7. Modelos de regresión de series de tiempo.

El concepto es que pronosticamos la serie de tiempo de interés y asumiendo que tiene una relación lineal con otras series de tiempo x.
Por ejemplo, podríamos desear pronosticar las ventas mensuales utilizando el gasto total en publicidad como predictor. O podemos pronosticar la demanda diaria de electricidad y usando la temperatura x_1 y el día de la semana x_2 como predictores

Las **variables de pronóstico** y a veces también se denomina variable de regreso, dependiente o explicada. Las **variables predictoras** a veces también se denominan regresores, variables independientes o explicativas. En este libro siempre nos referiremos a ellas como variable de "pronósticoz las variables "predictoras".

7.1 El modelo lineal

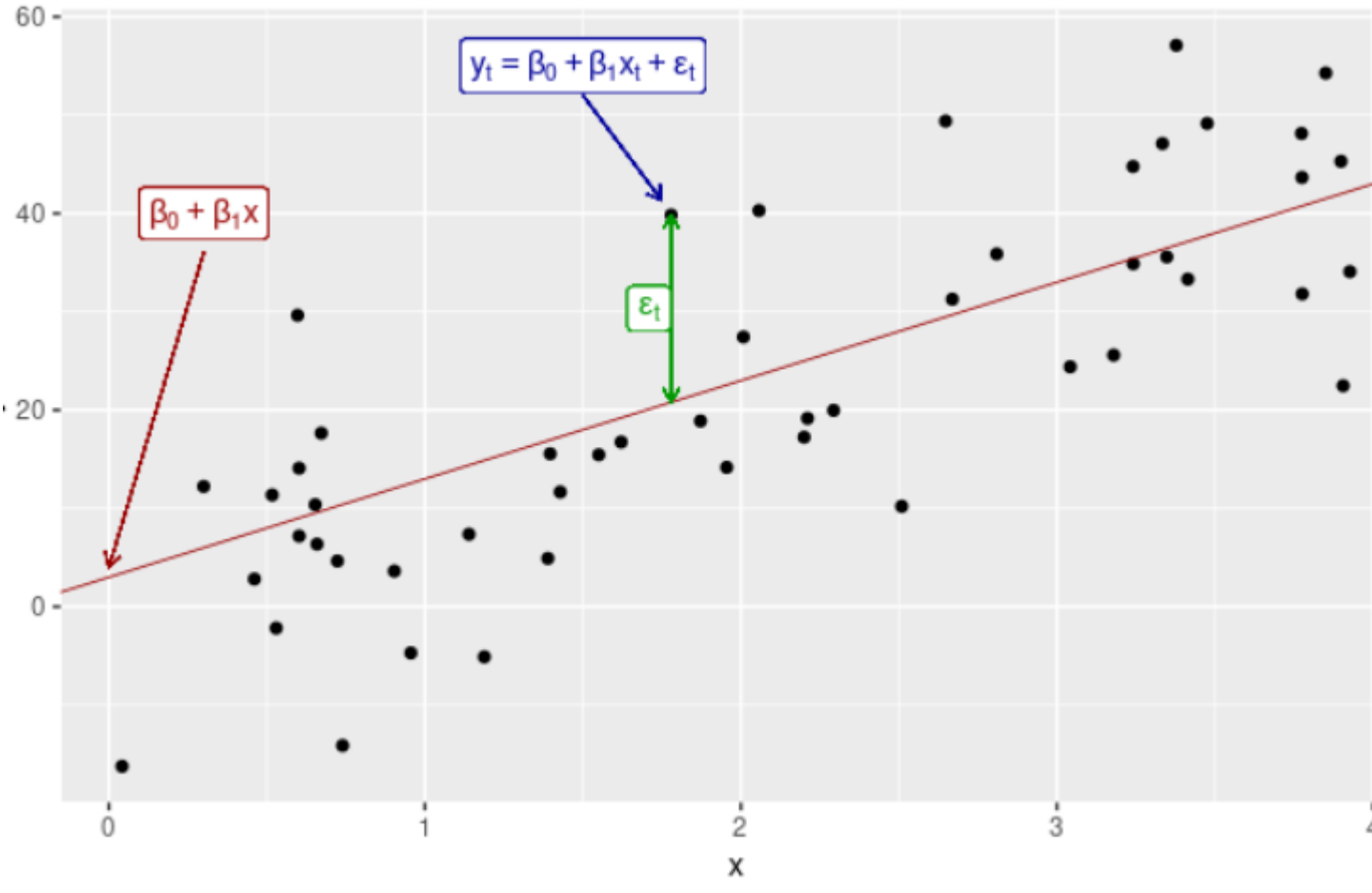
Regresión lineal simple

En el caso más simple, el modelo de regresión permite una relación lineal entre la variable de pronóstico y y una sola variable predictora x:

$$y_t = \beta_0 + \beta_1 x_1 + \varepsilon_t$$

En la siguiente figura se muestra un ejemplo artificial de datos de dicho modelo. Los coeficientes β_0 y β_1 denotan intersección y la pendiente de la línea respectivamente.

- El intercepto β_0 representa el valor pronosticado de y cuando $x = 0$. La pendiente
- La pendiente β_1 representa el cambio promedio pronosticado en y resultante de un aumento de una unidad en x.



Observe que las observaciones no se encuentran en la línea recta, sino que están dispersas a su alrededor. Podemos pensar que cada observación y_t consiste en la parte sistemática o explicada del modelo, $\beta_0 + \beta_1 x_1$, y el .errorrandom.aleatorio, ε_t. El término .errorrandom"no implica un error, sino una desviación del modelo de línea recta subyacente. Captura cualquier cosa que pueda afectar a y_t que no sea x_t.

Ejemplo: Datos de consumo de EE:UU

La sigueinte figura muestra series de tiempo de cambos porcentuales trimestrales(tasas de crecimiento) del gasto de consumo personal real, y_t, y el ingreso disponible personal real, x_t, para EE.UU, desde el primer trimestre de 1970 hasta el segundo trimestre de 2019

```
library(fpp3)
```

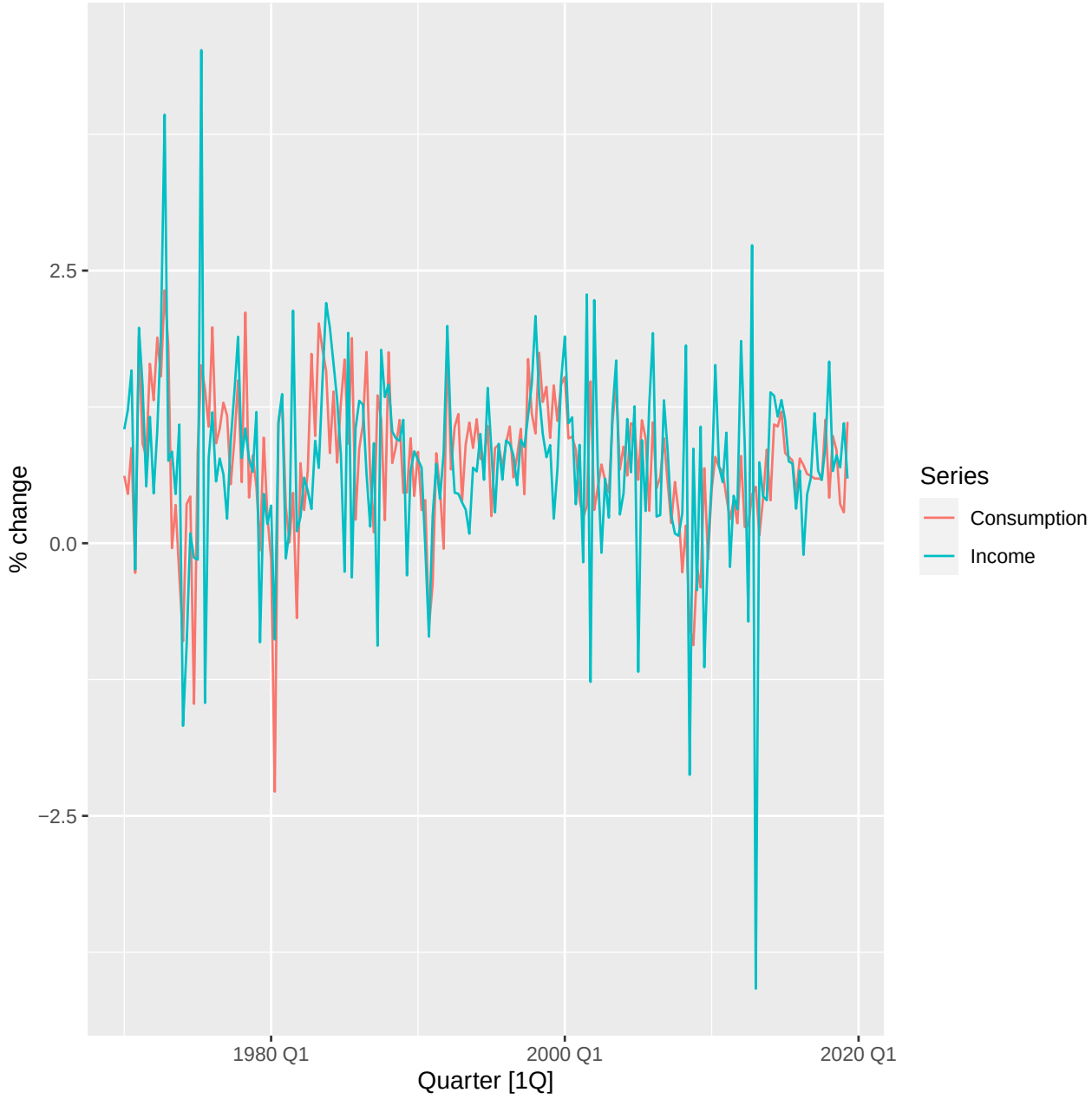
```
## -- Attaching packages --------------------- fpp3 0.4.0 -
## v tibble      3.1.6     v tsibble     1.1.1
## v dplyr       1.0.8     v tsibbledata 0.4.0
## v tidyr       1.2.0     v feasts      0.2.2
## v lubridate   1.8.0     v fable       0.3.1
## v ggplot2     3.3.5
## -- Conflicts ------------------------ 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(us_change, 5)
```

```
## # A tsibble: 5 x 6 [1Q]
##   Quarter Consumption Income Production Savings Unemployment
##     <qtr>       <dbl>  <dbl>      <dbl>   <dbl>        <dbl>
## 1 1970 Q1       0.619  1.04       -2.45    5.30         0.9
## 2 1970 Q2       0.452  1.23       -0.551   7.79         0.5
## 3 1970 Q3       0.873  1.59       -0.359   7.40         0.5
## 4 1970 Q4      -0.272 -0.240      -2.19    1.17         0.700
## 5 1971 Q1       1.90   1.98        1.91    3.54        -0.100
```

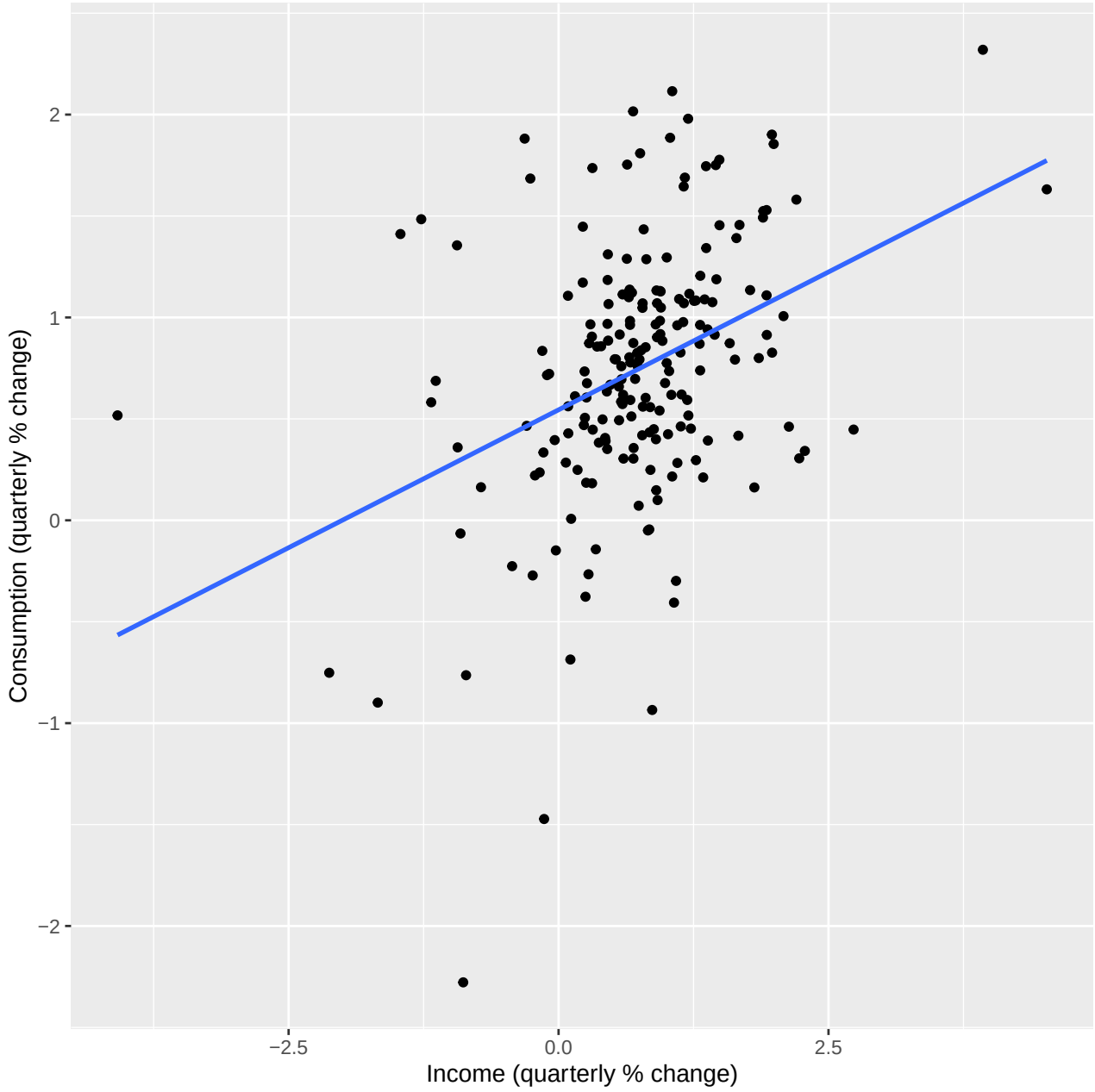
Gráfica del consumo trimestral en Estados unidos

```
us_change %>%
  pivot_longer(c(Consumption, Income), names_to = "Series") %>%
  autoplot(value) +
  labs(y = "% change")
```



(Podemos un "sombrero"sobre y para indicar que este es el valor de y predicho por el modelo).

```
us_change %>%
  ggplot(aes(x = Income, y = Consumption)) +
  labs(
    y = "Consumption (quarterly % change)",
    x = "Income (quarterly % change)"
  ) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)

## `geom_smooth()` using formula 'y ~ x'
```



La ecuación se estima usando la función **TSLM()**.

```
fit_cons <- us_change %>%
  model(lm = TSLM(Consumption ~ Income))
report(fit_cons)

## Series: Consumption
## Model: TSLM
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.58236 -0.27777  0.01862  0.32330  1.42229
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.54454    0.05403  10.079  < 2e-16 ***
## Income       0.27183    0.04673   5.817  2.4e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.5905 on 196 degrees of freedom
## Multiple R-squared: 0.1472,Adjusted R-squared: 0.1429
## F-statistic: 33.84 on 1 and 196 DF, p-value: 2.4022e-08
```

La línea ajustada tiene una pendiente positiva, lo que refleja la relación positiva entre el ingreso (income) y el consumo (consumption). El coeficiente de pendiente muestra que un aumento de una unidad den x (un aumento de 1 punto porcentual en el ingreso personal disponible) da como reslutado un aumento promedio de 0.27 unidades en y (un aumento promedio de 0.27 puntos porcentuales en el consumo personal de gasto). Alternativamente, la ecuación estimada muestra que un valor de 1 para x (el aumento personal en el ingreso personal disponible) dará como resultado un valor de pronóstico de $0..54 + 0.27(1) = 0.82$ para y (El porcentaje de aumento en el gasto de consumo personal).

