Project: The Forecasting Tourism 2010 Competition

EM1415

Marco Solari, 875475

Table of contents

---

1 Setup and Data Loading

1.1 Setup

```r
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
```

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

1.2 Loading Data

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

```r
data_main %>% dim
```

```
[1]  43 518
```

```r
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
  )
```

`tmelt` (Table 1) contains the *melted* data frame, which allows us to apply the tidy forecasting workflow all 518 time series at once. Its main variables are:

- `index`: `Year`, as in the original data frame.
- `key`: `Identifier`, a new categorical variable allowing us to transform the data frame the tidy format; it consists in a set of *labels* that identify each time series.
- `value`: the Y_t value for each time series.

```
tmelt ← reshape2::melt(
  tourism_full,
  id = "Year",
  variable.name = "Identifier",
  value.name = "Value"
  ) %>%
  as_tsibble(
    index = "Year",
    key = "Identifier"
  )
```

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

```
[1] 22274     3
```

Year	Identifier	Value
1998	Y518	1504
1999	Y518	1343
2000	Y518	1583
2001	Y518	1772
2002	Y518	1676
2003	Y518	1423
2004	Y518	1751
2005	Y518	1385
2006	Y518	1229
2007	Y518	1102

Table 1: Excerpt of melted `tsibble` containing all time series.

2 Assignment

2.1 Full Plot

In all the subsequent plots, a log_{10} transformation has been employed exclusively for representing the time series on the y-axis. This adjustment becomes necessary since the original data range[1] does not permit a clear

[1] 10^9, shown in Table 2.

and meaningful visualization of the series when plotted together.

```
tmelt %>%
  reframe(
    "Range" =  range(
      Value,
      na.rm = T,
      finite = T
      )
    ) %>%
  mutate(
    "Y" = c(
      "min",
      "max"
      ),
    .before = "Range"
    )
```

