Project: The Forecasting Tourism 2010 Competition

EM1415

Marco Solari, 875475

Table of contents

1 Setup and Data Loading

1.1 Setup

```
knitr::opts_chunk$set(
  echo = T,
  dev = "cairo_pdf"
)

libraries_list <- c(
  "tidyverse",
  "fpp3",
  "ggthemes"

)

lapply(
  X = libraries_list,
  FUN = require,
  character.only = TRUE
)
```

```
[[1]]
[1] TRUE

[[2]]
[1] TRUE

[[3]]
[1] TRUE
```

```
theme_set(
  ggthemes::theme_tufte(
    base_size = 16,
    base_family = "Atkinson Hyperlegible"
  )
)
```

1.2 Loading Data

```
data_main <- readr::read_csv(
  "Data/tourism_data.csv",
  show_col_types = F
)
```

```
data_main %>% dim
```

```
[1]  43 518
```

```
data_main %>% is.na() %>% sum
```

```
[1] 11668
```

We are missing 52.38% of the observations.

1.3 Creating **tsibble**

```
tourism_full ← data_main %>%
  mutate(
    Year = 1965:2007
  ) %>%
  as_tsibble(
    index = Year
  )
```

2 Assignment

2.1 Full Plot

> *Plot all the series (an advanced data visualization tool is recommended) - what type of components are visible? Are the series similar or different? Check for problems such as missing values and possible errors.*

```
tmelt ← reshape2::melt(tourism_full,id="Year")
```

```
tmelt %>% dim()
```

```
[1] 22274     3
```

`tmelt` (Table 1) contains the *melted* data frame, which allows us to visualize all 518 time series at once.

	Year	variable	value
22265	1998	Y518	1504
22266	1999	Y518	1343
22267	2000	Y518	1583
22268	2001	Y518	1772
22269	2002	Y518	1676
22270	2003	Y518	1423
22271	2004	Y518	1751
22272	2005	Y518	1385
22273	2006	Y518	1229
22274	2007	Y518	1102

Table 1: Melted `tsibble` containing all time series.

```
tmelt %>%
  ggplot(
    aes(
      x = Year,
      y = value,
      colour = variable,
      group = variable
      )
    ) +
  geom_line(
```

```
    alpha = .8
  ) +
  scale_y_log10() +
  scale_color_viridis_d(
    option = "cividis"
  ) +
  labs(
    title = "Tourism Time Series: Everything All At Once",
    y = "Value"
  ) +
  theme(
    legend.position = "none"
    )
```

```
Warning: Removed 11668 rows containing missing values (`geom_line()`).
```

