

# Fundamentos estadísticos - Tarea 2

José Roberto Pérez Chávez

```
knitr::opts_chunk$set(echo = TRUE, error=TRUE)

# cargamos paquetes
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.0 --

## v ggplot2 3.3.2    v purrr   0.3.4
## v tibble  3.0.3    v dplyr   1.0.1
## v tidyr   1.1.1    v stringr 1.4.0
## v readr   1.3.1    v forcats 0.5.0

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

## Series de tiempo

Consideramos la ventas semanales de un producto a lo largo de 5 años, transformaremos la variable de ventas utilizando el logaritmo.

1. Describe que observas en la gráfica.

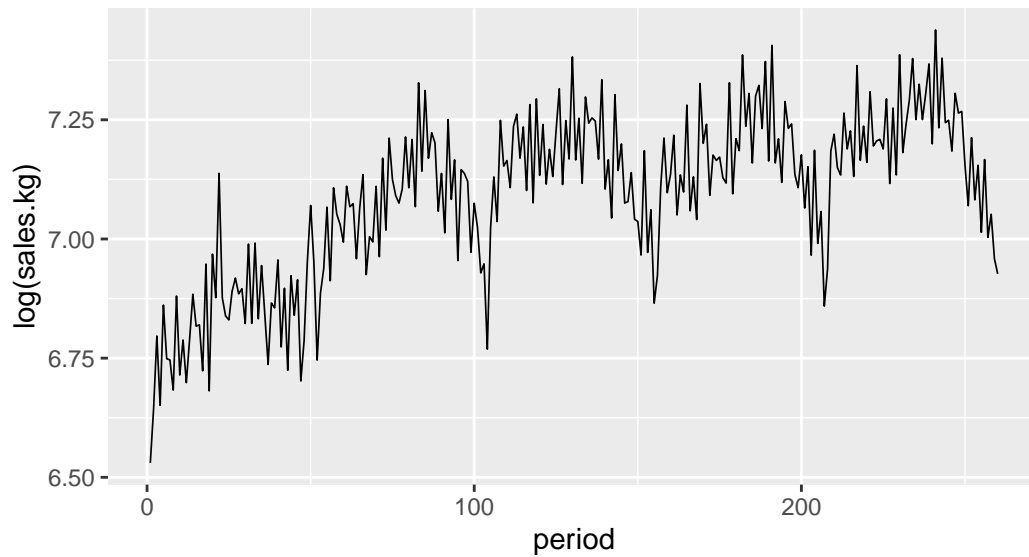
```
ventas <- read_csv("ventas_semanal.csv")

## Parsed with column specification:
## cols(
##   period = col_double(),
##   sales.kg = col_double()
## )

head(ventas)

## # A tibble: 6 x 2
##   period sales.kg
##   <dbl>   <dbl>
## 1     1     686.
## 2     2     768.
## 3     3     895.
## 4     4     774.
## 5     5     955.
## 6     6     853.

ggplot(ventas, aes(x = period, y = log(sales.kg))) +
  geom_line(size = 0.3)
```



En esta gráfica observo las siguientes características: - Hay una tendencia creciente en el valor de las ventas. - Hay una estacionalidad marcada por lapsos de 50 periodos.

Intentaremos usar suavizamiento para capturar los distintos tipos de variación que observamos en la serie.

2. Utiliza un suavizador *loess* para capturar la tendencia de la serie.

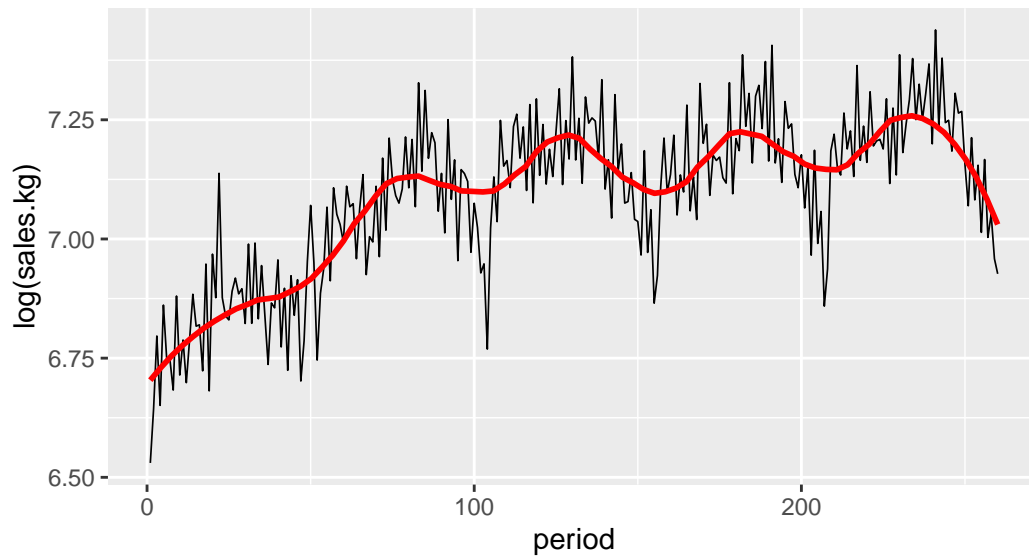
```
ggplot(ventas,
  aes(
    x = period,
    y = log(sales.kg)
  )
) +