Y	Range
min	5.810000e-02
max	5.200294e+07

Table 2: Range of Tourism Time Series

2.1.1 Everything, Everywhere, All At Once

> *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 %>%
  ggplot(
    aes(
      x = Year,
      y = Value,
      colour = Identifier,
      group = Identifier
      )
    ) +
  geom_line(
    alpha = .8
  ) +
  scale_y_log10() +
  scale_color_viridis_d(
    option = "cividis"
  ) +
  labs(
    title = "Tourism Time Series: Everything All At Once",
    y = expression(log[10](Value))
  ) +
  theme(
    legend.position = "none"
    )
```

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

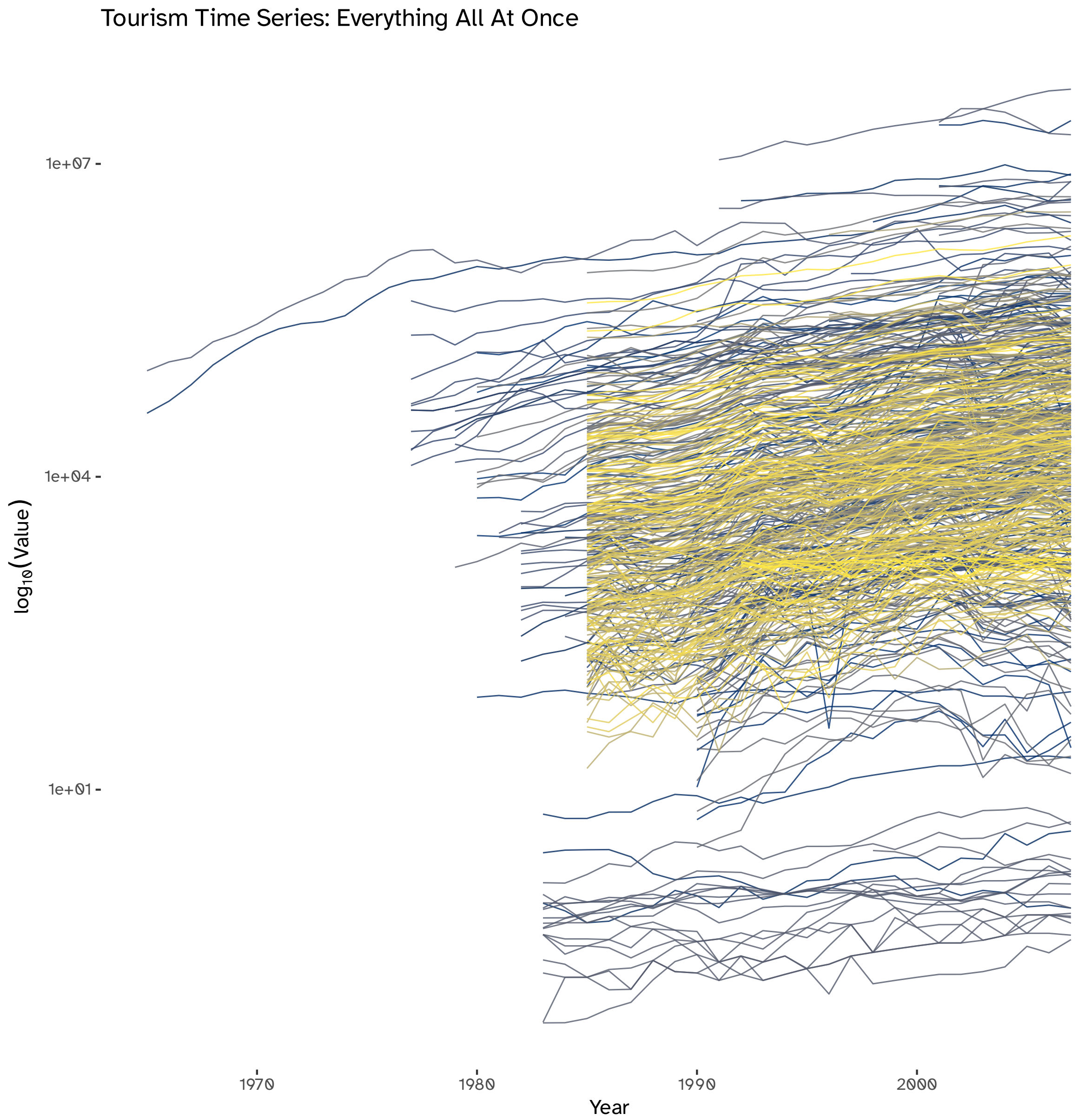

Figure 1: Printing a legend for 518 different series is not possible. However, color has been used only to differentiate the series and does not contain further information. Plotting the y-axis variable on the log scale was made necessary by the huge variation in the series values.

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 notable outliers, that should be further investigated, and some clues about the presence of cyclicality in some of the series.

A check for `NA`s 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. Clearly, this can be attributed to the distinct initial timestamps of the series. It is evident that we can categorize these series based on their respective starting years, indicating that an alternative visualization approach could be effectively implemented through this grouping method (Figure 2).

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

```r
      breaks = c(1964, 1975, 1985, 1995, 2003, 2007)
    )
  ) %>%
 # as_tibble() %>%
  select(-Year) %>%
  group_by(Time_Interval) %>%
  summarise(
    Available_Observations = sum(
      !is.na(Value)
      )
  )
```

Time_Interval	Year	Available_Observations
(1964,1975]	1965	2
(1964,1975]	1966	2
(1964,1975]	1967	2
(1964,1975]	1968	2
(1964,1975]	1969	2
(1964,1975]	1970	2
(1964,1975]	1971	2
(1964,1975]	1972	2
(1964,1975]	1973	2
(1964,1975]	1974	2
(1964,1975]	1975	2
(1975,1985]	1976	2
(1975,1985]	1977	13
(1975,1985]	1978	13
(1975,1985]	1979	18
(1975,1985]	1980	29
(1975,1985]	1981	31
(1975,1985]	1982	47
(1975,1985]	1983	66
(1975,1985]	1984	70
(1975,1985]	1985	336
(1985,1995]	1986	342
(1985,1995]	1987	342
(1985,1995]	1988	342
(1985,1995]	1989	342
(1985,1995]	1990	391
(1985,1995]	1991	406
(1985,1995]	1992	419
(1985,1995]	1993	419
(1985,1995]	1994	419
(1985,1995]	1995	419
(1995,2003]	1996	489
(1995,2003]	1997	494
(1995,2003]	1998	503
(1995,2003]	1999	503
(1995,2003]	2000	503
(1995,2003]	2001	518
(1995,2003]	2002	518
(1995,2003]	2003	518
(2003,2007]	2004	518
(2003,2007]	2005	518
(2003,2007]	2006	518
(2003,2007]	2007	518

Table 3: Missing observation grouped by time windows: binning the data suggests that the presence of missing observations is related to the scarcity of long-run time series.

2.1.2 Plotting Series By Starting Year

