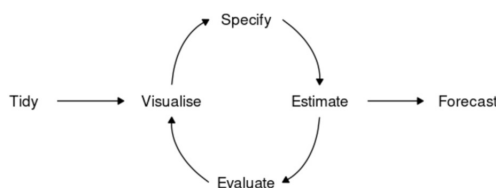


5. La caja del herramientas del pronosticador.

En este capítulo, discutimos algunas herramientas generales que son útiles para muchas situaciones de pronóstico diferentes. Describiremos algunos métodos de pronóstico de referencia, procedimientos para verificar si un método de pronóstico ha utilizado adecuadamente la información disponible, técnicas para calcular los intervalos de predicción y métodos para evaluar la precisión del pronóstico. Cada una de las herramientas discutidas en este capítulo se usará repetidamente en capítulos posteriores a medida que desarrollemos y exploremos una variedad de métodos de pronóstico.

5.1 Un flujo de trabajo de previsión ordenado

El proceso de producción de pronósticos para datos de series de tiempo se puede dividir en unos pocos pasos.



Para ilustrar el proceso, ajustaremos modelos de tendencia lineal a los datos del PIB nacional almacenados en `global_economy`

Preparación de datos (ordenado)

El primer paso en la previsión es preparar los datos en el formato correcto. Este proceso puede implicar la carga de datos, la identificación de valores faltantes, el filtrado de series temporales y otras tareas de preprocesamiento. La funcionalidad proporcionada por `tsibble` y otros paquetes en el `tidyverse` simplifica sustancialmente este paso.

Muchos modelos tienen diferentes requisitos de datos; algunos requieren que la serie esté en orden de tiempo, otros requieren que no falten valores. Verificar sus datos es un paso esencial para comprender sus características y siempre debe hacerse antes de estimar modelos.

Modelaremos el PIB per cápita a lo largo del tiempo; así que primero, debemos calcular la variable relevante.

```
library(fpp3)

## - Attaching packages ----- fpp3 0.4.0 -
## v tibble      3.1.6      v tsibble      1.1.1
## v dplyr       1.0.8      v tsibbledata 0.4.0
## v tidyr       1.2.0      v feasts      0.2.2
## v lubridate   1.8.0      v fable       0.3.1
## v ggplot2     3.3.5
## - 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()

head(global_economy,5) #Los primeros 5 filas
```

```
## # A tibble: 5 x 9 [1Y]
## # Key:      Country [1]
##   Country    Code  Year      GDP Growth  CPI Imports Exports Population
##   <fct>      <fct> <dbl>    <dbl>  <dbl> <dbl>  <dbl>  <dbl>  <dbl>
## 1 Afghanistan AFG    1960 537777811.    NA    NA    7.02    4.13    8996351
## 2 Afghanistan AFG    1961 548888896.    NA    NA    8.10    4.45    9166764
## 3 Afghanistan AFG    1962 546666678.    NA    NA    9.35    4.88    9345868
## 4 Afghanistan AFG    1963 751111191.    NA    NA    16.9    9.17    9533954
## 5 Afghanistan AFG    1964 800000044.    NA    NA    18.1    8.89    9731361

colnames(global_economy) #El nombre de las columnas

## [1] "Country"      "Code"          "Year"          "GDP"           "Growth"
## [6] "CPI"          "Imports"       "Exports"       "Population"

class(global_economy) #Tipo de clase de global_economy

## [1] "tbl_ts"       "tbl_df"       "tbl"          "data.frame"

gdppc <- global_economy %>% mutate(GDP_per_capita = GDP / Population )
colnames(gdppc) #Nombre de las columnas de gdpp

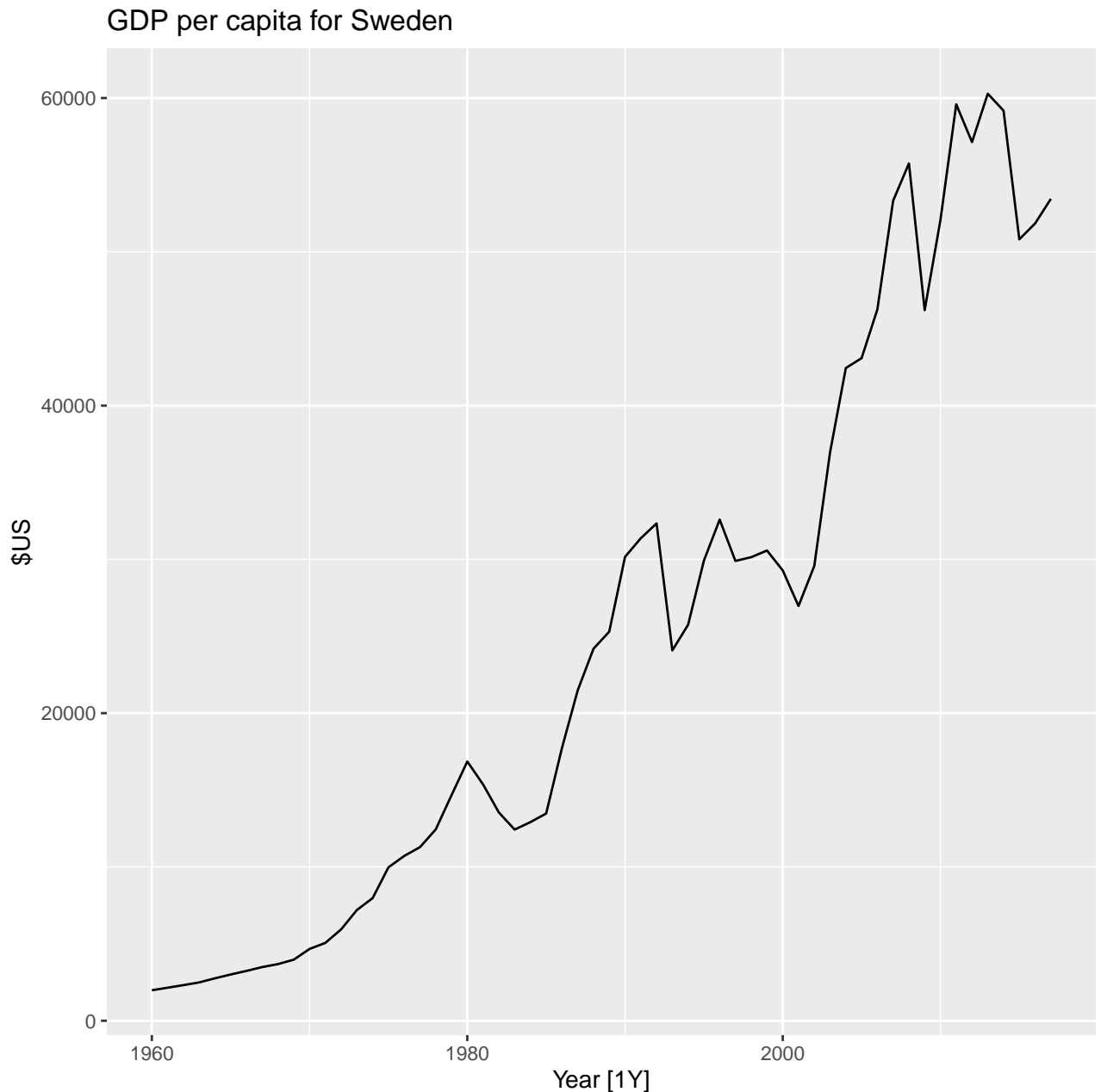
## [1] "Country"      "Code"          "Year"          "GDP"
## [5] "Growth"       "CPI"           "Imports"       "Exports"
## [9] "Population"   "GDP_per_capita"
```

