대한민국 인구 분석

20160131 김지현

2020 11 10

여자 인구 분석

자료읽기

- 1960~2017 :통계청 성별 인구추계 자료
    - 2017까지만 실제데이터
- 2018-2019 :인구주택총조사 자료

```
origianl_TSB  <- read_csv('data/krpop_MF.csv')
```

```
## Parsed with column specification:
## cols(
##   yy = col_double(),
##   male = col_double(),
##   female = col_double()
## )
```

```
origianl_TSB  <- pivot_longer(origianl_TSB ,
                    cols = c(-yy,male,female),
                    names_to ='sex',
                    values_to = 'y')
TSB <- origianl_TSB  %>%
    filter(sex == 'female') %>%
    select(yy,y)
```

```
tail(TSB)
```

```
## # A tibble: 6 x 2
##      yy        y
##   <dbl>    <dbl>
## 1  2014 25301582
## 2  2015 25429053
## 3  2016 25572567
## 4  2017 25654452
## 5  2018 25752317
## 6  2019 25827133
```

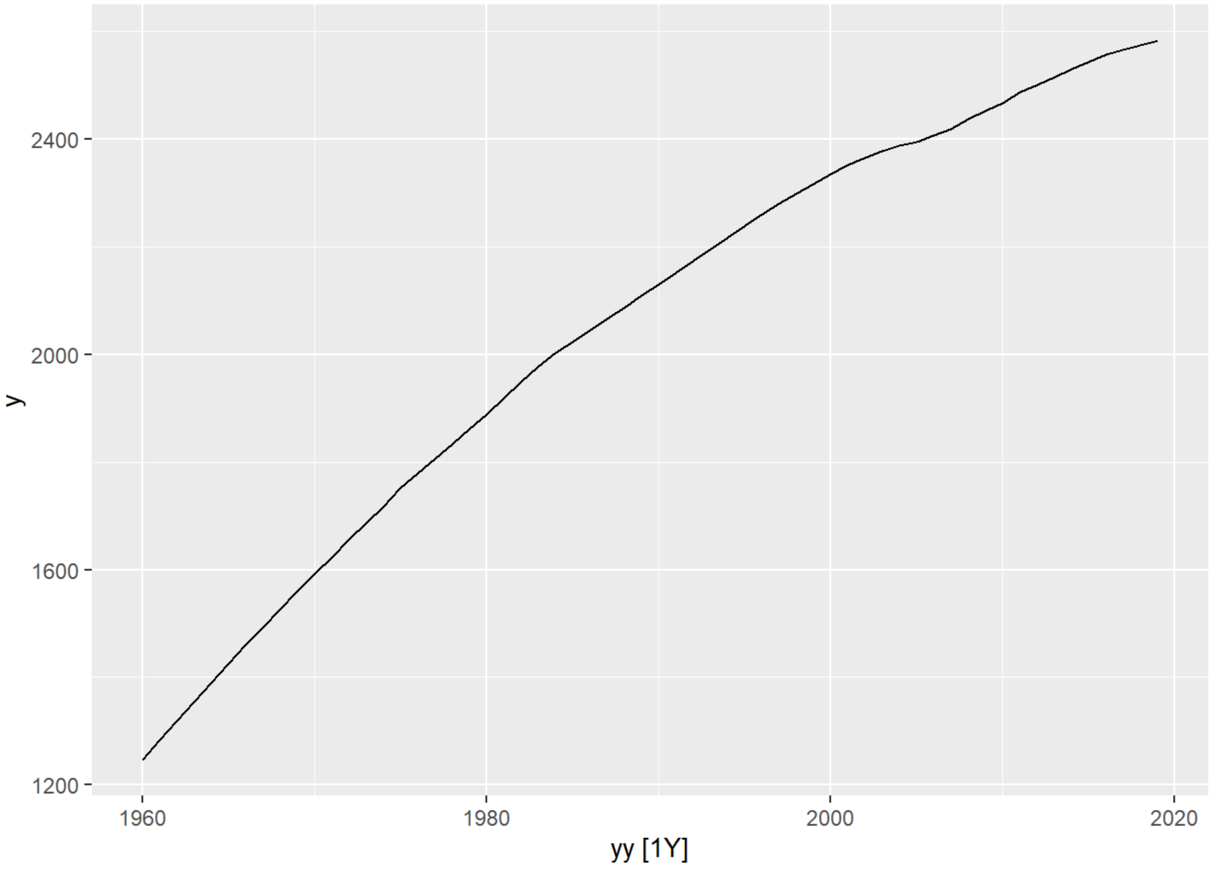
```
TSB <- mutate(TSB, y=y/10000)
#TSB <- mutate(TSB, y=y/10000, lin=1:length(y), quad=lin^2)
TSB <- as_tsibble(TSB,index=yy)
```

탐색/분할

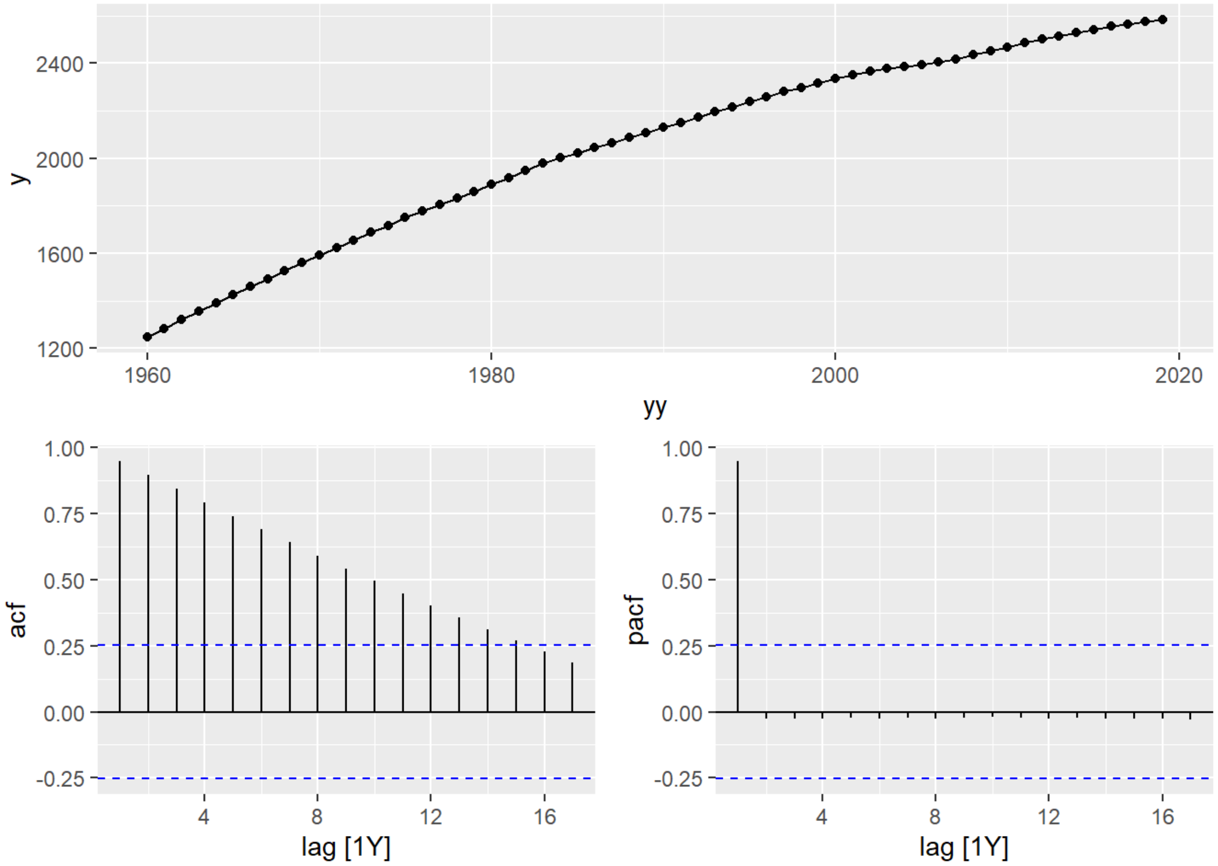
- 시계열 그림1

```
autoplot(TSB)
```

```
## Plot variable not specified, automatically selected `.vars = y`
```



```
gg_tsdisplay(TSB,y,plot_type = 'partial')
```

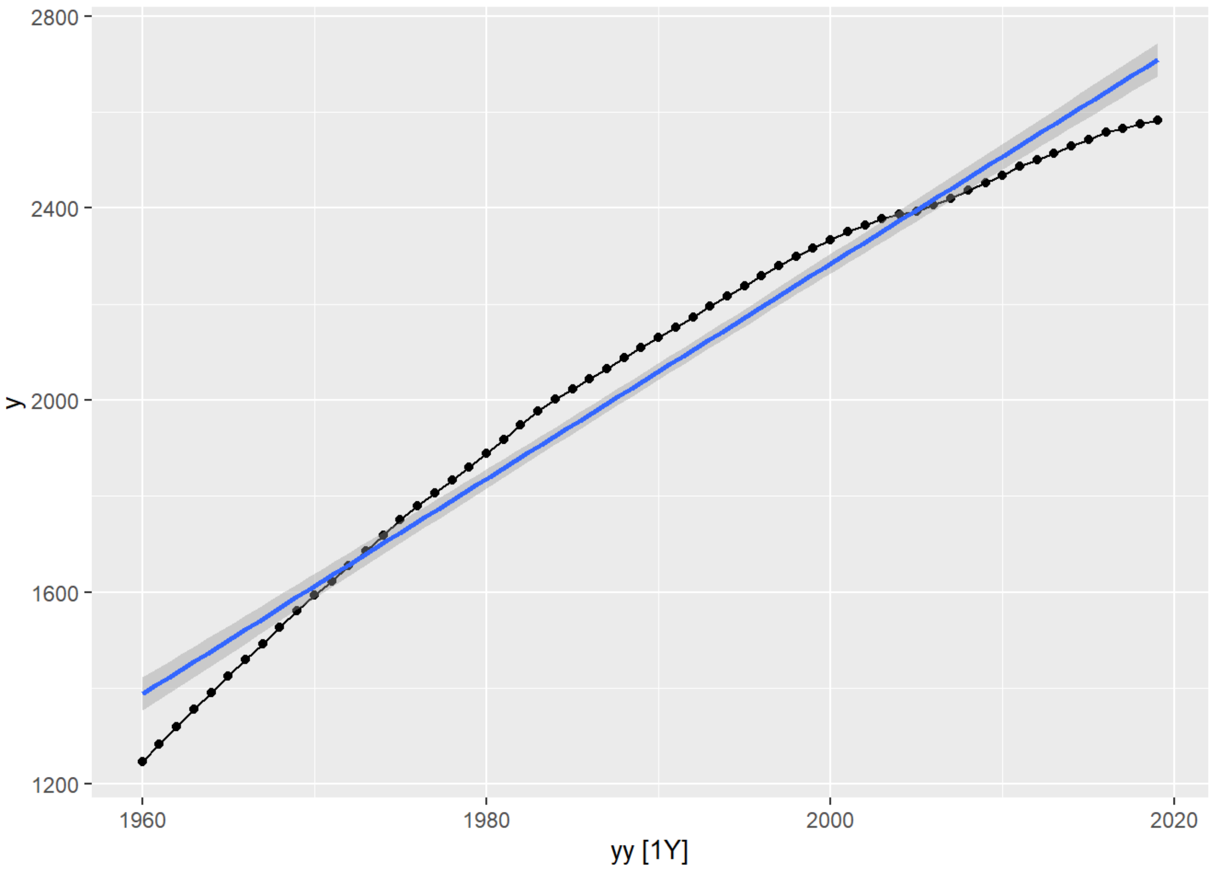


- 시계열그림2

```
TRN <- filter_index(TSB, .~2000)
TST <- filter_index(TSB, 2001~.)
autoplot(TSB) + geom_point() + geom_smooth(method = 'lm')
```

```
## Plot variable not specified, automatically selected `.vars = y`
```

```
## `geom_smooth()` using formula 'y ~ x'
```



모형

- 모형적합

```
# 한꺼번에 하기 1: 기본모형만
MM <- model(TRN,
            # ETS 자동선택
            ETS = ETS(log(y)),
            # ETS(E=A, T=A, S=N) = Holt Linear
            AAN = ETS(log(y)~error('A')+trend('A')+season('N')),
            #ETS(E=A,T=ad,S=N) = Holt
            ADN = ETS(log(y)~error('A')+trend('Ad', phi=0.9) + season('N')),
            LLT = TSLM(log(y)~trend()),
            LLTK = TSLM(log(y)~trend(knots=c(1990))))
            #(생략)LQT = TSLM(log(y)~lin+quad)
```