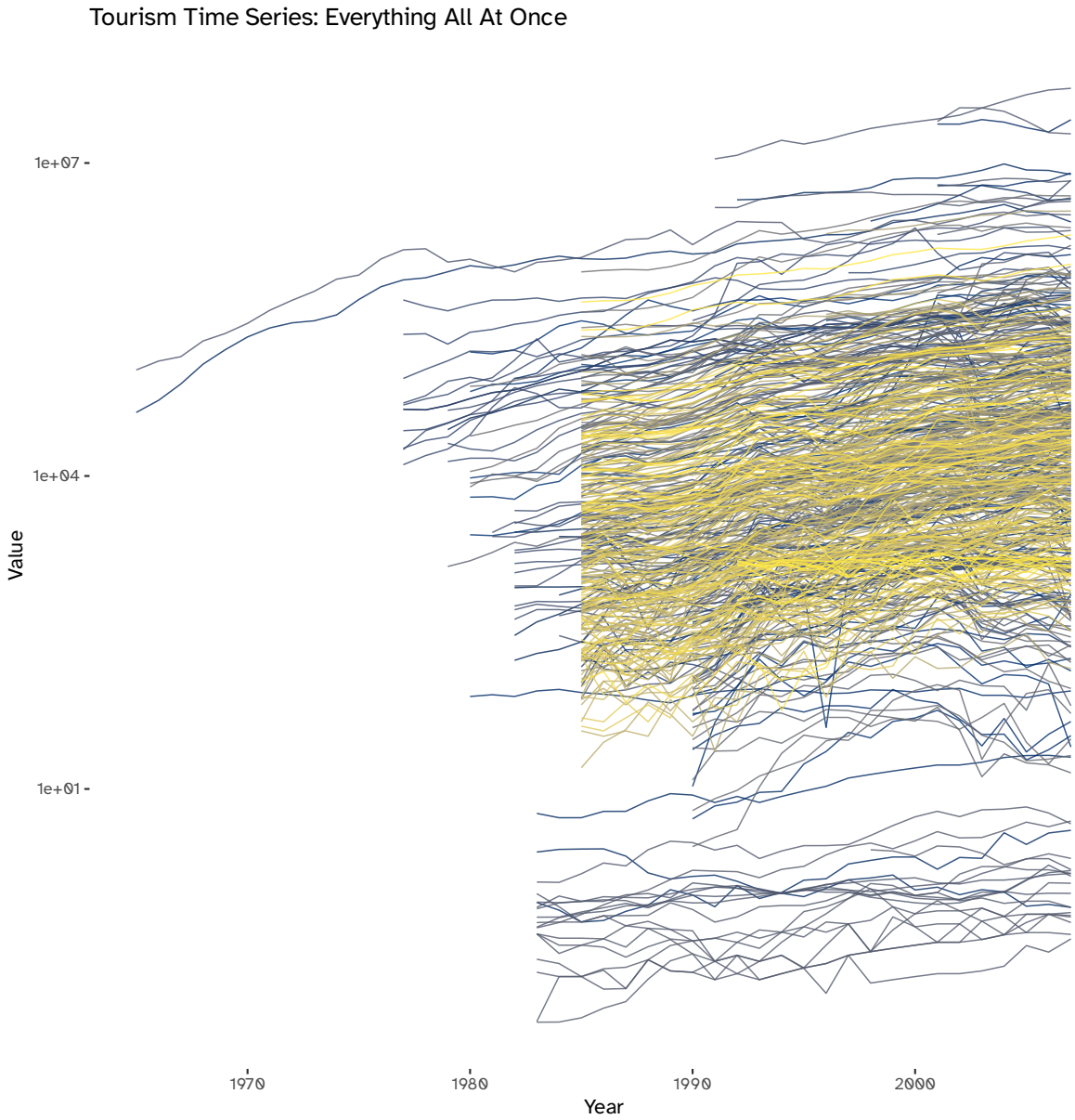

Figure 1: Printing a legend for 518 different series is not possible. However, color has been used only to differentiate the series and does not contain further information. Plotting the y-axis variable on the log scale was made necessary by the huge variation in the series values.

Plotting all 518 series does not allow to spot details, such as the presence of seasonal patterns. However, a general upward trend is clear; moreover, we can spot some probable outliers, that should be further investigated, and some clues about the presence of cyclicality in some of the series.

A check for NAs has already been made while loading data (Section 1.2) and it showed the presence of a large number of missing values, corresponding to 52.38 of all observations. While this is mostly to be attributed to the different starting time of the series, we can also see that some time series have missing values occurring between the starting and ending point; hence, some series present missing values *in between* available observations. From the visualization we can definitely group different starting points: this suggests that another approach to visualisation might be successfully conducted by grouping series by their starting date.

```
tmelt %>%
  mutate(
    Time_Interval = cut(
```

```
      Year,
      breaks = c(1964, 1975, 1985, 1995, 2003, 2008)
    )
  ) %>%
  group_by(Time_Interval) %>%
  summarise(
    Available_Observations = sum(
      !is.na(value)
      )
  )
```

Time_Interval	Available_Observations
(1964,1975]	22
(1975,1985]	625
(1985,1995]	3841
(1995,2003]	4046
(2003,2008]	2072