Trazar los datos (visualizar)

La visualización es un paso esencial para comprender los datos. Mirar los datos le permite identificar patrones comunes y posteriormente, especificar un modelo apropiado.

Los datos de un país en nuestro ejemplo:

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



La figura muestra datos del PIB per cápita de Suecia desde 1960 hasta 2017.

Definir un modelo (especificar)

Hay muchos modelos diferentes de series de tiempo que se pueden usar para pronosticar, y gran parte de este libro está dedicado a describir varios modelos. Especificar un modelo apropiado para los datos es esencial para producir pronósticos apropiados.

Los modelos en **fable** se especifican mediante funciones de modelo, cada una de las cuales utiliza una formula $y - x$. Las variables de respuesta se especifican a la izquierda de la formula y la estructura del modelo se escribe a la derecha.

Por ejemplo, un modelo de tendencia lineal para el PIB per cápita se puede especificar con:

```
TSLM(GDP_per_capita~trend())  
## <TSLM model definition>
```

En este caso, la función del modelo es **TSLM()** (modelo lineal de serie temporal), la variables de respuesta es **GDP_per_capita** y se está modelando mediante **trend()** (una función "especial" que especifica una tendencia lineal cuando se utiliza dentro de **TSLM()**).

Las funciones especiales utilizadas para definir la estructura del modelo varían entre modelos (uya que

cada modelo puede admitir diferentes estructuras). La sección `.Especiales` de la documentación para cada función de modelo enumera estas funciones especiales y como se pueden usar.

Entrenar el modelo (estimación)

Una vez que se especifica un modelo apropiado, luego entrenamos el modelo en algunos datos. Se pueden estimar una o más especificaciones del modelo utilizando la función `model()`. Para estimar el modelo en nuestro ejemplo, usamos

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

## Warning: 7 errors (1 unique) encountered for trend_model
## [7] 0 (non-NA) cases
```

Esto ajusta un modelo de tendencia lineal a los datos del PIB per cápita para cada combinación de variables claves en el `tsibble`. En este ejemplo, ajustará un modelo a cada uno de los 263 países del conjunto de datos. El objeto resultante es una mesa modelo.

```
head(fit, 5)

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

Cada fila corresponde a una combinación de las variables clave. La `trend_model` columna contiene información sobre el modelo ajustado para cada país.

Comprobar el rendimiento del modelo (evaluar)

Una vez que se ha ajustado un modelo, es importante verificar que tan bien se ha desempeñado en los datos. Hay varias herramientas de diagnóstico disponibles para verificar el comportamiento del modelo, y también medidas de precisión que permiten comparar un modelo con otro. Las secciones 5.8 y 5.9 dan más detalles.

Producir pronósticos (pronóstico)

Con un modelo apropiado especificado, estimado y verificado, es hora de producir los pronósticos usando `forecast()`. La forma más fácil de usar esta función es especificando el número de observaciones futuras para pronosticar. Por ejemplo, se pueden generar pronósticos para las próximas 10 observaciones usando $h = 10$. También podemos utilizar el lenguaje natural; por ejemplo, $h = 2\text{years}$ se puede utilizar para predecir 2 años en el futuro.

En otras situaciones, puede ser más conveniente proporcionar un conjunto de datos de períodos de tiempo futuros para pronosticar. Esto suele ser necesario cuando su modelo usa información adicional de los datos, como *regresores exógenos*. Los datos adicionales requeridos por el modelo se pueden incluir en el conjunto de datos de observaciones para pronosticar.