- TRN에서 모형적합도 비교
  - TRN에서 MAPE 기준 AAN=0.065 < ETS=0.066 < ADN=0.117 < LLTK = 1.91 < LLT=2.92
  - ALCC 기준 ETS=-410 < AAN=-408 < ADN=-383 < LLTK=-303 < LLT=-272

```
accuracy(MM)
```

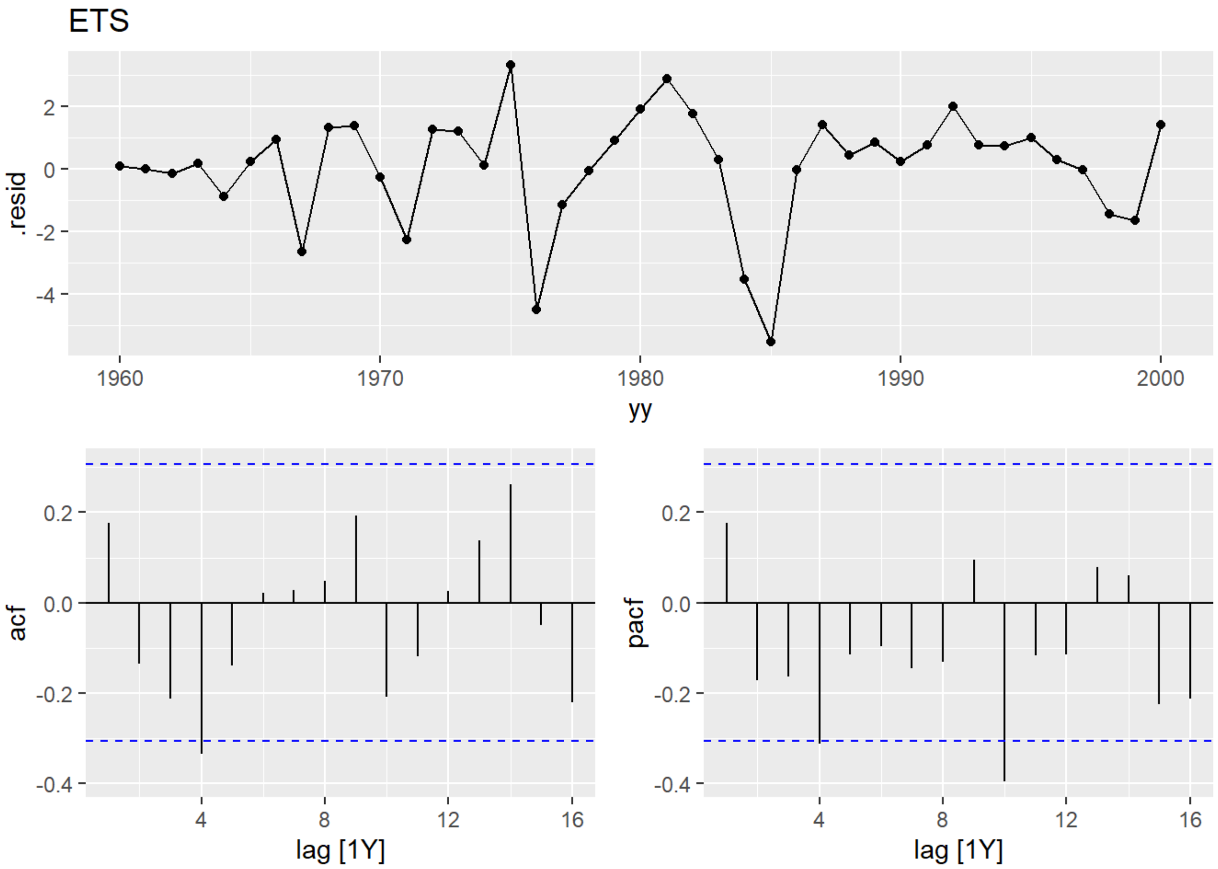
```
## # A tibble: 5 x 9
##   .model .type          ME  RMSE   MAE      MPE   MAPE   MASE  ACF1
##   <chr>  <chr>       <dbl> <dbl> <dbl>    <dbl>  <dbl>  <dbl> <dbl>
## 1 ETS    Training  0.0800   1.77  1.26  0.00365 0.0680 0.0464 0.176
## 2 AAN    Training -1.31     2.29  1.60 -0.0742  0.0899 0.0587 0.210
## 3 ADN    Training  2.39     3.01  2.67  0.134   0.148  0.0983 0.212
## 4 LLT    Training  0.0975  59.1  50.8  -0.0532  2.80   1.87   0.882
## 5 LLTK   Training  0.00949 36.4  31.5  -0.0222  1.78   1.16   0.916
```

```
glance(MM)
```

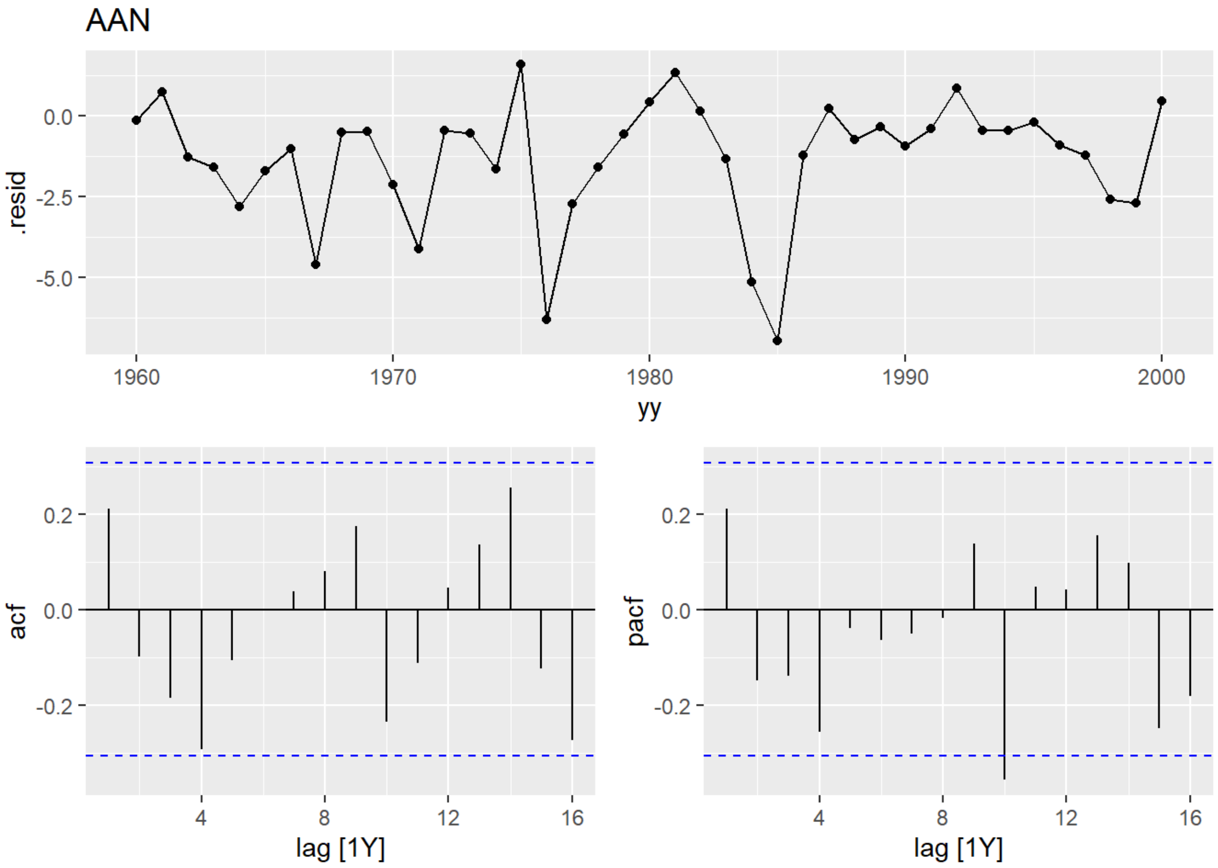
```
## # A tibble: 5 x 18
##   .model  sigma2 log_lik   AIC  AICc   BIC      MSE     AMSE      MAE r_squared
##   <chr>    <dbl>   <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl>    <dbl>     <dbl>
## 1 ETS    1.84e-8   209.  -406. -404. -396.  9.10e-7  5.17e-6  9.04e-5    NA
## 2 AAN    1.83e-6   197.  -384. -382. -375.  1.65e-6  1.29e-5  8.98e-4    NA
## 3 ADN    3.33e-6   185.  -360. -359. -352.  2.92e-6  1.27e-5  1.49e-3    NA
## 4 LLT    1.11e-3    82.3 -275. -274. -270. NA       NA       NA           0.968
## 5 LLTK   4.76e-4   100.  -309. -308. -302. NA       NA       NA           0.987
## # ... with 8 more variables: adj_r_squared <dbl>, statistic <dbl>,
## #   p_value <dbl>, df <int>, CV <dbl>, deviance <dbl>, df.residual <int>,
## #   rank <int>
```

- 적합값 저장/잔차 분석

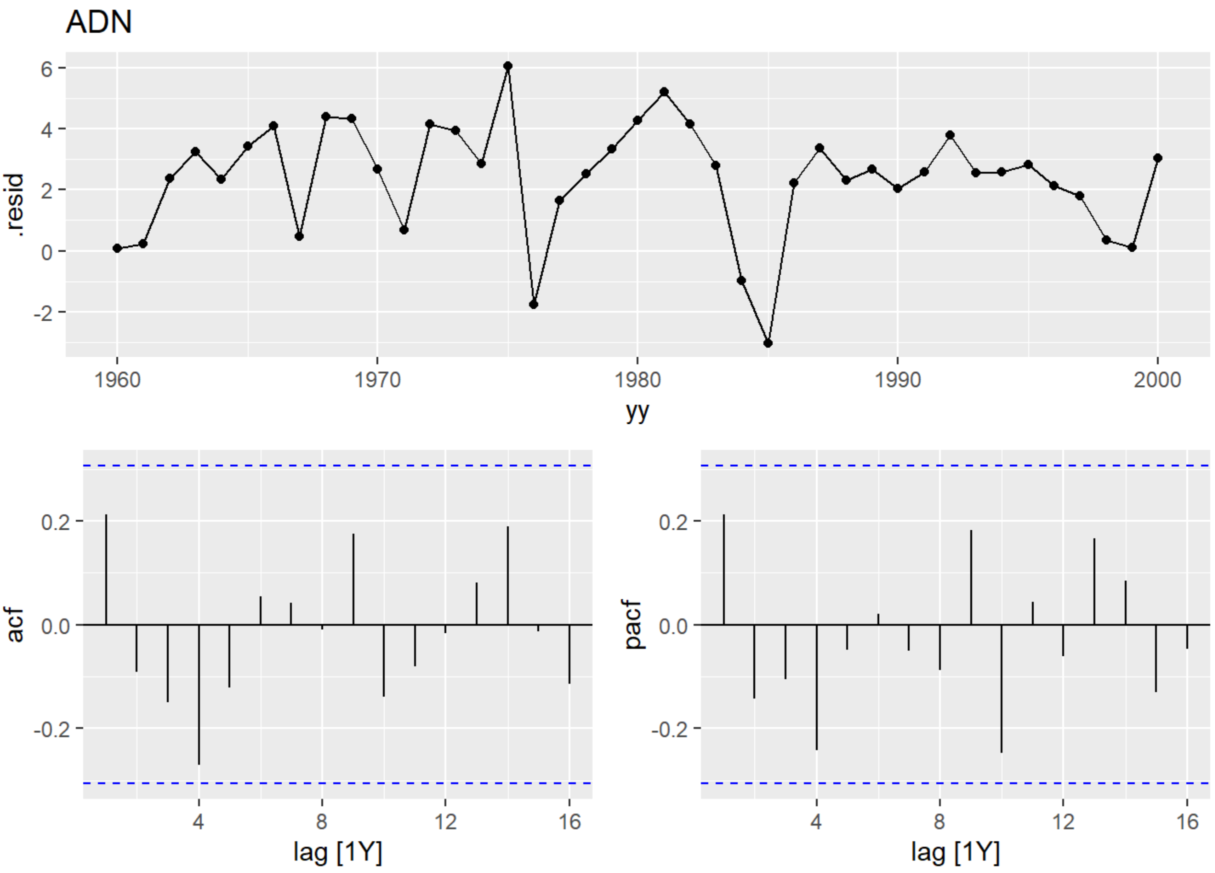
```
AA <- augment(MM)

# ETS(자동선택) 잔차분석
gg_tsdisplay(filter(AA, .model=="ETS"), .resid, plot_type = 'partial') + ggtitle(
  'ETS')
```



