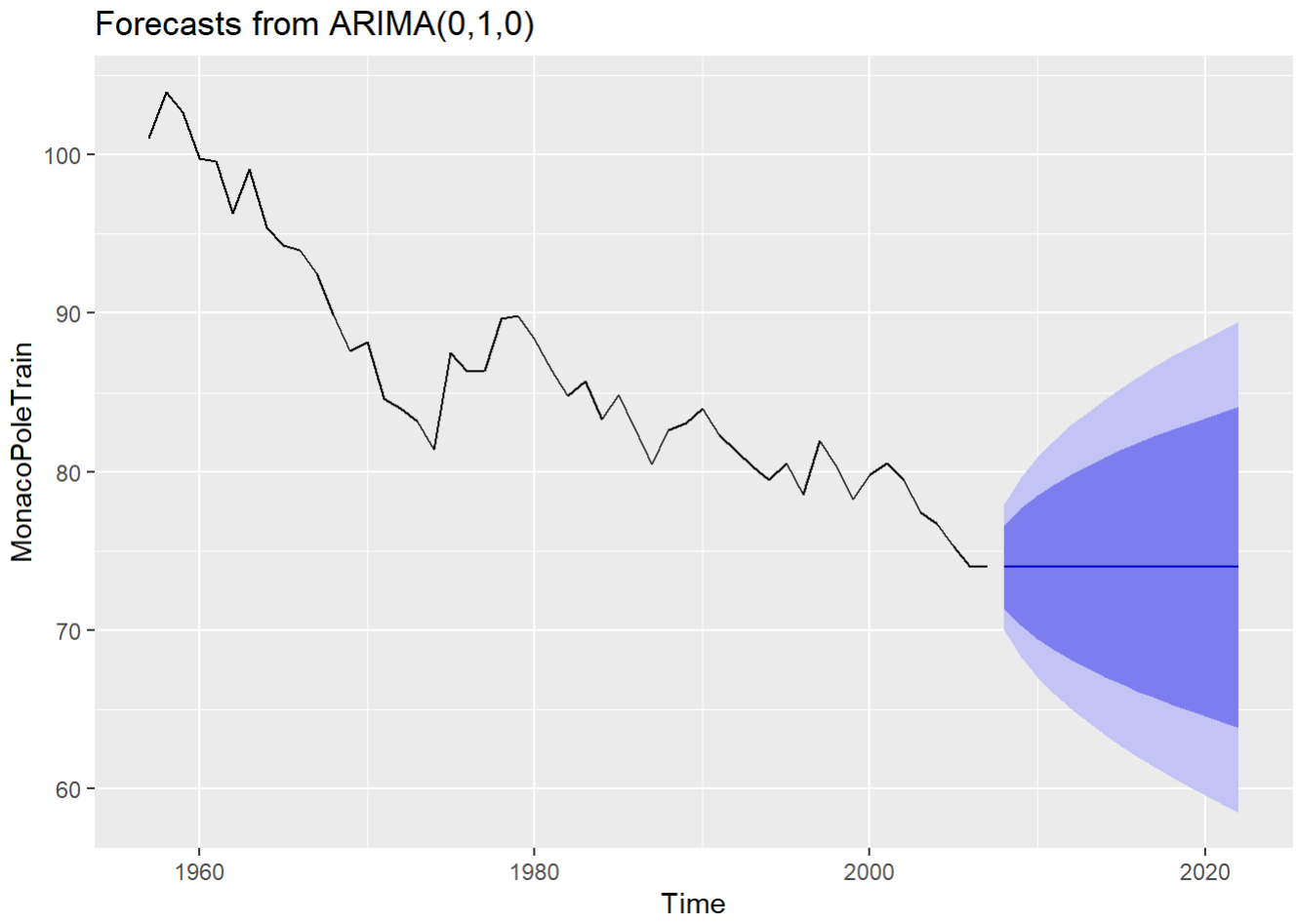

```
fit5 <- Arima(MonacoPoleTrain, order=c(2,2,1))
fit5 %>% forecast(h=15) %>% autoplot(include=80)
```




```
fit6 <- Arima(MonacoPoleTrain, order=c(2,0,0))
fit6 %>% forecast(h=15) %>% autoplot(include=80)
```




```
fit7 <- Arima(MonacoPoleTrain, order=c(0,1,0))
fit7 %>% forecast(h=15) %>% autoplot(include=80)
```




```
summary(fit)
```

```
## Series: MonacoPoleTrain
## ARIMA(1,1,0) with drift
##
## Coefficients:
##           ar1    drift
##       -0.2555  -0.5590
## s.e.   0.1398   0.2155
##
## sigma^2 = 3.778:  log likelihood = -103.19
## AIC=212.38   AICc=212.9   BIC=218.12
##
## Training set error measures:
##                       ME     RMSE      MAE         MPE     MAPE     MASE
## Training set 0.01706904 1.885737 1.486709 0.004796726 1.719795 0.878296
##                       ACF1
## Training set 0.0008589388
```

```
summary(fit2)
```

```
## Series: MonacoPoleTrain
## ARIMA(1,1,1)
##
## Coefficients:
##           ar1     ma1
##       -0.2611  0.1100
## s.e.   0.4389  0.4341
##
## sigma^2 = 4.239:  log likelihood = -106.04
## AIC=218.09   AICc=218.61   BIC=223.82
##
## Training set error measures:
##                      ME     RMSE      MAE        MPE     MAPE     MASE
## Training set -0.6033542 1.997308 1.670591 -0.7247327 1.940391 0.986927
##                     ACF1
## Training set -0.08889839
```

```
summary(fit3)
```

```