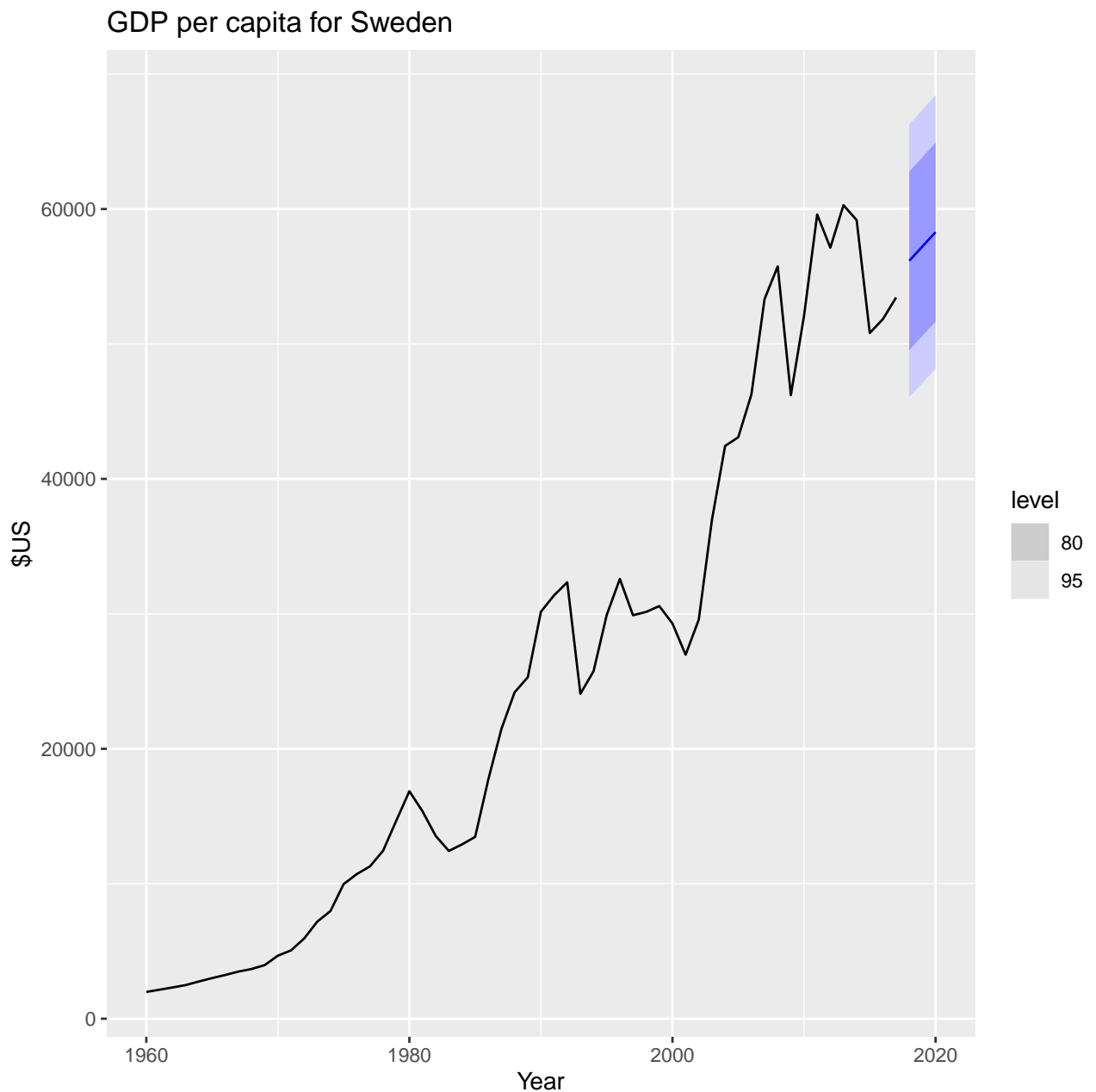
```
fc <- fit %>% forecast(h = "3 years")
```

Esta es una tabla de pronóstico. Cada fila corresponde a un período de pronóstico para cada país. La columna `gDP_per_capita` contiene la distribución del pronóstico, mientras que la columna `.mean`

contiene el pronóstico puntual. El pronóstico puntual es la media (o promedio) de la distribución del pronóstico.

Los pronósticos se pueden trazar junto con los datos históricos de la siguiente manera:

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



5.2 Métodos simples de pronóstico

Algunos métodos de pronóstico son extremadamente simples y sorprendentemente efectivos. Usaremos cuatro métodos simples de pronóstico como puntos de referencia a lo largo de este libro. Para ilustrarlos, utilizaremos la producción trimestral de ladrillos de arcilla de Australia entre 1970 y 2004.

```
head(aus_production, 5)  
  
## # A tsibble: 5 x 7 [1Q]
```

```
## Quarter Beer Tobacco Bricks Cement Electricity Gas
## <qtr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 1956 Q1 284 5225 189 465 3923 5
## 2 1956 Q2 213 5178 204 532 4436 6
## 3 1956 Q3 227 5297 208 561 4806 7
## 4 1956 Q4 308 5681 197 570 4418 6
## 5 1957 Q1 262 5577 187 529 4339 5
```

```
(aus_production %>%
  filter_index("1970 Q1"~"2004 Q4") %>% select(Bricks))
```

```
## # A tibble: 140 x 2 [1Q]
## Bricks Quarter
## <dbl> <qtr>
## 1 386 1970 Q1
## 2 428 1970 Q2
## 3 434 1970 Q3
## 4 417 1970 Q4
## 5 385 1971 Q1
## 6 433 1971 Q2
## 7 453 1971 Q3
## 8 436 1971 Q4
## 9 399 1972 Q1
## 10 461 1972 Q2
## # ... with 130 more rows
```

La función `filter_index()` es una abreviatura conveniente para extraer una sección de una serie de tiempo.

Aquí tendremos todos los modelos que veremos más adelante:

```
#Modelos
brick_fit <- aus_production %>%
  filter(!is.na(Bricks)) %>%
  model(
    Seasonal_naive = SNAIVE(Bricks~lag("year")),
    Naive = NAIVE(Bricks),
    Drift = RW(Bricks ~ drift()),
    Mean = MEAN(Bricks)
  )
#Pronóstico por 5 años
brick_fc <- brick_fit %>%
  forecast(h = "5 years")
```