  geom_line(
    size = 0.3
  ) +

  geom_smooth(
    method = "loess",
    span=0.3,
    degree = 1,
    se = FALSE,
    size = 1,
    color = "red"
  )
```

```
## Warning: Ignoring unknown parameters: degree
```

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



```
fit_trend_1 <- loess(
  log(sales.kg) ~ period,
  ventas,
  span = 0.3,
  degree = 1
)
```

3. Ahora calcula los residuales de este ajuste y descríbelos mediante un suavizamiento más fino. Verifica que se ha estimado la mayor parte de la tendencia, e intenta capturar la variación estacional de los residuales.

```
print(head(ventas))
```

```
## # A tibble: 6 x 2
##   period sales.kg
##   <dbl>   <dbl>
## 1     1     686.
## 2     2     768.
## 3     3     895.
## 4     4     774.
## 5     5     955.
## 6     6     853.
```

```
## Cálculo de los residuales 1
```

```
ventas_v1 <- ventas %>%
  mutate(ajuste_1 = fitted(fit_trend_1)) %>%
  mutate(res.trend.1 = log(sales.kg) - ajuste_1)
```

```
print(head(ventas_v1))
```

```
## # A tibble: 6 x 4
##   period sales.kg ajuste_1 res.trend.1
##   <dbl>   <dbl>   <dbl>   <dbl>
## 1     1     686.     6.74    -0.209
## 2     2     768.     6.74    -0.0996
## 3     3     895.     6.75     0.0485
## 4     4     774.     6.75    -0.101
## 5     5     955.     6.76     0.105
```

```
## 6      6      853.      6.76      -0.0122

## Ajuste estacional (ajuste 2)
fit_trend_2 <-loess(
  res.trend.1 ~ period,
  ventas_v1,
  span=0.15,
  degree=1
)

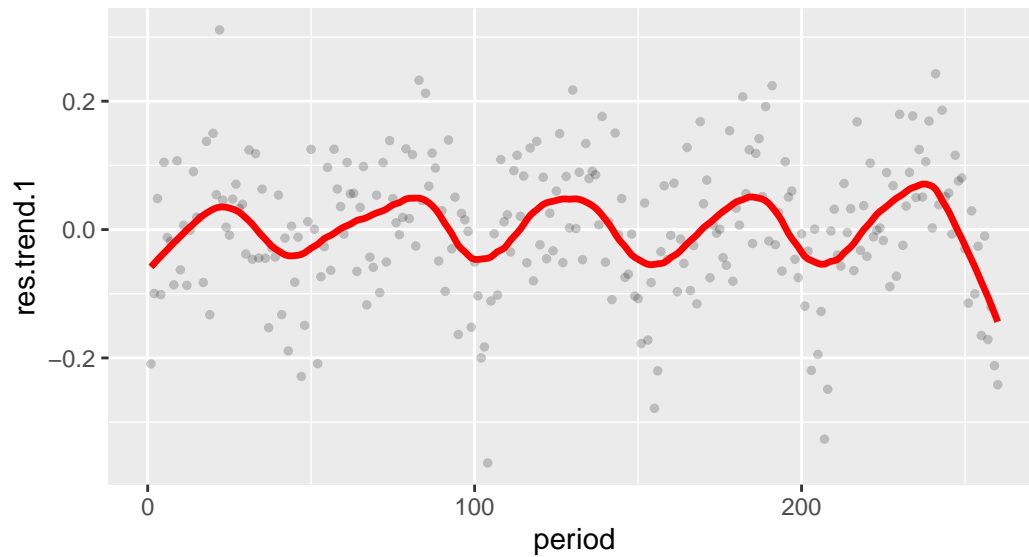
## Cálculo de los residuales 2
ventas_v2 <- ventas_v1 %>%
  mutate(ajuste_2 = fitted(fit_trend_2)) %>%
  mutate(res.trend.2 = res.trend.1 - ajuste_2)