```
tmelt %>%
  group_by(Identifier) %>%
  mutate(series_length = 43-  Value %>% is.na %>% sum) %>%
  ungroup() %>%
  arrange(desc(series_length)) %>%
  mutate(series_length = as_factor(series_length)) %>%
  ggplot(
    aes(x = Year)
  ) +
  facet_wrap(
    ~series_length,
    nrow = 6,
    ncol = 3,
    scales = "free"
  ) +
  geom_line(
    aes(
      y = Value,
      color = Identifier
    )
  ) +
  labs(
    title = "Tourism Time Series By Starting Year",
    y = expression(log[10](Value))
  ) +
  scale_y_log10() +
  scale_color_viridis_d(
    option = "cividis"
  ) +
  theme(
    legend.position = "none"
  )
```

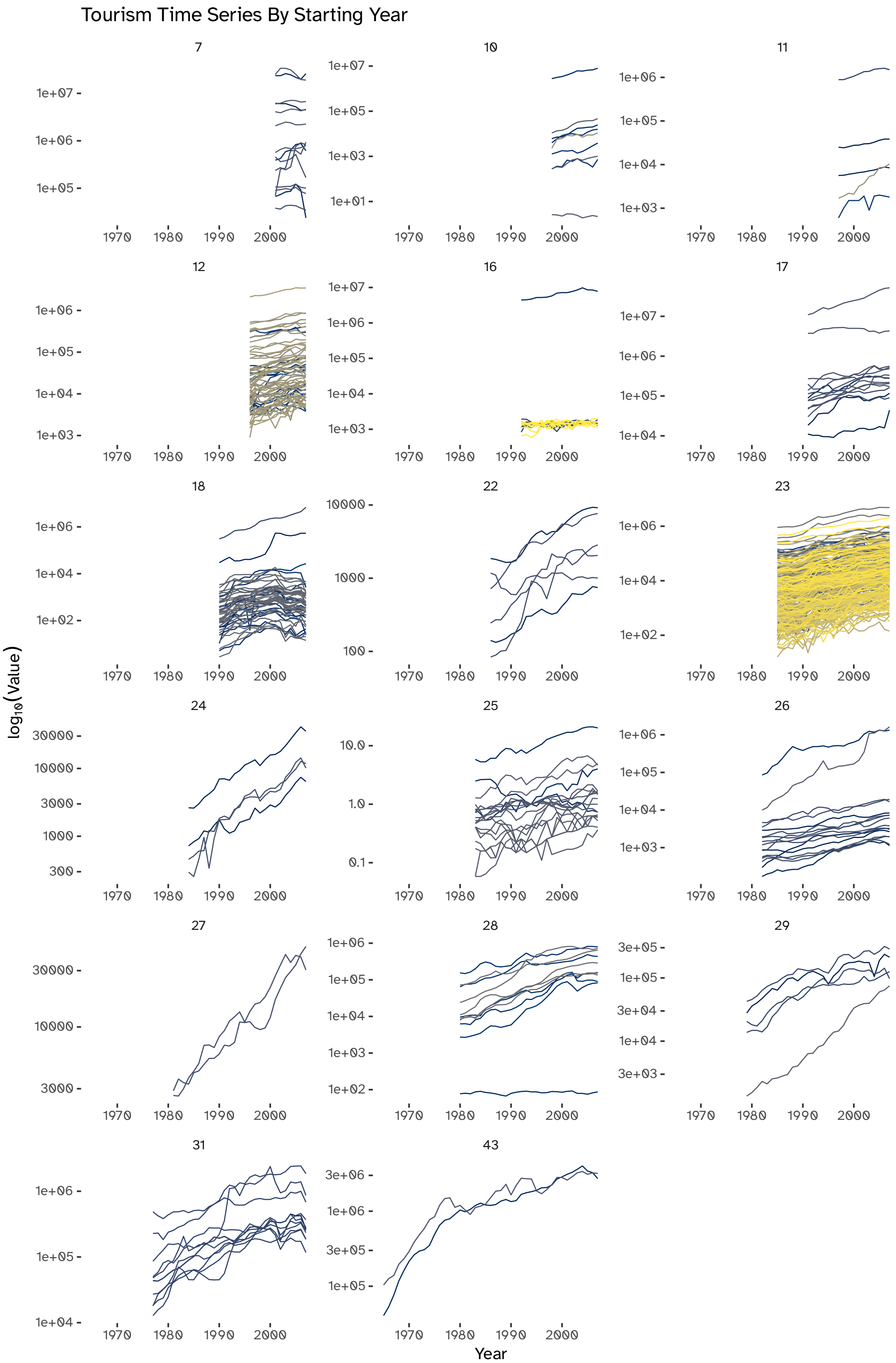

Figure 2: All series have been grouped by starting year and plotted to achieve more clarity. Each subtitle represent the number of periods for each subset. The same color mapping of Figure 1 has been used to differentiate the series.

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:

Derived from the definition of *naïve forecasts* outlined in Equation 1, it becomes evident that computing the Mean Absolute Percentage Error (MAPE) for the entire training dataset primarily involves a scaled version of the same data, with each row shifting by a single step. In this modified version, a row containing NAs symbolizes the (unavailable) naïve forecasts for $t = 1$.

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