## Series: MonacoPoleTrain
## ARIMA(2,1,0)
##
## Coefficients:
##           ar1     ar2
##       -0.1403  0.0728
## s.e.   0.1420  0.1419
##
## sigma^2 = 4.221:  log likelihood = -105.94
## AIC=217.88   AICc=218.41   BIC=223.62
##
## Training set error measures:
##                      ME    RMSE      MAE        MPE     MAPE     MASE
## Training set -0.5679947 1.99311 1.665496 -0.6827866 1.934229 0.983917
##                     ACF1
## Training set -0.1032427
```

```
summary(fit4)
```

```
## Series: MonacoPoleTrain
## ARIMA(3,1,0)
##
## Coefficients:
##           ar1     ar2     ar3
##       -0.1704  0.1118  0.2559
## s.e.   0.1379  0.1384  0.1404
##
## sigma^2 = 4.028:  log likelihood = -104.35
## AIC=216.7   AICc=217.59   BIC=224.35
##
## Training set error measures:
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.4392747 1.926693 1.612921 -0.5283535 1.868345 0.9528576
##                     ACF1
## Training set -0.03029443
```

```
summary(fit5)
```

```
## Series: MonacoPoleTrain
## ARIMA(2,2,1)
##
## Coefficients:
##           ar1      ar2      ma1
##       -0.2468  -0.0271  -0.9996
## s.e.   0.1473   0.1461   0.1013
##
## sigma^2 = 3.939:  log likelihood = -103.78
## AIC=215.56   AICc=216.47   BIC=223.12
##
## Training set error measures:
##                      ME     RMSE      MAE         MPE     MAPE      MASE
## Training set 0.06818714 1.884822 1.447895 0.08698981 1.675349 0.8553661
##                     ACF1
## Training set 0.04532382
```