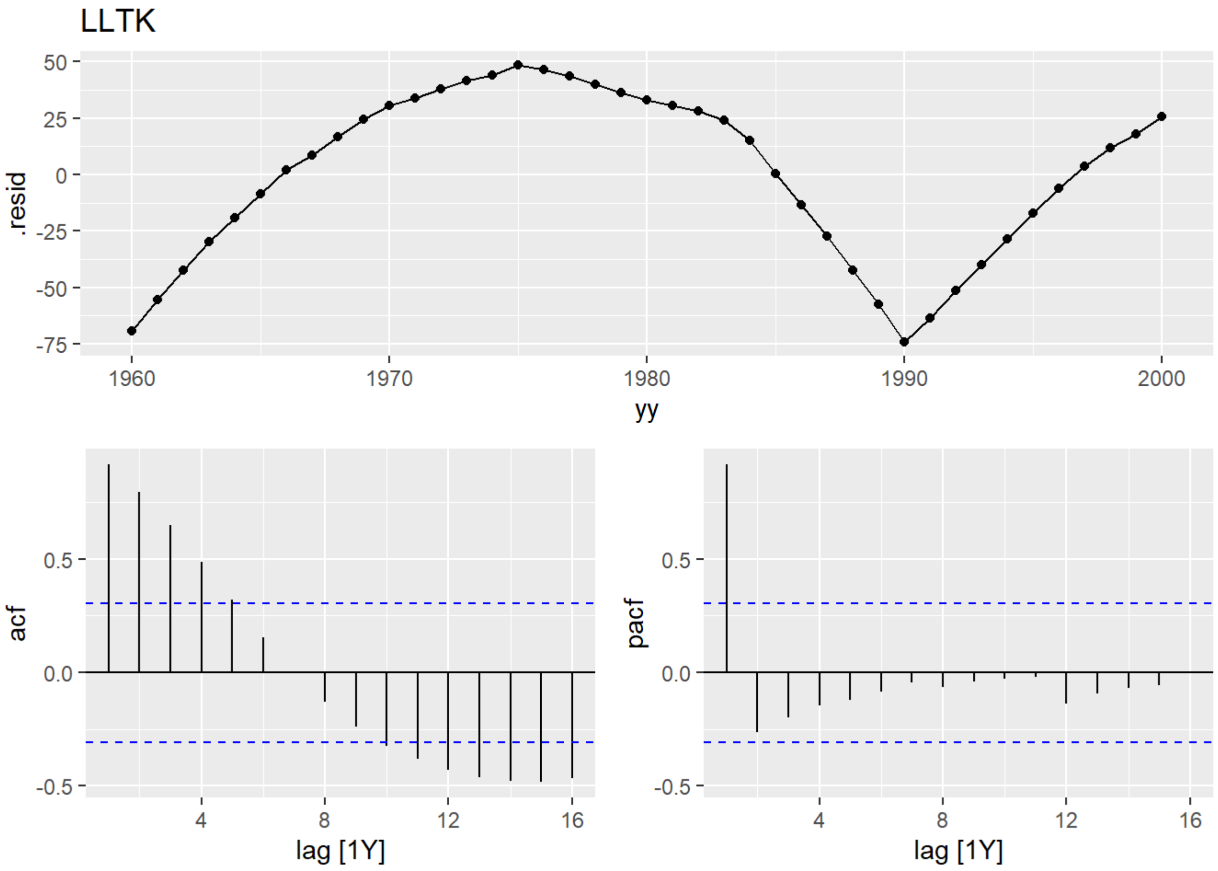
```
# ETS(A,A,N) 잔차분석
gg_tsdisplay(filter(AA, .model=="AAN"), .resid, plot_type = 'partial') + ggtitle(
  'AAN')
```



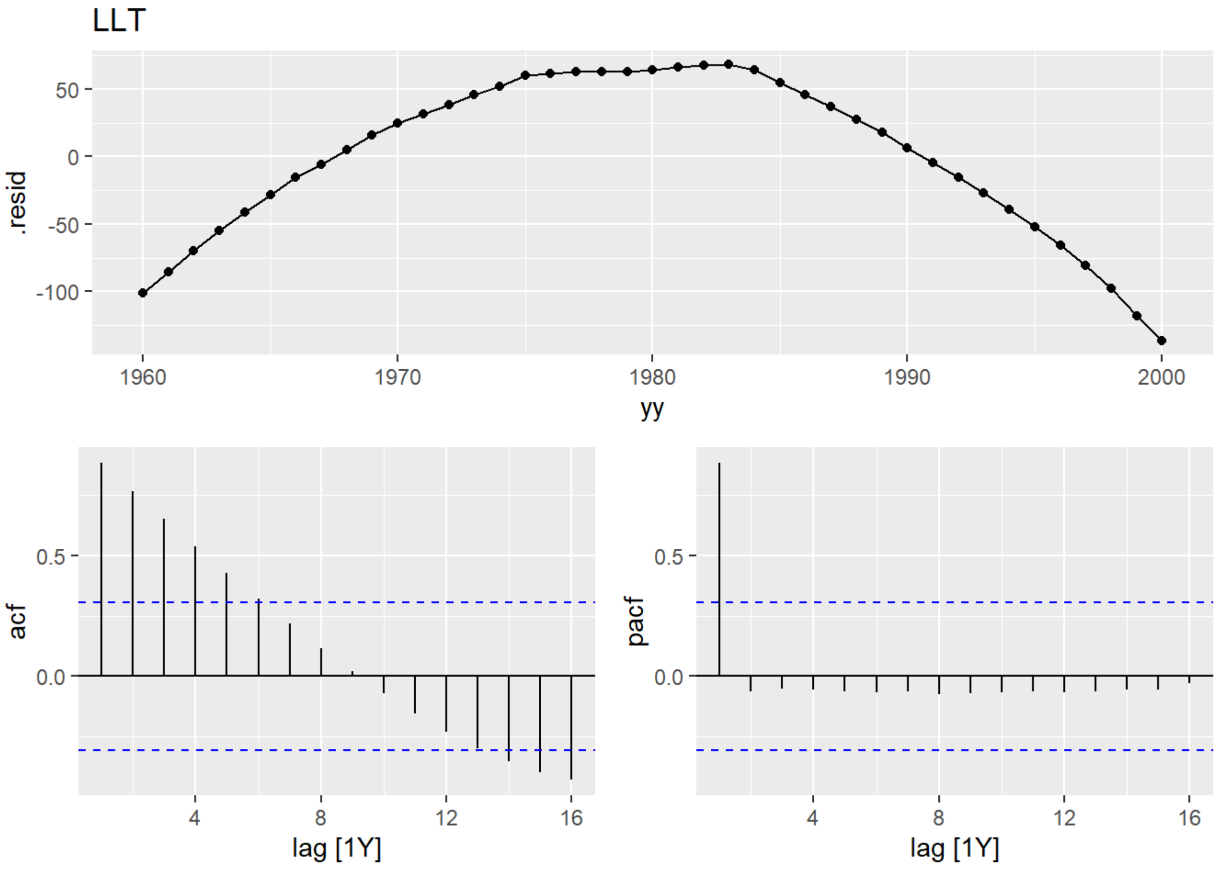
```
## ETS(A,Ad,N) 잔차분석
gg_tsdisplay(filter(AA, .model=="ADN"), .resid, plot_type = 'partial') + ggtitle(
  'ADN')
```



```
# LLTK 잔차분석
gg_tsdisplay(filter(AA, .model=="LLTK"), .resid, plot_type = 'partial') + ggtitle
  ('LLTK')
```



```
# LLT 잔차분석
gg_tsdisplay(filter(AA, .model=="LLT"), .resid, plot_type = 'partial') + ggtitle(
  'LLT')
```



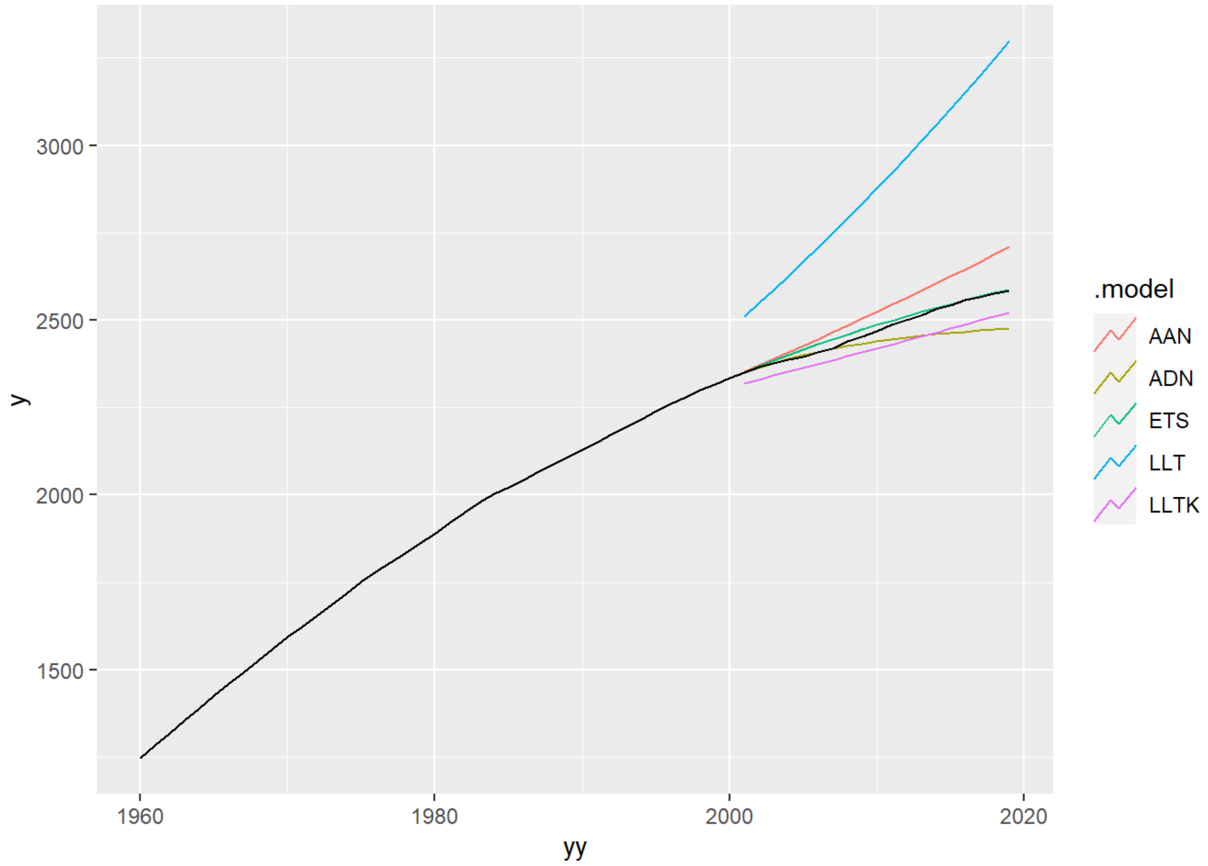
- 예측값 저장(TST)/ 모형평가(반드시 TST로 지정해야함)

```
FF <- forecast(MM, h=19)
# FF <- forecast(MM,new_data=TST)
accuracy(FF, data=TSB)
```