La interpretación del intercepto requiere que un valor de $x = 0$ tenga sentido. En este caso, cuando $x = 0$ (es decir, cuando no hay cambios en el ingreso personal disponible desde el último trimestre), el valor pronosticado de y es 0.54 (es decir, un aumento promedio en el gasto de consumo personal de 0.54 %) Incluso cuando $x = 0$ no tiene sentido, el intercepto es una parte importante del modelo. Sin el, el coeficiente de pendiente puede distorsionarse innecesariamente. La intersección simpre debe incluir a menos que el requisito sea forzar la línea de regresión .ª través del origen". En lo que sigue asumimos que siempre se incluye un intercepto en el modelo.

Regresión lineal múltiple

.Cuando hay dos o más variables predictoras, el modelo se denomina **modelo de regresión múltiple.** La forma general de un modelo de regresión múltiple es

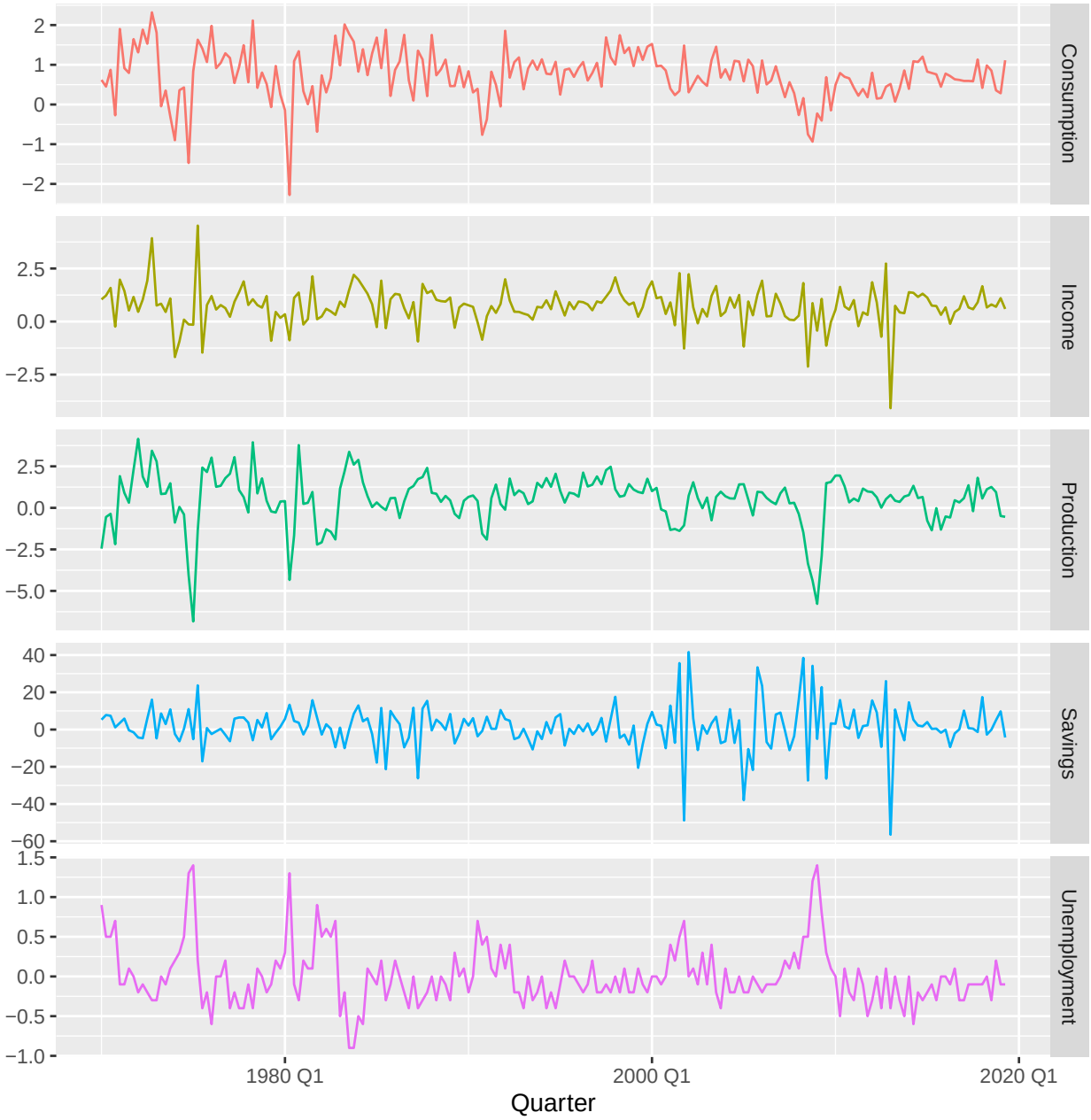
$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \cdots + \beta_k x_{k,t} + \varepsilon_t$$

donde y es la variable a pronosticar y $x_1, \ldots, x_k$ son las k variables predictoras. Cada una de las variables debe ser númerica. Los coeficientes $\beta_1, \ldots, \beta_k$ miden el efecto de cada predictor después de tener en cuenta los efectos de todos los demás predictores del modelo. Así, los coeficientes miden los **efectos marginales.**

Ejemplo: Gasto de consumo de EE.UU

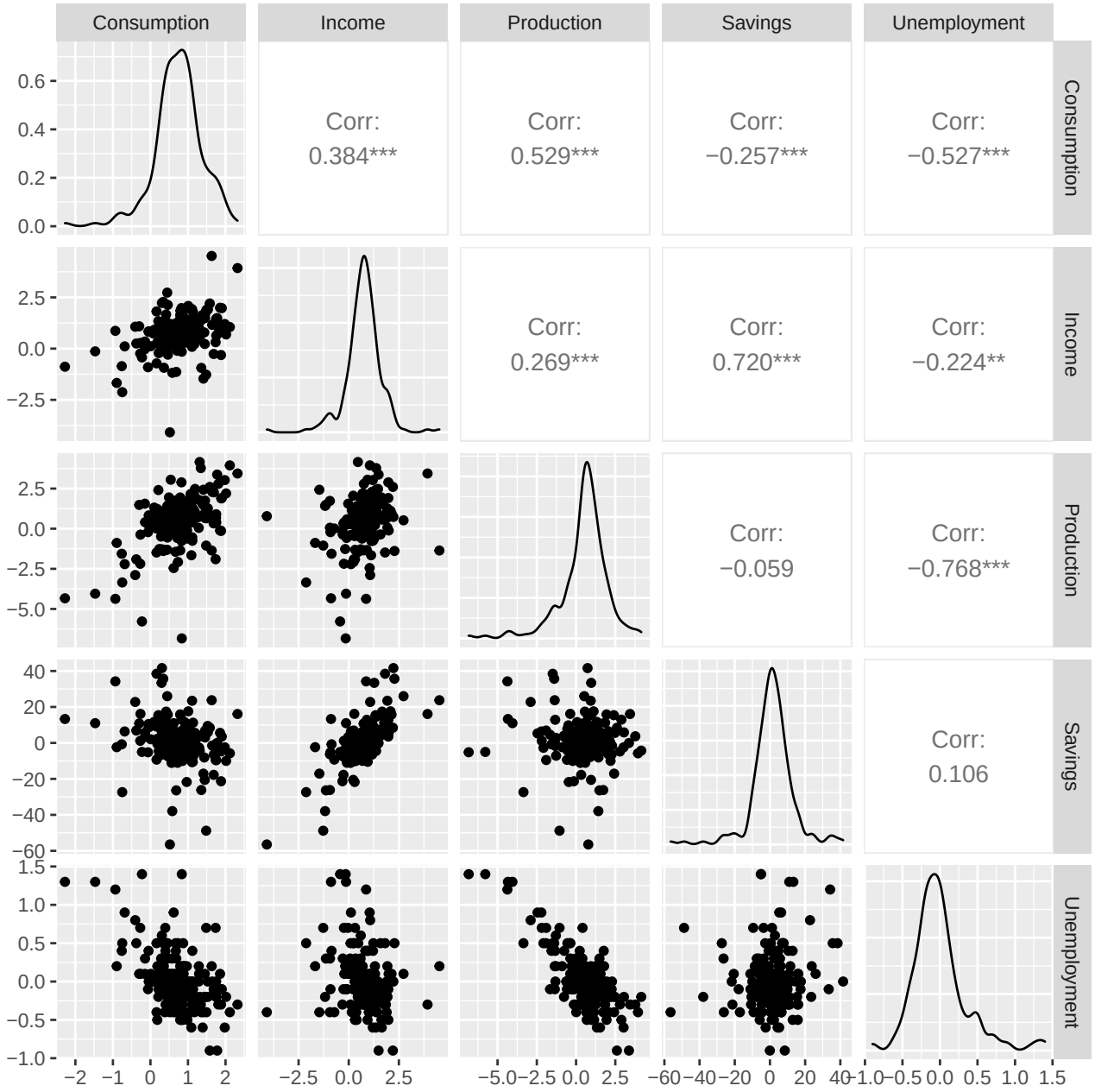
La siguiente figura muestra predictores adicionales que pueden ser útiles para pronosticar el gasto de consumo de $EE.UU$. Estos son cambios porcentualres trimestrales en la producción trimestral y el ahorro personal, y cambios trimestrales en la tasa de desempleo (que ya es un porcentaje). La construcción de un modelo de regresión lineal múltiple puede potencialmente generar pronósticos más precisos, ya que esperemos que el gasto de consumo NO solo dependa del ingreso personal sino también de otros predictores.

```
us_change %>%
  pivot_longer(-Quarter, names_to = "Measure", values_to = "Change") %>%
  ggplot(aes(x = Quarter, y = Change, colour = Measure)) +
  geom_line() +
  facet_grid(vars(Measure), scales = "free_y") +
  labs(y = "") +
  guides(colour = "none")
```



La siguiente figura es una **matriz de diagrama de dispersión (scatterplot) de cinco variables**. La primera columna muestra las relaciones entre la variable de pronóstico (consumo) y cada uno de los predictores. Los diagramas de dispersión muestran las relaciones positivas con el ingreso (income) y la producción industrial (Production), y las relaciones negativas con el ahorro () y el desempleo. La fuerza de estas relaciones se muestra mediante los coeficientes de correlación en la primera fila . Los diagramas de dispersión y los coeficientes de correlaciones restantes muestran las relaciones entre los predictores.

```
us_change %>%
  GGally::ggpairs(columns = 2:6)

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```



Suposiciones

Cuando usamos un modelo de regresión lineal, implícitamente hacemos algunas suposiciones sobre las variables en la ecuación de regresión lineal múltiple.

1. Asumimos que el modelo es una aproximación razonable a la realidad; es decir, la relación entre la variable de pronóstico y las variables predictoras satisface esta ecuación lineal.

2. Hacemos las siguientes suposiciones sobre los errores $(\varepsilon_1, \ldots, \varepsilon_T)$:

   - Tienen media cero: de lo contrario, los pronósticos estarán sistemáticamente sesgados.
   - No están correlacionados; de lo contrario, los pronósticos serán ineficientes, ya que hay más información en los datos que se puede explotar.
   - No están relacionados con las variables predictoras; de lo contrario, habría más infomación que debería incluirse en la parte sistemática del modelo.

   También es útil que los errores se distribuyen normalmente con una varinza constante para producir fácilmente intervalos de predicción. $\varepsilon_t \sim N(0, \sigma^2)$

Otra suposición importante en el modelo de regresión lineal es que cada predictor x no es una variable aleatoria. Si estuviéramos realizando un experimento controlado en un laboratorio, podríamos controlar los valores resultantes de x (para que no fueran aleatorios) y observar los valores resultantes de y. Con los datos observacionales (incluyendo la mayoría de los datos en negocio y econmía), no es posible controlar el valor de x simplemente lo observamos. Por lo tanto hacemos de esto una suposición.

7.2 Estimación por mínimos cuadrados

En la práctica, por supuesto, tenemos una colección de observacones pero no conocemos los valores de los coeficientes $\beta_0, \beta_1, \ldots, \beta_k$. Estos deben estimarse a partir de los datos.
El principio de los mínimos cuadrados proporciona una forma de elegir los coeficientes de manera efectiva al minimizar la suma de los errores al cuadrado. Es decir, elegimos los valores de $\beta_0, \beta_1, \ldots, \beta_k$ que minimizan

$$\sum_{t=1}^{T} \varepsilon_t^2 = \sum_{t=1}^{T} (y_t - \beta_0 - \beta_1 x_{1,t} - \beta_2 x_{2,t} - \cdots - \beta_k x_{k.t})^2$$

Esto se llama estimación de **mínimos cuadrados** porque da el valor mínimo para la suma de los errores al cuadrado. Encontrar las mejores estimaciones de los coeficientes a menudo se denomina .ªjustar.el modelo a los datos o, a veces, .ªprender.o .entrenar.el modelo.
Cuando nos referimos a los coeficientes **estimados**, usaremos la notación $\hat{\beta}_1, \ldots, \hat{\beta}_k$.

La función **TSLM()** ajusta un modelo de regresión lineal a los datos de series temporales. Es similar a la función **lm()** que se usa ampliamente para los modelos lienales, pero **TSLM()** proporciona funciones adicionales para el manejor de series de tiempo.

Ejemplo: Gasto de consumo de EE.UU

Un modelo de regresión lineal múltiple para el consumo de $EE.UU$ es:

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_1 x_{2,t} + \beta_3 x_{3,x} + \beta_4 x_{4,t} + \varepsilon_t$$

Donde

- y es el cambio porcentual en el gasto de consumo personal real.
- x_1 es el cambio porcentual en el ingreso disponible personal real.
- x_2 es el cambio porcentual en la producción industrial.
- x_3 es el cambio porcentual en los ahorros personales.
- x_4 es el cambio en la tasa de desempleo.

El siguiente resultado proporciona información sobre el modelo ajustado

1. La primera columna de **Coefficients** da una estimación de cada β.
2. La segunda columna da su error estándar (es decir, la desviación estándar que se obtendría al estimar repetidamente los β coeficientes en datos similares).
   El error estándar da una medida de la incertidumbre en la estimación β coeficiente.

