Forecasting on Tourism data using Exponential Smoothing

Problem Background: You have been hired by a company in the hospitality business to help them plan the staffing levels for the following year. The company operates resorts in three regions of the New South Wales of Australia; the three regions are the **Sydney**, the **South Coast** and the **North Coast NSW** areas.

As it takes time to hire new personnel and it is necessary for any new employee to undergo a detailed training program before starting to work, the company needs to plan its personnel requirements one year in advance. Furthermore, as it is possible for the company to transfer qualified personnel between regions, they are interested only in an aggregate forecast of their demand

As the company caters to **Holiday** travelers, and it has been growing faster than the market (i.e., it has been gaining market share), the Chief Commercial Officer estimates that next year they will have respectively (3%, 4%, 4%) of only the **Holiday** travelers in the (**Sydney**, **South Coast**, and **North Coast NSW**) regions respectively. Furthermore based on prior experience they anticipate that each traveler will stay respectively (5,2,2) hotel-nights in (**Sydney**, **South Coast**, and **North Coast NSW**) respectively

To forecast demand in hotel-nights, we use the **tourism** data set in **fpp3**. This data set reports the quarterly trips (in thousands) to different destinations, and as this data set has a *tsibble* structure, we use **tidyverse** functions to subset the time-series of interest.

For the purpose of this project, we are ignoring all data before **2008 Q1** and use the data from **2008 Q1** through **2016 Q4** as a training set and the four quarters of **2017** as a testing set.

Part I. Model-Aggregation Forecast

1. After sub-setting for the time-series of interest in the **tourism** data set (a *tsibble*), we add to the restricted set the corresponding demand time-series, by creating a column called *Demand* for each of the corresponding regions of interest. The *Demand* column contains the hotel-nights (in thousands) corresponding to each of the *Trips* observations. After creating the *Demand* column, we first fit automatically the best **ETS** model for each *Demand* time-series. In addition to the automatic fit, we try the "AAM" model and the "AAdM" models as they may be preferred under the *BIC* criterion.

```
library(fpp3)
```

```
## Warning: package 'fpp3' was built under R version 4.1.1
```

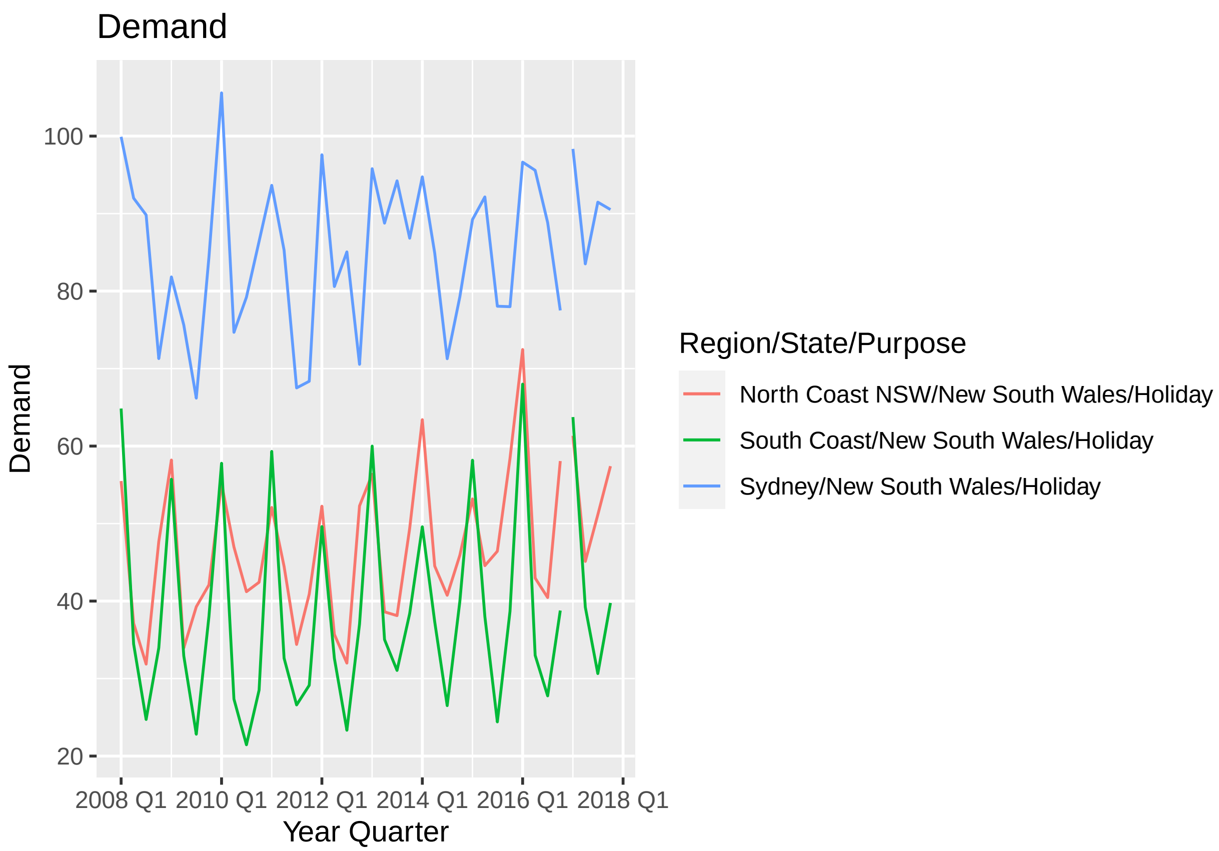
```
## -- Attaching packages -------------------------------------------- fpp3 0.4.0 --
```

```
## v tibble      3.1.2      v tsibble     1.0.1
## v dplyr       1.0.7      v tsibbledata 0.3.0
## v tidyr       1.1.3      v feasts      0.2.2
## v lubridate   1.7.10     v fable       0.3.1
## v ggplot2     3.3.5
```

```
## Warning: package 'tsibble' was built under R version 4.1.1
```

```
## Warning: package 'tsibbledata' was built under R version 4.1.1
## Warning: package 'feasts' was built under R version 4.1.1

## Warning: package 'fabletools' was built under R version 4.1.1

## Warning: package 'fable' was built under R version 4.1.1

## -- Conflicts ------------------------------------------------- fpp3_conflicts --
## x lubridate::date()    masks base::date()
## x dplyr::filter()      masks stats::filter()
## x tsibble::intersect() masks base::intersect()
## x tsibble::interval()  masks lubridate::interval()
## x dplyr::lag()         masks stats::lag()
## x tsibble::setdiff()   masks base::setdiff()
## x tsibble::union()     masks base::union()
```

```r
# Subset the appropriate data and create the "Demand" time-series
tourism %>%
  filter(Quarter >= yearquarter("2008 Q1")) %>%
  filter(Purpose == "Holiday" & State == "New South Wales") %>%
  filter(Region %in% c("North Coast NSW","South Coast","Sydney")) %>%
  mutate(Demand = case_when(
    Region == "Sydney" ~ 0.03*Trips*5,
    Region == "South Coast" ~ 0.04*Trips*2,
    Region == "North Coast NSW" ~ 0.04*Trips*2
  )) -> D

# D <- subset(D, select = -c(State,Purpose) )

# Break into Training and Testing sets.

DTR <- D %>%
  filter(Quarter <= yearquarter("2016 Q4"))
DTE <- D %>%
  filter(Quarter >= yearquarter("2017 Q1"))

autoplot(DTR,Demand) +
  autolayer(DTE, Demand) +
  labs(title = "Demand",
       x = "Year Quarter")
```