print(head(ventas_v2))

## # A tibble: 6 x 6
##   period sales.kg ajuste_1 res.trend.1 ajuste_2 res.trend.2
##   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1     1     686.     6.74   -0.209  -0.0583  -0.151
## 2     2     768.     6.74   -0.0996 -0.0527  -0.0469
## 3     3     895.     6.75    0.0485  -0.0471   0.0955
## 4     4     774.     6.75   -0.101  -0.0416  -0.0597
## 5     5     955.     6.76    0.105  -0.0364   0.141
## 6     6     853.     6.76   -0.0122 -0.0313   0.0191

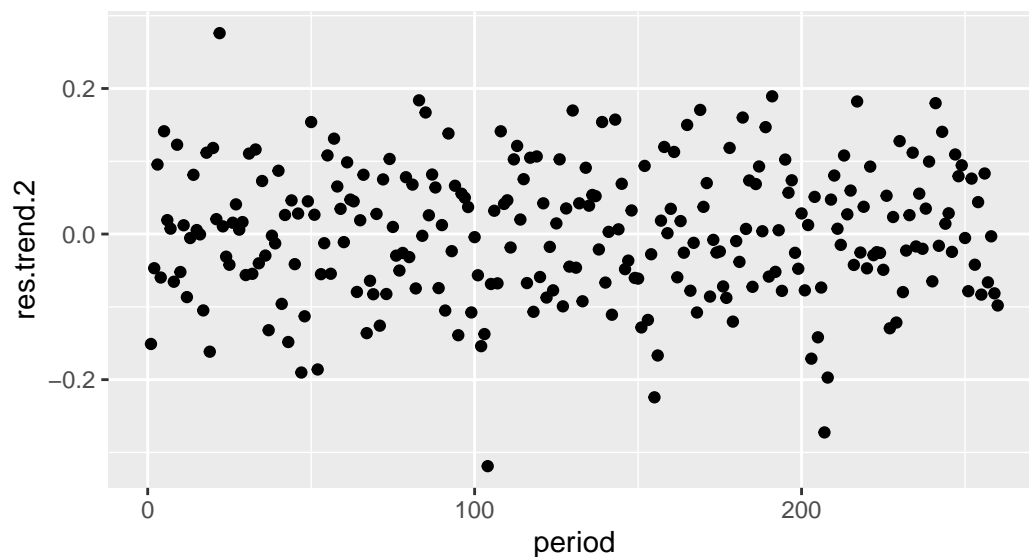
## Gráfica de estacionalidad de las ventas
ggplot(
  ventas_v2,
  aes(
    x=period
  )
) +
  geom_point(
    aes(
      y=res.trend.1
    ),
    alpha=0.2,
    size=1
  ) +
  geom_line(
    aes(
      y=ajuste_2
    ),
    colour="red",
    size=1.2
  )

```



4. Grafica los residuales obtenidos después de ajustar el componente estacional para estudiar la componente de mayor frecuencia.

```
ggplot(
  ventas_v2,
  aes(
    x = period,
    y = res.trend.2
  )
) +
  geom_point(
  )
```



```
# fit_season_high <-
```

5. **Extra opcional.** Ahora que tenemos nuestra primera estimación de cada una de las componentes, podemos regresar a hacer una mejor estimación de la tendencia. La ventaja de volver es que ahora podemos suavizar más sin que en nuestra muestra compita tanto la variación estacional. Por tanto puedes suavizar un poco menos.