Método medio

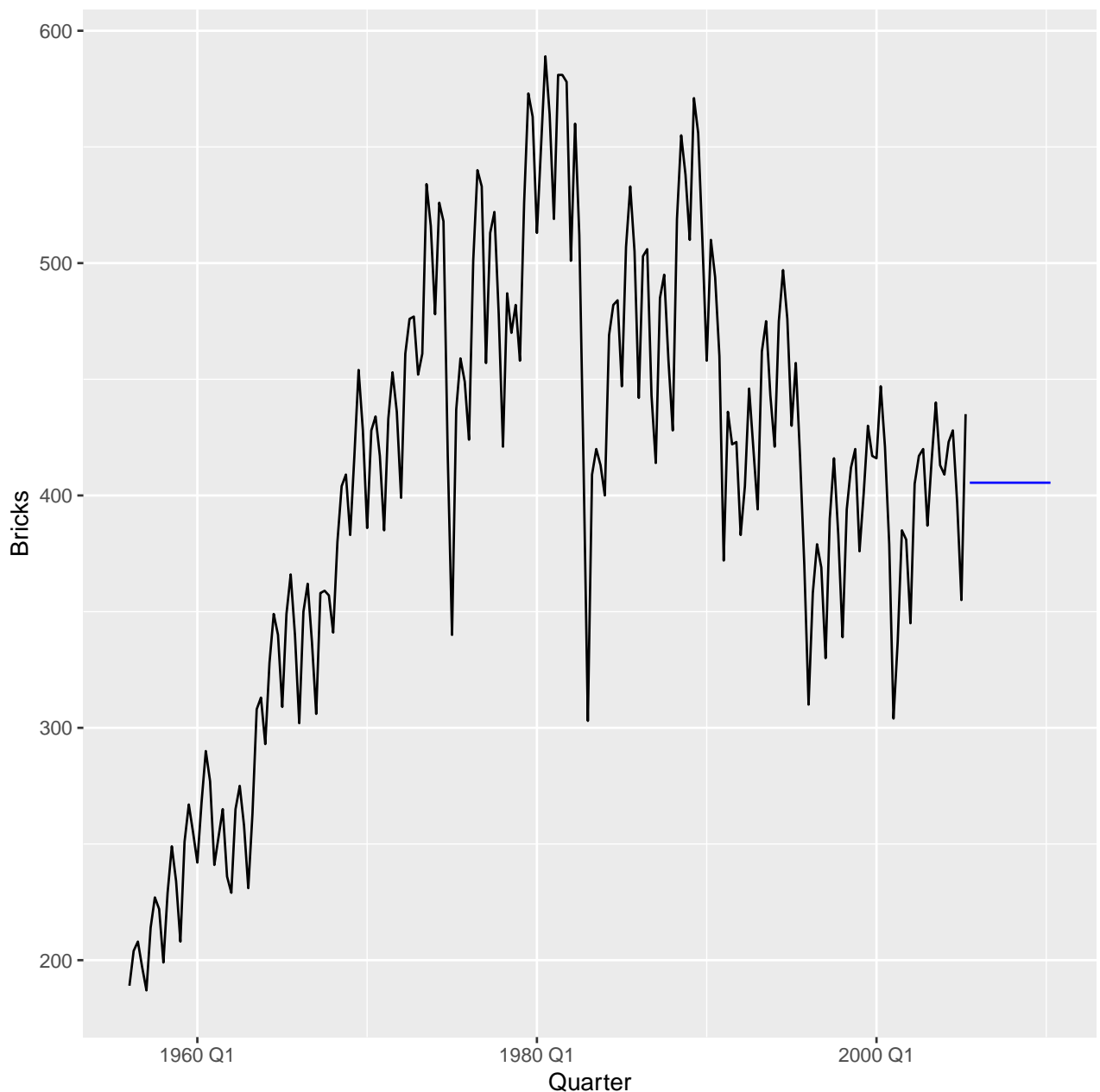
Aquí, los pronósticos de todos los valores futuros son iguales al promedio (o "medio") de los datos históricos. Si denotamos los datos históricos como y_1, \dots, y_T , entonces podemos escribir los pronósticos como :

$$\hat{y}_{T+h|T} = \bar{y} = \frac{y_1, \dots, y_T}{T}$$

La notación $\hat{y}_{T+h|T}$ wa una forma abreviada de la estimación y_{T+h} basada en los datos y_1, \dots, Y_T

```
brick_fc %>%
  filter(.model == "Mean") %>%
  autoplot(aus_production, level = NULL)
```

```
## Warning: Removed 20 row(s) containing missing values (geom_path).
```



Método ingenuo

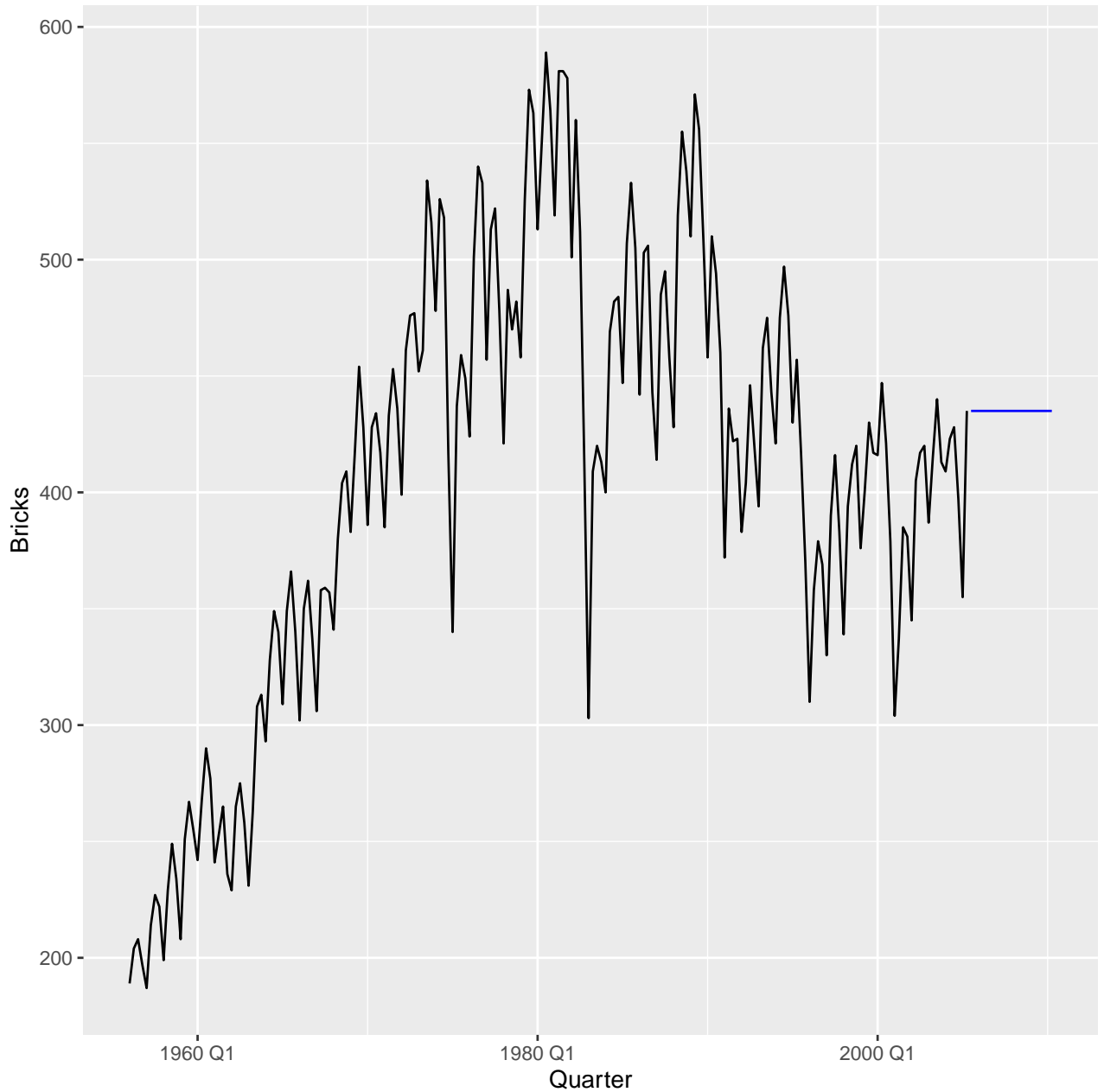
Para los pronósticos ingenuos, simplemente configuramos todos los pronósticos para que sea el valor de la última observación. Es decir:

$$\hat{y}_{T+h|T} = y_T$$

Este método funciona muy bien para muchas series temporales económicas y financieras.

```
brick_fc %>%  
  filter(.model == "Naive") %>%  
  autoplot(aus_production, level = NULL)
```

```
## Warning: Removed 20 row(s) containing missing values (geom_path).
```