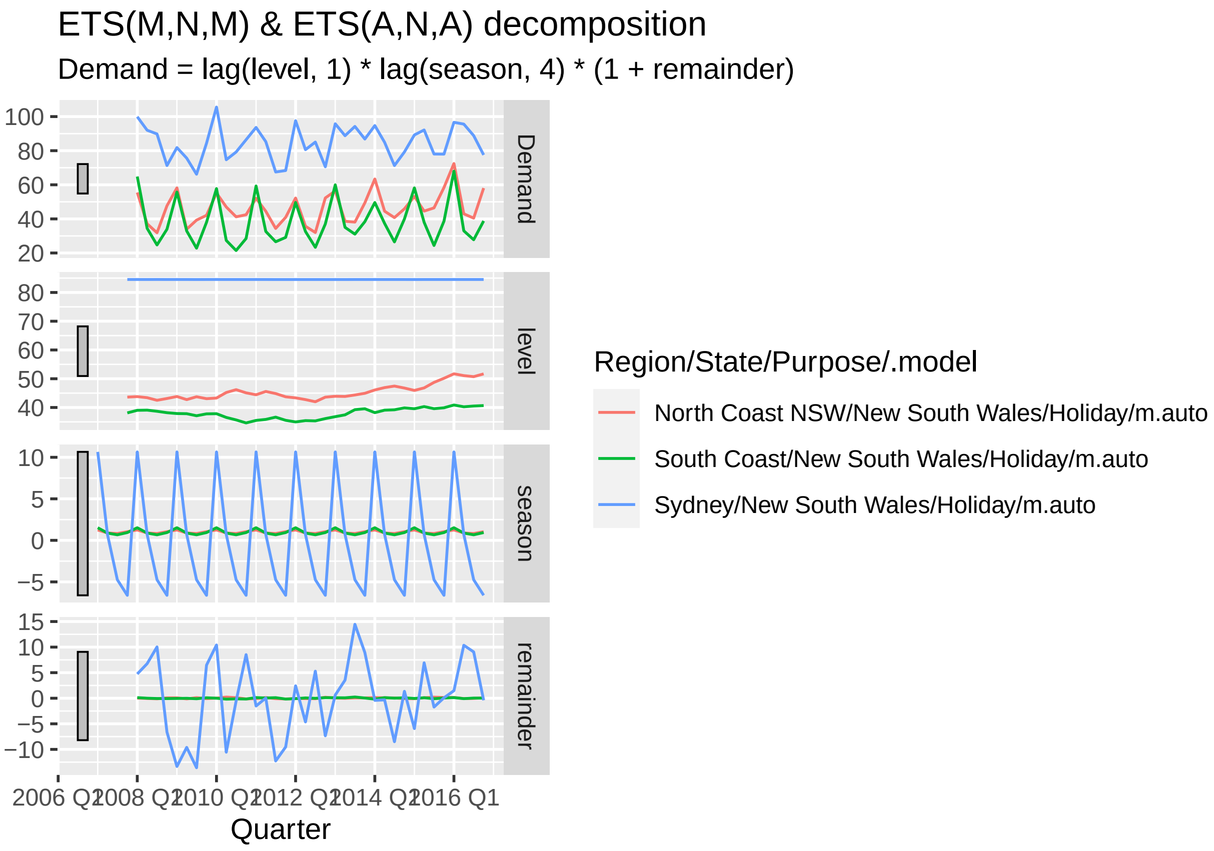


```
## Fitting model automatically on Demand

m <- DTR %>%
  model(m.auto = ETS(Demand))

m %>%
  tidy()
```