```
## Suavización de los puntos resultantes después de las dos suavizaciones previas
```

```
#### Creación de curva de Loess
```

```
fit_trend_f <- loess(
  res.trend.2 ~ period,
  ventas_v2,
  span=0.2,
  degree=1
)
```

```
#### Ajuste de la curva con los datos
```

```
ventas_vf <- ventas_v2 %>%
  mutate(ajuste_f = fitted(fit_trend_f)) %>%
  mutate(res.trend.f = res.trend.2 - ajuste_f)
```

```
print(head(ventas_vf))
```

```
## # A tibble: 6 x 8
```

```
##   period sales.kg ajuste_1 res.trend.1 ajuste_2 res.trend.2 ajuste_f res.trend.f
##   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1     1     686.     6.74   -0.209  -0.0583  -0.151  -1.49e-3  -0.150
## 2     2     768.     6.74   -0.0996 -0.0527  -0.0469 -7.33e-4  -0.0462
## 3     3     895.     6.75    0.0485  -0.0471   0.0955  7.98e-5   0.0955
## 4     4     774.     6.75   -0.101  -0.0416  -0.0597  9.17e-4  -0.0606
## 5     5     955.     6.76    0.105  -0.0364   0.141   1.75e-3   0.139
## 6     6     853.     6.76   -0.0122 -0.0313   0.0191  2.53e-3   0.0166
```

6. Visualiza el ajuste, genera una gráfica de paneles, en cada uno muestra una componente de la serie de tiempo y los residuales.

```
## Datos originales + ajuste a tendencia
```

```
g1 <- ggplot(
  ventas_vf,
  aes(
    x=period
  )
) +
  geom_point(
    aes(
      y=log(sales.kg)
    )
  ) +
  geom_line(
    aes(
      y=ajuste_1
    ),
    colour="red"
  ) +
  labs(
    title="Datos originales + ajuste a tendencia"
  )
)
```

```
## Datos sin tendencia + ajuste estacional
```

```

g2 <- ggplot(
  ventas_vf,
  aes(
    x=period
  )
) +
  geom_point(
    aes(
      y=res.trend.1
    )
  ) +
  geom_line(
    aes(
      y=ajuste_2
    ),
    colour="red"
  ) +
  labs(
    title="Datos sin tendencia + ajuste estacional"
  )

## Datos sin tendencia y sin estacionalidad
g3 <- ggplot(
  ventas_vf,
  aes(
    x=period
  )
) +
  geom_point(
    aes(
      y=res.trend.2
    )
  ) +
  geom_line(
    aes(
      y=ajuste_f
    ),
    colour="red"
  ) +
  labs(
    title="Datos sin tendencia y sin estacionalidad"
  )