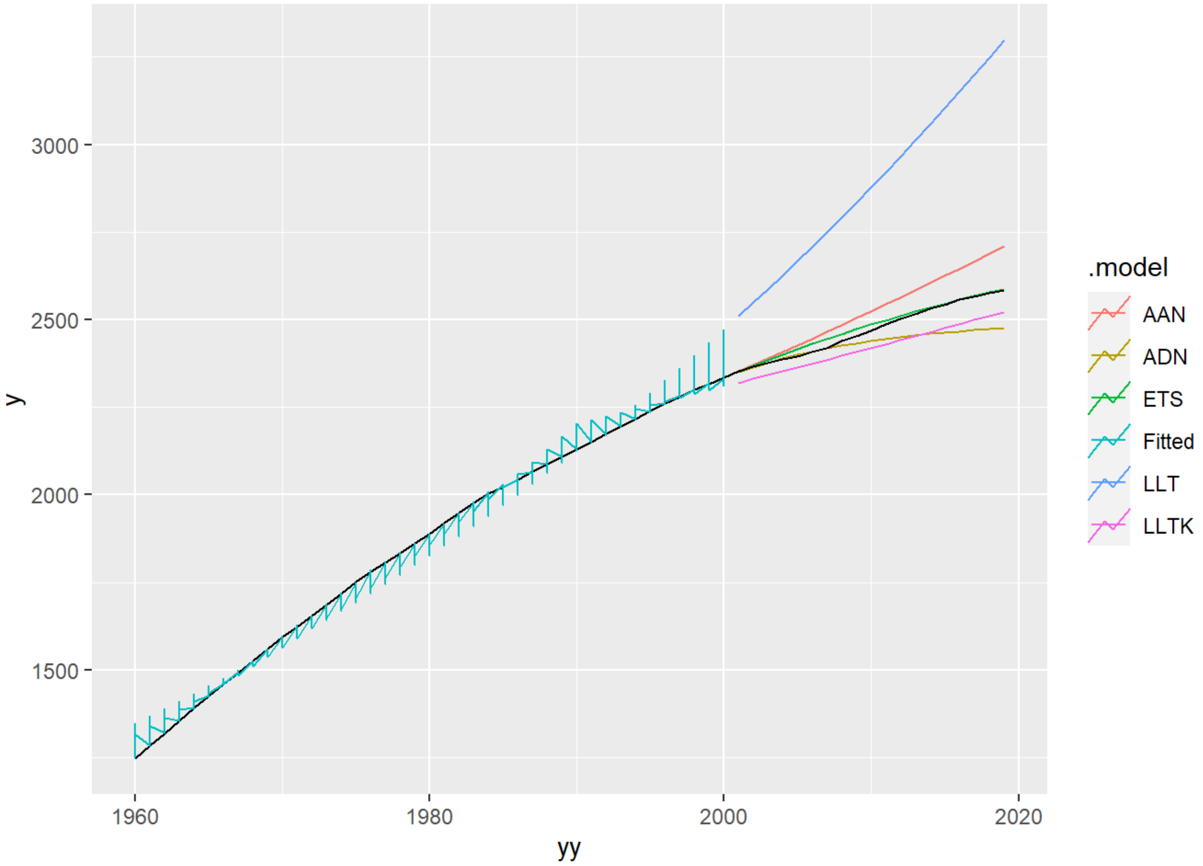
```
## # A tibble: 5 x 9
##   .model .type     ME  RMSE   MAE    MPE  MAPE  MASE  ACF1
##   <chr>  <chr>  <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1 AAN    Test   -57.0  66.5  57.0 -2.27  2.27  2.10  0.796
## 2 ADN    Test    39.7  56.1  40.7  1.56  1.60  1.49  0.882
## 3 ETS    Test   -10.5  13.4  10.5 -0.428 0.431 0.387 0.849
## 4 LLT    Test  -418.  449.  418.  -16.7  16.7  15.4  0.833
## 5 LLTK   Test    49.8  51.8  49.8  2.00  2.00  1.83  0.909
```

- 예측값 시각화/ 개별모형 검토

```
# 과거값, 예측값을 한번에 찍는 방법
autoplot(FF, data=TSB, level = NULL)
```



```
# 과거값, 적합값, 예측값을 한번에 찍는 방법(불안정)
autoplot(FF, TSB, level = NULL) +
  geom_line(aes(y=.fitted, color='Fitted'), data=AA)
```



```
# 개별모형 검토
# ETS의 과거값, 적합값, 예측값 시각화
METS <- select(MM, ETS)
report(METS)
```

```
## Series: y
## Model: ETS(M,Ad,N)
## Transformation: log(.x)
##   Smoothing parameters:
##     alpha = 0.9998998
##     beta  = 0.6705683
##     phi   = 0.9638782
##
##   Initial states:
##         l         b
##  7.097662 0.0312273
##
##   sigma^2:  0
##
##        AIC      AICc       BIC
## -406.0903 -403.6197 -395.8089
```

```
# gg_tsresiduals(ETS) : 위 결과와 동일
features(filter(AA, .model=='ETS'), .resid, ljung_box, lag=10, dof=3)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 ETS       15.3    0.0322
```

```
G1 <- autoplot(filter(FF, .model=='ETS'),data=TSB)+
  geom_line(aes(y=.fitted, color='Fitted'), data = filter(AA, .model=='ETS')) + gg
  title('ETS')

# AAN의 과거값, 적합값, 예측값 시각화
MAAN <- select(MM, AAN)
report(MAAN)
```

```
## Series: y
## Model: ETS(A,A,N)
## Transformation: log(.x)
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 0.6834533
##
##   Initial states:
##         l          b
##  7.099391 0.02855318
##
##   sigma^2:  0
##
##       AIC      AICc       BIC
## -383.6573 -381.9430 -375.0894
```

```
# gg_tsresiduals(MAAN) : 위 결과와 동일
features(filter(AA, .model=='AAN'), .resid, ljung_box, lag=10, dof=2)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 AAN       13.8    0.0874
```

```
G2 <- autoplot(filter(FF, .model=='AAN'),data=TSB)+
  geom_line(aes(y=.fitted, color='Fitted'), data =filter(AA, .model=='AAN')) + ggt
  itle('ETS(AAN)')

# ADN의 과거값, 적합값, 예측값 시각화
MADN <- select(MM, ADN)
report(MADN)
```

```
## Series: y
## Model: ETS(A,Ad,N)
## Transformation: log(.x)
##   Smoothing parameters:
##     alpha = 0.9993705
##     beta  = 0.7109512
##     phi   = 0.9
##
##   Initial states:
##         l          b
##  7.095709 0.0356079
##
##   sigma^2:  0
##
##       AIC      AICc       BIC
## -360.2171 -358.5028 -351.6492
```

```
# gg_tsresiduals(MADN) : 위 결과와 동일
features(filter(AA, .model=='ADN'), .resid, ljung_box, lag=10, dof=3)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 ADN       10.6     0.155
```

```
G3 <- autoplot(filter(FF, .model=='ADN'), data=TSB)+geom_line(aes(y=.fitted, colo
  r='Fitted'), data=filter(AA, .model=='ADN'))+ggtitle('ETS(ADN)')

# LLT의 과거값, 적합값, 예측값 시각화
MLLT <- select(MM, LLT)
report(MLLT)
```

```
## Series: y
## Model: TSLM
## Transformation: log(.x)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.078237 -0.022884  0.008426  0.030537  0.035496
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.1909092  0.0106037  678.15   <2e-16 ***
## trend()     0.0151571  0.0004399   34.45   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03333 on 39 degrees of freedom
## Multiple R-squared: 0.9682,  Adjusted R-squared: 0.9674
## F-statistic:  1187 on 1 and 39 DF, p-value: < 2.22e-16
```

```
# gg_tsresiduals(MLLT) : 위 결과와 동임
features(filter(AA, .model=='LLT'),  .resid, ljung_box, lag=10, dof=2)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 LLT       112.         0
```

```
G4 <- autoplot(filter(FF, .model=='LLT'), data=TSB)+ geom_line(aes(y=.fitted ,col
  or='Fitted'), data=filter(AA, .model=='LLT'))+ggtitle('LLT:Linear Trend')

# LLTK의 과거값, 적합값, 예측값 시각화
MLLTK <-select(MM, LLTK)
report(MLLTK)
```

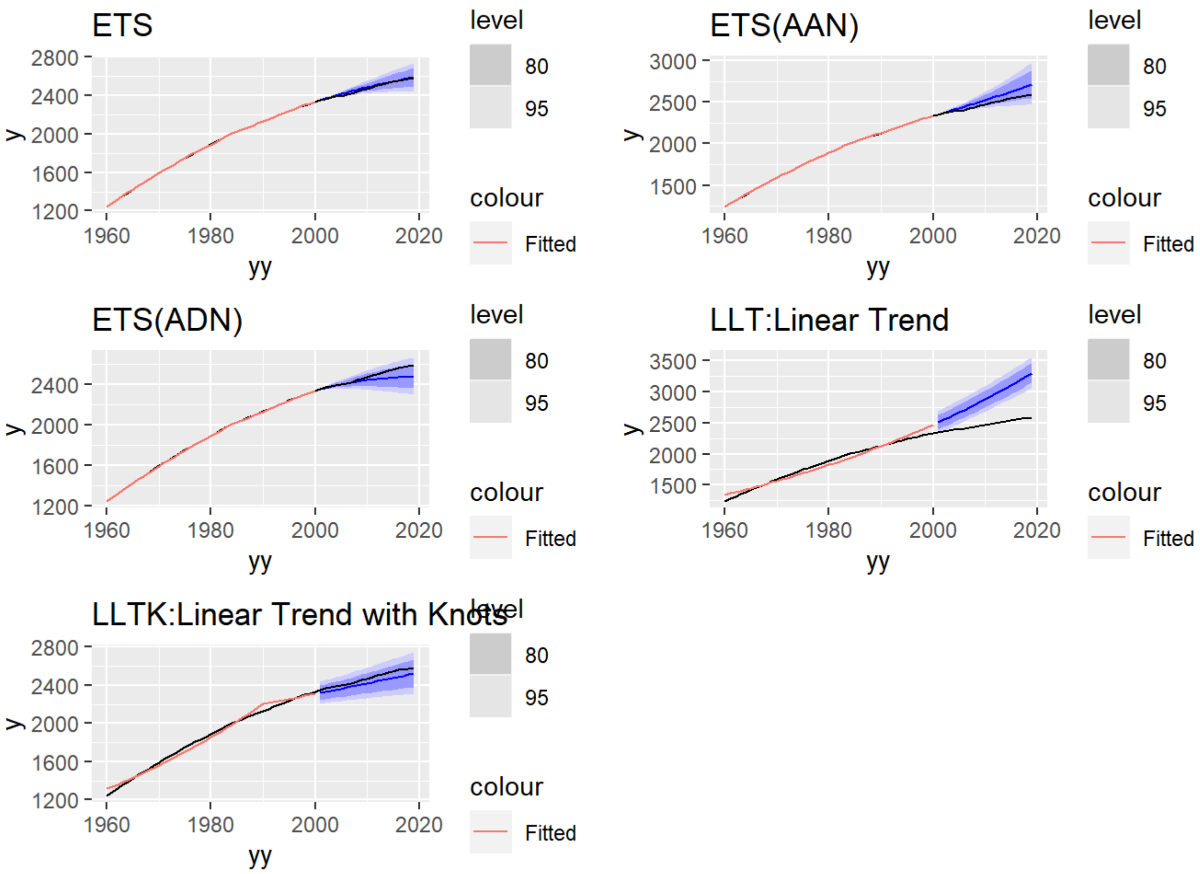
```
## Series: y
## Model: TSLM
## Transformation: log(.x)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.054089 -0.013539  0.005185  0.017588  0.028159
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  7.164709   0.007819  916.31  < 2e-16 ***
## trend(knots = c(1990))trend     0.017210   0.000403   42.70  < 2e-16 ***
## trend(knots = c(1990))trend_31 -0.012601   0.001731   -7.28 1.03e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02182 on 38 degrees of freedom
## Multiple R-squared: 0.9867,  Adjusted R-squared: 0.986
## F-statistic:  1412 on 2 and 38 DF, p-value: < 2.22e-16
```

```
# gg_tsresiduals(MLLTK) : 위 결과와 동일
features(filter(AA, .model=='LLTK'), .resid, ljung_box, lag=10, dof=3)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 LLTK      113.         0
```

```
G5 <- autoplot(filter(FF, .model=='LLTK'), data=TSB)+geom_line(aes(y=.fitted,colo
  r='Fitted'), data=filter(AA, .model=='LLTK'))+ ggtitle('LLTK:Linear Trend with Kn
  ots')
```

```
gridExtra::grid.arrange(G1,G2,G3,G4,G5, nrow=3)
```



```
# 예측값 확인
cbind(
  tail(TSB)[,c('yy','y')],
  ADN = tail(filter(FF,.model=='ADN')$.mean),
  ETS = tail(filter(FF,.model=='ETS')$.mean),
  LLTK= tail(filter(FF,.model=='LLTK')$.mean),
  AAN = tail(filter(FF,.model=='AAN')$.mean),
  LLT = tail(filter(FF,.model=='LLT')$.mean))
```

```
##     yy        y      ADN      ETS     LLTK      AAN      LLT
## 1 2014 2530.158 2459.372 2534.585 2464.020 2604.216 3057.121
## 2 2015 2542.905 2463.303 2545.873 2475.518 2624.841 3103.831
## 3 2016 2557.257 2466.868 2556.814 2487.076 2645.666 3151.256
## 4 2017 2565.445 2470.103 2567.417 2498.692 2666.694 3199.406
## 5 2018 2575.232 2473.039 2577.693 2510.369 2687.931 3248.293
## 6 2019 2582.713 2475.707 2587.651 2522.105 2709.381 3297.927
```