Método ingenuo estacional

Un método similar es útil para datos altamente estacionales. En este caso, fijamos cada pronóstico para que sea igual al último valor observado de la misma temporada (por ejemplo, el mismo mes del año anterior). Formalmente, el pronóstico para el tiempo $T + h$ se escribe como :

$$\hat{y}_{T+h|T} = y_{T+h-m(k+1)}$$

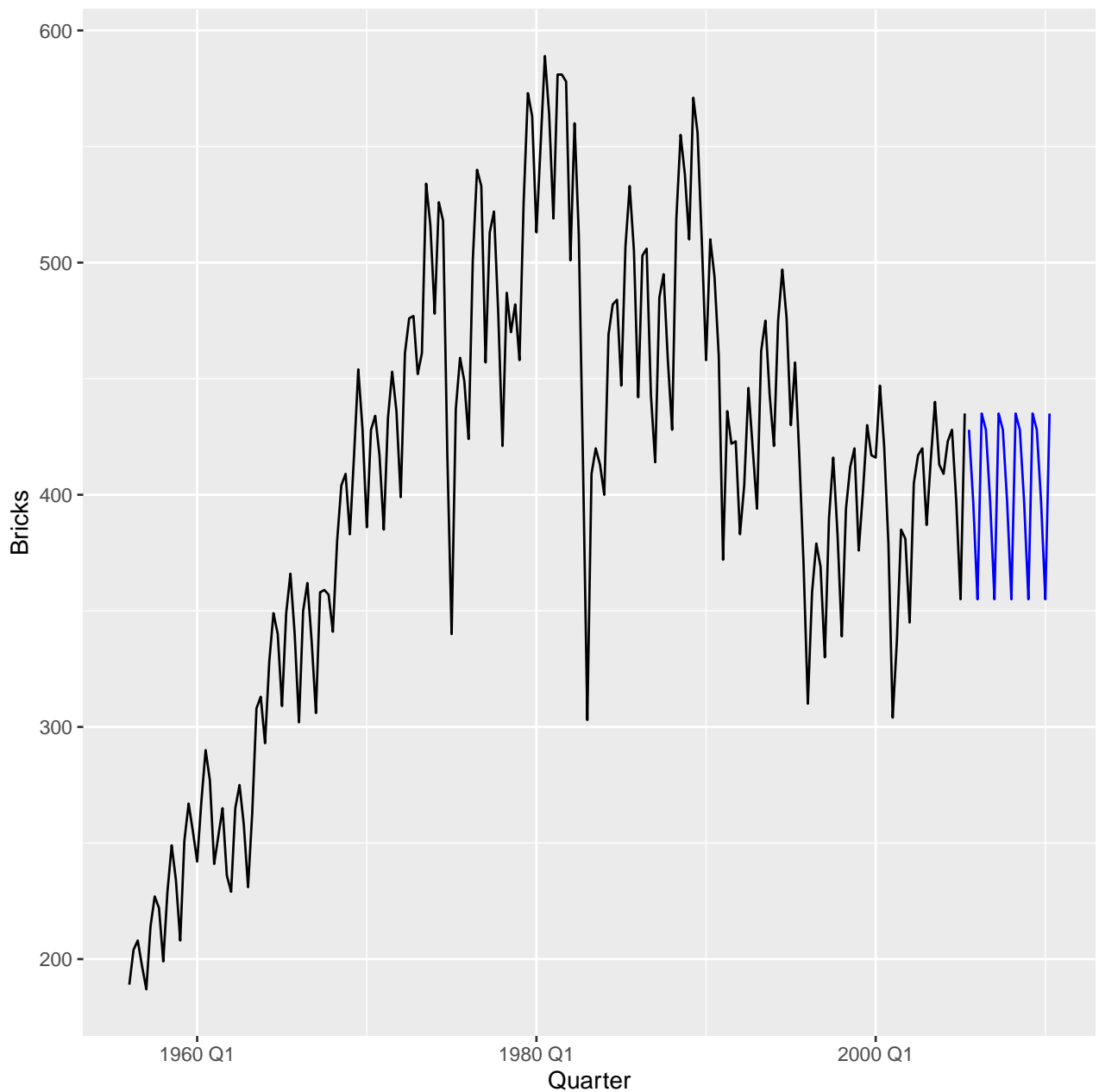
Donde:

- m : Es el período estacional.
- k : Es la parte tenera de $\frac{h-1}{m}$ (es decir, el número de años completos en el período de pronóstico antes del tiempo $T + h$).

Esto parece más complicado de lo que realmente es. Por ejemplo, con datos mensuales, el pronóstico para todos los valores futuros de febrero es igual al último valor de febrero observado- Con datos trimestrales, el pronóstico de todos los valores futuros del segundo trimestre es igual al último valor observado del segundo trimestre . Se aplican reglas similares para otros meses y trimestres, y para otros períodos estacionales.


```
brick_fc %>%
  filter(.model == "Seasonal_naive") %>%
  autoplot(aus_production, level = NULL)

## Warning: Removed 20 row(s) containing missing values (geom_path).
```



La función **lag()** es opcional aquí, al igual que los datos trimestrales **bricks**, por lo que un método ingenuo estacional necesitará un retraso de un año. Sin embargo, para algunas series de tiempo hay más de un período estacional y luego se debe especificar el retraso requerido.