```
fit_consMR <- us_change %>%
  model(lm = TSLM(Consumption ~ Income + Production + Unemployment + Savings))
report(fit_consMR)
```

```
## Series: Consumption
## Model: TSLM
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.90555 -0.15821 -0.03608  0.13618  1.15471
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.253105   0.034470   7.343 5.71e-12 ***
## Income        0.740583   0.040115  18.461  < 2e-16 ***
## Production    0.047173   0.023142   2.038   0.0429 *
## Unemployment -0.174685   0.095511  -1.829   0.0689 .
## Savings      -0.052890   0.002924 -18.088  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3102 on 193 degrees of freedom
## Multiple R-squared: 0.7683,Adjusted R-squared: 0.7635
## F-statistic:   160 on 4 and 193 DF, p-value: < 2.22e-16
```

A efectos de previsión, las dos columnas finales tienen un interés limitado. El "t value.[es] la relación entre un coeficiente βestimado y su error estándar y la última columna da el "p-value": la probabilidad de que el coeficiente estimado β sea tan grande como si no existiera una relación real entre el consumo y el predictor correspondiente. Es to es útil cuando se estudia el efecto de cada predictor pero no es particularmente útil para la previsión.

Valores ajustados

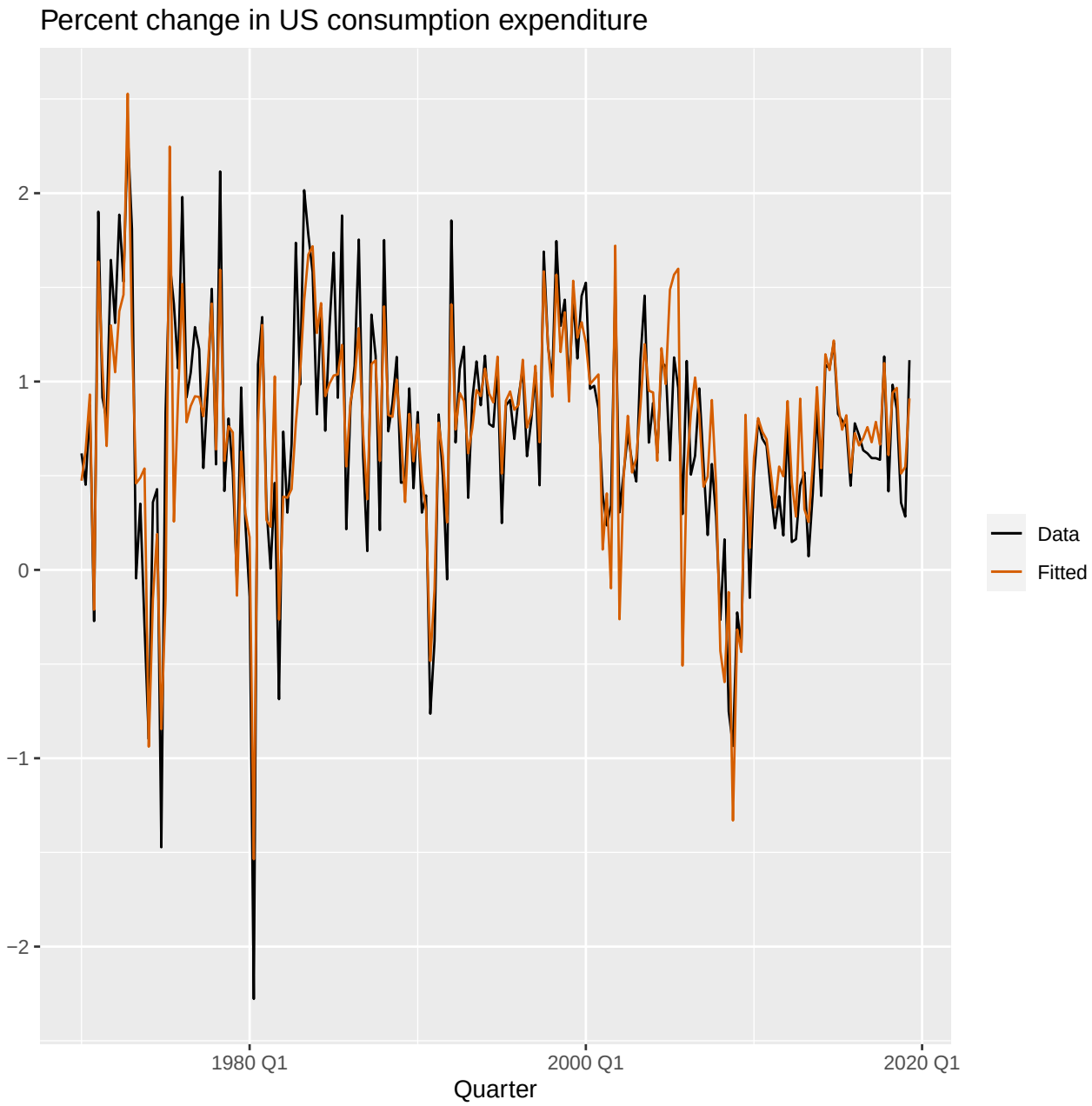
Las predicciones de y se pueden obtener utilizando los coeficientes estimados en la ecuación de regresión y estableciendo el término de error en 0. En general escribimos:

$$\hat{y}_t = \hat{\beta}_0 + \hat{\beta}_1 x_{1,t} + \hat{\beta}_2 x_{2,t} + \cdots + \hat{\beta}_k x_{k,t}$$

Sustituyendo los valores de $x_{1,t}, \ldots, x_{k,t}$ para $t = 1, \ldots, T$ devuelve predicciones de y_t dentro del conjunto de entrenamiento, denominados **valores ajustados.** Tenga en cuenta que estas son predicciones de los datos utilizados para estimar el modelo, NO pronósticomos genuinos de valores futuros de y.

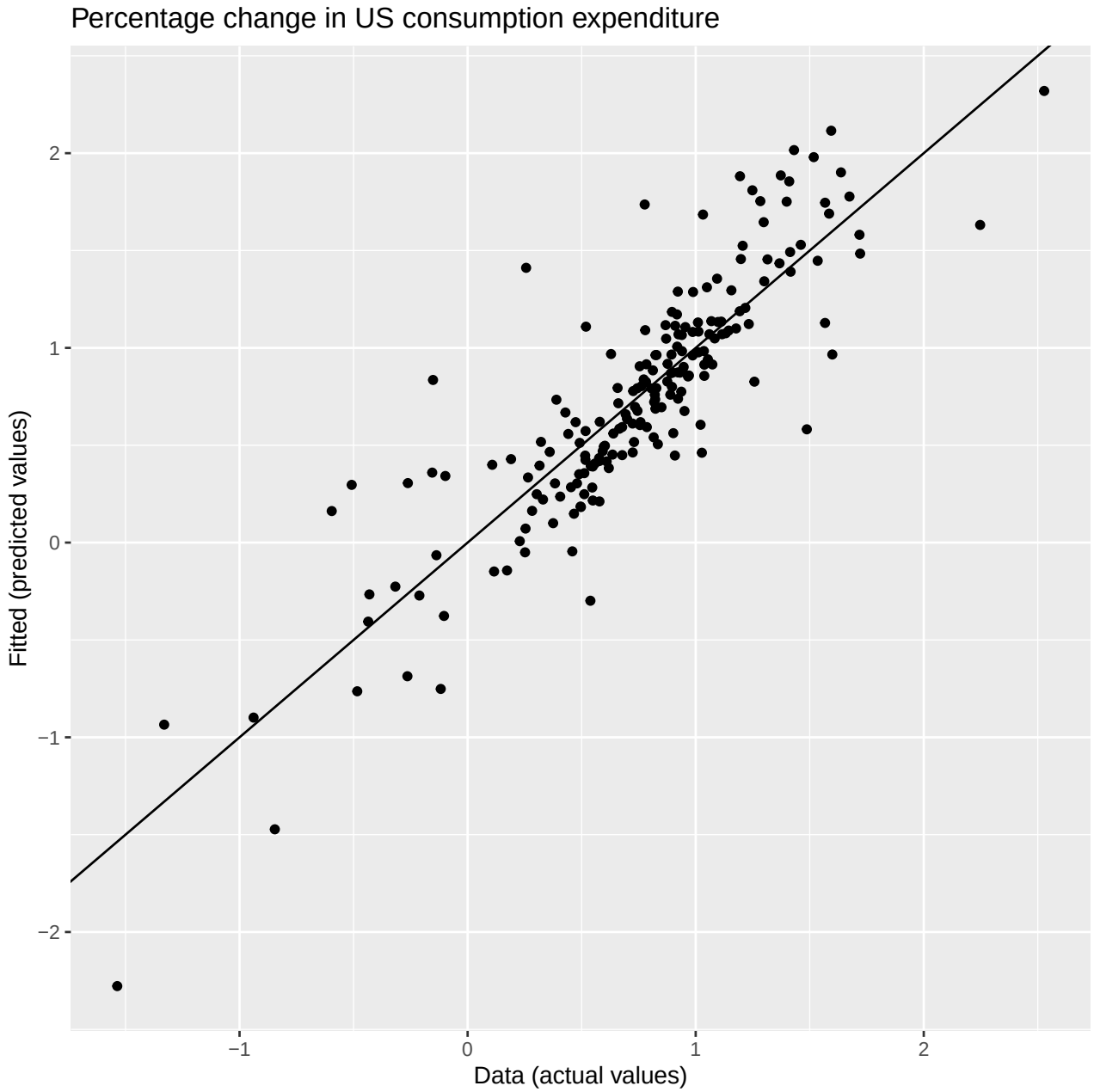
Los siguiente gráficos muestran los valores reales comparados con los valores ajustados para el cambio porcentual en la serie de gastos de consumo de $EE.UU$. El siguiente gráfico muestra que los valores ajustados siguen bastante de cerca a los datos reales.

```
augment(fit_consMR) %>%
  ggplot(aes(x = Quarter)) +
  geom_line(aes(y = Consumption, colour = "Data")) +
  geom_line(aes(y = .fitted, colour = "Fitted")) +
  labs(
    y = NULL,
    title = "Percent change in US consumption expenditure"
  ) +
  scale_colour_manual(values = c(Data = "black", Fitted = "#D55E00")) +
  guides(colour = guide_legend(title = NULL))
```



La siguiente gráfica verifica mediante la fuerte relación positiva que muestra el diagrama de dispersión.

```
augment(fit_consMR) %>%
  ggplot(aes(x = .fitted, y = Consumption)) +
  geom_point() +
  labs(
    y = "Fitted (predicted values)",
    x = "Data (actual values)",
    title = "Percentage change in US consumption expenditure"
  ) +
  geom_abline(intercept = 0, slope = 1)
```



Bondad de ajuste

Una forma común de resumir que tan bien se ajusta un modelo de regresión lineal a los datos es a través del coeficiente de determinación, R^2. Esto se puede calcular como el cuadrado de la correlación entre los valores observados y y los valores de predición $\hat{y}$.
Alternativamente, también se puede calcular como :

$$R^2 = \frac{\sum(\hat{y}_t - \bar{y})^2}{\sum(y_t - \hat{y})^2}$$

Donde la suma es sobre todas las observaciones. Por lo tanto, refleja la proporción de variación en la variable de pronóstico que se explica por el modelo de regresión.

En la regresión lineal simple, el valor de R^2 es igual al cuadrado de la correlación de entre y y x (siempre que se haya incluido una intersección).

Si las predicciones están cerca de los valores reales, esperaríamos que R^2 estuviera cerca de 1. Por otro lado, si las predicciones no están relacionadas con los valores reales entonces $R^2 = 0$ (nuevamente,

asumiendo que hay una intersección). En todos los casos, R^2 se encuentra entre 0 y 1.

El R^2 se usa con frecuencia, aunque a menudo de forma incorrecta, en la predicción. El valor de R^2 nunca disminuirá al agregar un predictor adicional al modelo y esto puede provocar un ajuste excesivo. No hay reglas fijas sobre lo que es un buen R^2, y los valores típicos de R^2 dependen del tipo de datos utilizados. Validar el rendimiento de pronóstico de un modelo en los datos de prueba es mucho mejor que medir el R^2 en los datos de entrenamiento.

Ejemplo: Gatos de consumo de EE.UU

La anterior figura traza los valores de gatos de consumo real frente a los valores ajustados. La correlación entre estas variables es, $\varphi = 0.877$ por lo tanto $R^2 = 0.768$ (que se muestra en el resultado anterior). En este caso el modelo hace un excelente trabajo ya que explica el 76.8 % de la variación en los datos de consumo. Compare eso con el R^2 de 0.15 obtenido de la regresión simple con el mismo conjunto de datos. Agregar los tres predictores adicionales ha permitido explicar mucho más la variación en los datos del consumo.

Error estándar de la regresión.

Otra medida de qué tan bien se ha ajustado el modelo a los datos es la **desviación estándar de los residuos**, que a menudo se conoce como el .$^{\text{er}}$ror estándar residual". Esto se muestra en la salida anterior con el valor 0.31. Se calcula usando :

$$\hat{\sigma_\varepsilon} = \sqrt{\frac{1}{T-k-1}\sum_{t=1}^{T}\varepsilon_t^2}$$

Donde:

- k es el número de predictores en el modelo.
- Observe que dividimos entre $T-k-1$ porque hemos estimado $k+1$ parámetros (el intercepto y un coeficiente para cada variable predictora) al calcular los residuos.

El error estándar está relacionado con el tamaño del error promedio que produce el modelo. Podemos comparar este error con la media muestral de y o con la desviación estándar de y para obtener cierta perspectiva sobre la precisión del modelo.
El error estándar se usará cuando se generen intervalos de predicción.

7.3 Evaluación del modelo de regresión

Las diferencias entre los valores y observados y los valores $\hat{y}$ ajustados correspondientes son los errores del conjunto de entrenamiento o residuales"definidos como :

$$\varepsilon_t = y_t - \hat{y}_t$$
$$\varepsilon_t = y_t - \hat{\beta}_0 - \hat{\beta}_1 x_{1,t} - \hat{\beta}_2 x_{2,t} - \cdots - \hat{\beta}_k x_{k,t}$$

para $t = 1, \ldots, T$. Cada residual es el componente impredecible de la observación asociada.
Los residuos tiene algunas propiedades útiles, incluidas las dos siguientes:

1. $\sum_{t=1}^{T} \varepsilon_t = 0$
2. $\sum_{t=1}^{T} x_{k,t}\varepsilon_t = 0 \qquad \forall k$

Como resultado de estas propiedades, es claro que el promedio de los residuales es 0, y que la correlación entre residuales y las observaciones para la variable predictora también es 0. (Esto no es necesariamente cierto cuando se omite la intersección del modelo).
Después de seleccionar las variables de regresión y ajustar un modelo de regresión, es necesario graficar los residuos para comprobar que se han satisfecho los supuestos del modelo. Hay una serie de gráficos que se deben producir para verificar diferentes aspectos del modelo ajustado y los supuestos subyacentes.

Gráfica de residuos ACF

Con datos de series temporales, es muy probable que el valor de una variable observada en el período de tiempo actual sea similar a su valor en el período anterior, o incluso al período anterior, así sucesivamente. Por lo tanto cuando se ajusta un modelo de regresión a datos de series de tiempo, es común encontrar autocorrelación en lo residuos. En este caso, el modelo etimado viola el supuesto de que no hay autocorrelación en los errores, y nuestros pronósticos pueden ser ineficientes: queda algo de información que debe tenerse en cuenta en el modelo para obtener mejores pronósticos. Los pronósticos de un modelo con errores autocorrelacionados aún no están sesgados y, por lo tanto, no son .$^{\text{er}}$róneos", pero generalmente tendrán intervalos de predicción más grandes de lo necesario. Por lo tanto, siempre debemos mirar una gráfica ACF de los residuos.

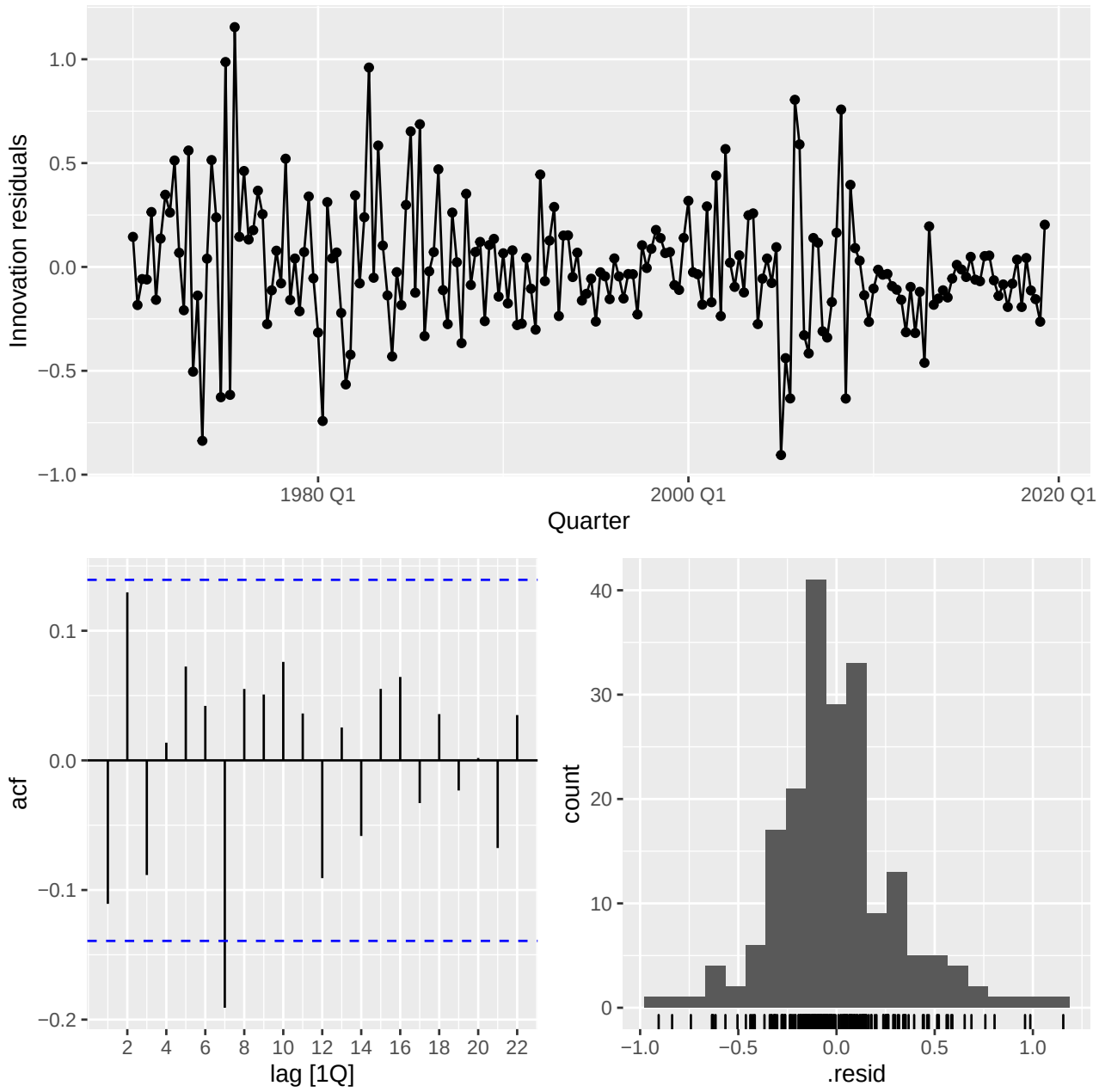
Histograma de residuos

Siempre es una buena idea comprobar si los residuos se distribuyen normalmente. Como explicaos anteriormente, esto no es esencia para la previsión, pero facilita el cálculo de los intervalos de predicción.

Ejemplo

Usando la función **gg_tsresiduals()** podemos obtener los diagnósticos residuales útiles mencionados anteriormente.