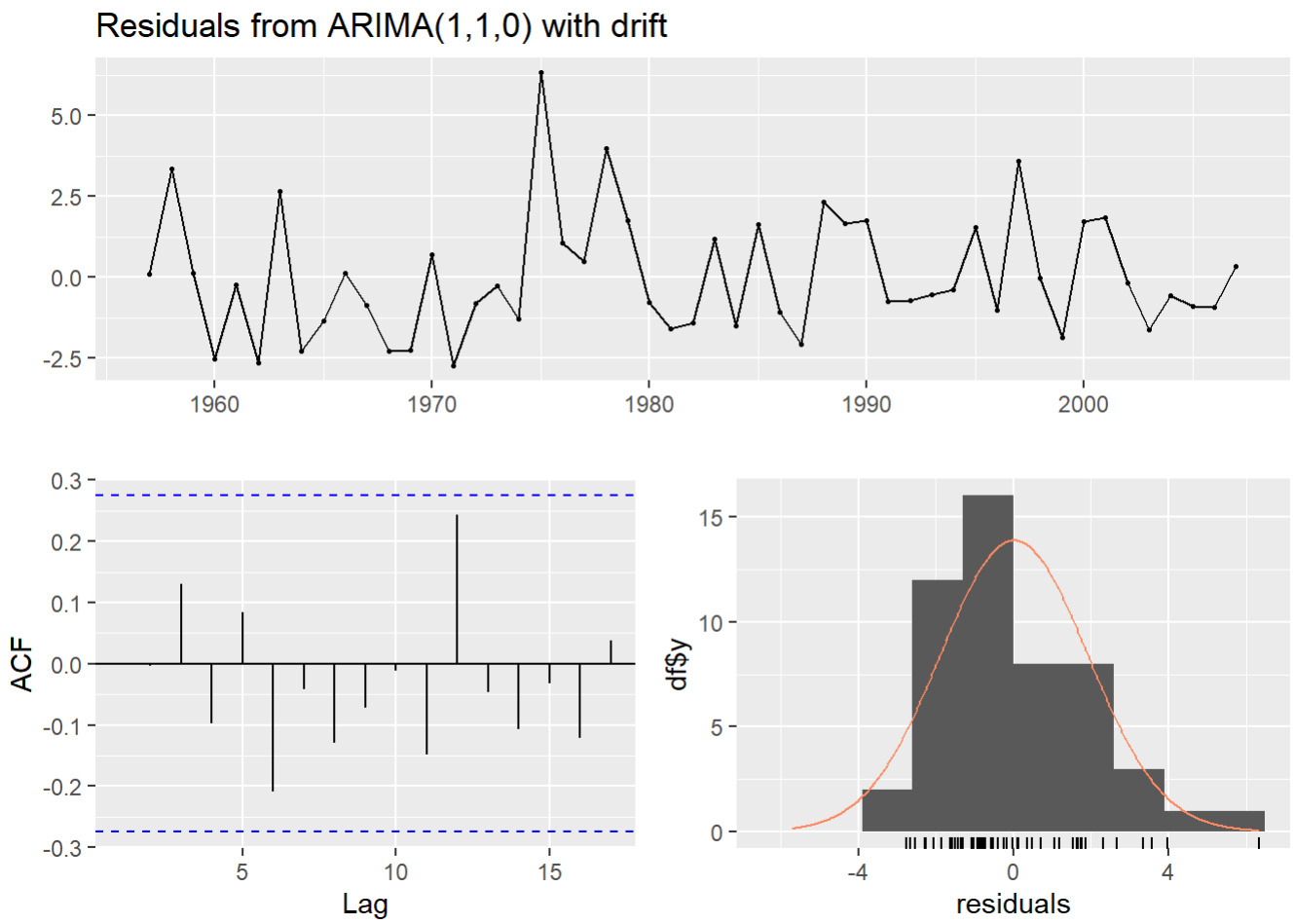
```
summary(fit6)
```

```
## Series: MonacoPoleTrain
## ARIMA(2,0,0) with non-zero mean
##
## Coefficients:
##          ar1     ar2     mean
##       0.8487  0.1386  87.3079
## s.e.  0.1409  0.1431  10.5037
##
## sigma^2 = 4.32:  log likelihood = -109.92
## AIC=227.83   AICc=228.7   BIC=235.56
##
## Training set error measures:
##                      ME     RMSE      MAE        MPE     MAPE     MASE
## Training set -0.5738235 2.016388 1.710012 -0.7067715 1.983155 1.010215
##                     ACF1
## Training set -0.05687589
```

```
summary(fit7)
```

```
## Series: MonacoPoleTrain
## ARIMA(0,1,0)
##
## sigma^2 = 4.168:  log likelihood = -106.63
## AIC=215.27   AICc=215.35   BIC=217.18
##
## Training set error measures:
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.5301353 2.021544 1.661512 -0.6378665 1.926756 0.9815633
##                     ACF1
## Training set -0.2321968
```