```
mape_training %>% length
```

```
[1] 518
```

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
```

2.5.3 Comparison Table

```
bind_rows(
mape_training[1:10] %>%
  round(., digits = 2) %>%
  t() %>%
  as_tibble() %>%
  mutate(
    Set = "Training",
    .before = Y1
  ) %>%
  tail(),
mape_validation[1:10] %>%
  round(., digits = 2) %>%
  t() %>%
  as_tibble() %>%
  mutate(
    Set = "Validation"
  )
)
```

Set	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10
Training	-2.98	-7.25	-6.44	-9.69	-1.62	-4.48	-3.53	-3.94	-7.57	-9.46
Validation	-16.59	-6.05	4.08	-20.52	-24.44	-15.28	-17.13	-12.30	-13.39	-16.17

Table 4: Naïve forecasts training and MAPEs for the first 10 time series.

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.

2.6.1 Training Period:

```r
qhat_training <- (dim(uhat_full)[1] - 1) * uhat_full / sum(abs(uhat_full),
↪  na.rm = T)
```

```r
mase_training <- apply(
    X = qhat_training %>% select(-Year),
    FUN = mean,
    MARGIN = 2,
    na.rm = T
  )
```

```r
mase_training %>% length
```

```
[1] 518
```

This is the MASE for the first 10 series.

2.6.2 Validation Period:

```r
qhat_validation <- (dim(uhat_validation)[1] - 1) * uhat_validation /
↪  sum(abs(uhat_validation), na.rm = T)
```

```r
mase_validation <- apply(
    X = qhat_validation %>% select(-Year),
    FUN = mean,
    MARGIN = 2,
    na.rm = T
  )
```

```r
mase_validation %>% length
```

```
[1] 518
```

2.6.3 Comparison Table:

```r
tibble(
  "Time Series Identifier" = mase_training[1:10] %>% names(),
  "Training MASE" = mase_training[1:10],
  "Validation MASE" = mase_validation[1:10]
)
```

Time Series Identifier	Training MASE	Validation MASE
Y1	-0.0002733	-0.0001112
Y2	-0.0037849	-0.0007173
Y3	-0.0019614	-0.0001876
Y4	-0.0016011	-0.0003841
Y5	-0.0000849	-0.0001560
Y6	-0.0000227	-0.0000077

Time Series Identifier	Training MASE	Validation MASE
Y7	-0.0000539	-0.0000246
Y8	-0.0000822	-0.0000186
Y9	-0.0001114	-0.0000219
Y10	-0.0000353	-0.0000092

Table 5: Naïve forecasts training and MAPEs for the first 10 time series.