Table 2: Missing observation grouped by time windows.

```
tmelt %>%
  mutate(
    Time_Interval = cut(
      Year,
      breaks = c(1964, 1975, 1985, 1995, 2003, 2008)
    )
  ) %>%
  group_by(Year) %>%
  mutate(
    median_value = median(value, na.rm = T),
    mean_value = mean(value, na.rm = T),
    q_0.25 = quantile(value, probs = .25, na.rm = T),
    q_0.75 = quantile(value, probs = .75, na.rm = T)
  ) %>%
  ggplot(
    aes(
      x = Year
      )
    ) +
  geom_line(
    aes(
      y = mean_value,
    color = viridisLite::cividis(1, begin = 1),
    ),
    linewidth = 1,
    linetype = "dotted"
  ) +
  geom_line(
    aes(
      y = median_value,
    color = viridisLite::cividis(1, begin = .5)
    ),
    linewidth = 1
  ) +
  geom_line(
    aes(
      y = q_0.25,
```

```
    color = viridisLite::cividis(1, begin = .25)
    ),
    linewidth = 1
  ) +
  geom_line(
    aes(
      y = q_0.75,
    color = viridisLite::cividis(1, begin = .75)
    ),
    linewidth = 1
  ) +
  geom_ribbon(
    aes(
      ymin = q_0.25,
      ymax = q_0.75,
    ),
    fill = "grey95",
    alpha = .5
  ) +
  scale_y_log10()  +
  scale_color_viridis_d(
    labels = c(
      expression(q[0.25]),
      expression(q[0.50]),
      expression(q[0.75]),
      expression(mu)
      ),
    option = "cividis",
    direction = -1,
    end = .9
  ) +
  labs(
    title = "Tourism Time Series: Quartiles and Mean",
    y = "Value",
    colour = "Index"
  )
```