```
fit_consMR %>% gg_tsresiduals()
```



El gráfico de tiempo muestra alguna variación cambiante a lo largo del tiempo, pero por lo demás es relativamente anodino. Esta heterocedasticidad potencialmente hará que la cobertura del intervalo de predicción sea inexacta.
El histograma muestra que los residuos parecen estar ligeramente sesgados, lo que también puede afectar la probabilidad de cobertura de los intervalos de predicción.
El gráfico de autocorrelación muestra un pico significativo en el desfase 7 y una prueba de **Ljung-Box** significativa en el nivel del 5 % . Sin embargo, la autocorrelación no es particularmente grande, y en el desase 7 es poco probable que tenga un impacto notable en los pronósticos o los intervalos de predicción.

Gráficas residuales contra predictores.

Esperaríamos que los residuos se dispersaran aleatoriamente sin mostrar ningún patrón sistemático. Una forma sencilla y rápida de verificar esto es examinar los diagramas de dispersión de los residuos frente a cada una de las variables predictoras. Si estos diagramas de dispersión muestran un patrón, entonces la relación puede ser NO lineal y el modelo deberá modificarse en consecuenca.

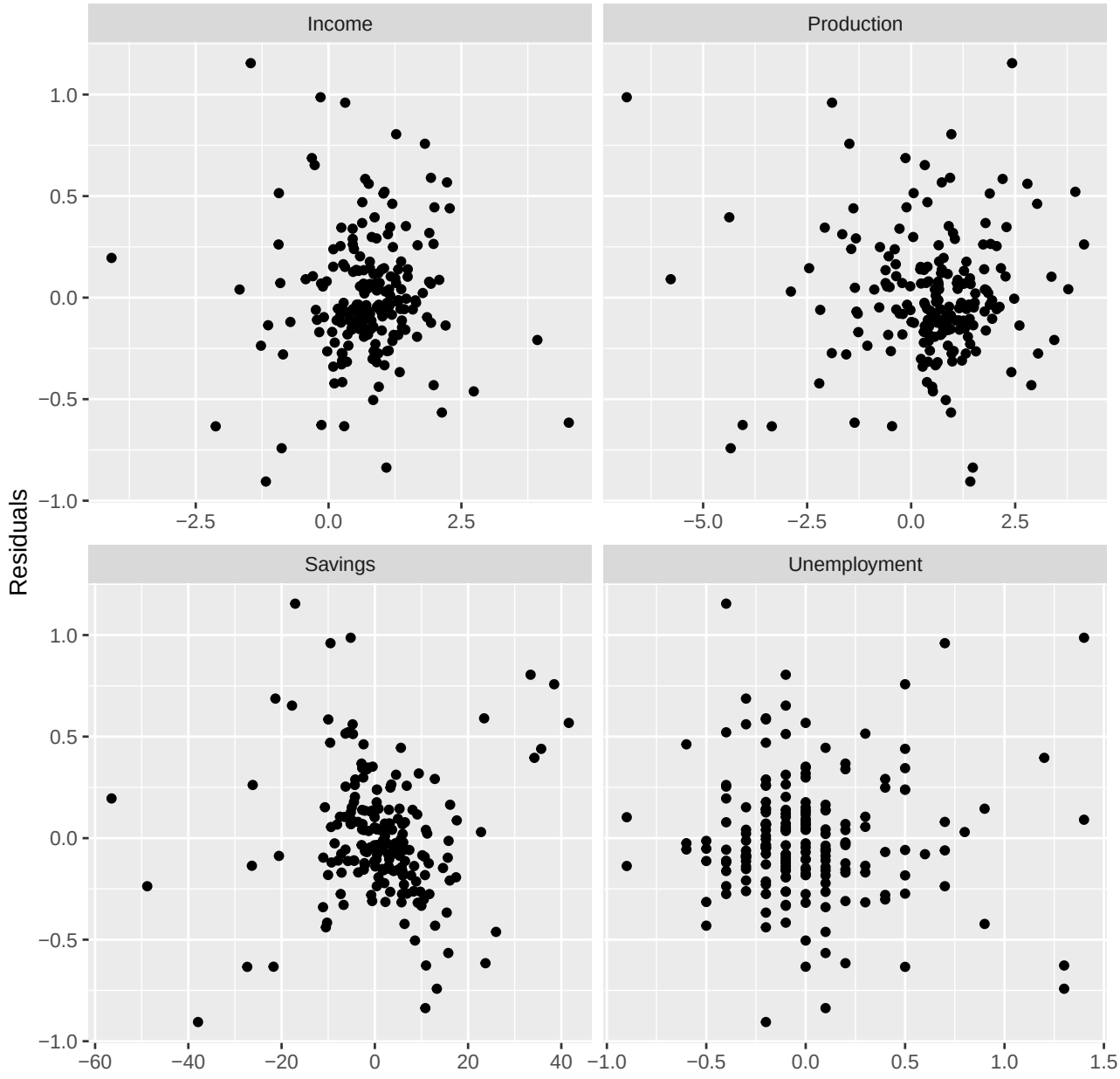
También es necesario graficar los residuales contra cualquier predictor que NO esté en el modelo.

Si alguno de estos muestra un patrón, es posible que sea necesario agregar el predictor correspondiente al modelo (posiblemente en una forma no lineal).

Ejemplo

Los residuos del modelo de regresión múltiple para pronosticar el consumo de EE.UU trazados contra cada predictor parecen estar dispersos al azar. Por lo tanto, estamos satisfechos con estos en este caso.

```
us_change %>%
  left_join(residuals(fit_consMR), by = "Quarter") %>%
  pivot_longer(Income:Unemployment,
               names_to = "regressor", values_to = "x") %>%
  ggplot(aes(x = x, y = .resid)) +
  geom_point() +
  facet_wrap(. ~ regressor, scales = "free_x") +
  labs(y = "Residuals", x = "")
```



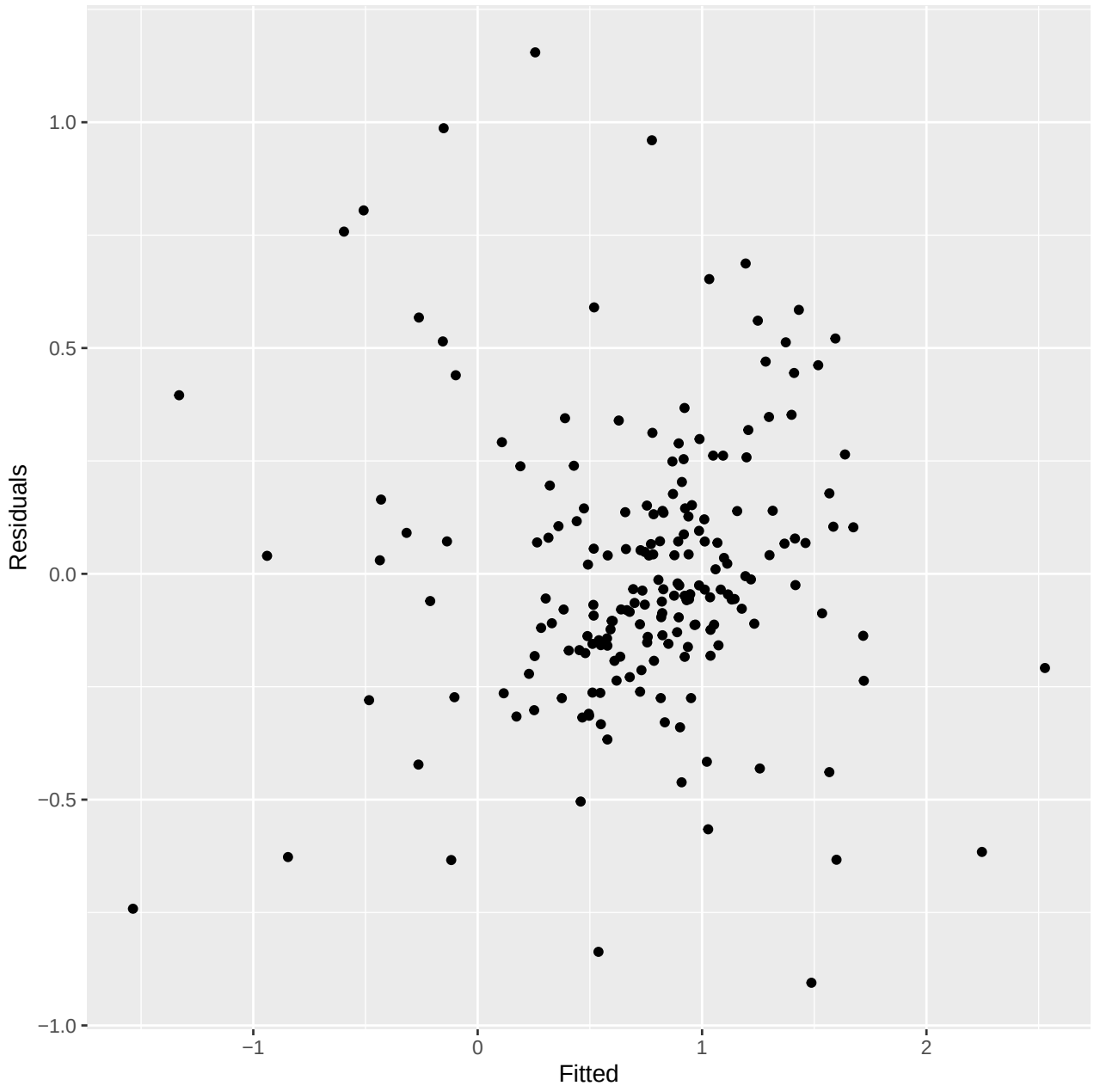
Gráficas residuales contra valores ajustados

Una gráfica de los residuos contra los valores ajustados tampoco debería mostrar ningún patrón. Si se observa un patrón, puede haber "**heterocedasticidad**.[en] los errores, lo que significa que la varianza de los residuos puede no ser constante. Si ocurre este problema, es posible que se requiera una transformación de la variable de pronóstico, como un logarítmo o una raíz cuadrada.

Ejemplo

Continuando con el ejemplo anterior, la siguiente figura muestra los residuos graficados contra los valores ajustados. La dispersión aleatoria sugiere que los errores son homocedásticos.