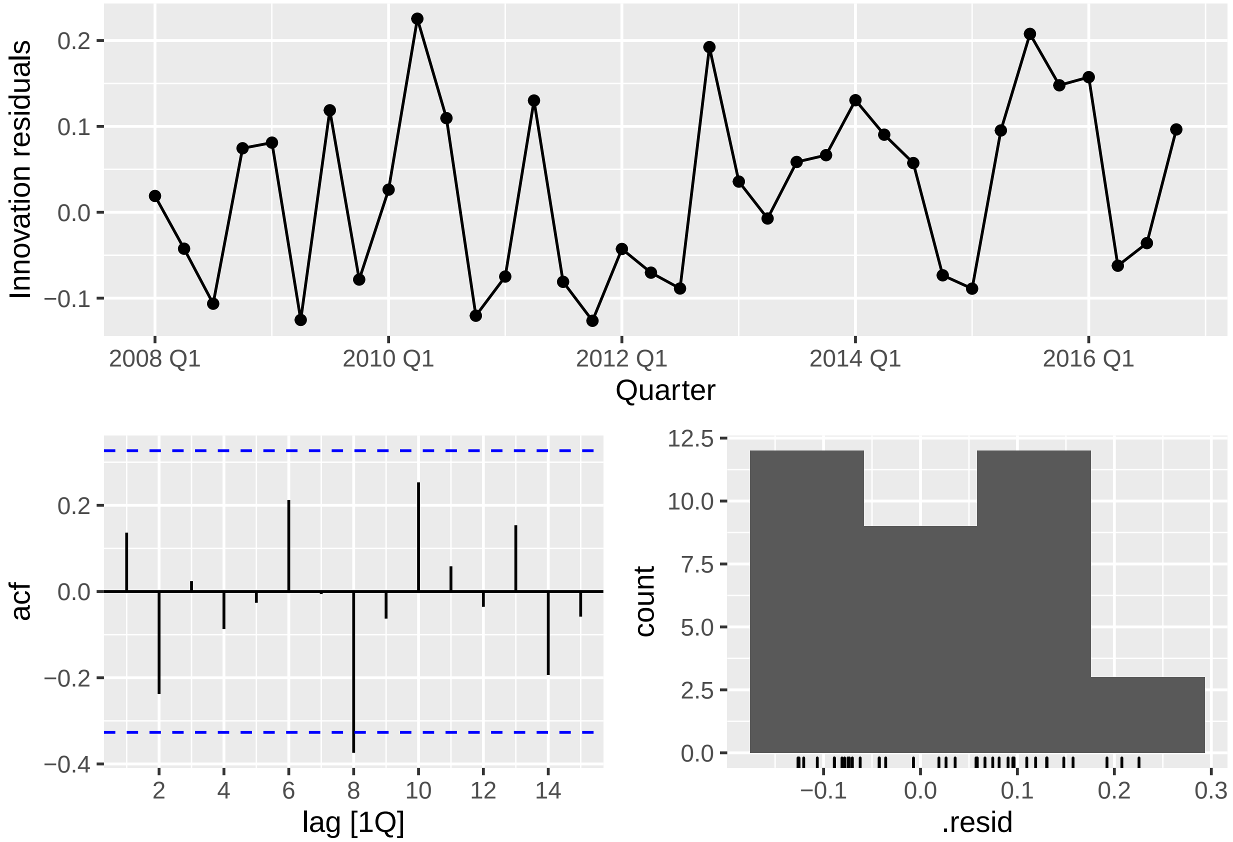
```
## # A tibble: 21 x 6
##    Region          State           Purpose .model term   estimate
##    <chr>           <chr>           <chr>   <chr>  <chr>     <dbl>
##  1 North Coast NSW New South Wales Holiday m.auto alpha  0.198
##  2 North Coast NSW New South Wales Holiday m.auto gamma  0.000100
##  3 North Coast NSW New South Wales Holiday m.auto l[0]  43.6
##  4 North Coast NSW New South Wales Holiday m.auto s[0]   1.04
##  5 North Coast NSW New South Wales Holiday m.auto s[-1]  0.821
##  6 North Coast NSW New South Wales Holiday m.auto s[-2]  0.886
##  7 North Coast NSW New South Wales Holiday m.auto s[-3]  1.25
##  8 South Coast     New South Wales Holiday m.auto alpha  0.197
##  9 South Coast     New South Wales Holiday m.auto gamma  0.000100
## 10 South Coast     New South Wales Holiday m.auto l[0]  38.1
## # ... with 11 more rows
```

```
m %>%
  components() %>%
  autoplot()
```

```
## Warning: Removed 12 row(s) containing missing values (geom_path).
```



```
## Sample Residual Plot for North Coast NSW
m %>% filter(Region == "North Coast NSW") %>% gg_tsresiduals()
```



```
m %>% glance()
```

```
## # A tibble: 3 x 12
##   Region     State    Purpose .model   sigma2 log_lik   AIC  AICc   BIC   MSE  AMSE
##   <chr>      <chr>    <chr>   <chr>     <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 North Co~ New So~ Holiday m.auto  0.0134    -120.  254.  258.  266.  22.5  25.5
## 2 South Co~ New So~ Holiday m.auto  0.0120    -111.  235.  239.  247.  15.6  15.0
## 3 Sydney    New So~ Holiday m.auto 67.0       -137.  288.  292.  299.  55.8  56.6
## # ... with 1 more variable: MAE <dbl>
```

```
## Fitting AAM and AAdM models as suggested by the colleague
m <- DTR %>%
  model(m.auto = ETS(Demand),
    m.AAM = ETS(Demand ~ error("A") + trend("A") + season("M")),
    m.AAdM = ETS(Demand ~ error("A") + trend("Ad") + season("M")))

m %>%
  glance()
```

```
## # A tibble: 9 x 12
##   Region     State    Purpose .model   sigma2 log_lik   AIC  AICc   BIC   MSE  AMSE
##   <chr>      <chr>    <chr>   <chr>     <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 North Co~ New So~ Holiday m.auto  0.0134    -120.  254.  258.  266.  22.5  25.5
## 2 North Co~ New So~ Holiday m.AAM  27.2       -119.  257.  264.  271.  21.1  23.4
## 3 North Co~ New So~ Holiday m.AAdM 28.8       -120.  260.  268.  276.  21.6  24.0
## 4 South Co~ New So~ Holiday m.auto  0.0120   -111.  235.  239.  247.  15.6  15.0
## 5 South Co~ New So~ Holiday m.AAM  19.6      -113.  245.  252.  259.  15.2  15.9
## 6 South Co~ New So~ Holiday m.AAdM 20.1      -113.  247.  256.  263.  15.1  15.7
## 7 Sydney    New So~ Holiday m.auto 67.0      -137.  288.  292.  299.  55.8  56.6
## 8 Sydney    New So~ Holiday m.AAM  80.1      -139.  296.  303.  310.  62.3  65.6
## 9 Sydney    New So~ Holiday m.AAdM 82.8      -139.  298.  306.  313.  62.1  65.9
## # ... with 1 more variable: MAE <dbl>
```

```
m %>%
  accuracy()
```

```
## # A tibble: 9 x 13
##   Region    State   Purpose .model .type      ME  RMSE   MAE     MPE  MAPE  MASE
##   <chr>     <chr>   <chr>   <chr>  <chr>   <dbl> <dbl> <dbl>   <dbl> <dbl> <dbl>
## 1 North Co~ New So~ Holiday m.auto Trai~  1.13    4.74  4.16  1.45    8.94 0.808
## 2 North Co~ New So~ Holiday m.AAM  Trai~  0.0376  4.60  4.07 -0.841   8.90 0.791
## 3 North Co~ New So~ Holiday m.AAdM Trai~  0.271   4.65  4.12 -0.381   8.95 0.800
## 4 South Co~ New So~ Holiday m.auto Trai~  0.398   3.95  3.19  0.0102  8.45 0.729
## 5 South Co~ New So~ Holiday m.AAM  Trai~ -0.0293  3.90  3.27 -1.03    8.90 0.748
## 6 South Co~ New So~ Holiday m.AAdM Trai~  0.0969  3.89  3.25 -0.744   8.85 0.743
## 7 Sydney    New So~ Holiday m.auto Trai~  0.131   7.47  6.07 -0.668   7.43 0.636
## 8 Sydney    New So~ Holiday m.AAM  Trai~ -0.749   7.89  6.35 -1.76    7.87 0.665
## 9 Sydney    New So~ Holiday m.AAdM Trai~ -0.686   7.88  6.41 -1.69    7.92 0.671
## # ... with 2 more variables: RMSSE <dbl>, ACF1 <dbl>
```