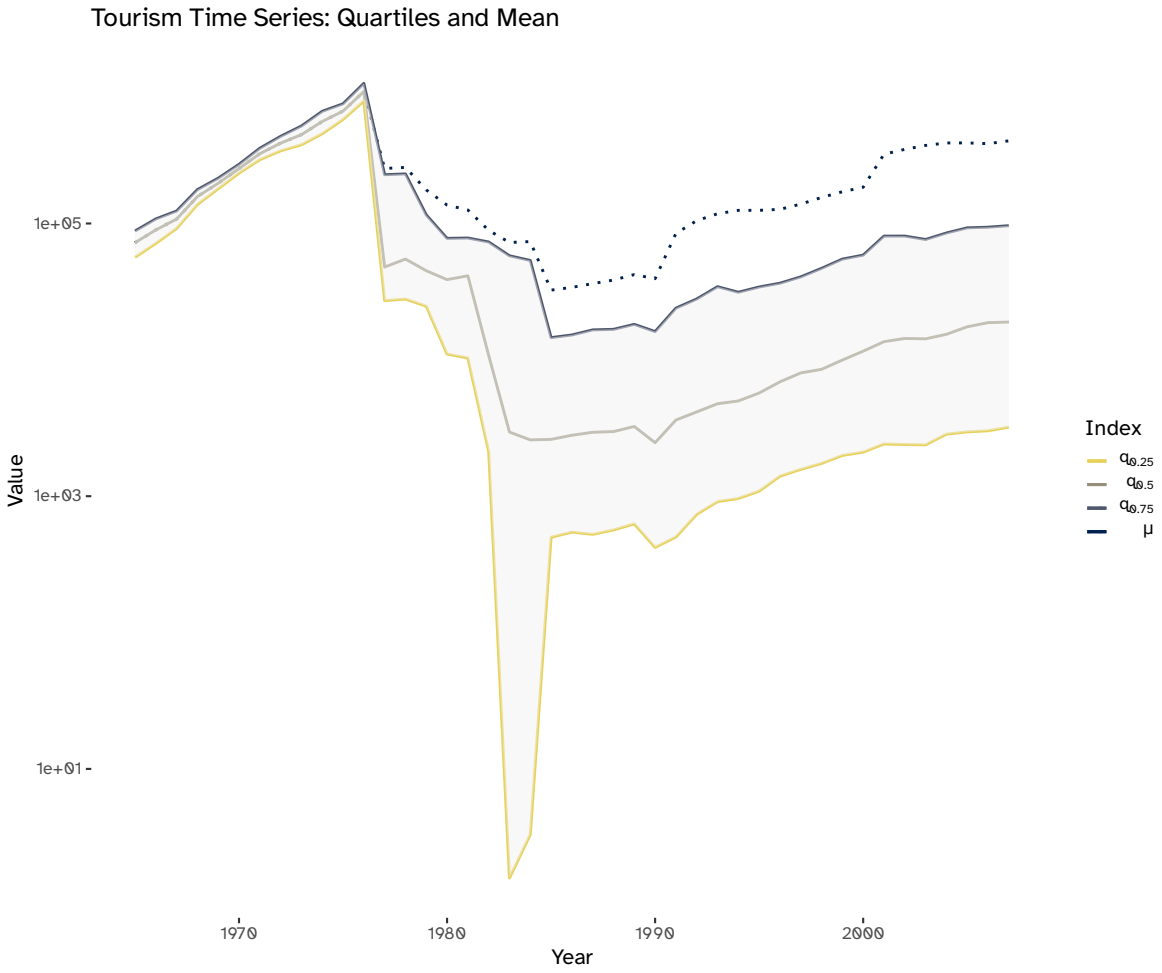

Figure 2: The following plot represents the position indexes given by the mean and the quartile, computed by considering all the series as observations and discarding missing observations.

2.2 Creating Validation Set

Partition the series into training and validation, so that the last 4 years are in the validation period for each series. What is the logic of such a partitioning? What is the disadvantage?

```
train ← tourism_full %>%
  filter(Year < 2004)
validation ← tourism_full %>%
  filter(Year ⩾ 2004)
```

The logic behind partitioning the series into a *training* and *validation* set is to *estimate the forecasting error*: we can train a model or apply a filter to the train set and use it to assess its performance with out-of-sample data. The main disadvantage with this approach is that we are not using all the information available to train our model; moreover, we are not computing *true forecasts*, therefore the accuracy measures from the residuals will be smaller.

2.3 Naïve Forecasts

Generate naïve forecasts for all series for the validation period. For each series, create forecasts with horizons of 1, 2, 3, and 4 years ahead ($F_{t+1}, F_{t+2}, F_{t+3}$, and F_{t+4}).

We know that *naïve* forecasts consist in the last observation, $\forall h$.

$$y_{T+h \mid T} = y_T \tag{1}$$

It follows that we can produce the forecasts with the following code:

```
naive_forecast ← train %>%
  filter(
    Year == 2003
  ) %>% as_tibble()
```

`naive_forecast` will contain a `tsibble` with y_t for all the series.

```
naive_forecast %>% dim
```

```
[1]   1 519
```

To obtain $F_{t+1}, F_{t+2}, F_{t+3}$, and F_{t+4}:

```
naive_2004_2007 ← merge(
    x = 2003 + 1:4,
    y = naive_forecast
  ) %>%
  mutate(
    Year = x
  ) %>%
  select(
    -x
  ) %>%
  as_tsibble(
    index = Year
  )
```

2.4 Choosing Measures

> *Which measures are suitable if we plan to combine the results for the 518 series? Consider MAE, Average error, MAPE and RMSE.*

2.5 Computing MAPE

> *For each series, compute MAPE of the naive forecasts once for the training period and once for the validation period.*

2.5.1 Training Period:

It follows from the definition of *naïve forecasts* (Equation 1) that what is needed to compute the MAPE for all the training dataset is just a modified version of it in which all rows change their position by 1. The last forecast can be added by binding it to maintain the same dimensions and have a full forecast matrix.

```
train_forecasts ← train %>%
  select(-Year) %>%
  as_tibble()
first_forecast ← rep(
  NA,
```

```
  518
)
naive_train <- rbind(
  c(
    rep(
      NA,
      dim(train_forecasts)[2]
      )
  ),
  train_forecasts[1:(dim(train_forecasts)[1] -1), ]
)
```

```
uhat_full <- naive_train - train
p_uhat_full <- uhat_full/train
```

```
mape_full <-  100*apply(
    X = p_uhat_full %>% select(-Year),
    FUN = mean,
    MARGIN = 2,
    na.rm = T
  )
```

```
mape_full %>% length
```

```
[1] 518
```

```
mape_full[1:10] %>%
  round(., digits = 2) %>%
  t() %>%
  as_tibble() %>%
  tail()
```

Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10
-2.98	-7.25	-6.44	-9.69	-1.62	-4.48	-3.53	-3.94	-7.57	-9.46

Table 3: Naïve forecasts training MAPE for the first 10 time series from 1998 to 2003.