2.7 MAPE & MASE 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_training,
    Validation_MAPE = mape_validation,
    Series_Identifier = names(mape_training)
  ),
  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"
  )
```

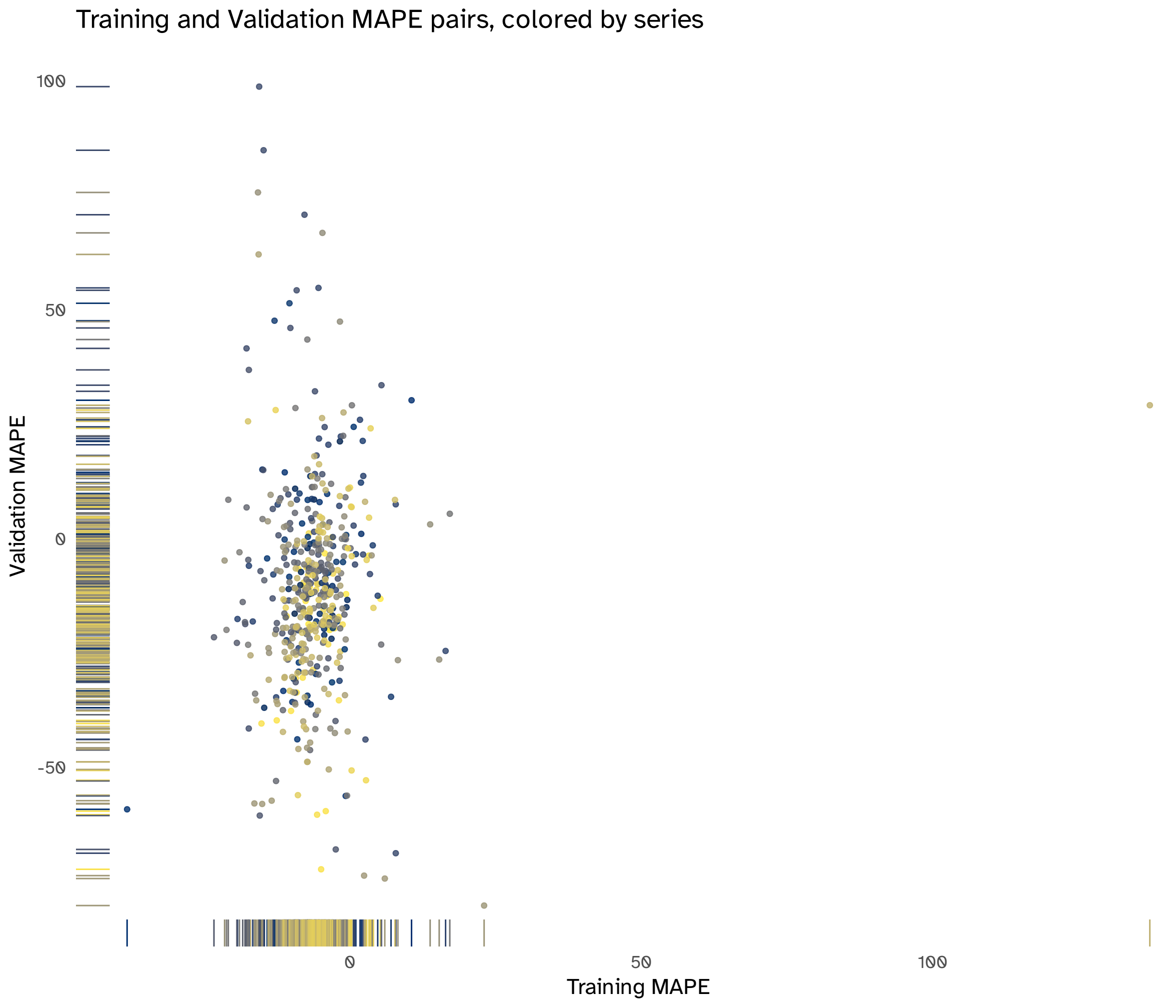

Figure 3: Scatterplot of training and validation MAPE pairs: on both axis, the distribution of values. The time series have been colored using the same mapping seen in Figure 1.

```
ggplot(
  data = tibble(
    Training_MASE = mase_training,
    Validation_MASE = mase_validation,
    Series_Identifier = names(mase_training)
  ),
  aes(
    x = Training_MASE,
    y = Validation_MASE,
  color = Series_Identifier
  ),
) + geom_point(
  alpha = .8
) +
  geom_rug() +
  labs(
    title = "Training and Validation MASE pairs, colored by series",
    x = "Training MASE",
    y = "Validation MASE"
  ) +
  scale_color_viridis_d(
```