Método de deriva

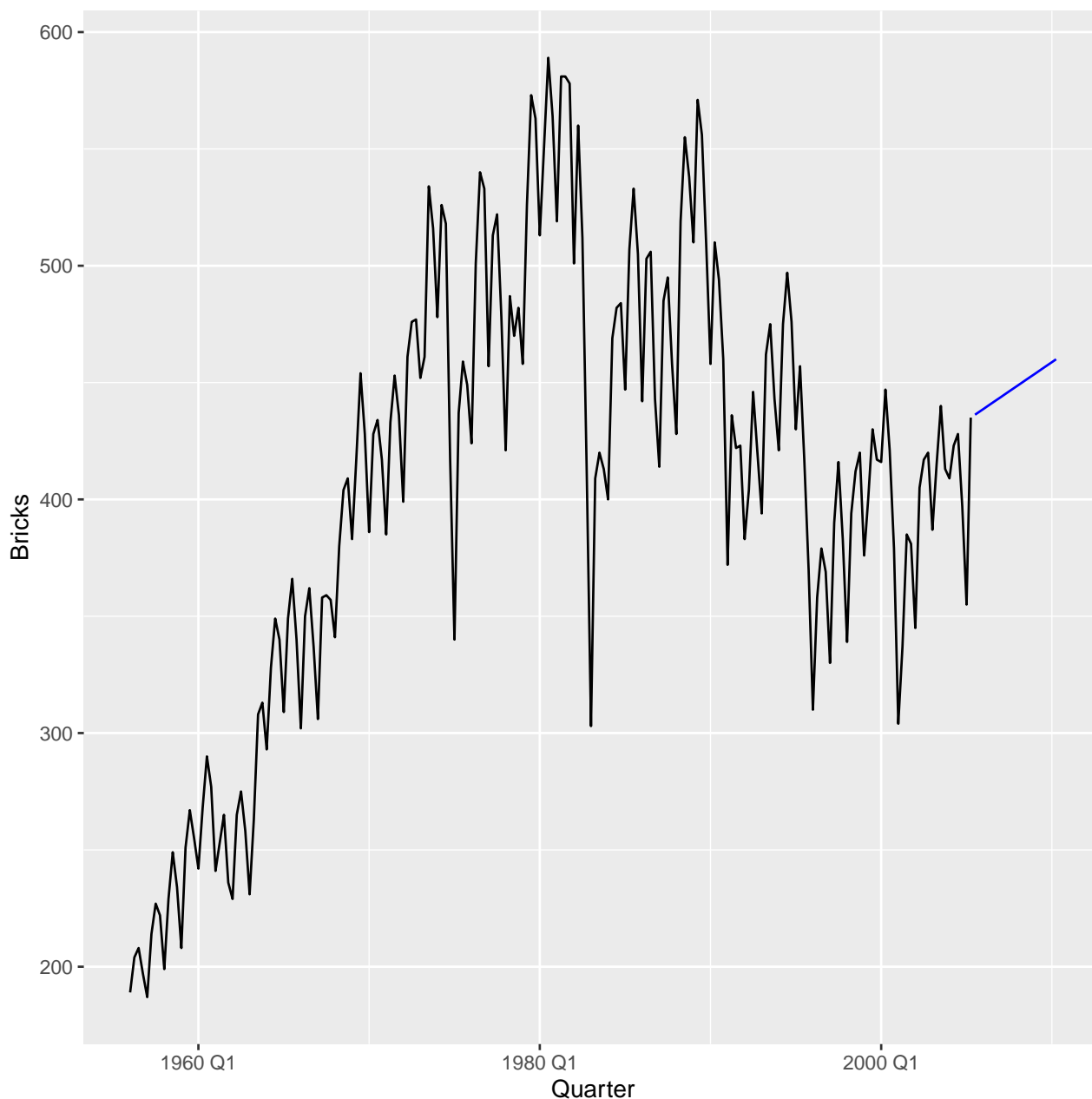
Una variación del método ingenuo es permitir que los pronósticos aumenten o disminuyan con el tiempo, donde la cantidad de cambio con el tiempo (llamada **deriva**) se establece como el cambio promedio observado en los datos históricos. Así, el pronóstico para el tiempo $T + h$ viene dado por:

$$\hat{y}_{T+h|T} = y_T + \frac{h}{T-1} \sum_{t=2}^T + h \left(\frac{y_T - y_1}{T-1} \right)$$

Esto es equivalente a trazar una línea entre la primera y última observación, y extrapolarla al futuro.

```
brick_fc %>%
  filter(.model == "Drift") %>%
  autoplot(aus_production, level = NULL)

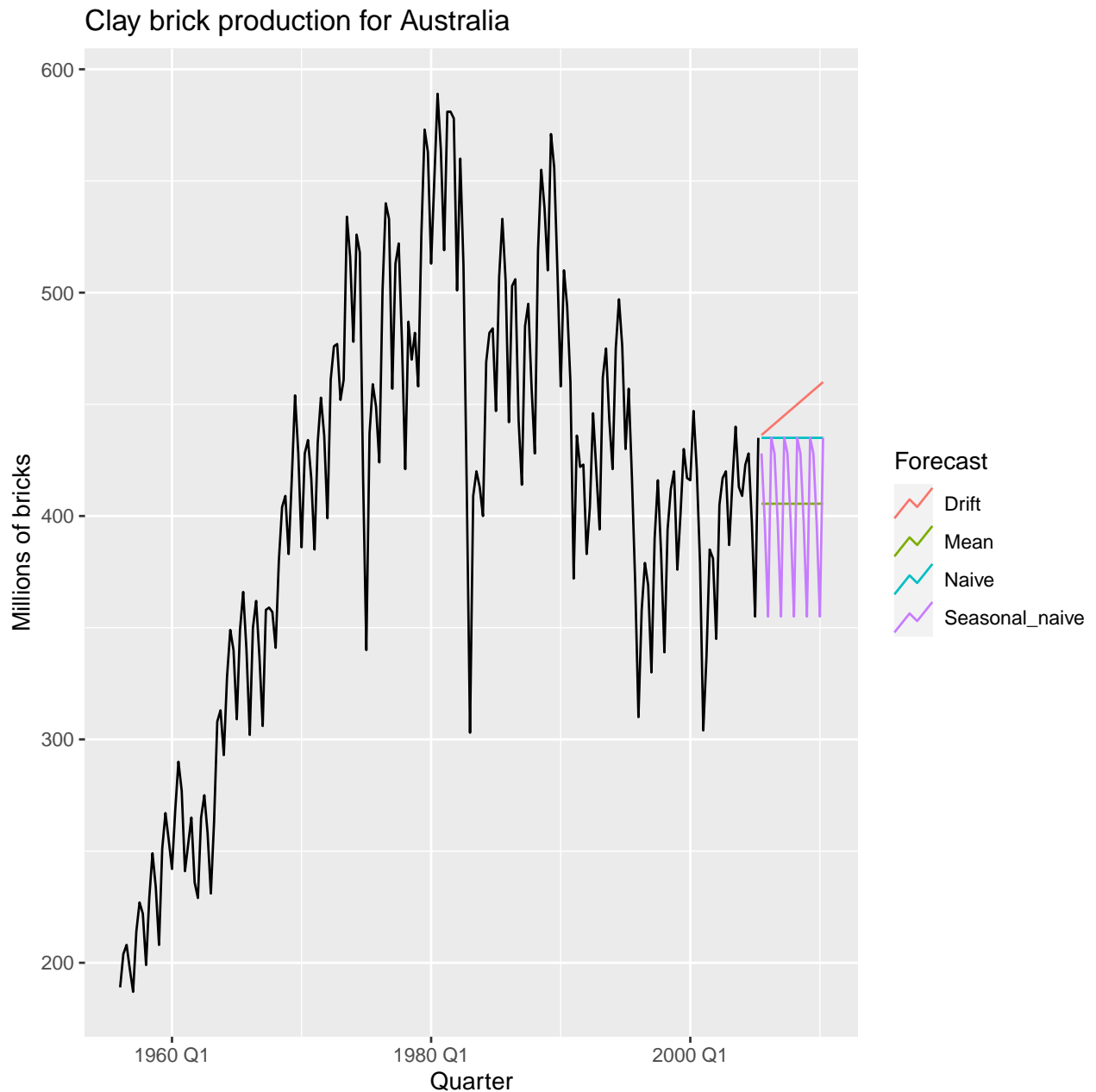
## Warning: Removed 20 row(s) containing missing values (geom_path).
```



