

5. The forecaster's toolbox

5.1 A tidy forecasting workflow

OTexts.org/fpp3/

Rob J Hyndman
George Athanasopoulos

FORECASTING

PRINCIPLES AND PRACTICE

A comprehensive introduction to the latest forecasting methods using R. Learn to improve your forecast accuracy using dozens of real data examples.



3RD EDITION

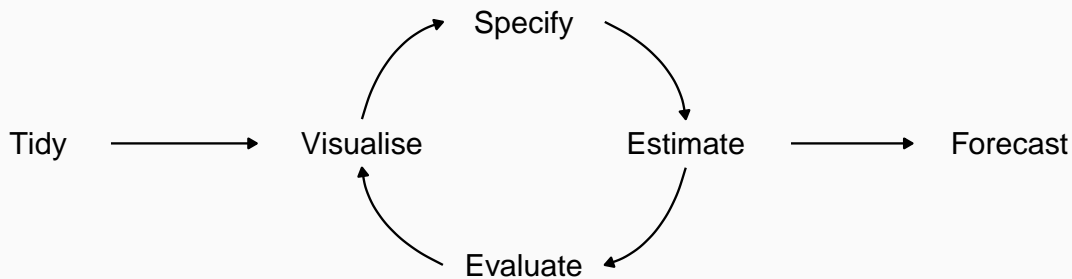
 **OTexts**
OPEN. APPLIED. ACCESS. FORECASTING.

A tidy forecasting workflow

The process of producing forecasts can be split up into a few fundamental steps.

- 1 Preparing data
- 2 Data visualisation
- 3 Specifying a model
- 4 Model estimation
- 5 Accuracy & performance evaluation
- 6 Producing forecasts

A tidy forecasting workflow



Data preparation (tidy)

```
gdppc <- global_economy |>
  mutate(GDP_per_capita = GDP / Population) |>
  select(Year, Country, GDP, Population, GDP_per_capita)
gdppc
```

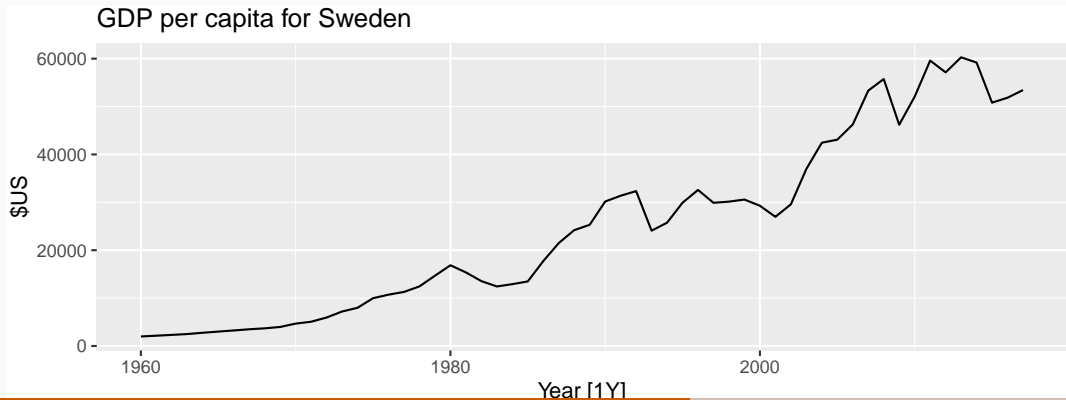
```
## # A tsibble: 15,150 x 5 [1Y]
```

```
## # Key:      Country [263]
```

##	Year	Country	GDP	Population	GDP_per_capita
##	<dbl>	<fct>	<dbl>	<dbl>	<dbl>
## 1	1960	Afghanistan	537777811.	8996351	59.8
## 2	1961	Afghanistan	548888896.	9166764	59.9
## 3	1962	Afghanistan	546666678.	9345868	58.5
## 4	1963	Afghanistan	751111191.	9533954	78.8
## 5	1964	Afghanistan	800000044.	9731361	82.2
## 6	1965	Afghanistan	1006666638.	9938414	101.

Data visualisation

```
gdppc |>  
  filter(Country == "Sweden") |>  
  autoplot(GDP_per_capita) +  
  labs(title = "GDP per capita for Sweden", y = "$US")
```



Model estimation

The `model()` function trains models to data.

```
fit <- gdppc |>
  model(trend_model = TSLM(GDP_per_capita ~ trend()))
fit
```

```
## # A mable: 263 x 2
## # Key:      Country [263]
##   Country      trend_model
##   <fct>         <model>
## 1 Afghanistan <TSLM>
## 2 Albania     <TSLM>
## 3 Algeria     <TSLM>
## 4 American Samoa <TSLM>
## 5 Andorra     <TSLM>
```

Model estimation

The `model()` function trains models to data.

```
fit <- gdppc |>
  model(trend_model = TSLM(GDP_per_capita ~ trend()))
fit
```

```
## # A mable: 263 x 2
## # Key:      Country [263]
##   Country      trend_model
##   <fct>         <model>
## 1 Afghanistan <TSLM>
## 2 Albania      <TSLM>
## 3 Algeria      <TSLM>
## 4 American Samoa <TSLM>
## 5 Andorra      <TSLM>
```

A mable is a model table, each cell corresponds to a fitted model.

Producing forecasts

```
fit |> forecast(h = "3 years")
```

```
## # A tibble: 789 x 5 [1Y]
```

```
## # Key:      Country, .model [263]
```

##	Country	.model	Year	GDP_per_capita	.mean
##	<fct>	<chr>	<dbl>	<dist>	<dbl>
##	1 Afghanistan	trend_model	2018	N(526, 9653)	526.
##	2 Afghanistan	trend_model	2019	N(534, 9689)	534.
##	3 Afghanistan	trend_model	2020	N(542, 9727)	542.
##	4 Albania	trend_model	2018	N(4716, 476419)	4716.
##	5 Albania	trend_model	2019	N(4867, 481086)	4867.
##	6 Albania	trend_model	2020	N(5018, 486012)	5018.
##	7 Algeria	trend_model	2018	N(4410, 643094)	4410.
##	8 Algeria	trend_model	2019	N(4489, 645311)	4489.

Producing forecasts

```
fit |> forecast(h = "3 years")
```

A fable is a forecast table with
point forecasts and distributions.

```
## # A fable: 789 x 5 [1Y]
```

```
## # Key:      Country, .model [263]
```

##	Country	.model	Year	GDP_per_capita	.mean
##	<fct>	<chr>	<dbl>	<dist>	<dbl>
##	1 Afghanistan	trend_model	2018	N(526, 9653)	526.
##	2 Afghanistan	trend_model	2019	N(534, 9689)	534.
##	3 Afghanistan	trend_model	2020	N(542, 9727)	542.
##	4 Albania	trend_model	2018	N(4716, 476419)	4716.
##	5 Albania	trend_model	2019	N(4867, 481086)	4867.
##	6 Albania	trend_model	2020	N(5018, 486012)	5018.
##	7 Algeria	trend_model	2018	N(4410, 643094)	4410.
##	8 Algeria	trend_model	2019	N(4489, 645311)	4489.

Visualising forecasts

```
fit |>  
  forecast(h = "3 years") |>  
  filter(Country == "Sweden") |>  
  autoplot(gdppc) +  
  labs(title = "GDP per capita for Sweden", y = "$US")
```