Inference:

From the results above, considering the metrics AICc and BIC, the models fitted **automatically** on all regions are the best. Summarizing the AICc and BIC for the models below:

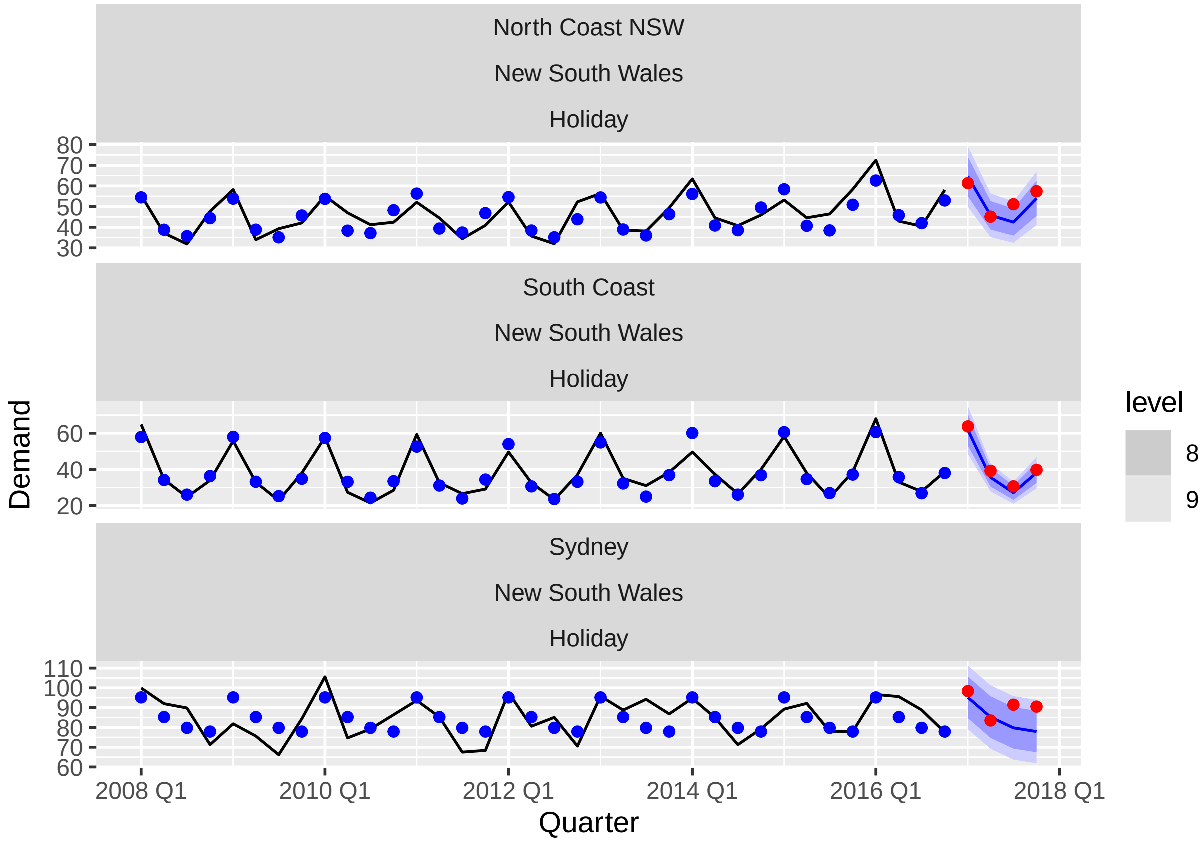
AICc: North Coast NSW : 254.4963 South Coast : 235.4490 Sydney : 287.8126

BIC: North Coast NSW : 265.5809 South Coast : 246.5336 Sydney : 298.8972

The models that have been fitted automatically can be obtained by looking at **m**, and they are as follows: North Coast NSW : <ETS(M,N,M)> South Coast : <ETS(M,N,M)> Sydney : <ETS(A,N,A)>