2.5.2 Validation Period:

```
uhat_validation <- naive_2004_2007 - validation
p_uhat <- uhat_validation/validation
```

```
mape_validation <-  100*apply(
    X = p_uhat %>% select(-Year),
    FUN = mean,
    MARGIN = 2,
    na.rm = T
  )
```

```
mape_validation %>% length
```

```
[1] 518
```

This is the MAPE for the first 10 series.

```
mape_validation[1:10] %>%
  round(., digits = 2) %>%
  t() %>%
  as_tibble()
```

Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10
-16.59	-6.05	4.08	-20.52	-24.44	-15.28	-17.13	-12.3	-13.39	-16.17

Table 4: Naïve forecasts validation MAPE for the first 10 time series for the years 2004, 2005, 2006, 2007.

2.6 Computing MASE

The performance measure used in the competition is Mean Absolute Scaled Error (MASE). Explain the advantage of MASE and compute the training and validation MASE for the naive forecasts.

2.7 MAPE Pairs

Create a scatter plot of the MAPE pairs, with the training MAPE on the x-axis and the validation MAPE on the y-axis. Create a similar scatter plot for the MASE pairs. Now examine both plots. What do we learn? How does performance differ between the training and validation periods? How does performance range across series?

```
ggplot(
  data = tibble(
    Training_MAPE = mape_full,
    Validation_MAPE = mape_validation,
    Series_Identifier = names(mape_full)
  ),
  aes(
    x = Training_MAPE,
    y = Validation_MAPE,
  color = Series_Identifier
  ),
) + geom_point(
  alpha = .8
) +
  geom_rug() +
  labs(
    title = "Training and Validation MAPE pairs, colored by series",
    x = "Training MAPE",
    y = "Validation MAPE"
  ) +
  scale_color_viridis_d(
    option = "cividis"
  ) +
  ggthemes::theme_tufte(
    base_size = 16,
    base_family = "Atkinson Hyperlegible",
    ticks = F
  ) +
  theme(
    legend.position = "none"
```

```
)
```