```
augment(fit_consMR) %>%
  ggplot(aes(x = .fitted, y = .resid)) +
  geom_point() + labs(x = "Fitted", y = "Residuals")
```



Valores atípicos y observaciones influyentes

Las observaciones que toman valores extremos en comparación con la mayoría de los datos se denominan **valores atípicos (outliers)**. Las observaciones que tienen una gran influenca en los coeficientes estimados de un modelo de regresión se denominan **observaciones influyentes**. Por lo general, las observaciones influyentes son valores atípicos que son extremos en la dirección x
Hay métodos formales para detectar valores atípicos y observaciones influyentes, por eso familiarizarse con sus datos antes de realizar cualquier análisis es de vital importancia. Un diagrama de dispersión de y contra cada x es un punto de partida útil en el análisis de regresión y, a menudo, ayuda a identificar observaciones inusuales.

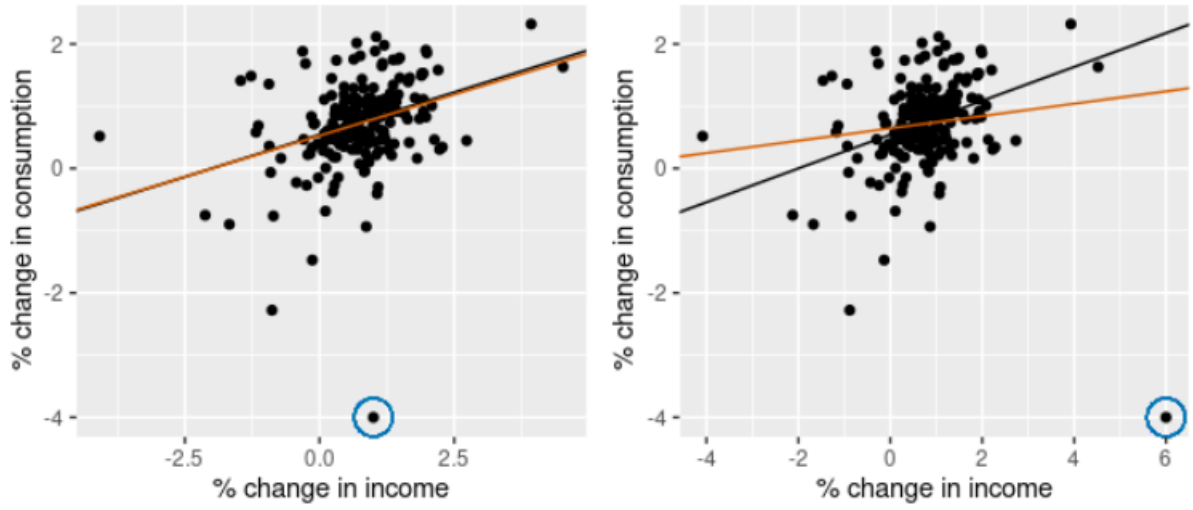
Una fuente de valores atípicos es la entrada incorrecta de datos. Las estadísticas descriptivas simples de sus datos pueden identificar mínimos y máximos que no son sensibles. Si se identificar tal observación y se ha registrado incorrectamente, ddebe corregirse o eliminarse de la muestra de inmediato.

Los valores atípicos también ocurren cuando algunas observaciones son simplemente diferentes. En este caso, puede que no sea prudente eliminar estas observaciones. Si una observación ha sido identificada como un posible valor atípico, es importante estudiarla y analizar las posibles razones detrás de ella. La decisión de eliminar o retener una observación puede ser desafiante (especialmente cuando los valores atípicos son observaciones influyentes). Es aconsejable informar los resultados con o sin eliminación de tales observaciones.

Ejemplo

La siguiente figura destaca el efecto de un solo valor atípico al hacer una regresión del consumo de EE.UU sobre el ingreso (El ejemplo presentado en la sección 7.1). En el panel de la izquierda, el valor atípico solo es extremo en la dirección de y, ya que el cambio porcentual en el consumo se registró incorrectamente como $-4\,\%$.

La línea naranja es la línea de regresión ajustada a los datos que incluyen el valor atípico, en comparación con la línea negra, que es la línea ajustada a los datos sin el valor atípico.
En el panel de la derecha, el valor atípico ahora también es extremo en la dirección de $x$con una disminución del $4\,\%$ en el consumo que corresponde a un aumento del $6\,\%$ en los ingresos. En este caso, el valor atípico es extremadamente influyente ya que la línea naranja ahora se desvía sustancialmente de la línea negra.

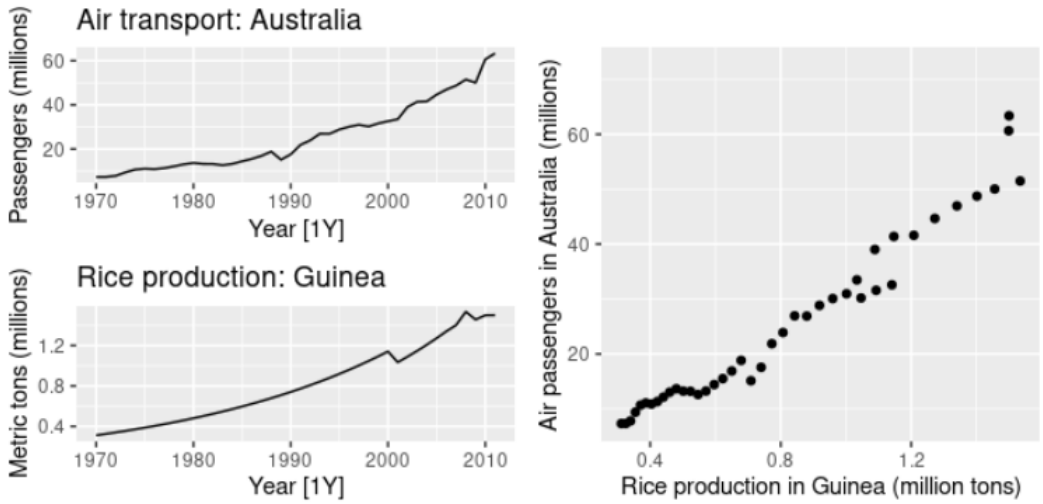


Regresión espuria

La mayoría de las veces, los datos de series de tiempo son "**NO estacionarios**"; es decir, los valores de la serie de tiempo no fluctúan alrededor de una media constante o con una varianza constante. Aquí

se abordará el efecto que los datos NO estacionarios pueden tener en los modelos de regresión.

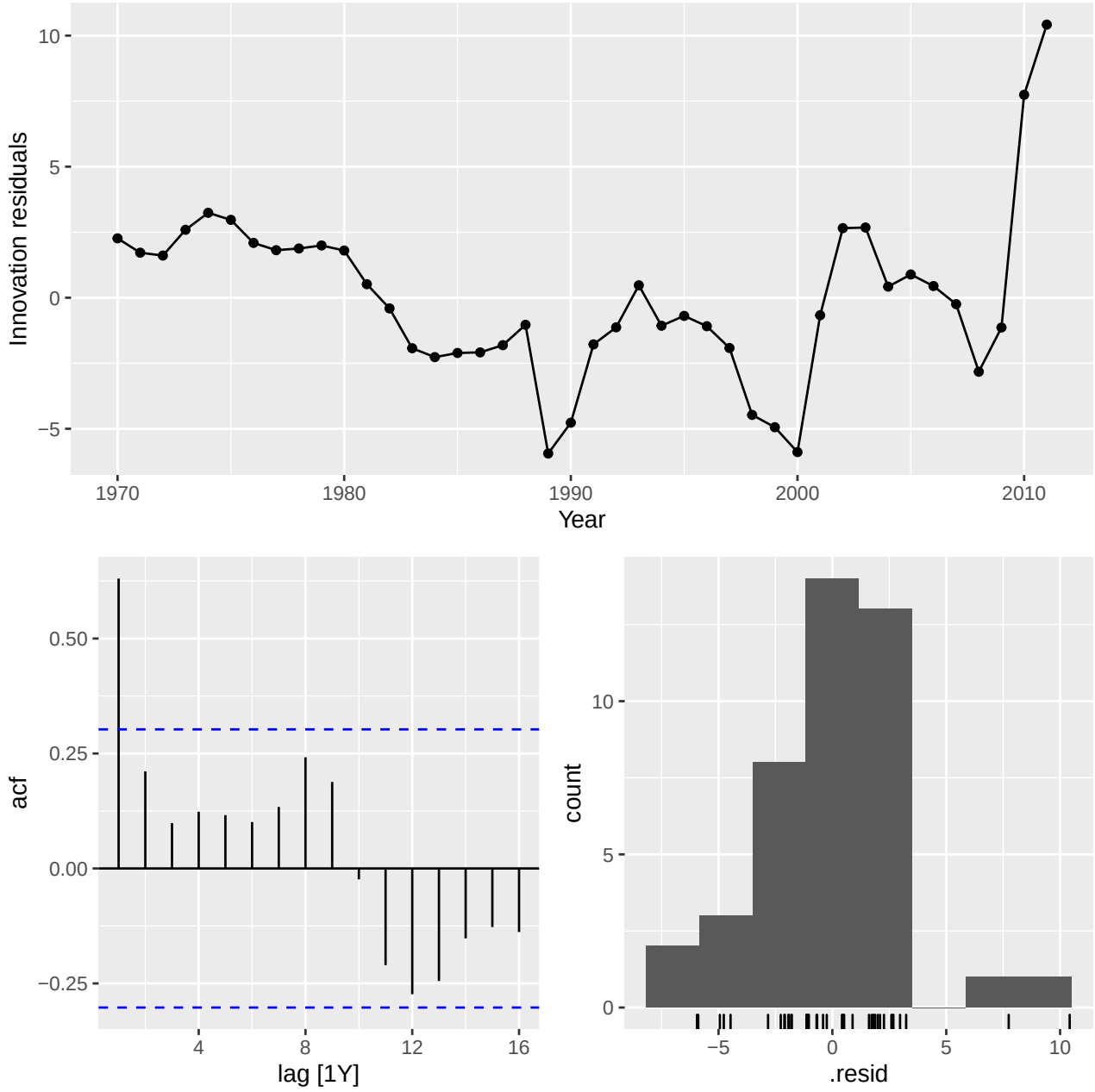
Por ejemplo, considere las dos variables trazadas en la siguiente figura. Estos parecen estar relacionados simplemente porque ambos tienden hacía arriba de la misma manera. Sin embargo, el tráfico aéreo de pasajeros en Australia no tiene nada que ver con la producción de arroz en Guinea.



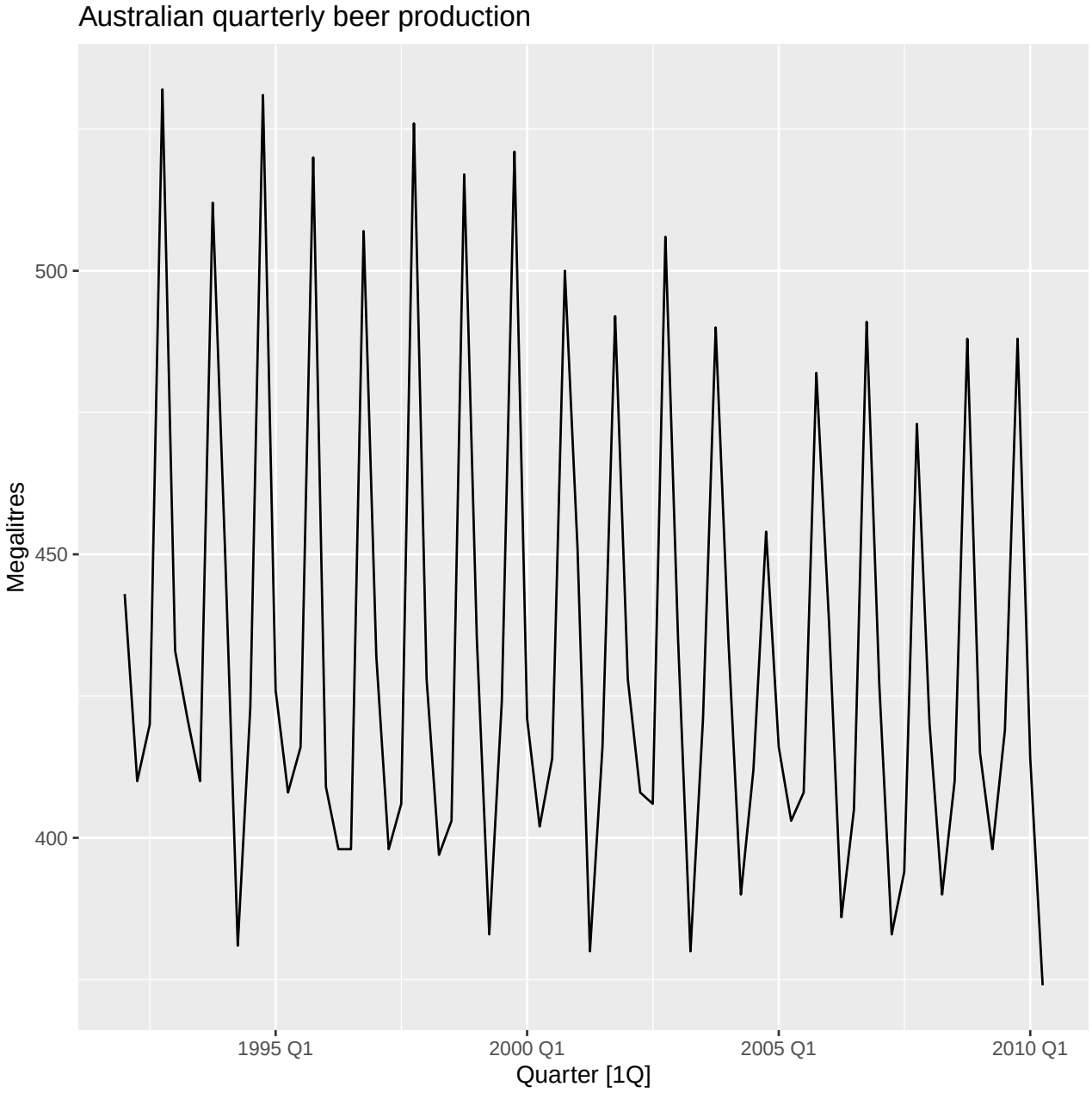
La regresión de series de tiempo NO estacionarias pueden conducir a regresiones espurias. En la siguiente figura se muestra el resultado de la regresión de los pasajeros aéreos australianos sobre la producción de arroz en Guinea. Alta R^2 y una autocorrelación residual alta pueden ser signos de una regresión espuria. Observe estas características en el resultado acontinuación.

Puede parecer que los casos de regresión espuria dan pronósticos razonables a corto plazo, pero generalmente no seguirán funcionando en el futuro.

```
fit <- aus_airpassengers %>%
  filter(Year <= 2011) %>%
  left_join(guinea_rice, by = "Year") %>%
  model(TSLM(Passengers ~ Production))
report(fit)

## Series: Passengers
## Model: TSLM
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9448 -1.8917 -0.3272  1.8620 10.4210
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -7.493      1.203  -6.229 2.25e-07 ***
## Production    40.288      1.337  30.135  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.239 on 40 degrees of freedom
## Multiple R-squared: 0.9578,Adjusted R-squared: 0.9568
## F-statistic: 908.1 on 1 and 40 DF, p-value: < 2.22e-16

fit %>% gg_tsresiduals()
```



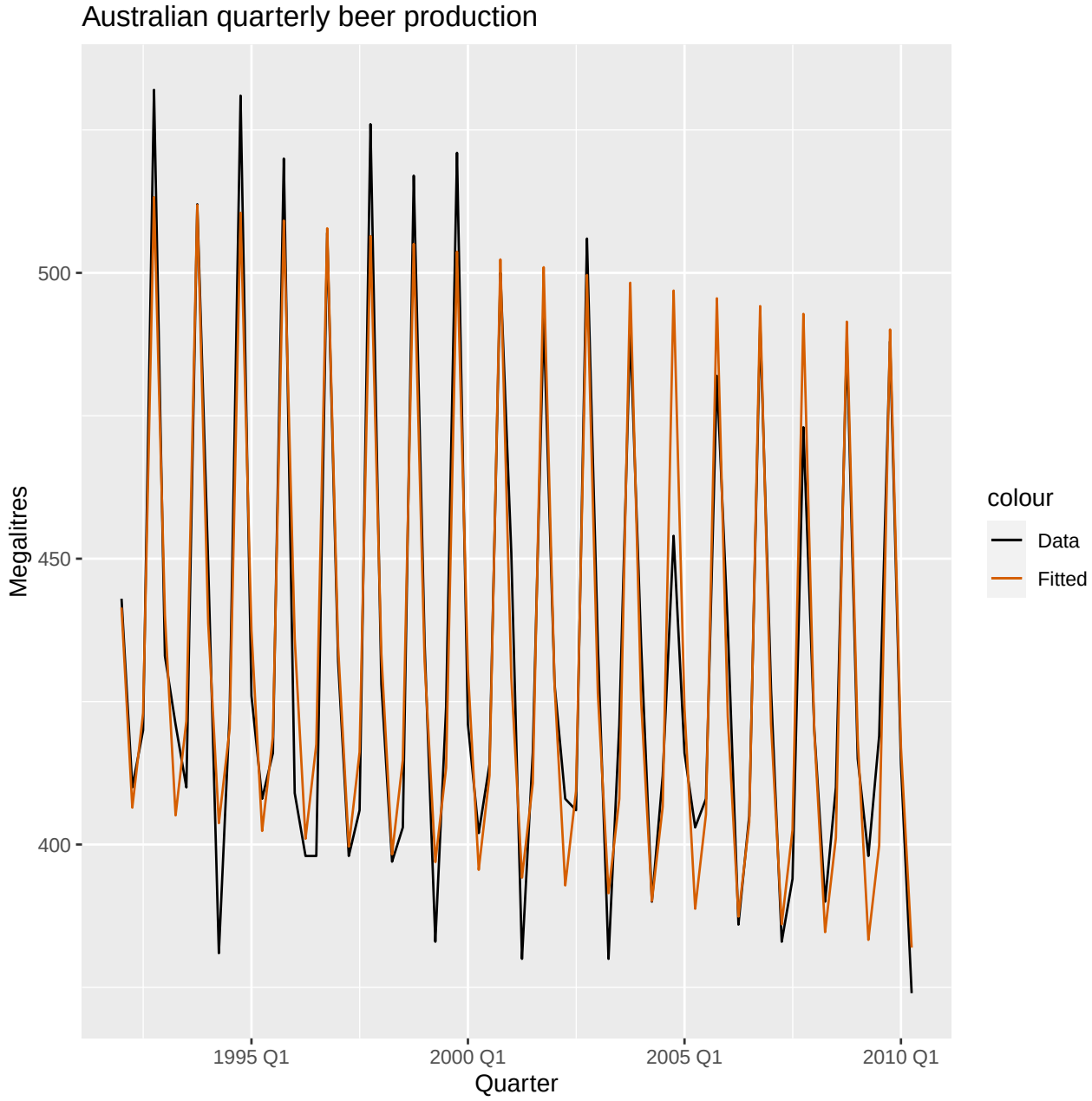

```
# Australian beer production

recent_production <- aus_production %>% filter(year(Quarter) >= 1992)
recent_production %>%
  autoplot(Beer) +
  labs(y = "Megalitres", title = "Australian quarterly beer production")
```




```
fit_beer <- recent_production %>%
  model(TSLM(Beer ~ trend() + season()))
report(fit_beer)