남자 인구 분석

자료읽기

```
TSB <- origianl_TSB  %>%
    filter(sex == 'male') %>%
    select(yy,y)
```

```
tail(TSB)
```

```
## # A tibble: 6 x 2
##      yy        y
##   <dbl>    <dbl>
## 1  2014 25445077
## 2  2015 25585894
## 3  2016 25696987
## 4  2017 25768055
## 5  2018 25877195
## 6  2019 25952070
```

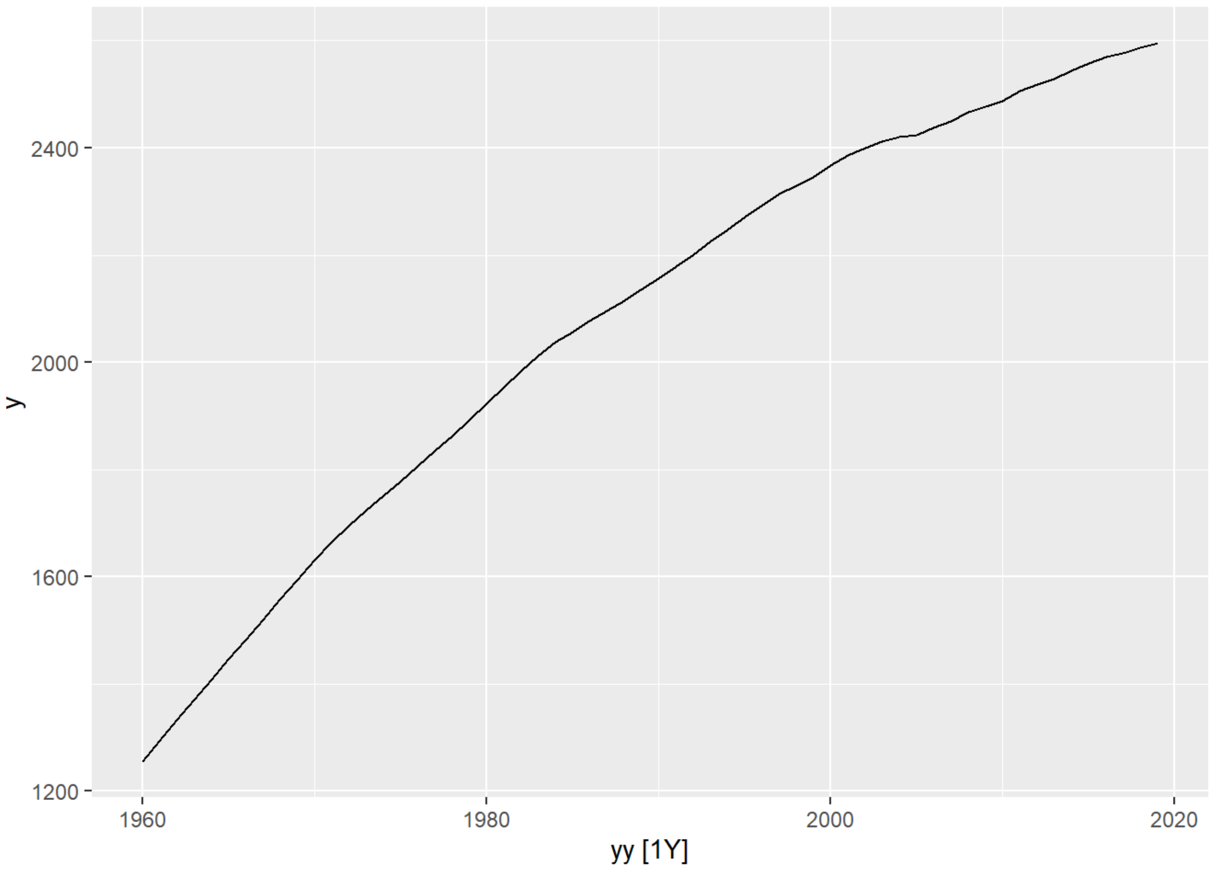
```
TSB <- mutate(TSB, y=y/10000)
#TSB <- mutate(TSB, y=y/10000, lin=1:length(y), quad=lin^2)
TSB <- as_tsibble(TSB,index=yy)
```

탐색/분할

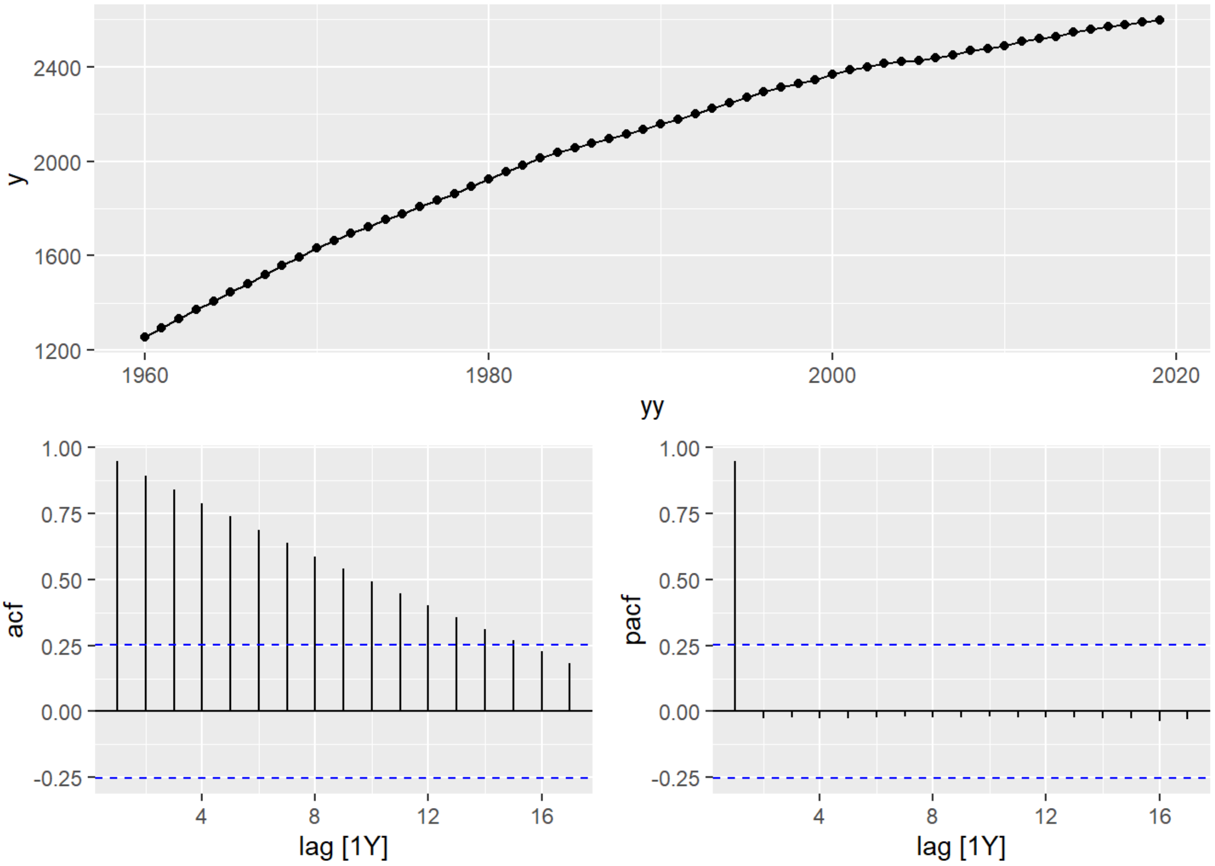
- 시계열 그림1

```
autoplot(TSB)
```

```
## Plot variable not specified, automatically selected `.vars = y`
```



```
gg_tsdisplay(TSB,y,plot_type = 'partial')
```

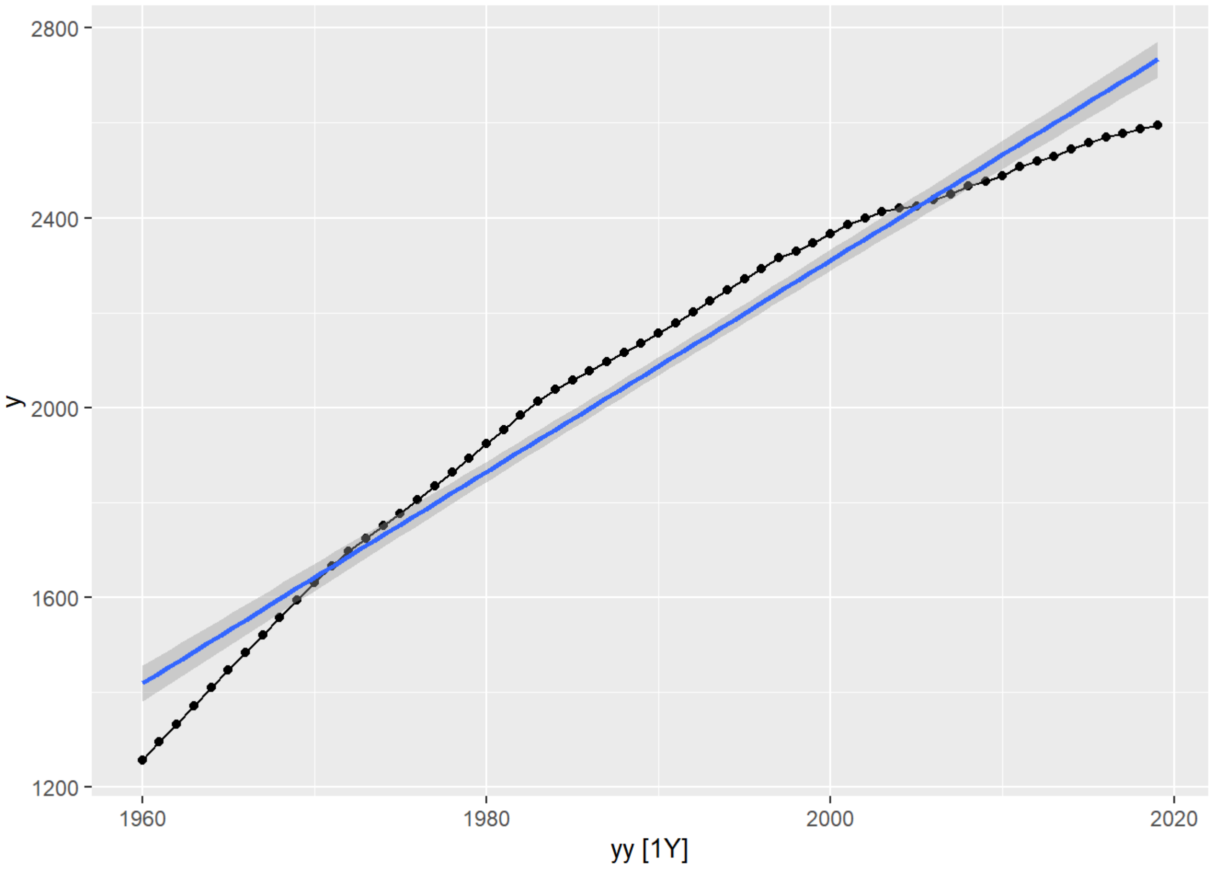


- 시계열그림2

```
TRN <- filter_index(TSB, .~2000)
TST <- filter_index(TSB, 2001~.)
autoplot(TSB) + geom_point() + geom_smooth(method = 'lm')
```

```
## Plot variable not specified, automatically selected `.vars = y`
```

```
## `geom_smooth()` using formula 'y ~ x'
```



모형

- 모형적합

```
# 한꺼번에 하기 1: 기본모형만
MM <- model(TRN,
            # ETS 자동선택
            ETS = ETS(log(y)),
            # ETS(E=A, T=A, S=N) = Holt Linear
            AAN = ETS(log(y)~error('A')+trend('A')+season('N')),
            #ETS(E=A,T=ad,S=N) = Holt
            ADN = ETS(log(y)~error('A')+trend('Ad', phi=0.9) + season('N')),
            LLT = TSLM(log(y)~trend()),
            LLTK = TSLM(log(y)~trend(knots=c(1990))))
            #(생략)LQT = TSLM(log(y)~lin+quad)
```