## Datos finales
g4 <- ggplot(
  ventas_vf,
  aes(
    x=period
  )
) +
  geom_point(

```

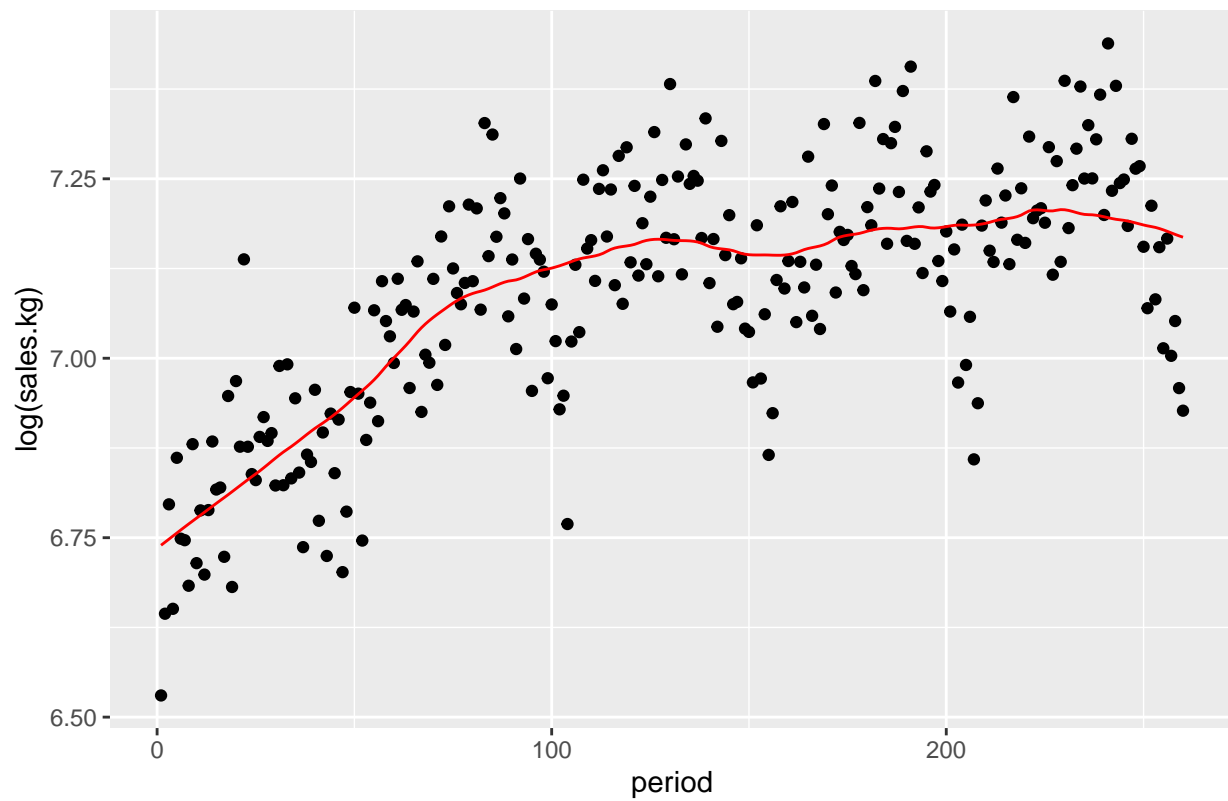
```

aes(
  y=res.trend.f
)
) +
labs(
  title="Datos finales"
)

print(g1)

```

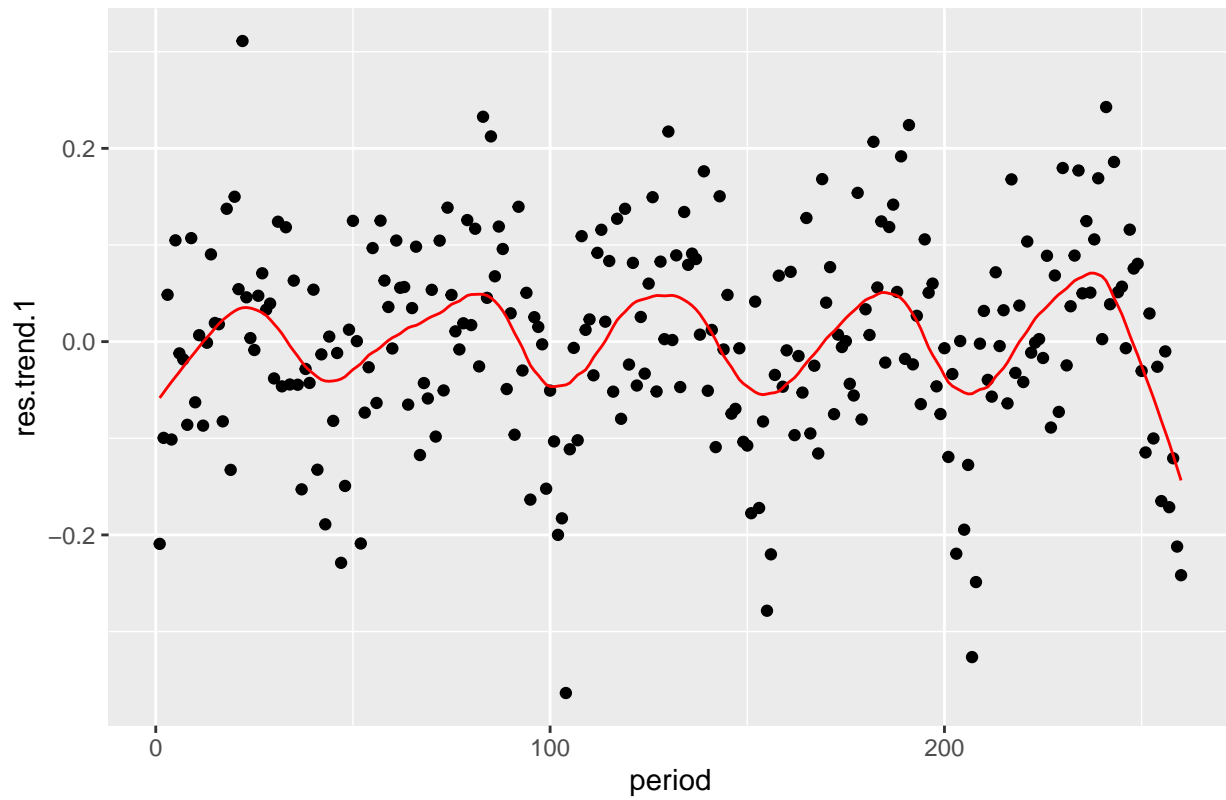
Datos originales + ajuste a tendencia