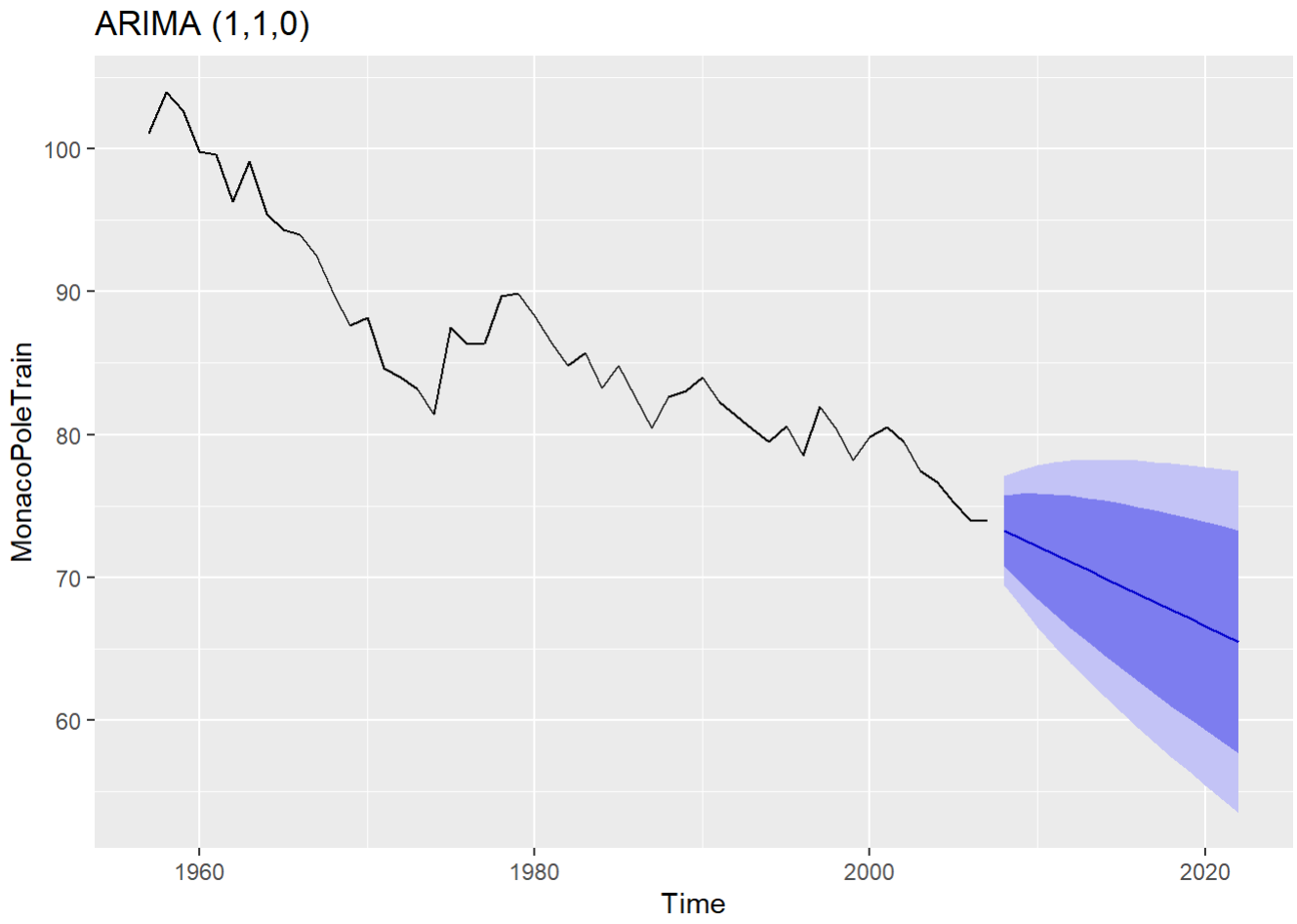
```
checkresiduals(fit)
```



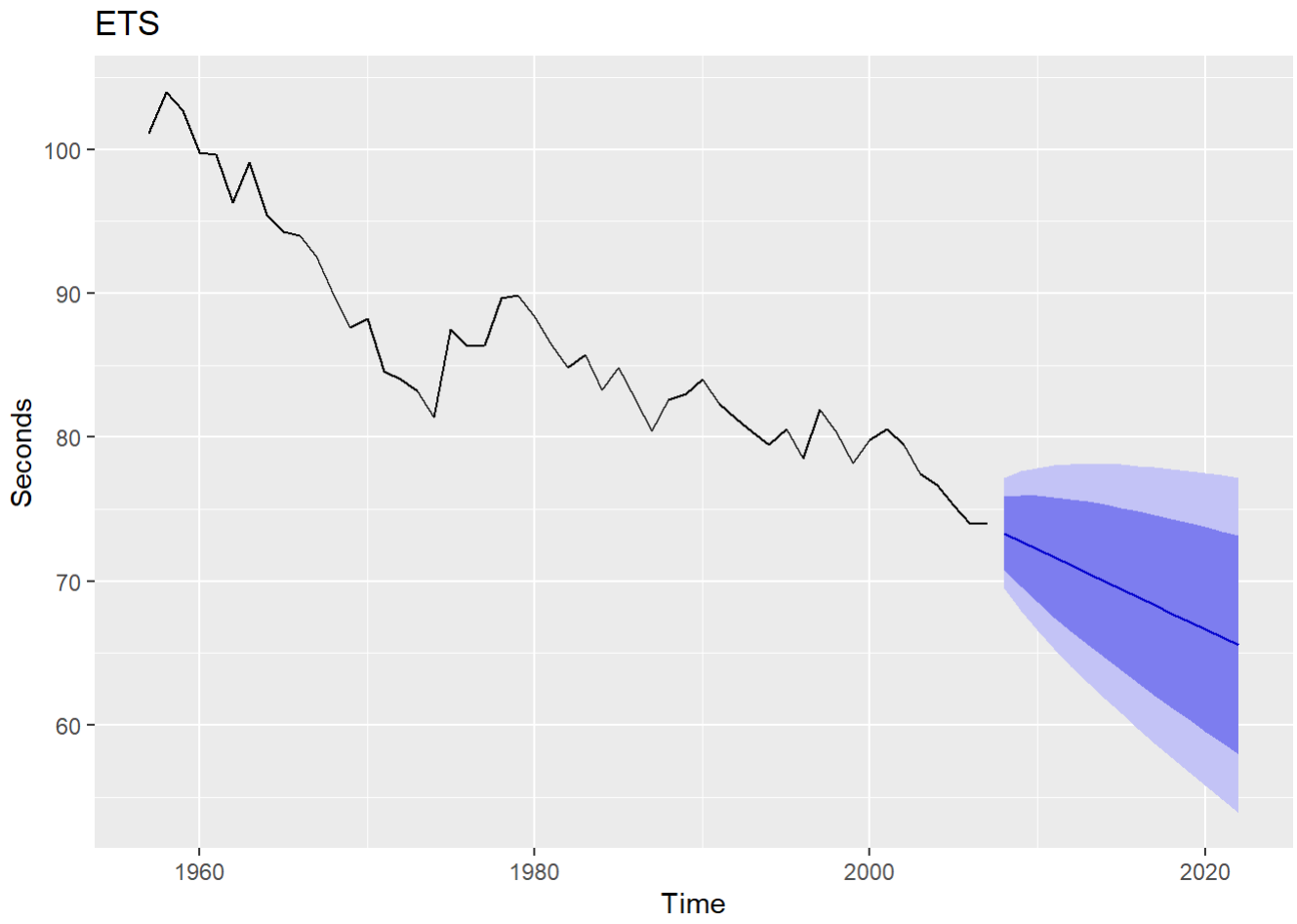

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,0) with drift