2. Using the best model selected in (1), we prepare a forecast for the four quarters of 2017 and report for each time series the in-sample (training) MAPE, and out-of-sample (testing) MAPE.

```
## Running the best model as identified by Part 1

m <- DTR %>%
  model(m.auto = ETS(Demand))

mg <- m %>% augment()

## Preparing a forecast for the test dataset
f <- m %>%
  forecast(h = 4)

f %>% filter(.model == "m.auto") %>% autoplot(DTR) +
  geom_point(data = mg, mapping = aes(y = .fitted), col = "blue") +
  geom_point(data = DTE, mapping = aes(y = Demand), col = "red")
```



```
# Examining In-Sample and Out-of-Sample Accuracy Statistics

rbind(m %>% accuracy(),
        f %>% accuracy(data = DTE)) %>% select(Region, .model, .type, MAPE)
```

```
## # A tibble: 6 x 4
##   Region          .model .type     MAPE
##   <chr>           <chr>  <chr>    <dbl>
## 1 North Coast NSW m.auto Training  8.94
## 2 South Coast     m.auto Training  8.45
## 3 Sydney          m.auto Training  7.43
## 4 North Coast NSW m.auto Test      7.42
## 5 South Coast     m.auto Test      6.94
## 6 Sydney          m.auto Test      8.01
```

3. Next, we add the three forecasts of each region for the selected model to obtain the total forecast and compute the fitted (training) MAPE and the testing MAPE.

```
## Computing fitted Train MAPE
mg_agg <- mg %>% summarize(Demand = sum(.fitted))
train_agg <- DTR %>% summarize(Demand = sum(Demand))
train_err <- train_agg %>% left_join(mg_agg, by="Quarter")
cat('In Sample MAPE:',mean(abs((train_err$Demand.x-train_err$Demand.y)/train_err$Demand.x)) * 100)
```

```
## In Sample MAPE: 4.969444
## Computing fitted Test MAPE
test = f$.mean
f_agg <- f %>% group_by(f$Quarter) %>% summarize(sum(.mean)) %>% rename(Demand = `sum(.mean)`)
test_agg <- DTE %>% summarize(Demand = sum(Demand))
test_err <- test_agg %>% left_join(f_agg, by="Quarter")
cat('\nOut Sample MAPE:',mean(abs((test_err$Demand.x-test_err$Demand.y)/test_err$Demand.x)) * 100)
```

```
##
## Out Sample MAPE: 6.198686
```

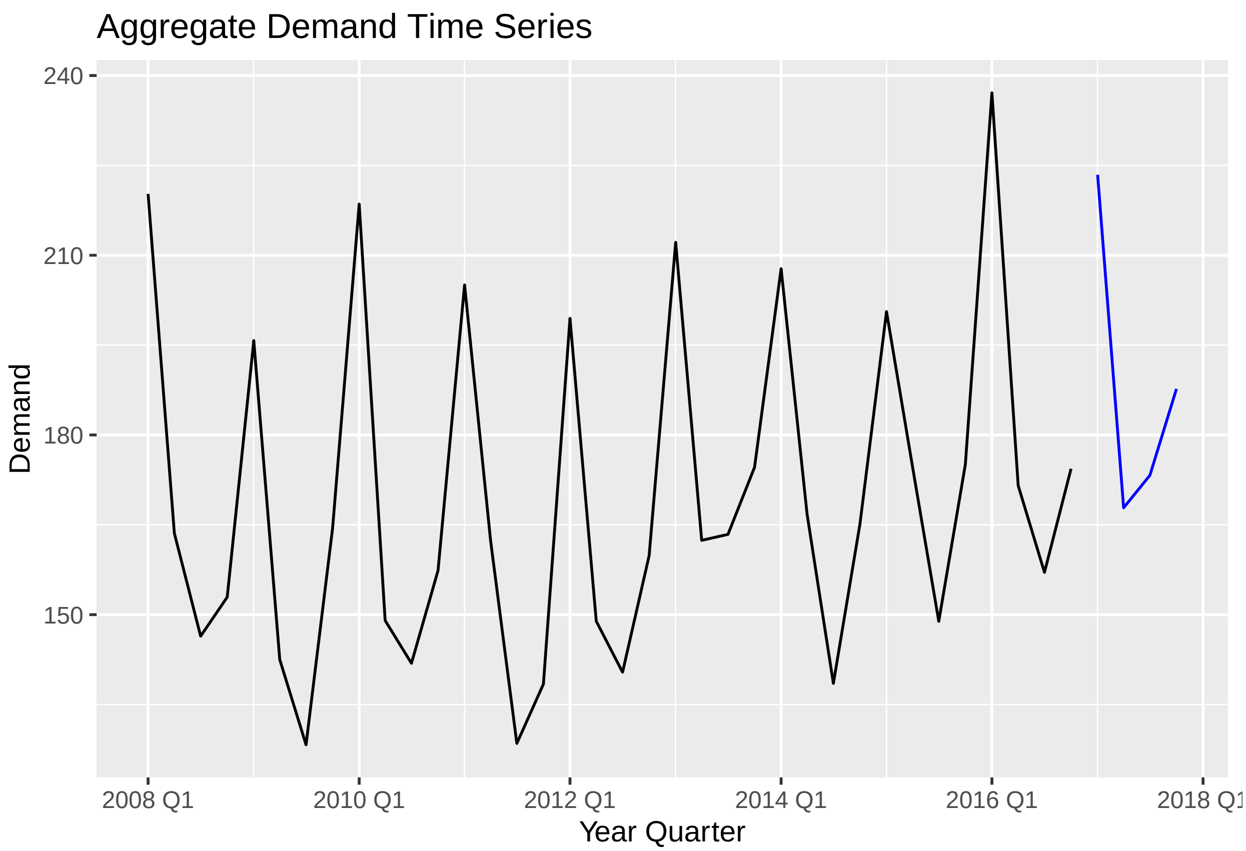
Inference:

From the above results, we see that the aggregated forecasts have a much lower MAPE than the regional forecasts. This is because aggregated forecasts tend to be more accurate than regional forecasts, as they have a lower standard deviation of error with respect to the mean when compared to regional forecasts.