## Series: Beer
## Model: TSLM
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -42.9029  -7.5995  -0.4594   7.9908  21.7895
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   441.80044    3.73353 118.333  < 2e-16 ***
## trend()        -0.34027    0.06657  -5.111 2.73e-06 ***
## season()year2 -34.65973    3.96832  -8.734 9.10e-13 ***
## season()year3 -17.82164    4.02249  -4.430 3.45e-05 ***
## season()year4  72.79641    4.02305  18.095  < 2e-16 ***
```

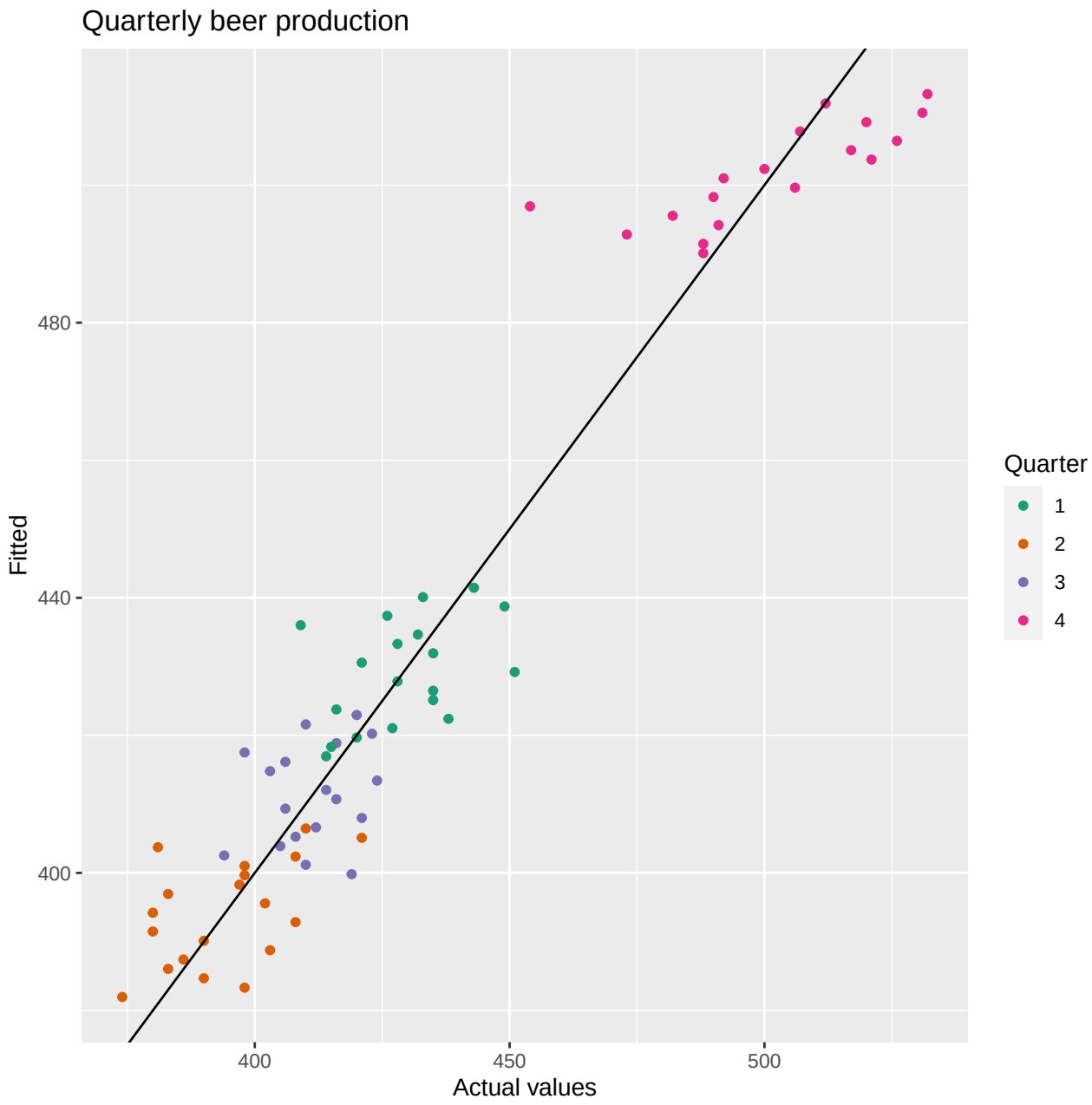
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.23 on 69 degrees of freedom
## Multiple R-squared: 0.9243,Adjusted R-squared: 0.9199
## F-statistic: 210.7 on 4 and 69 DF, p-value: < 2.22e-16
```

```
augment(fit_beer) %>%
  ggplot(aes(x = Quarter)) +
  geom_line(aes(y = Beer, colour = "Data")) +
  geom_line(aes(y = .fitted, colour = "Fitted")) +
  labs(y = "Megalitres", title = "Australian quarterly beer production") +
  scale_colour_manual(values = c(Data = "black", Fitted = "#D55E00"))
```

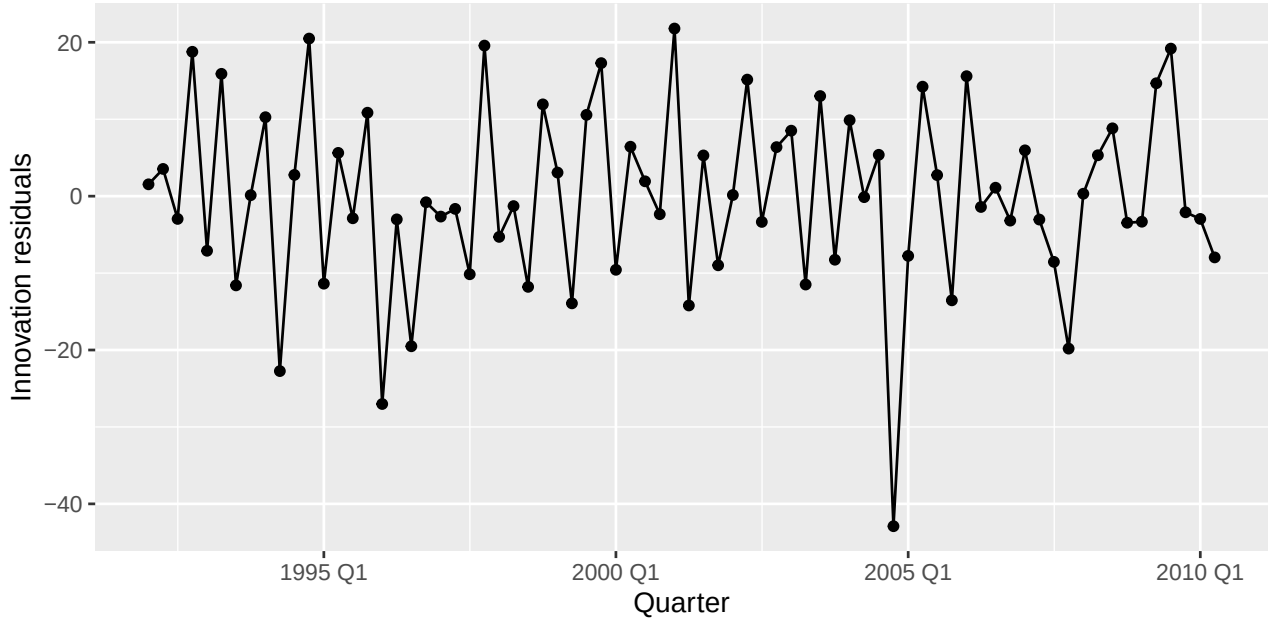


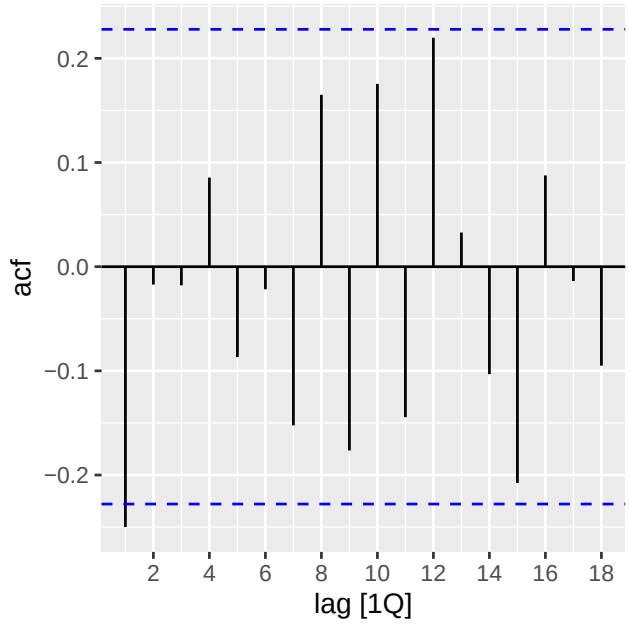

```
augment(fit_beer) %>%
  ggplot(aes(x = Beer, y = .fitted, colour = factor(quarter(Quarter)))) +
  geom_point() +
  labs(
```

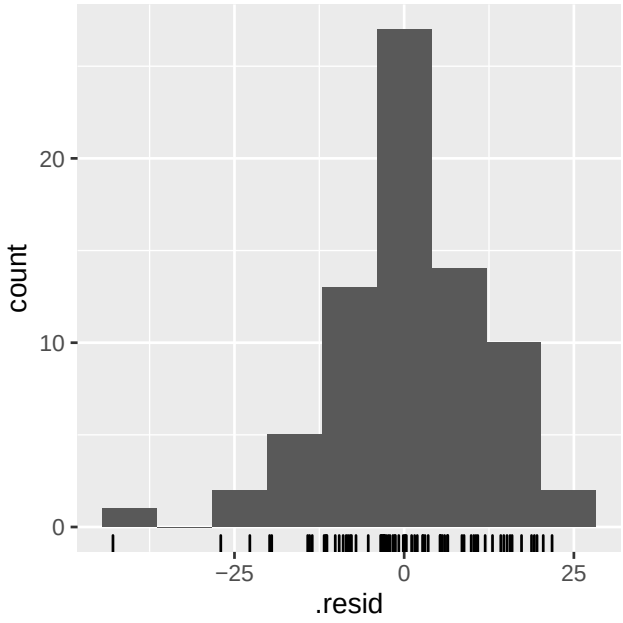
```
    y = "Fitted", x = "Actual values",
    title = "Quarterly beer production"
  ) +
  scale_colour_brewer(palette = "Dark2", name = "Quarter") +
  geom_abline(intercept = 0, slope = 1)
```



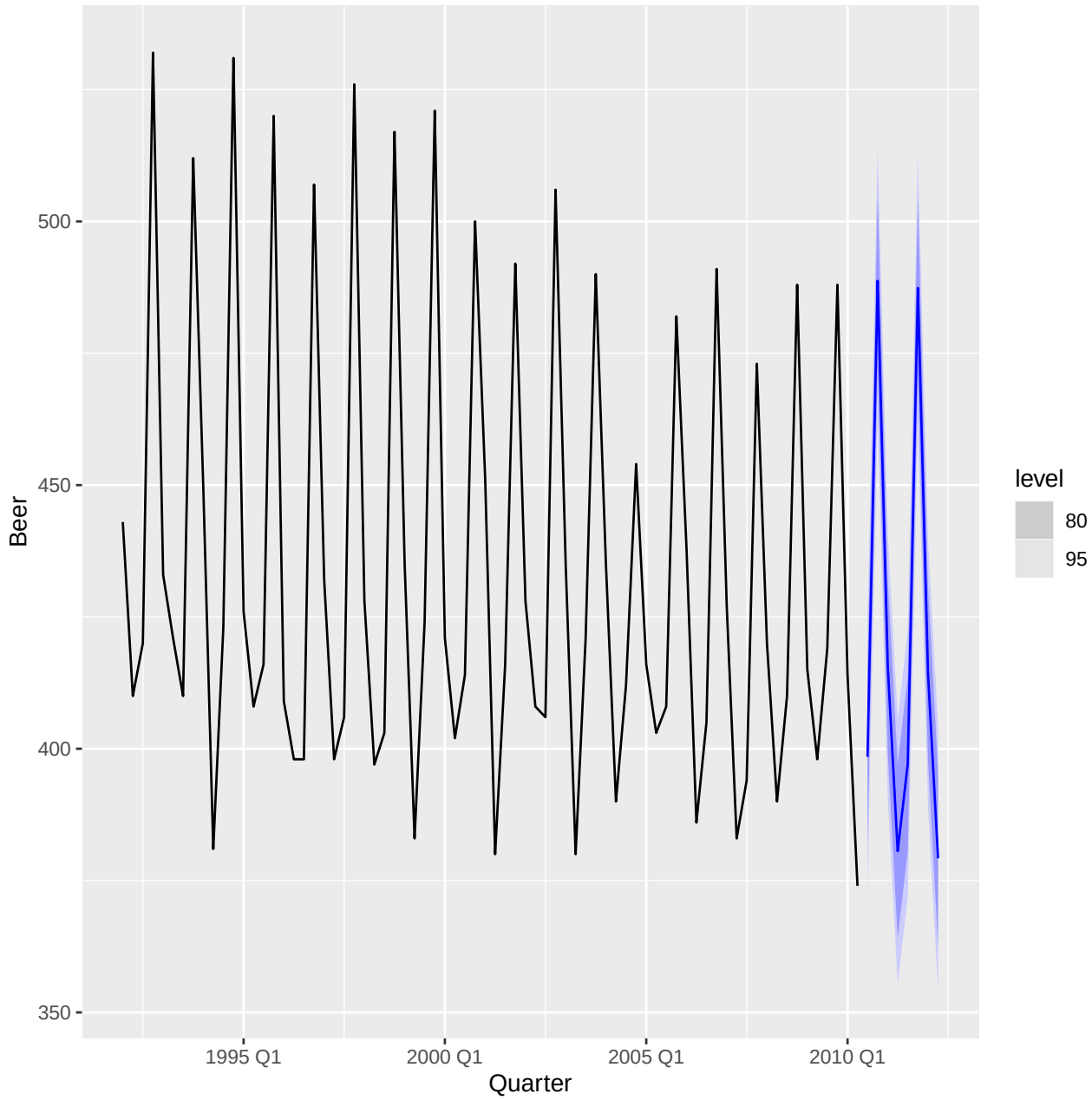

```
fit_beer %>% gg_tsresiduals()
```





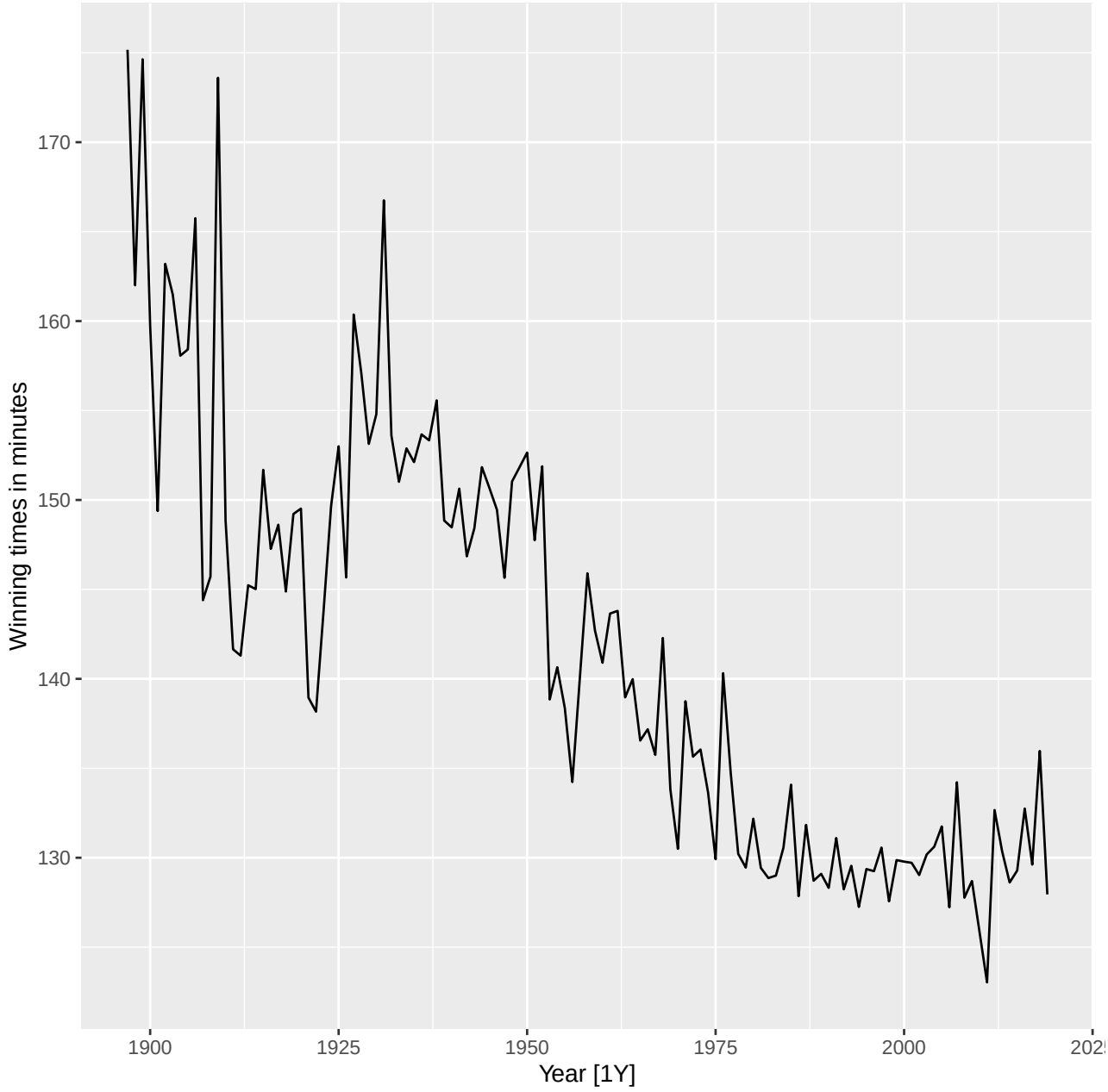



```
fit_beer %>%
  forecast() %>%
  autoplot(recent_production)
```