- TRN에서 모형적합도 비교

  - TRN에서 MAPE 기준 AAN=0.065 < ETS=0.066 < ADN=0.117 < LLTK = 1.91 < LLT=2.92
  - ALCc 기준 ETS=-410 < AAN=-408 < ADN=-383 < LLTK=-303 < LLT=-272

```
accuracy(MM)
```

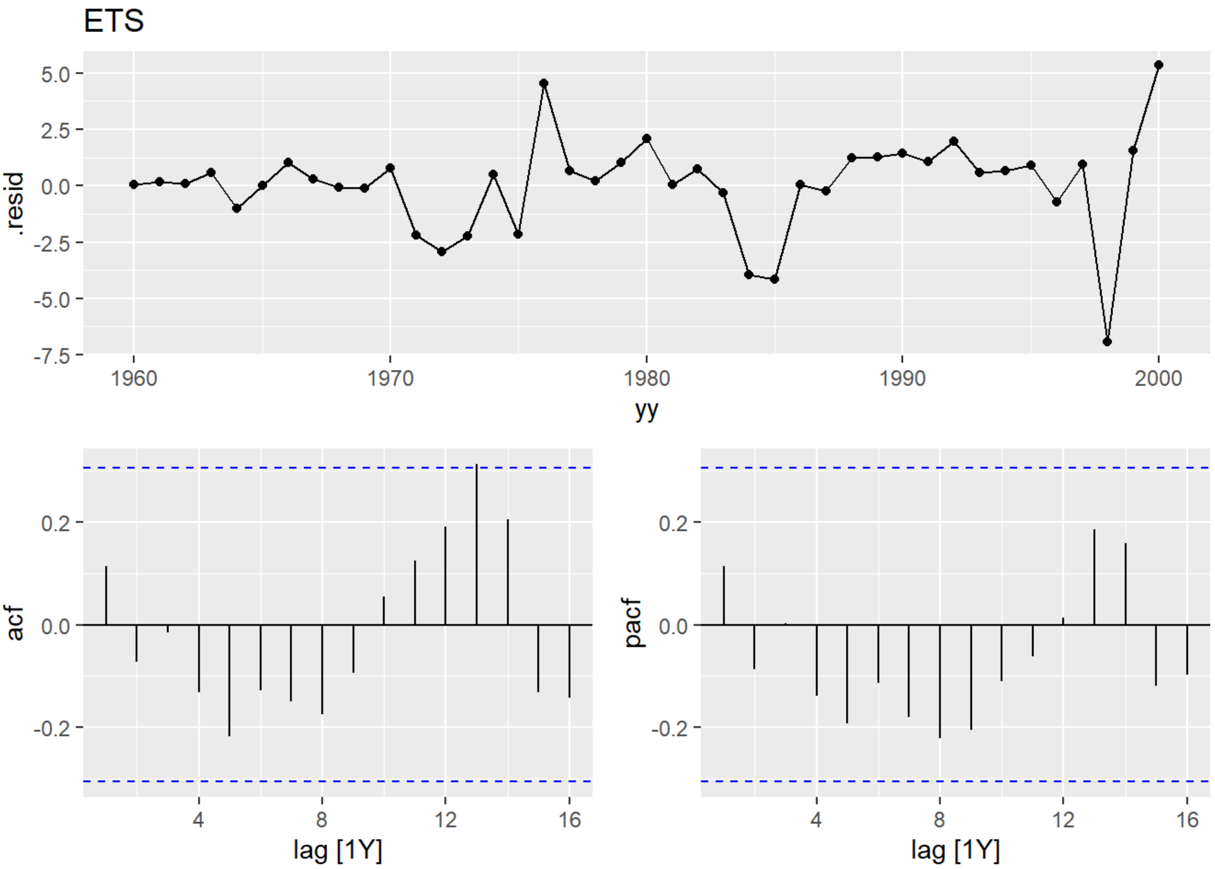
```
## # A tibble: 5 x 9
##   .model .type        ME  RMSE   MAE      MPE   MAPE   MASE  ACF1
##   <chr>  <chr>     <dbl> <dbl> <dbl>    <dbl>  <dbl>  <dbl> <dbl>
## 1 ETS    Training  0.0749  2.09  1.39  0.00270 0.0704 0.0500 0.114
## 2 AAN    Training -0.903   2.35  1.60 -0.0519  0.0855 0.0577 0.109
## 3 ADN    Training  2.38    3.28  2.91  0.130   0.156  0.105  0.328
## 4 LLT    Training  0.191  63.8  55.4  -0.0625  3.03   1.99   0.884
## 5 LLTK   Training  0.0737 42.0  36.5  -0.0294  2.04   1.31   0.914
```

```
glance(MM)
```

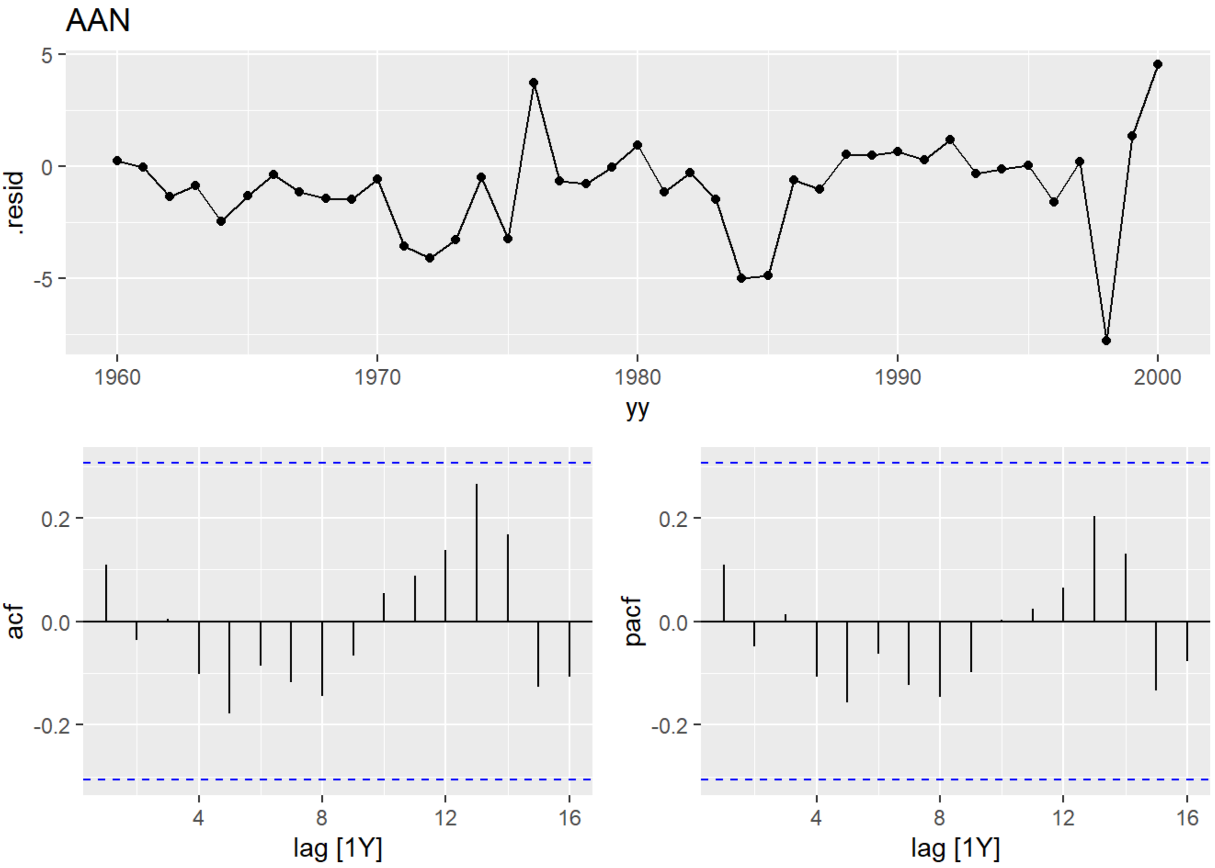
```
## # A tibble: 5 x 18
##   .model  sigma2 log_lik   AIC  AICc   BIC      MSE     AMSE      MAE r_squared
##   <chr>    <dbl>   <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl>    <dbl>     <dbl>
## 1 ETS    2.05e-8   207.  -401. -399. -391.  1.04e-6  7.62e-6  9.29e-5    NA
## 2 AAN    1.60e-6   200.  -389. -387. -381.  1.44e-6  1.35e-5  8.54e-4    NA
## 3 ADN    3.59e-6   184.  -357. -355. -349.  3.15e-6  1.49e-5  1.57e-3    NA
## 4 LLT    1.30e-3    79.0 -268. -268. -263. NA       NA       NA           0.963
## 5 LLTK   6.30e-4    94.4 -297. -296. -290. NA       NA       NA           0.982
## # ... with 8 more variables: adj_r_squared <dbl>, statistic <dbl>,
## #   p_value <dbl>, df <int>, CV <dbl>, deviance <dbl>, df.residual <int>,
## #   rank <int>
```

- 적합값 저장/잔차 분석

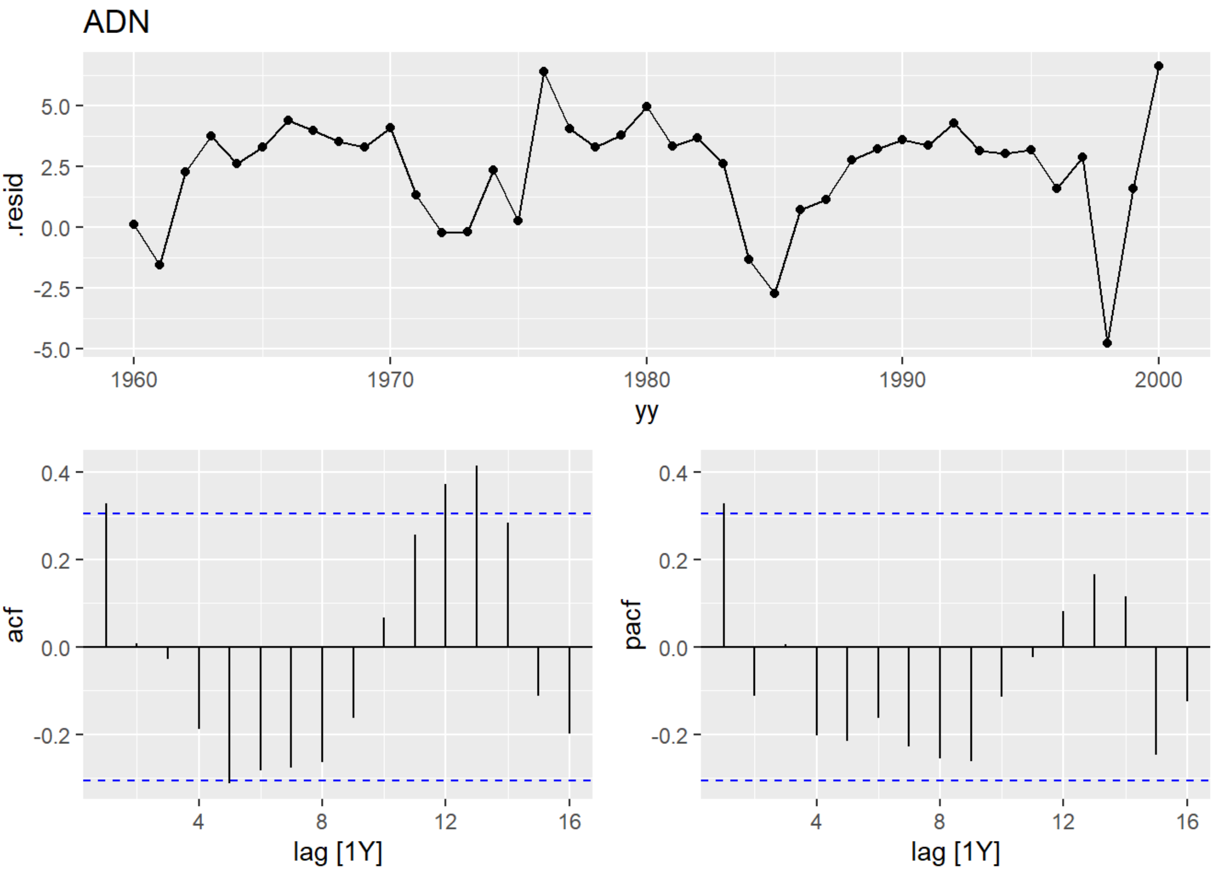
```
AA <- augment(MM)

# ETS(자동선택) 잔차분석
gg_tsdisplay(filter(AA, .model=="ETS"), .resid, plot_type = 'partial') + ggtitle(
  'ETS')
```