```
    option = "cividis"
  ) +
  ggthemes::theme_tufte(
    base_size = 16,
    base_family = "Atkinson Hyperlegible",
    ticks = F
  ) +
  theme(
    legend.position = "none"
  )
```

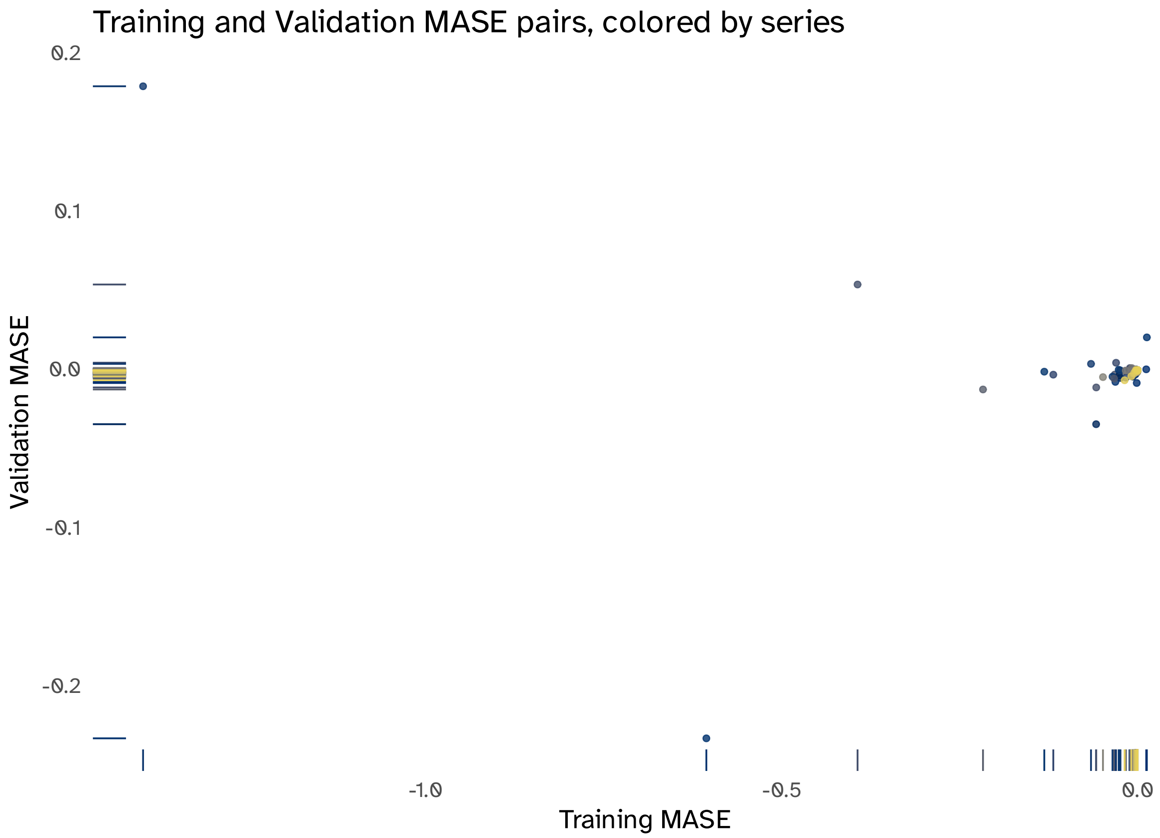

Figure 4: Scatterplot of training and validation MASE pairs: on both axis, the distribution of values. The time series have been colored using the same mapping seen in Figure 1.

Most of the pairs cluster around (0,0), with some notable outliers.

We can filter out these outliers in the MASE training/validation pair to "zoom in" the area in which most of them are clustering:

```
ggplot(
  data = tibble(
    Training_MASE = mase_training,
    Validation_MASE = mase_validation,
    Series_Identifier = names(mase_training)
  ) %>% filter(
    Training_MASE ⩽ 3/2*quantile(Training_MASE, probs = .75) &
    Training_MASE ⩾ 3/2*quantile(Training_MASE, probs = .25) &
    Validation_MASE ⩽ 3/2*quantile(Training_MASE, probs = .75) &
    Validation_MASE ⩾ 3/2*quantile(Training_MASE, probs = .25)
```