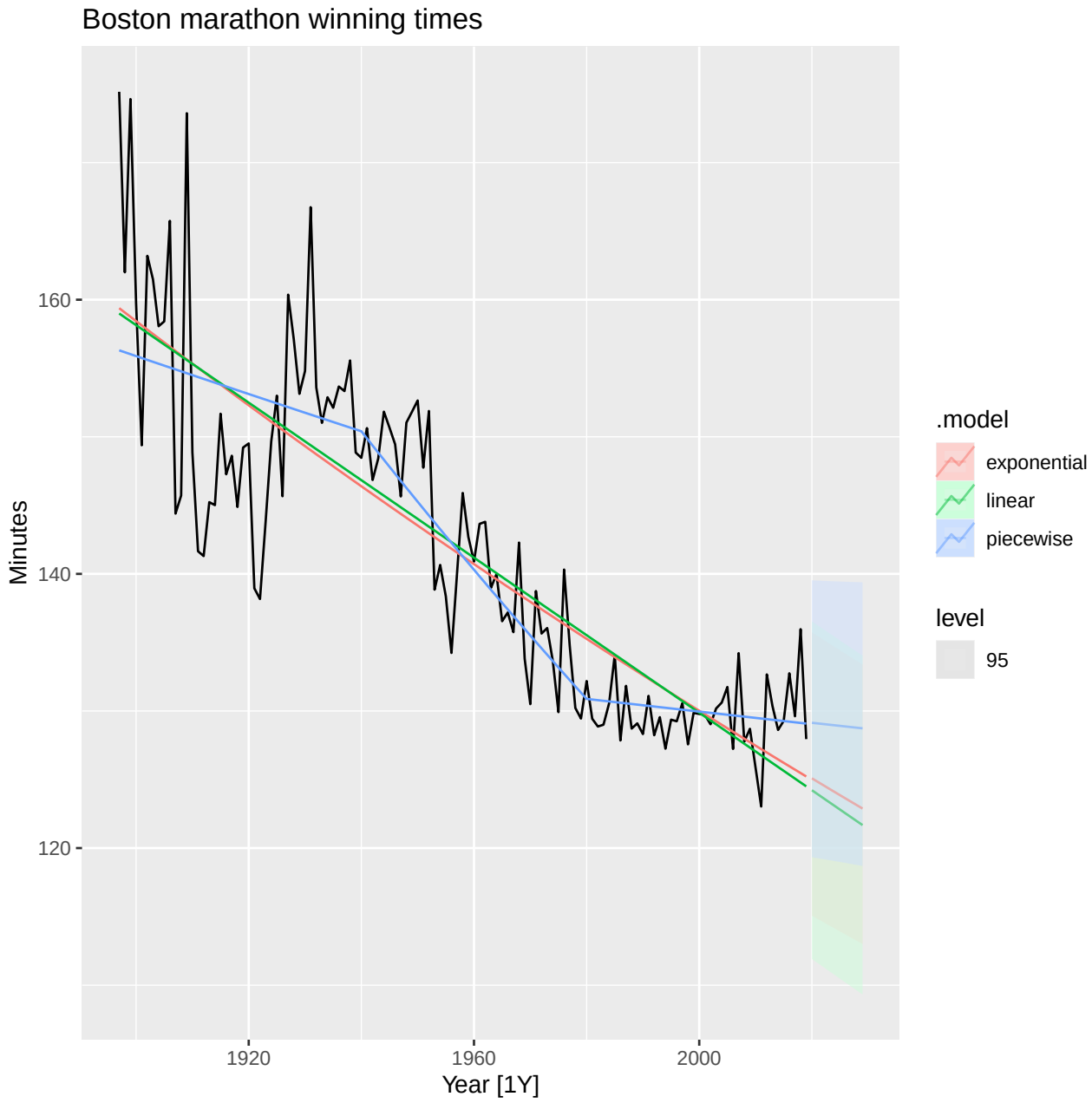

```
fourier_beer <- recent_production %>%
  model(TSLM(Beer ~ trend() + fourier(K = 2)))
report(fourier_beer)

## Series: Beer
## Model: TSLM
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -42.9029  -7.5995  -0.4594   7.9908  21.7895
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        446.87920    2.87321 155.533  < 2e-16 ***
## trend()             -0.34027    0.06657  -5.111 2.73e-06 ***
## fourier(K = 2)C1_4   8.91082    2.01125   4.430 3.45e-05 ***
## fourier(K = 2)S1_4 -53.72807    2.01125 -26.714  < 2e-16 ***
## fourier(K = 2)C2_4 -13.98958    1.42256  -9.834 9.26e-15 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.23 on 69 degrees of freedom
## Multiple R-squared: 0.9243,Adjusted R-squared: 0.9199
## F-statistic: 210.7 on 4 and 69 DF, p-value: < 2.22e-16

recent_production %>%
  model(
    f1 = TSLM(Beer ~ trend() + fourier(K = 1)),
    f2 = TSLM(Beer ~ trend() + fourier(K = 2)),
    season = TSLM(Beer ~ trend() + season())
  ) %>%
  glance()

## # A tibble: 3 x 15
##   .model r_squared adj_r_squared sigma2 statistic  p_value    df log_lik   AIC
##   <chr>      <dbl>         <dbl>  <dbl>     <dbl>    <dbl> <int>   <dbl> <dbl>
## 1 f1         0.818         0.810   354.      105. 7.41e-26     4   -320.  440.
## 2 f2         0.924         0.920   150.      211. 6.97e-38     5   -288.  377.
## 3 season     0.924         0.920   150.      211. 6.97e-38     5   -288.  377.
## # ... with 6 more variables: AICc <dbl>, BIC <dbl>, CV <dbl>, deviance <dbl>,
## #   df.residual <int>, rank <int>

## Boston Marathon

marathon <- boston_marathon %>%
  filter(Event == "Men's open division") %>%
  select(-Event) %>%
  mutate(Minutes = as.numeric(Time) / 60)
marathon %>%
  autoplot(Minutes) +
  labs(y = "Winning times in minutes")
```

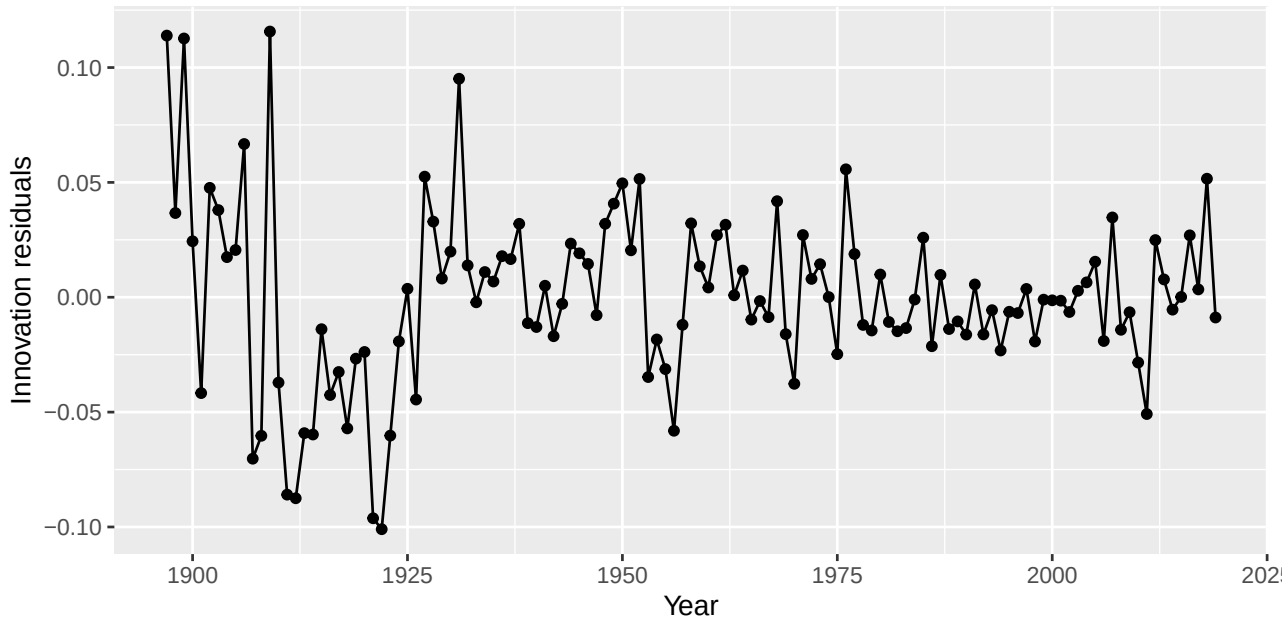


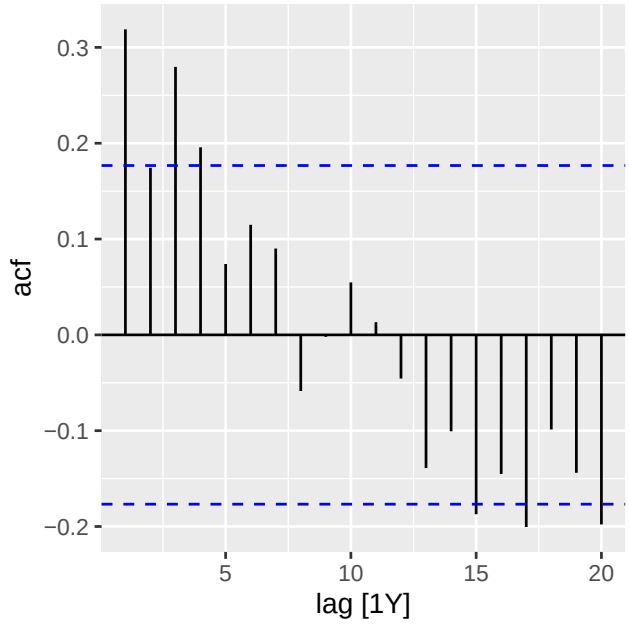

```
fit_trends <- marathon %>%
  model(
    # Linear trend
    linear = TSLM(Minutes ~ trend()),
    # Exponential trend
    exponential = TSLM(log(Minutes) ~ trend()),
    # Piecewise linear trend
    piecewise = TSLM(log(Minutes) ~ trend(knots = c(1940, 1980)))
  )

fit_trends

## # A mable: 1 x 3
##    linear exponential piecewise
##   <model>     <model>   <model>
## 1  <TSLM>      <TSLM>    <TSLM>

fc_trends <- fit_trends %>%
```

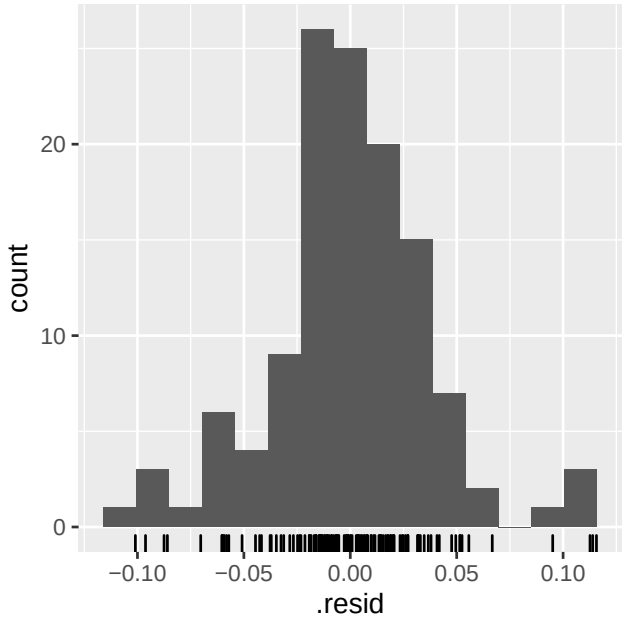
```
  forecast(h = 10)
marathon %>%
  autoplot(Minutes) +
  geom_line(
    data = fitted(fit_trends),
    aes(y = .fitted, colour = .model)
  ) +
  autolayer(fc_trends, alpha = 0.5, level = 95) +
  labs(
    y = "Minutes",
    title = "Boston marathon winning times"
  )
```




```
fit_trends %>%
  select(piecewise) %>%
  gg_tsresiduals()
```