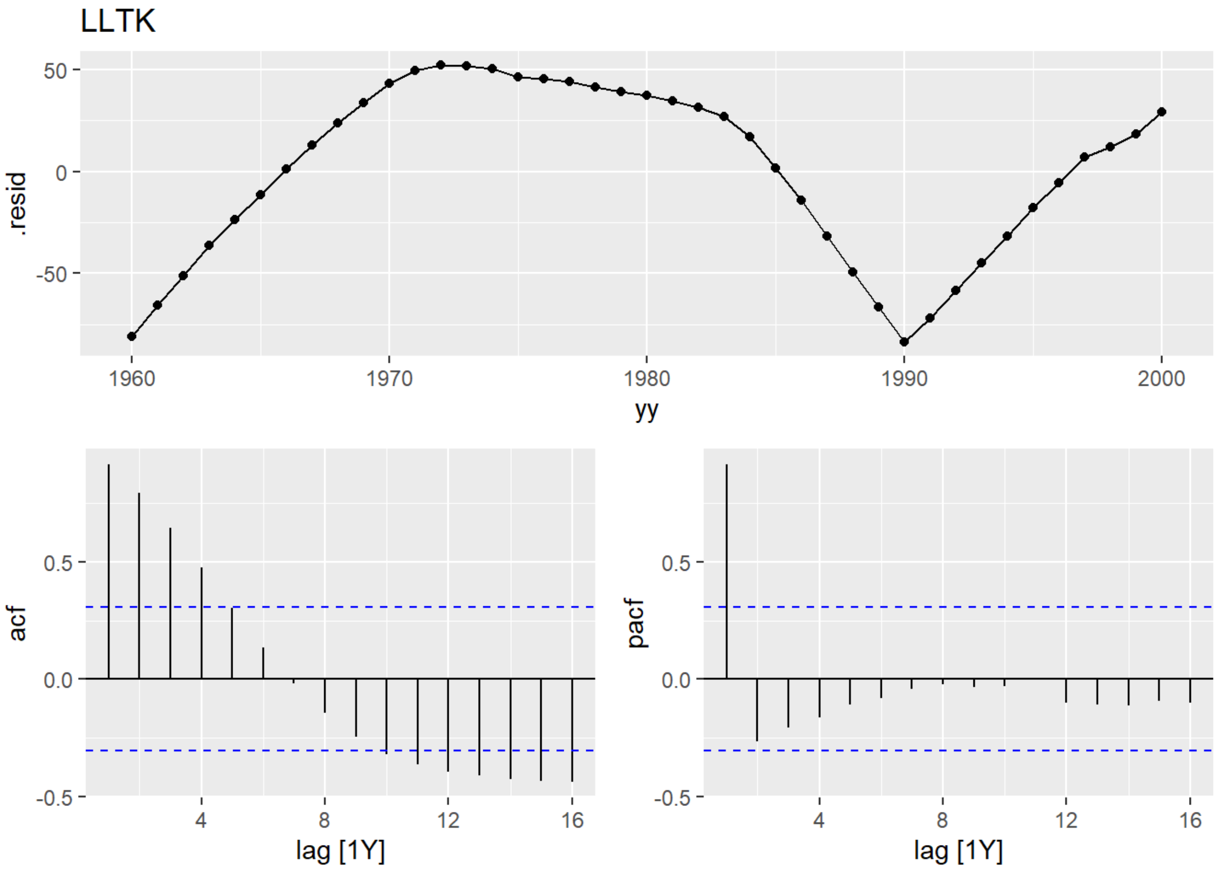
```
# ETS(A,A,N) 잔차분석
gg_tsdisplay(filter(AA, .model=="AAN"), .resid, plot_type = 'partial') + ggtitle(
  'AAN')
```



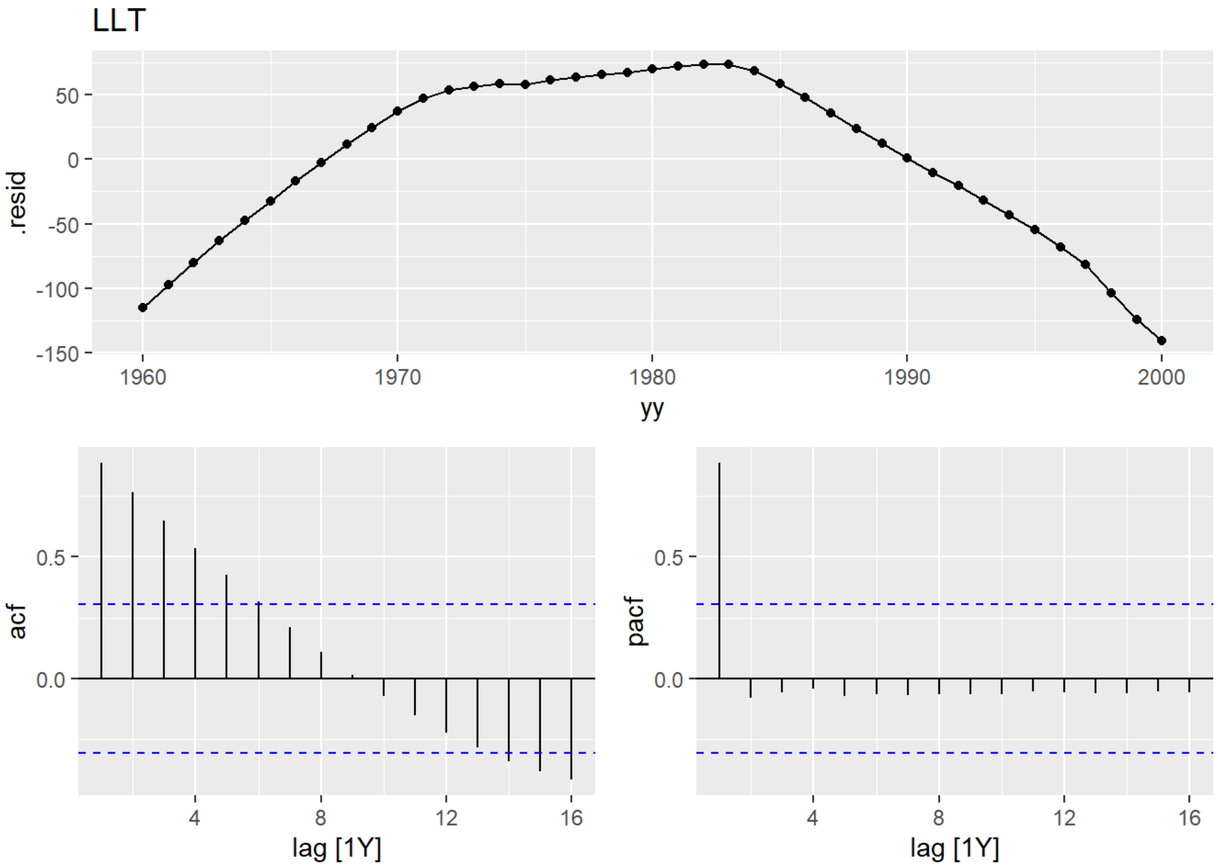
```
## ETS(A,Ad,N) 잔차분석
gg_tsdisplay(filter(AA, .model=="ADN"), .resid, plot_type = 'partial') + ggtitle(
  'ADN')
```



```
# LLTK 잔차분석
gg_tsdisplay(filter(AA, .model=="LLTK"), .resid, plot_type = 'partial') + ggtitle
  ('LLTK')
```



```
# LLT 잔차분석
gg_tsdisplay(filter(AA, .model=="LLT"), .resid, plot_type = 'partial') + ggtitle(
  'LLT')
```



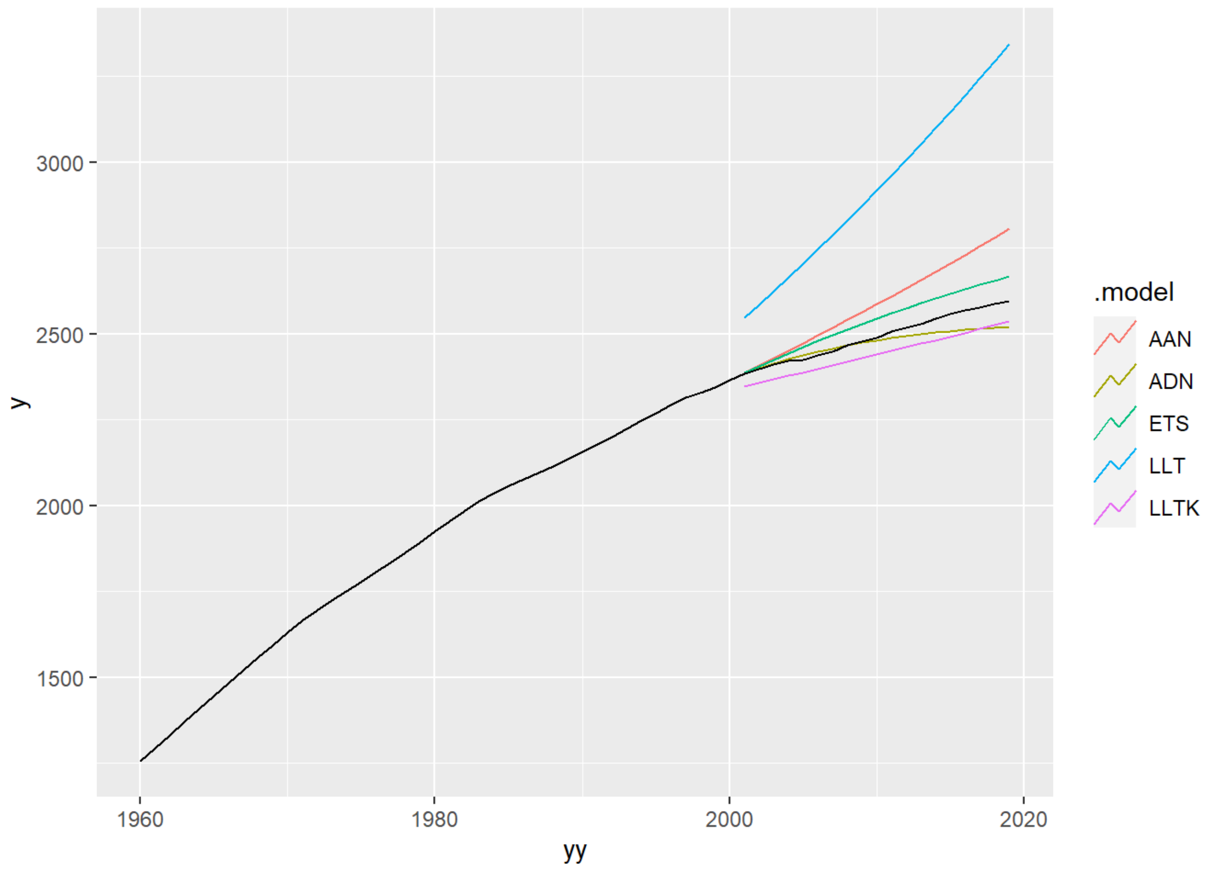
- 예측값 저장(TST)/ 모형평가(반드시 TST로 지정해야함)

```
FF <- forecast(MM, h=19)
# FF <- forecast(MM,new_data=TST)
accuracy(FF, data=TSB)
```