## Q* = 6.0432, df = 9, p-value = 0.7356
##
## Model df: 1.   Total lags used: 10
```

###Comparing all forcasting methods

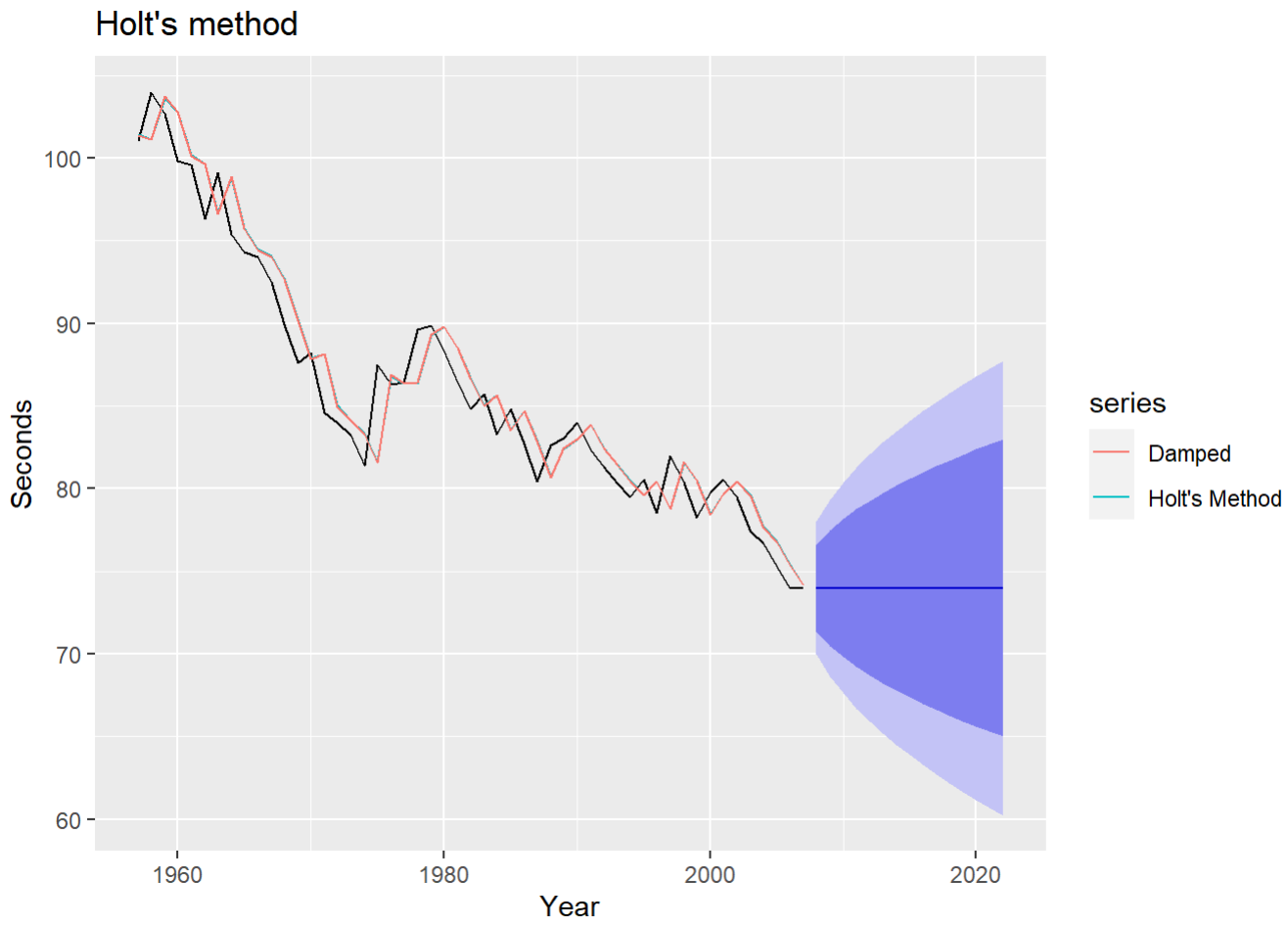
```
fit %>% forecast(h=15) %>% autoplot(include=80)+
  ggtitle("ARIMA (1,1,0)")
```



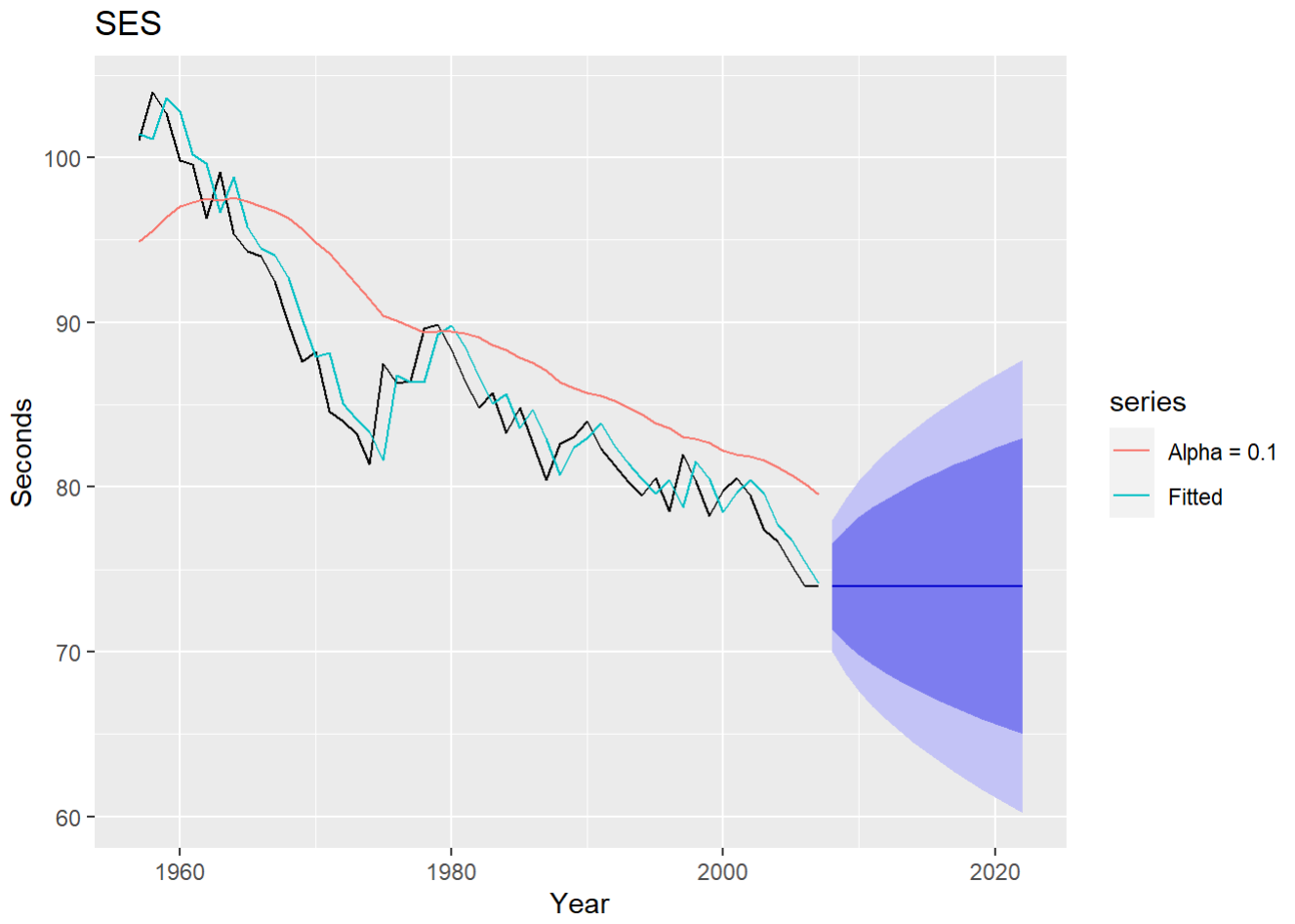