Part II. Data-Aggregation Forecast

4. Now, we aggregate the region-specific demand data to compile an aggregate demand time series, the aggregated demand into training and testing time-series, and fit the automatic model, plus the two models we fitted in (1)

```
new_D <- D %>% summarize(Demand = sum(Demand))

DTR <- new_D %>% filter(Quarter <= yearquarter("2016 Q4"))
DTE <- new_D %>% filter(Quarter >= yearquarter("2017 Q1"))

autoplot(DTR,Demand) +
  autolayer(DTE, Demand, col = 'blue') +
  labs(title = "Aggregate Demand Time Series",
  x = "Year Quarter")
```



```
m <- DTR %>%
    model(m.auto = ETS(Demand),
    m.AAM = ETS(Demand ~ error("A") + trend("A") + season("M")),
    m.AAdM = ETS(Demand ~ error("A") + trend("Ad") + season("M")))


m %>%
  glance() %>% select(.model, AICc, BIC)
```

```
## # A tibble: 3 x 3
##   .model  AICc   BIC
##   <chr>  <dbl> <dbl>
## 1 m.auto  311.  318.
## 2 m.AAM   320.  327.
## 3 m.AAdM  322.  329.
```

```
m %>% accuracy() %>% select(.model, .type, MAPE)
```

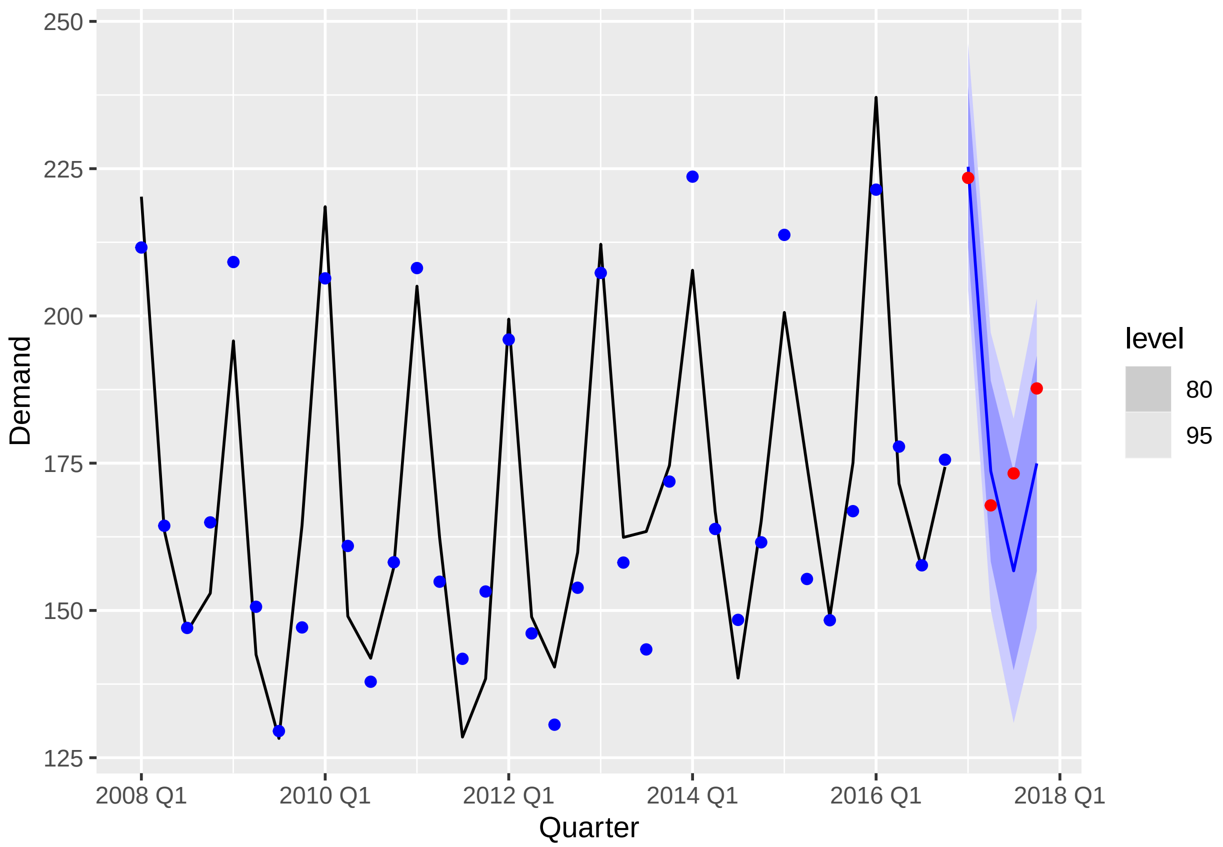
```
## # A tibble: 3 x 3
##   .model .type     MAPE
##   <chr>  <chr>    <dbl>
## 1 m.auto Training  4.63
## 2 m.AAM  Training  4.89
## 3 m.AAdM Training  4.60
```

Inference:

Looking at the AICc and BIC scores, we can see that again, the model fit using the automatic method is the best.

5. Using the best model selected in (4), we prepare a forecast for the four quarters of 2017 and report the in-sample (training) MAPE, and out-of-sample (testing) MAPE.

```
## Running the best model as identified by Part 1

m <- train_agg %>%
  model(m.auto = ETS(Demand))

mg <- m %>% augment()

## Preparing a forecast for the test dataset
f <- m %>%
  forecast(h = 4)

f %>% filter(.model == "m.auto") %>% autoplot(train_agg) +
  geom_point(data = mg, mapping = aes(y = .fitted), col = "blue") +
  geom_point(data = test_agg, mapping = aes(y = Demand), col = "red")
```



```
rbind(m %>% accuracy(), f %>% accuracy(data = test_agg)) %>% select(.model, .type, MAPE)
```

```
## # A tibble: 2 x 3
##   .model .type     MAPE
##   <chr>  <chr>    <dbl>
## 1 m.auto Training  4.63
## 2 m.auto Test      5.16
```

Part III. Forecasting Model Analysis and Aggregate Forecast

6. Using the best modeling approach (model-aggregation vs data-aggregation) and the best ETS model(s) selected, and using all the data available fit the model(s), we report the model parameters, the in-sample MAPE, and plot the forecast for the four quarters of 2018.