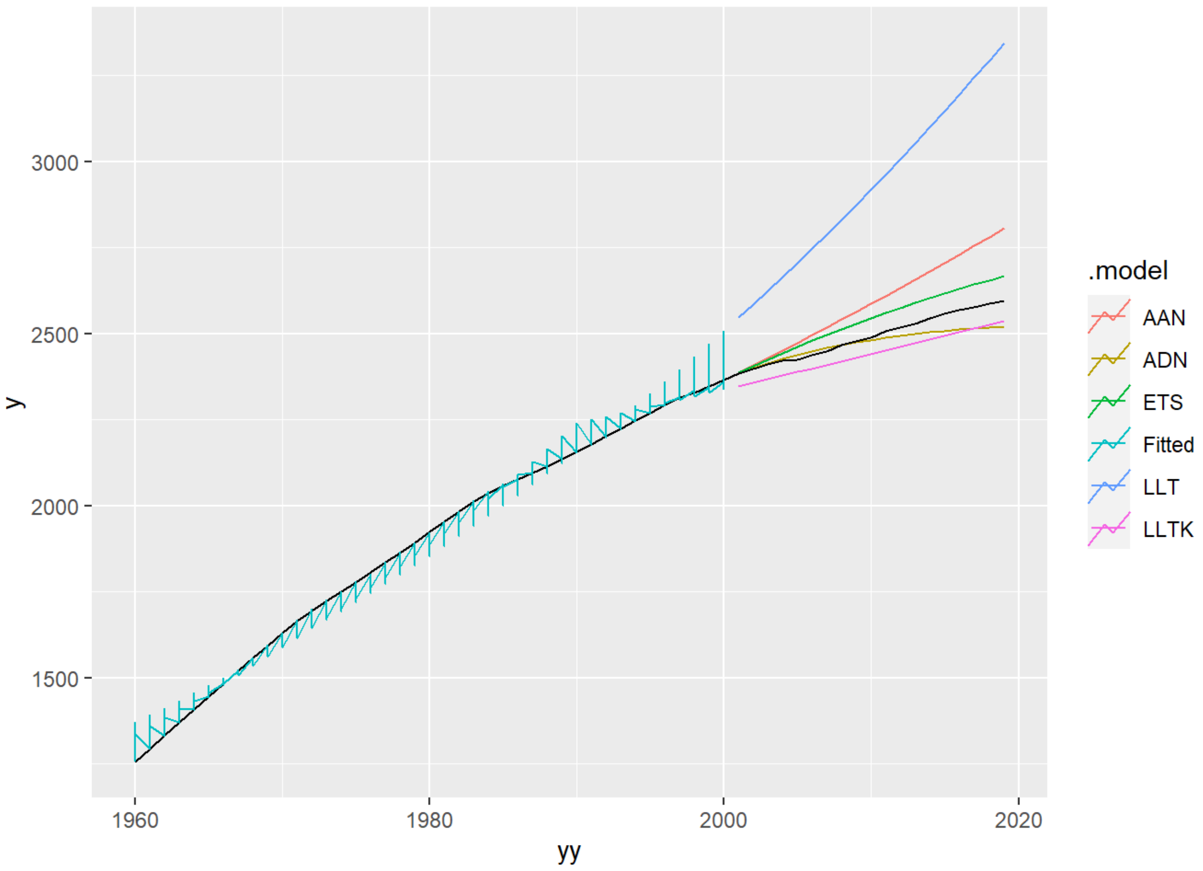
```
## # A tibble: 5 x 9
##   .model .type    ME  RMSE   MAE    MPE  MAPE  MASE  ACF1
##   <chr>  <chr>  <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1 AAN    Test   -98.7 116.   98.7 -3.90  3.90  3.55  0.825
## 2 ADN    Test    20.9  35.4  25.3  0.811 0.990 0.909 0.874
## 3 ETS    Test   -46.7  51.0  46.7 -1.86  1.86  1.68  0.788
## 4 LLT    Test  -437.  471.  437.  -17.4 17.4  15.7   0.836
## 5 LLTK   Test    51.1  52.1  51.1  2.04  2.04  1.84  0.864
```

- 예측값 시각화/ 개별모형 검토

```
# 과거값, 예측값을 한번에 찍는 방법
autoplot(FF, data=TSB, level = NULL)
```



```
# 과거값, 적합값, 예측값을 한번에 찍는 방법(불안정)
autoplot(FF, TSB, level = NULL) +
  geom_line(aes(y=.fitted, color='Fitted'), data=AA)
```



```
# 개별모형 검토
# ETS의 과거값, 적합값, 예측값 시각화
METS <- select(MM, ETS)
report(METS)
```

```
## Series: y
## Model: ETS(M,Ad,N)
## Transformation: log(.x)
##   Smoothing parameters:
##     alpha = 0.9782058
##     beta  = 0.9782051
##     phi   = 0.9642255
##
##   Initial states:
##         l         b
##  7.103666 0.0323802
##
##   sigma^2:  0
##
##       AIC      AICc       BIC
## -401.4823 -399.0117 -391.2009
```

```
# gg_tsresiduals(ETS) : 위 결과와 동일
features(filter(AA, .model=='ETS'), .resid, ljung_box, lag=10, dof=3)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 ETS       8.23     0.313
```

```
G1 <- autoplot(filter(FF, .model=='ETS'),data=TSB)+
  geom_line(aes(y=.fitted, color='Fitted'), data = filter(AA, .model=='ETS')) + gg
  title('ETS')

# AAN의 과거값, 적합값, 예측값 시각화
MAAN <- select(MM, AAN)
report(MAAN)
```

```
## Series: y
## Model: ETS(A,A,N)
## Transformation: log(.x)
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 0.9998999
##
##   Initial states:
##         l          b
##  7.104634 0.03012878
##
##   sigma^2:  0
##
##       AIC      AICc       BIC
## -389.1667 -387.4524 -380.5988
```

```
# gg_tsresiduals(MAAN) : 위 결과와 동일
features(filter(AA, .model=='AAN'), .resid, ljung_box, lag=10, dof=2)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 AAN       5.21     0.734
```

```
G2 <- autoplot(filter(FF, .model=='AAN'),data=TSB)+
  geom_line(aes(y=.fitted, color='Fitted'), data =filter(AA, .model=='AAN')) + ggt
  itle('ETS(AAN)')

# ADN의 과거값, 적합값, 예측값 시각화
MADN <- select(MM, ADN)
report(MADN)
```

```
## Series: y
## Model: ETS(A,Ad,N)
## Transformation: log(.x)
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 0.7074936
##     phi   = 0.9
##
##   Initial states:
##         l          b
##  7.099922 0.03881148
##
##   sigma^2:  0
##
##       AIC      AICc       BIC
## -357.1048 -355.3906 -348.5370
```

```
# gg_tsresiduals(MADN) : 위 결과와 동일
features(filter(AA, .model=='ADN'), .resid, ljung_box, lag=10, dof=3)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 ADN       24.7  0.000848
```

```
G3 <- autoplot(filter(FF, .model=='ADN'), data=TSB)+geom_line(aes(y=.fitted, colo
  r='Fitted'), data=filter(AA, .model=='ADN'))+ggtitle('ETS(ADN)')

# LLT의 과거값, 적합값, 예측값 시각화
MLLT <- select(MM, LLT)
report(MLLT)
```

```
## Series: y
## Model: TSLM
## Transformation: log(.x)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.087672 -0.023721  0.007404  0.033169  0.037686
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.2075077  0.0114866  627.47   <2e-16 ***
## trend()     0.0151107  0.0004765   31.71   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0361 on 39 degrees of freedom
## Multiple R-squared: 0.9627,  Adjusted R-squared: 0.9617
## F-statistic:  1005 on 1 and 39 DF, p-value: < 2.22e-16
```

```
# gg_tsresiduals(MLLT) : 위 결과와 동임
features(filter(AA, .model=='LLT'),  .resid, ljung_box, lag=10, dof=3)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 LLT       111.         0
```

```
G4 <- autoplot(filter(FF, .model=='LLT'), data=TSB)+ geom_line(aes(y=.fitted ,col
  or='Fitted'), data=filter(AA, .model=='LLT'))+ggtitle('LLT:Linear Trend')

# LLTK의 과거값, 적합값, 예측값 시각화
MLLTK <-select(MM, LLTK)
report(MLLTK)
```

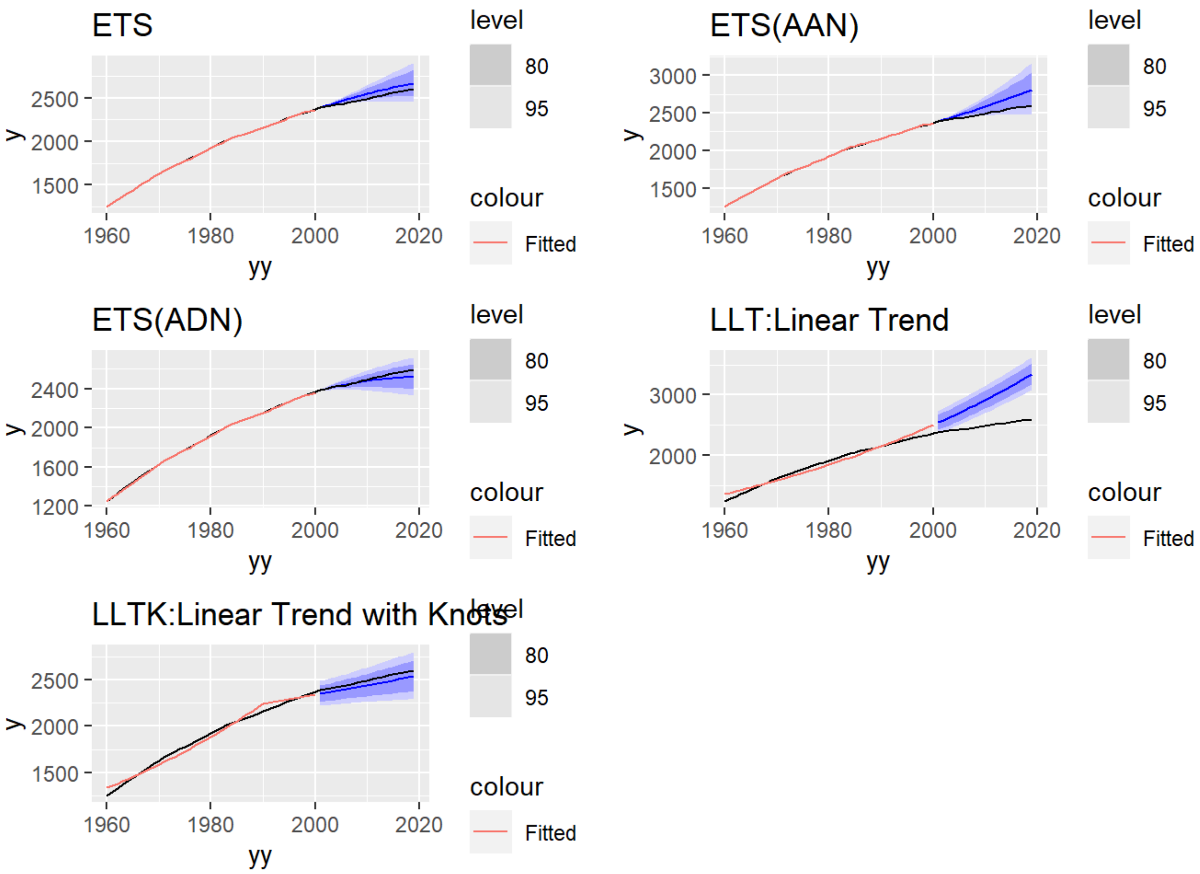
```
## Series: y
## Model: TSLM
## Transformation: log(.x)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.062749 -0.016547  0.005082  0.020836  0.031314
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   7.1804656  0.0089982  797.99  < 2e-16 ***
## trend(knots = c(1990))trend     0.0172292  0.0004638   37.15  < 2e-16 ***
## trend(knots = c(1990))trend_31 -0.0130056  0.0019918   -6.53 1.07e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02511 on 38 degrees of freedom
## Multiple R-squared: 0.9824,  Adjusted R-squared: 0.9815
## F-statistic:  1061 on 2 and 38 DF, p-value: < 2.22e-16
```

```
# gg_tsresiduals(MLLTK) : 위 결과와 동일
features(filter(AA, .model=='LLTK'), .resid, ljung_box, lag=10, dof=3)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 LLTK      111.         0
```

```
G5 <- autoplot(filter(FF, .model=='LLTK'), data=TSB)+geom_line(aes(y=.fitted,colo
  r='Fitted'), data=filter(AA, .model=='LLTK'))+ ggtitle('LLTK:Linear Trend with Kn
  ots')
```

```
gridExtra::grid.arrange(G1,G2,G3,G4,G5, nrow=3)
```



```
# 예측값 확인
cbind(
  tail(TSB)[,c('yy','y')],
  ADN = tail(filter(FF,.model=='ADN')$.mean),
  ETS = tail(filter(FF,.model=='ETS')$.mean),
  LLTK= tail(filter(FF,.model=='LLTK')$.mean),
  AAN = tail(filter(FF,.model=='AAN')$.mean),
  LLT = tail(filter(FF,.model=='LLT')$.mean))
```

```
##     yy        y      ADN      ETS     LLTK      AAN      LLT
## 1 2014 2544.508 2503.705 2603.809 2482.205 2681.663 3100.728
## 2 2015 2558.589 2508.014 2617.376 2492.865 2706.033 3147.962
## 3 2016 2569.699 2511.922 2630.554 2503.579 2730.693 3195.917
## 4 2017 2576.805 2515.468 2643.354 2514.346 2755.649 3244.603
## 5 2018 2587.720 2518.688 2655.788 2525.166 2780.911 3294.031
## 6 2019 2595.207 2521.614 2667.866 2536.040 2806.487 3344.213
```