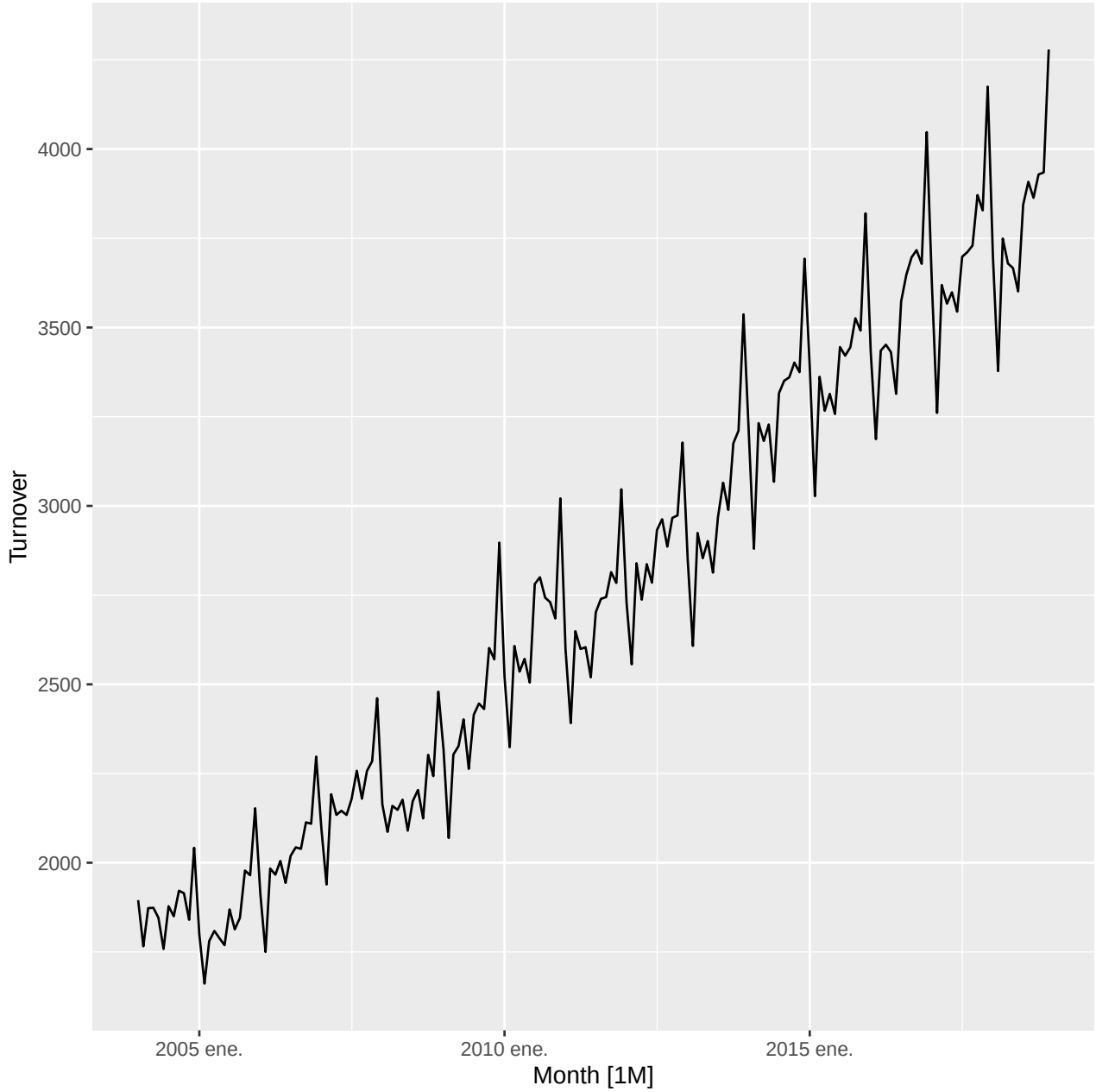




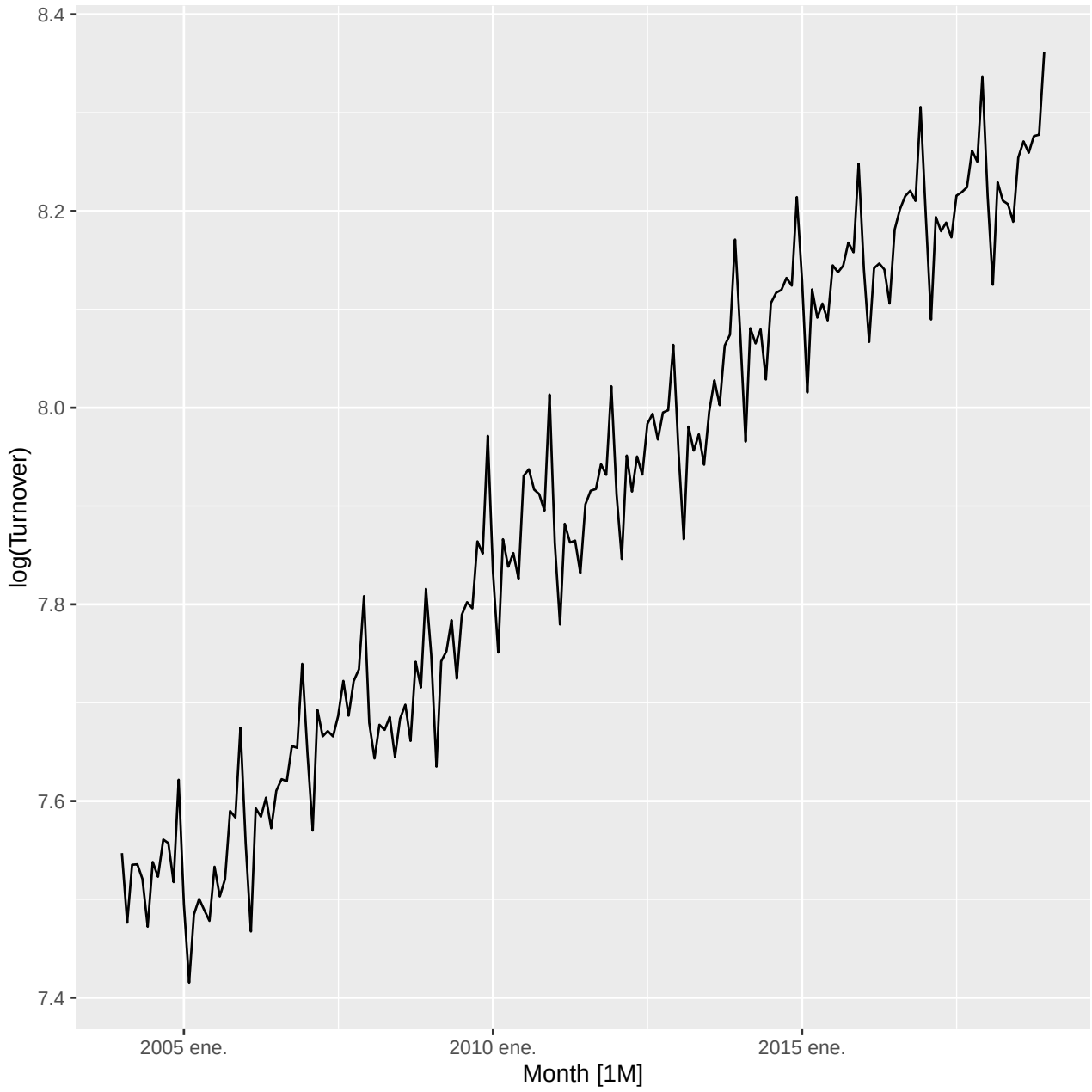



```
glance(fit_trends) %>%
  select(.model, r_squared, adj_r_squared, AICc, CV)

## # A tibble: 3 x 5
##   .model      r_squared adj_r_squared  AICc       CV
##   <chr>           <dbl>         <dbl> <dbl>    <dbl>
## 1 linear          0.728         0.726  452. 39.1
## 2 exponential     0.744         0.742 -779.  0.00176
## 3 piecewise       0.787         0.781 -797.  0.00152

# Fourier terms for cafe data

aus_cafe <- aus_retail %>%
  filter(
    Industry == "Cafes, restaurants and takeaway food services",
    year(Month) %in% 2004:2018
  ) %>%
  summarise(Turnover = sum(Turnover))
```

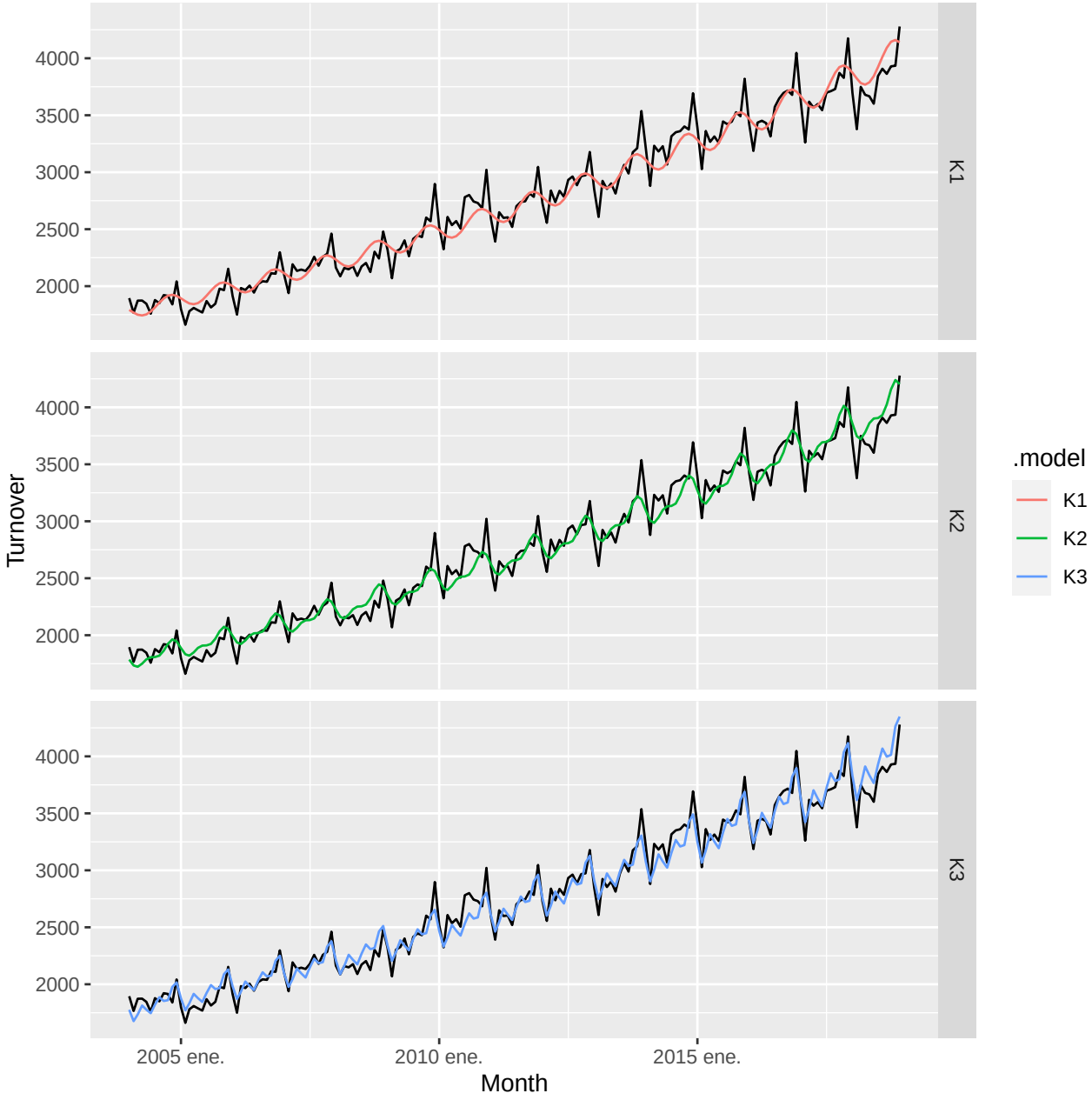
```
aus_cafe %>%
  autoplot(Turnover)
```




```
aus_cafe %>%
  autoplot(log(Turnover))
```




```
fit <- aus_cafe %>%
  model(
    K1 = TSLM(log(Turnover) ~ trend() + fourier(K = 1)),
    K2 = TSLM(log(Turnover) ~ trend() + fourier(K = 2)),
    K3 = TSLM(log(Turnover) ~ trend() + fourier(K = 3)),
    K4 = TSLM(log(Turnover) ~ trend() + fourier(K = 4)),
    K5 = TSLM(log(Turnover) ~ trend() + fourier(K = 5)),
    K6 = TSLM(log(Turnover) ~ trend() + fourier(K = 6))
  )

augment(fit) %>%
  filter(.model %in% c("K1", "K2", "K3")) %>%
  ggplot(aes(x = Month, y = Turnover)) +
  geom_line() +
  geom_line(aes(y = .fitted, col = .model)) +
  facet_grid(.model ~ .)
```




```
glance(fit) %>%
  select(.model, sigma2, log_lik, AIC, AICc, BIC)

## # A tibble: 6 x 6
##   .model   sigma2 log_lik    AIC   AICc    BIC
##   <chr>     <dbl>   <dbl>  <dbl>  <dbl>  <dbl>
## 1 K1     0.00232     292. -1086. -1085. -1070.
## 2 K2     0.00213     301. -1100. -1099. -1077.
## 3 K3     0.00150     334. -1161. -1160. -1132.
## 4 K4     0.00130     348. -1185. -1183. -1149.
## 5 K5     0.000966    376. -1236. -1234. -1195.
## 6 K6     0.000969    376. -1235. -1232. -1190.

# US consumption quarterly changes

fit_all <- us_change %>%
  model(
    TSLM(Consumption ~ Income + Production + Unemployment + Savings),
```

```
    TSLM(Consumption ~ Production + Unemployment + Savings),
    TSLM(Consumption ~ Income + Unemployment + Savings),
    TSLM(Consumption ~ Income + Production + Savings),
    TSLM(Consumption ~ Income + Production + Unemployment),
    TSLM(Consumption ~ Income + Production),
    TSLM(Consumption ~ Income + Unemployment),
    TSLM(Consumption ~ Income + Savings),
    TSLM(Consumption ~ Production + Unemployment),
    TSLM(Consumption ~ Production + Savings),
    TSLM(Consumption ~ Unemployment + Savings),
    TSLM(Consumption ~ Income),
    TSLM(Consumption ~ Production),
    TSLM(Consumption ~ Unemployment),
    TSLM(Consumption ~ Savings),
    TSLM(Consumption ~ 1),
  )

us_change %>%
  model(
    TSLM(Consumption ~ Income * Savings + Production + Unemployment),
  ) %>%
  report()

## Series: Consumption
## Model: TSLM
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9123 -0.1579 -0.0350  0.1377  1.1519
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.2514027  0.0359393   6.995 4.29e-11 ***
## Income          0.7408481  0.0402455  18.408  < 2e-16 ***
## Savings        -0.0529245  0.0029383 -18.012  < 2e-16 ***
## Production      0.0474158  0.0232432   2.040   0.0427 *
## Unemployment   -0.1736946  0.0959238  -1.811   0.0717 .
## Income:Savings  0.0001641  0.0009515   0.172   0.8632
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.311 on 192 degrees of freedom
## Multiple R-squared: 0.7683,Adjusted R-squared: 0.7623
## F-statistic: 127.3 on 5 and 192 DF, p-value: < 2.22e-16

fit_all %>%
  glance() %>%
  select(.model, adj_r_squared, AICc, BIC, CV) %>%
  arrange(CV)

## # A tibble: 16 x 5
##    .model                                          adj_r_squared  AICc   BIC    CV
##    <chr>                                                   <dbl> <dbl> <dbl> <dbl>
##  1 TSLM(Consumption ~ Income + Production + Une~        0.763  -456. -437. 0.104
##  2 TSLM(Consumption ~ Income + Unemployment + S~        0.760  -454. -438. 0.104
```

```
##  3 TSLM(Consumption ~ Income + Production + Sav~        0.761  -455. -439. 0.105
##  4 TSLM(Consumption ~ Income + Savings)                0.735  -436. -423. 0.114
##  5 TSLM(Consumption ~ Income + Production + Une~       0.366  -262. -246. 0.271
##  6 TSLM(Consumption ~ Income + Unemployment)           0.345  -257. -244. 0.276
##  7 TSLM(Consumption ~ Production + Unemployment~       0.349  -257. -241. 0.279
##  8 TSLM(Consumption ~ Income + Production)             0.336  -254. -241. 0.282
##  9 TSLM(Consumption ~ Production + Savings)            0.324  -250. -238. 0.287
## 10 TSLM(Consumption ~ Unemployment + Savings)          0.311  -247. -234. 0.291
## 11 TSLM(Consumption ~ Production + Unemployment)       0.308  -246. -233. 0.293
## 12 TSLM(Consumption ~ Unemployment)                    0.274  -237. -228. 0.303
## 13 TSLM(Consumption ~ Production)                      0.276  -238. -228. 0.304
## 14 TSLM(Consumption ~ Income)                          0.143  -204. -195. 0.356
## 15 TSLM(Consumption ~ Savings)                         0.0611 -186. -177. 0.388
## 16 TSLM(Consumption ~ 1)                               0      -175. -168. 0.409
```

```
fit_consBest <- us_change %>%
  model(
    TSLM(Consumption ~ Income + Production + Unemployment + Savings),
  )

fit_consBest %>% report()
```

```
## Series: Consumption
## Model: TSLM
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.90555 -0.15821 -0.03608  0.13618  1.15471
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.253105   0.034470   7.343 5.71e-12 ***
## Income        0.740583   0.040115  18.461  < 2e-16 ***
## Production    0.047173   0.023142   2.038   0.0429 *
## Unemployment -0.174685   0.095511  -1.829   0.0689 .
## Savings      -0.052890   0.002924 -18.088  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3102 on 193 degrees of freedom
## Multiple R-squared: 0.7683,Adjusted R-squared: 0.7635
## F-statistic:   160 on 4 and 193 DF, p-value: < 2.22e-16
```

```
future_scenarios <- scenarios(
  Increase = new_data(us_change, 4) %>%
    mutate(Income = 1, Savings = 0.5, Unemployment = 0, Production = 0),
  Decrease = new_data(us_change, 4) %>%
    mutate(Income = -1, Savings = -0.5, Unemployment = 0, Production = 0),
  names_to = "Scenario"
)

fc <- forecast(fit_consBest, new_data = future_scenarios)

us_change %>% autoplot(Consumption) +
  labs(y = "% change in US consumption") +
```

```
autolayer(fc) +
labs(title = "US consumption", y = "% change")
```