```r
    ),
    aes(
      x = Training_MASE,
      y = Validation_MASE,
    color = Series_Identifier
    ),
  ) + geom_point(
    alpha = .8
  ) +
    geom_rug() +
    labs(
      title = "Training and Validation MASE pairs, colored by series",
      x = "Training MASE",
      y = "Validation MASE"
    ) +
    scale_color_viridis_d(
      option = "cividis"
    ) +
    ggthemes::theme_tufte(
      base_size = 16,
      base_family = "Atkinson Hyperlegible",
      ticks = F
    ) +
    theme(
      legend.position = "none",
      plot.margin = margin(0, 1, 0, 0, "cm")
    )
```

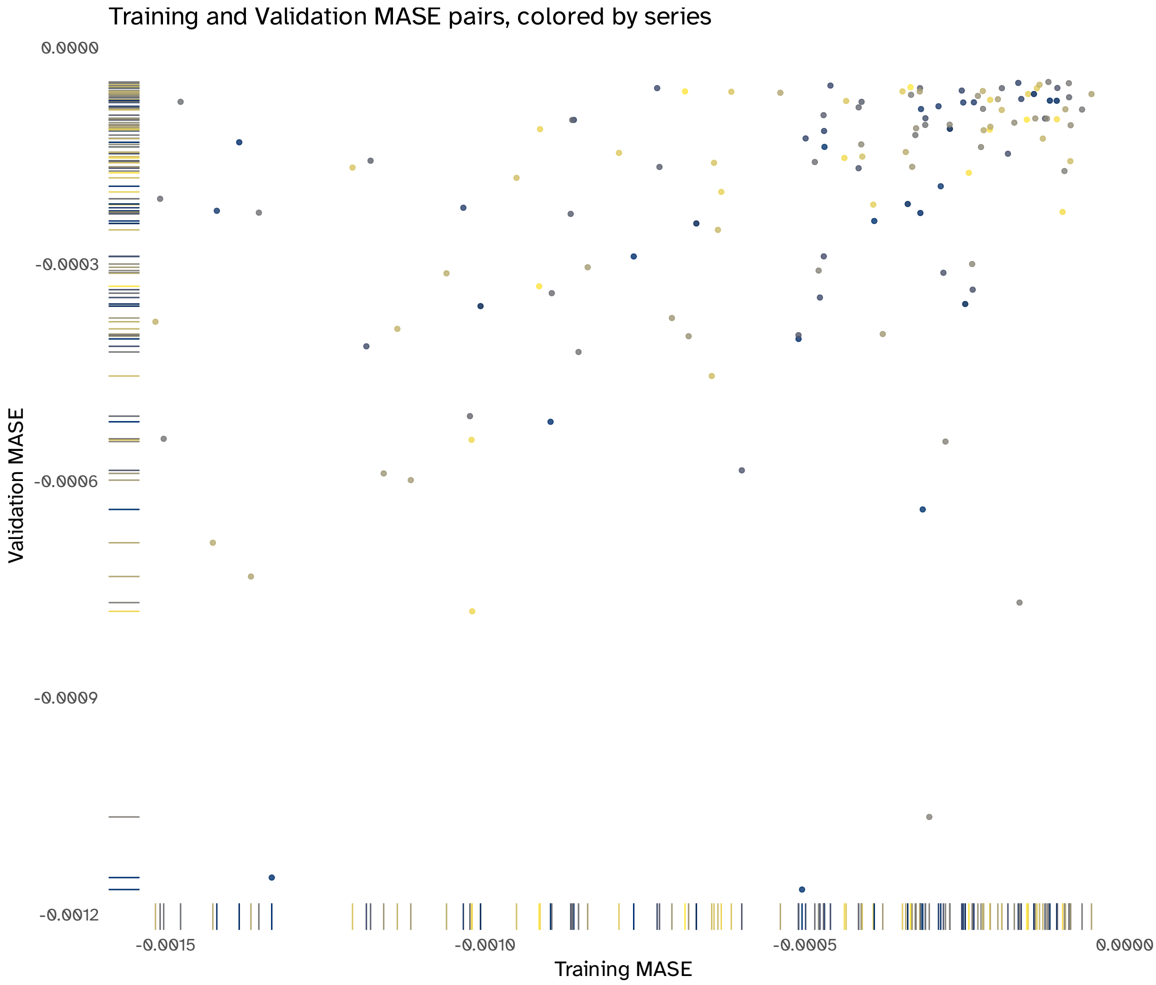

Figure 5: Scatterplot of all MASE pairs, as in Figure 4, with outliers exceeding 3/2 times the IQR filtered out.

2.8 Ensemble Methods

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

- *Naive forecasts multiplied by a constant trend*[2].
- Linear regression.
- Exponentially-weighted linear regression.

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

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

b. *What is the rational behind multiplying the naive forecasts by a constant?*[3]

[3] Hint: think empirical and domain knowledge.

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

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
  )
```

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?*

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

g. *The winner concludes with possible improvements one being àn 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?*

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?*