```
MonacoPoleETS %>% forecast(h=15) %>%
  autoplot() +
  ylab("Seconds") +
  ggtitle("ETS")
```




```
holtpole %>%
  autoplot() +
  autolayer(fitted(holtpole), series = "Holt's Method") +
  autolayer(fitted(holt1pole), series = "Damped") +
  ylab("Seconds") +
  xlab("Year") +
  ggtitle("Holt's method")
```




```
SESpole %>%
  autoplot() +
  autolayer(fitted(SESpole), series = "Fitted") +
  autolayer(fitted(SES1pole), series = "Alpha = 0.1") +
  ylab("Seconds") +
  xlab("Year") +
  ggtitle("SES")
```




```
summary(fit)
```

```
## Series: MonacoPoleTrain
## ARIMA(1,1,0) with drift
##
## Coefficients:
##           ar1    drift
##       -0.2555  -0.5590
## s.e.   0.1398   0.2155
##
## sigma^2 = 3.778:  log likelihood = -103.19
## AIC=212.38   AICc=212.9   BIC=218.12
##
## Training set error measures:
##                       ME     RMSE      MAE          MPE     MAPE      MASE
## Training set 0.01706904 1.885737 1.486709 0.004796726 1.719795 0.878296
##                      ACF1
## Training set 0.0008589388
```

```
summary(MonacoPoleETS)
```

```
## ETS(A,A,N)
##
## Call:
##  ets(y = MonacoPoleTrain)
##
##   Smoothing parameters:
##     alpha = 0.7589
##     beta  = 1e-04
##
##   Initial states:
##     l = 102.4359
##     b = -0.5561
##
##   sigma:  1.9673
##
##      AIC     AICc      BIC
## 275.3759 276.7092 285.0350
##
## Training set error measures:
##                       ME     RMSE      MAE         MPE     MAPE      MASE
## Training set -0.004390536 1.888554 1.491227 -0.01681682 1.723302 0.8809648
##                     ACF1
## Training set -0.02113771
```

```
accuracy(holtpole, MonacoPoleTest)
```

```
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.6133220 2.002742 1.675943 -0.7348174 1.945320 0.9900888
## Test set     -0.5442133 2.014831 1.605829 -0.8121323 2.217238 0.9486680
##                    ACF1 Theil's U
## Training set -0.1231987        NA
## Test set      0.7755574  1.965637
```

```
accuracy(holt1pole, MonacoPoleTest)
```

```
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.5968741 2.003414 1.672328 -0.7155580 1.940802 0.9879530
## Test set     -0.5359538 2.012615 1.604177 -0.8008782 2.214758 0.9476921
##                    ACF1 Theil's U
## Training set -0.1445582        NA
## Test set      0.7755574  1.962684
```