La gráfica de los 4 métodos está dada por:

```
brick_fc %>%
  autoplot(aus_production, level = NULL) +
  labs(
    title = "Clay brick production for Australia",
    y = "Millions of bricks"
  ) +
  guides(colour = guide_legend(title = "Forecast"))

## Warning: Removed 20 row(s) containing missing values (geom_path).
```



Ejemplo: Producción de cerveza trimestral de Australia

La siguiente figura muestra los tres primeros métodos aplicados a la producción trimestral de cerveza en Australia desde 1992 hasta 2006, con las previsiones comparadas con los valores reales en los próximos 3.5 años.

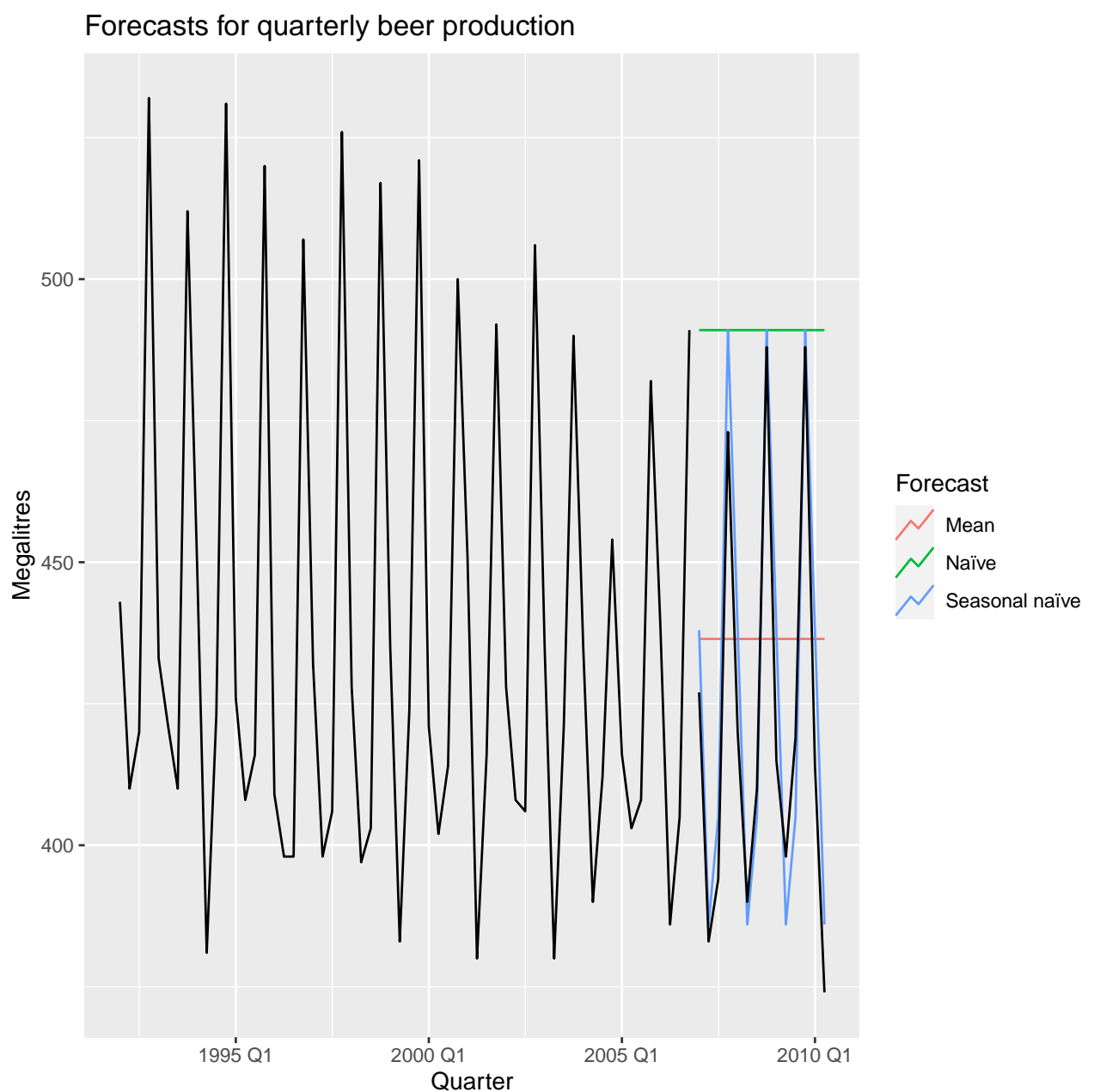
```
# Establecer datos de entrenamiento de 1992 a 2006.
train <- aus_production %>%
  filter_index("1992 Q1" ~ "2006 Q4")
# Ajustando el modelo.
beer_fit <- train %>%
  model(
    Mean = MEAN(Beer),
    `Naïve` = NAIVE(Beer),
    `Seasonal naïve` = SNAIVE(Beer)
  )
# Generando pronóstico para 14 trimestres.
```

```

beer_fc <- beer_fit %>% forecast(h = 14)
# Graficando pronóstico contra los valores reales.
beer_fc %>%
  autoplot(train, level = NULL) +
  autolayer(
    filter_index(aus_production, "2007 Q1" ~ .),
    colour = "black"
  ) +
  labs(
    y = "Megalitres",
    title = "Forecasts for quarterly beer production"
  ) +
  guides(colour = guide_legend(title = "Forecast"))

## Plot variable not specified, automatically selected '.vars = Beer'