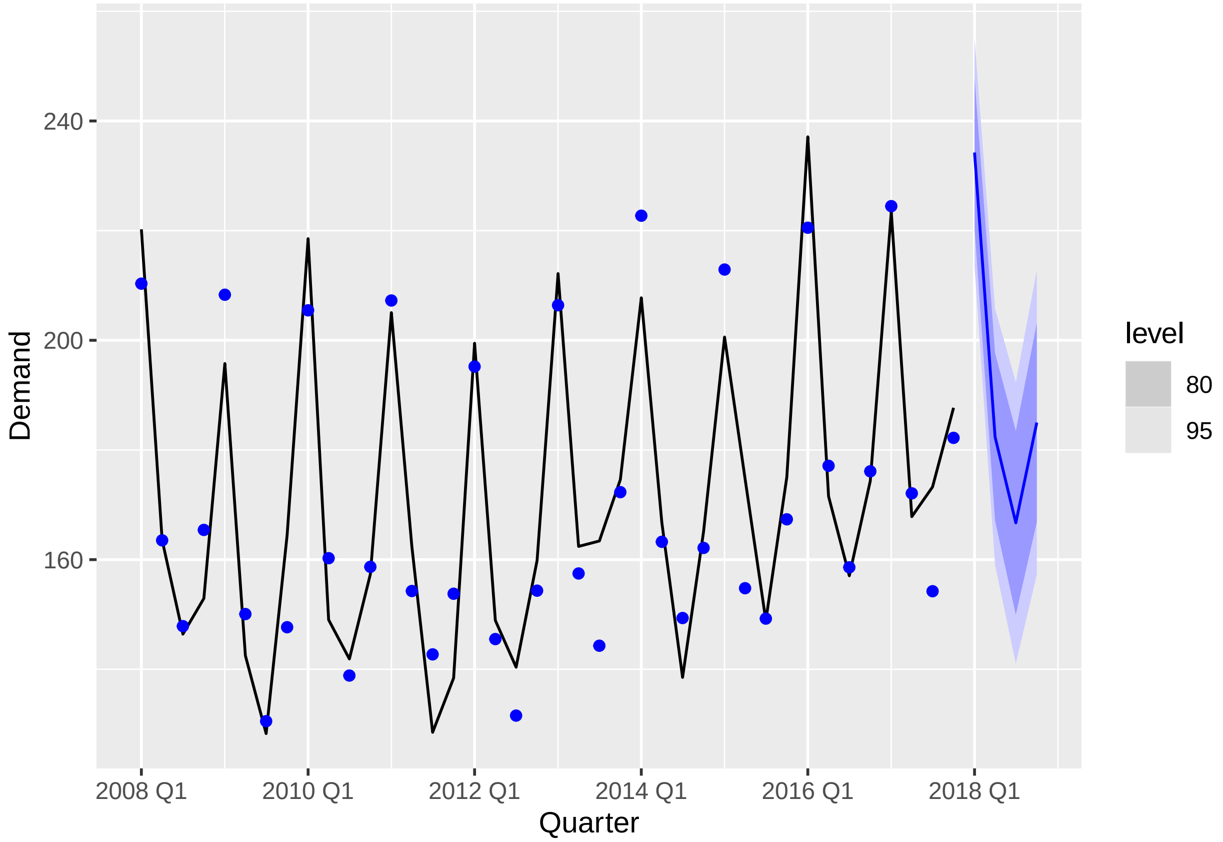
```
new_D <- D %>% summarize(Demand = sum(Demand))

m <- new_D %>%
  model(m.auto = ETS(Demand))

m %>% tidy()
```

```
## # A tibble: 7 x 3
##   .model term    estimate
##   <chr>  <chr>      <dbl>
## 1 m.auto alpha   0.509
## 2 m.auto gamma   0.000100
## 3 m.auto l[0]  168.
## 4 m.auto s[0]    -7.09
## 5 m.auto s[-1] -25.4
## 6 m.auto s[-2]  -9.69
## 7 m.auto s[-3]  42.2
```

```
m %>% accuracy() %>% select(.model, .type, MAPE)
```

```
## # A tibble: 1 x 3
##   .model .type     MAPE
##   <chr>  <chr>    <dbl>
## 1 m.auto Training  4.63
```

```
f <- m %>%
  forecast(h = 4)

mg <- m %>%
        augment()

mgH <- mg %>%
        filter(.model == "m.auto")

f %>%
    filter(.model == "m.auto") %>% autoplot(new_D) +
    geom_point(data = mgH, mapping = aes(y = .fitted), col = "blue")
```



7. As it is very costly to be short of personnel, we need to plan the staffing levels according to a forecast that we anticipate it will not be exceeded with a probability of 99%. Below are the quarterly demand levels:

```
f %>%
  filter(.model == "m.auto") %>%
  hilo(level =c(99)) %>%
  unpack_hilo("99%") %>%
  select(Quarter,"99%_lower","99%_upper")
```

```
## # A tsibble: 4 x 3 [1Q]
##   Quarter ‘99%_lower‘ ‘99%_upper‘
##     <qtr>       <dbl>       <dbl>
## 1 2018 Q1        207.        262.
## 2 2018 Q2        152.        213.
## 3 2018 Q3        133.        200.
## 4 2018 Q4        149.        221.
```