```
accuracy(SESpole, MonacoPoleTest)
```

```
##                       ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.6133220 2.002742 1.675943 -0.7348174 1.945320 0.9900888
## Test set     -0.5442133 2.014831 1.605829 -0.8121323 2.217238 0.9486680
##                    ACF1 Theil's U
## Training set -0.1231987        NA
## Test set      0.7755574  1.965637
```

```
accuracy(SES1pole, MonacoPoleTest)
```

```
##                      ME     RMSE      MAE       MPE     MAPE     MASE      ACF1
## Training set -3.117346 4.860777 4.233726 -3.890137 4.988834 2.501138 0.7800208
## Test set     -5.571048 5.899148 5.571048 -7.661541 7.661541 3.291181 0.7755574
##              Theil's U
## Training set        NA
## Test set      5.778652
```

```
accuracy(MonacoPoleTrain_naive, MonacoPoleTest)
```

```
##                     ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.54276 2.041609 1.692720 -0.6526238 1.963291 1.0000000
## Test set     -0.51940 2.008271 1.600867 -0.7783225 2.209787 0.9457363
##                    ACF1 Theil's U
## Training set -0.2441621        NA
## Test set      0.7755574  1.956846
```

###Comparing models

```
summary(fit)
```

```
## Series: MonacoPoleTrain
## ARIMA(1,1,0) with drift
##
## Coefficients:
##           ar1    drift
##       -0.2555  -0.5590
## s.e.   0.1398   0.2155
##
## sigma^2 = 3.778:  log likelihood = -103.19
## AIC=212.38   AICc=212.9   BIC=218.12
##
## Training set error measures:
##                       ME     RMSE      MAE         MPE     MAPE     MASE
## Training set 0.01706904 1.885737 1.486709 0.004796726 1.719795 0.878296
##                       ACF1
## Training set 0.0008589388
```

```
summary(MonacoPoleETS)
```

```
## ETS(A,A,N)
##
## Call:
##  ets(y = MonacoPoleTrain)
##
##   Smoothing parameters:
##     alpha = 0.7589
##     beta  = 1e-04
##
##   Initial states:
##     l = 102.4359
##     b = -0.5561
##
##   sigma:  1.9673
##
##      AIC     AICc      BIC
## 275.3759 276.7092 285.0350
##
## Training set error measures:
##                       ME     RMSE      MAE         MPE     MAPE      MASE
## Training set -0.004390536 1.888554 1.491227 -0.01681682 1.723302 0.8809648
##                     ACF1
## Training set -0.02113771
```

```
accuracy(holtpole, MonacoPoleTest)
```

```
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.6133220 2.002742 1.675943 -0.7348174 1.945320 0.9900888
## Test set     -0.5442133 2.014831 1.605829 -0.8121323 2.217238 0.9486680
##                    ACF1 Theil's U
## Training set -0.1231987        NA
## Test set      0.7755574  1.965637
```

```
accuracy(holt1pole, MonacoPoleTest)
```

```
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.5968741 2.003414 1.672328 -0.7155580 1.940802 0.9879530
## Test set     -0.5359538 2.012615 1.604177 -0.8008782 2.214758 0.9476921
##                    ACF1 Theil's U
## Training set -0.1445582        NA
## Test set      0.7755574  1.962684
```

```
accuracy(SESpole, MonacoPoleTest)
```

```
##                       ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.6133220 2.002742 1.675943 -0.7348174 1.945320 0.9900888
## Test set     -0.5442133 2.014831 1.605829 -0.8121323 2.217238 0.9486680
##                    ACF1 Theil's U
## Training set -0.1231987        NA
## Test set      0.7755574  1.965637
```

```
accuracy(SES1pole, MonacoPoleTest)
```

```
##                      ME     RMSE      MAE       MPE     MAPE     MASE      ACF1
## Training set -3.117346 4.860777 4.233726 -3.890137 4.988834 2.501138 0.7800208
## Test set     -5.571048 5.899148 5.571048 -7.661541 7.661541 3.291181 0.7755574
##              Theil's U
## Training set        NA
## Test set      5.778652
```

```
accuracy(MonacoPoleTrain_naive, MonacoPoleTest)
```

```
##                     ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.54276 2.041609 1.692720 -0.6526238 1.963291 1.0000000
## Test set     -0.51940 2.008271 1.600867 -0.7783225 2.209787 0.9457363
##                    ACF1 Theil's U
## Training set -0.2441621        NA
## Test set      0.7755574  1.956846
```

```
```