```



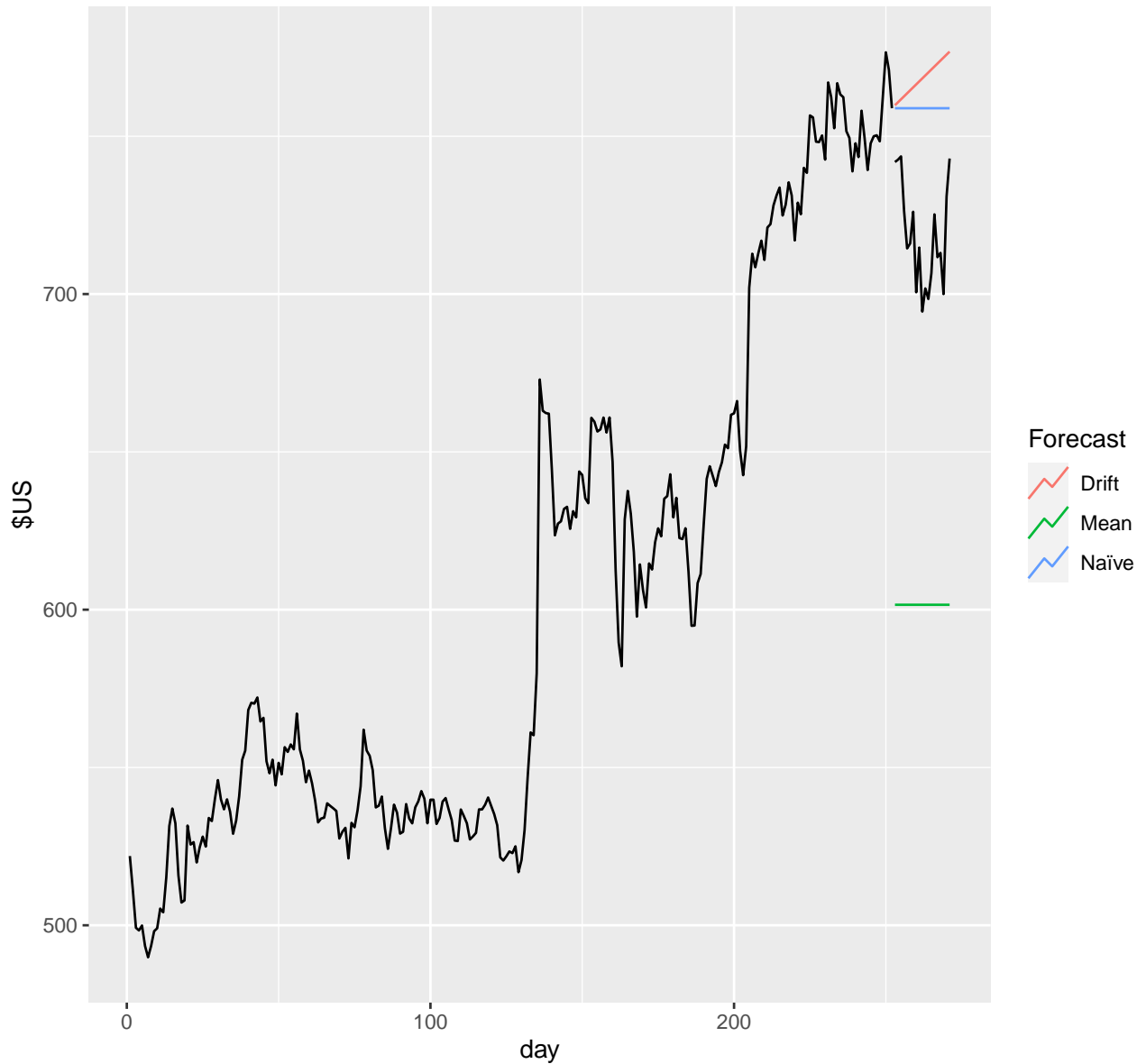
En este caso, solo los pronósticos ingenuos estacionales se acercan a los valores observados a partir de 2007

Ejemplo: Precio de cierre diario de las acciones de Google

En la siguiente figura, los métodos NO estacionales se aplican al precio de cierre diario de las acciones de Google en 2015 y se usan para pronosticar con un mes de anticipación. Debido a que los precios de las acciones no se observan todos los días, primero configuramos un nuevo índice de tiempo basado en los días de negociación en lugar de los días calendario.

```
#Volver a indexar según el día de negociación.
google_stock <- gafa_stock %>%
  filter(Symbol == "GOOG", year(Date) >= 2015) %>%
  mutate(day = row_number()) %>%
  update_tsibble(index = day, regular = TRUE)
# Filtrando conforme al año de interés.
google_2015 <- google_stock %>% filter(year(Date) == 2015)
#Ajustando los modelos
google_fit <- google_2015 %>%
  model(
    Mean = MEAN(Close),
    `Naïve` = NAIVE(Close),
    Drift = NAIVE(Close ~ drift())
  )
# Producir pronóstico para los días hábiles en enero de 2016
google_jan_2016 <- google_stock %>%
  filter(yearmonth(Date) == yearmonth("2016 Jan"))
google_fc <- google_fit %>%
  forecast(new_data = google_jan_2016)
# Plot the forecasts
google_fc %>%
  autoplot(google_2015, level = NULL) +
  autolayer(google_jan_2016, Close, colour = "black") +
  labs(y = "$US",
       title = "Google daily closing stock prices",
       subtitle = "(Jan 2015 - Jan 2016)") +
  guides(colour = guide_legend(title = "Forecast"))
```

Google daily closing stock prices
(Jan 2015 – Jan 2016)



A veces, uno de estos métodos simples será el mejor método de pronóstico disponible; pero en muchos casos, estos métodos servirán como puntos de referencia en lugar del método de elección. Es decir, cualquier método de pronóstico que desarrollemos se comparará con estos métodos simples para garantizar que el nuevo método sea mejor que estas alternativas simples. Si no, no vale la pena considerar el nuevo método.