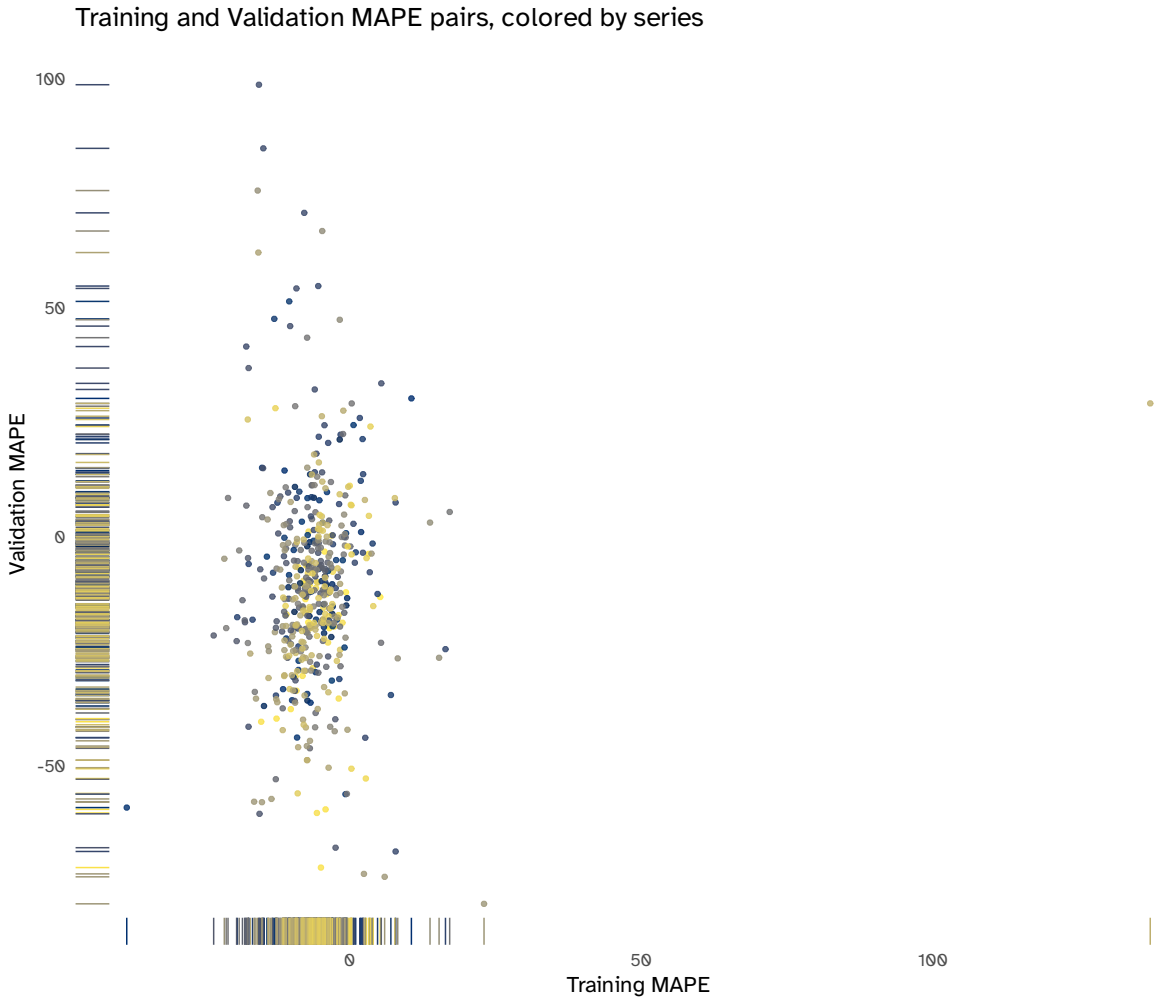

Figure 3: Scatterplot of training and validation MAPE pairs: on both axis, the distribution of values. The time series have been colored using the same mapping seen in Figure 1.

2.8 Ensemble Methods

The competition winner, Lee Baker, used an ensemble of three methods:

- *Naive forecasts multiplied by a constant trend*[1].
- Linear regression.
- Exponentially-weighted linear regression.

[1] Global/local trend: "globally tourism has grown "at a rate of 6% annually."

2.8.1 a. *Write the exact formula used for generating the first method, in the form* $F_{t+k} = ...(k = 1, 2, 3, 4)$,

2.8.2 b. *What is the rational behind multiplying the naive forecasts by a constant?*[2]

[2] Hint: think empirical and domain knowledge.

2.8.3 c. *What should be the dependent variable and the predictors in a linear regression model for this data? Explain..*

2.8.4 d. *Fit the linear regression model to the first five series and compute forecast errors for the validation period.*

```
train_subset ← train %>%
  select(
    Y1,
    Y2,
    Y3,
    Y4,
    Y5
  )
```

2.8.5 e. *Before choosing a linear regression, the winner described the following process:*

"I examined fitting a polynomial line to the data and using the line to predict future values. I tried using first through fifth order polynomials to find that the lowest MASE was obtained using a first order polynomial (simple regression line). This best fit line was used to predict future values. I also kept the R^2 value of the fit for use in blending the results of the prediction."

What are two flaws in this approach?

2.8.6 f. *If we were to consider exponential smoothing, what particular type(s) of exponential smoothing are reasonable candidates?*

2.8.7 g. *The winner concludes with possible improvements one being "an investigation into how to come up with a blending ensemble method that doesn't use much manual twerking would also be of benefit". Can you suggest methods or an approach that would lead to easier automation of the ensemble step?*

2.8.8 h. *The competition focused on minimizing the average MAPE of the next four values across all 518 series. How does this goal differ from goals encountered in practice when considering tourism demand? Which steps in the forecasting process would likely be different in a real-life tourism forecasting scenario?*