8. Sometimes not all the data available is representative of the recent and future business conditions. We redefine the training data set ***DTR*** to exclude all data older than 2010 and reevaluate the recommendation in (6) and (7).

```
DTR <- D %>%
  filter(Quarter >= yearquarter("2010 Q1"),
         Quarter <= yearquarter("2016 Q4"))
DTE <- D %>% filter(Quarter >= yearquarter("2017 Q1"))

train_agg <- DTR %>% summarize(Demand = sum(Demand))
test_agg <- DTE %>% summarize(Demand = sum(Demand))

m <- train_agg %>%
  model(m.auto = ETS(Demand))

m %>% tidy()
```

```
## # A tibble: 7 x 3
##   .model term    estimate
##   <chr>  <chr>      <dbl>
## 1 m.auto alpha   0.454
## 2 m.auto gamma   0.000100
## 3 m.auto l[0]  169.
## 4 m.auto s[0]   -7.86
## 5 m.auto s[-1] -25.4
## 6 m.auto s[-2]  -8.28
## 7 m.auto s[-3]  41.5
```

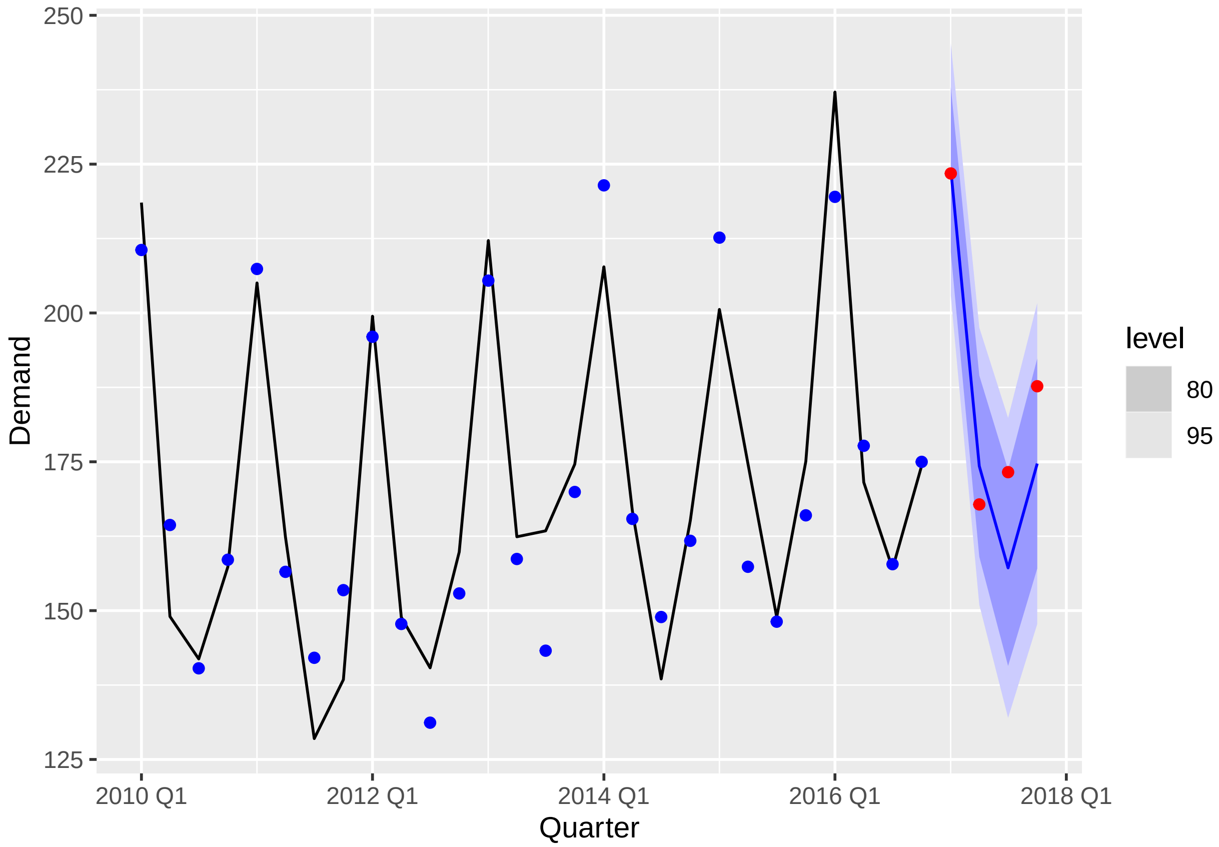
```
m %>% accuracy() %>% select(.model, .type, MAPE)
```

```
## # A tibble: 1 x 3
##   .model .type     MAPE
##   <chr>  <chr>    <dbl>
## 1 m.auto Training  4.51
```

```
f <- m %>%
  forecast(h = 4)

mg <- m %>%
        augment()

mgH <- mg %>%
        filter(.model == "m.auto")

rbind(m %>% accuracy(), f %>% accuracy(data = test_agg))
```

```
## # A tibble: 2 x 10
##   .model .type       ME  RMSE   MAE   MPE  MAPE    MASE   RMSSE    ACF1
##   <chr>  <chr>    <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>   <dbl>   <dbl>
## 1 m.auto Training  1.06  9.59  7.58 0.356  4.51   0.601   0.646 -0.0384
## 2 m.auto Test      5.49 10.8   9.04 3.02   5.08 NaN     NaN       0.0756
```

```
f %>%
    filter(.model == "m.auto") %>% autoplot(train_agg) +
    geom_point(data = mgH, mapping = aes(y = .fitted), col = "blue") +
    geom_point(data = test_agg, mapping = aes(y = Demand), col = "red")
```



```
f %>%
hilo(level =c(99,100)) %>%
unpack_hilo("99%") %>%
select(Quarter,"99%_lower", "99%_upper")
```

```
## # A tsibble: 4 x 3 [1Q]
##   Quarter ‘99%_lower‘ ‘99%_upper‘
##     <qtr>       <dbl>       <dbl>
## 1 2017 Q1        196.        252.
## 2 2017 Q2        144.        205.
## 3 2017 Q3        124.        190.
## 4 2017 Q4        139.        210.
```