```
print(g2)
```

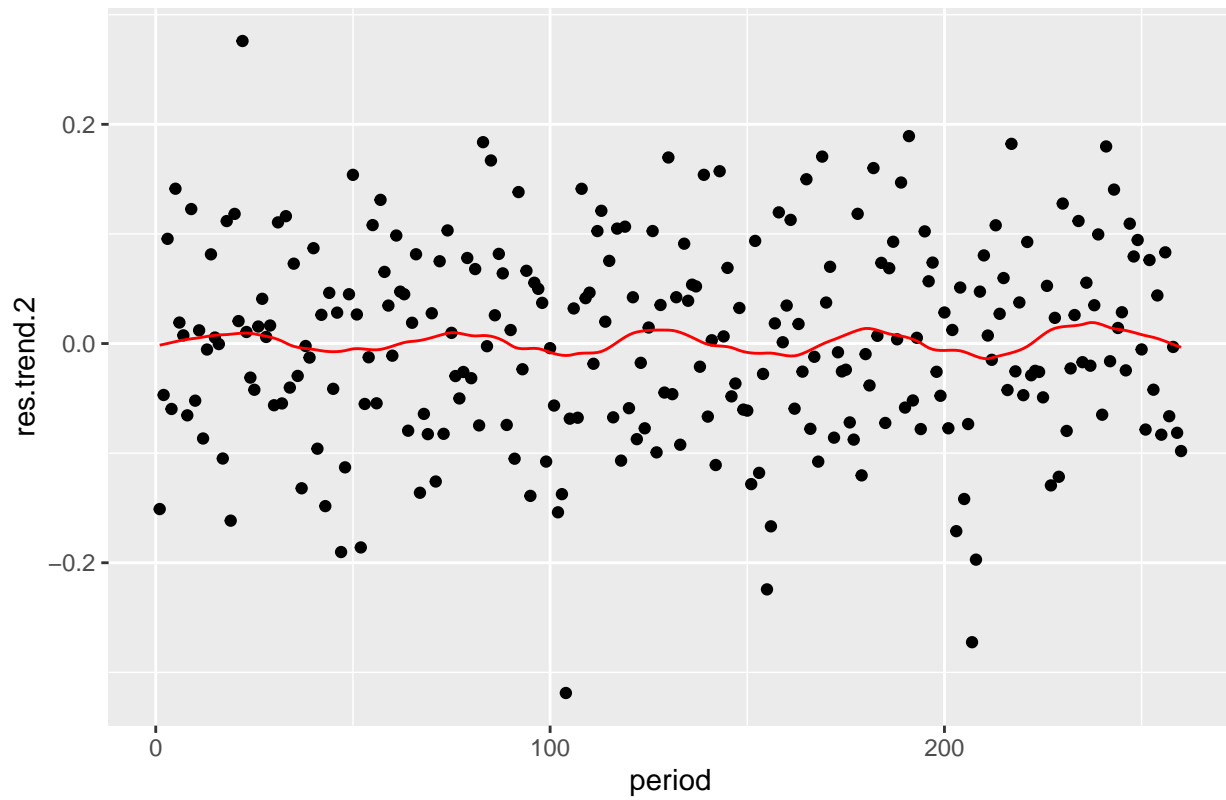


Datos sin tendencia + ajuste estacional

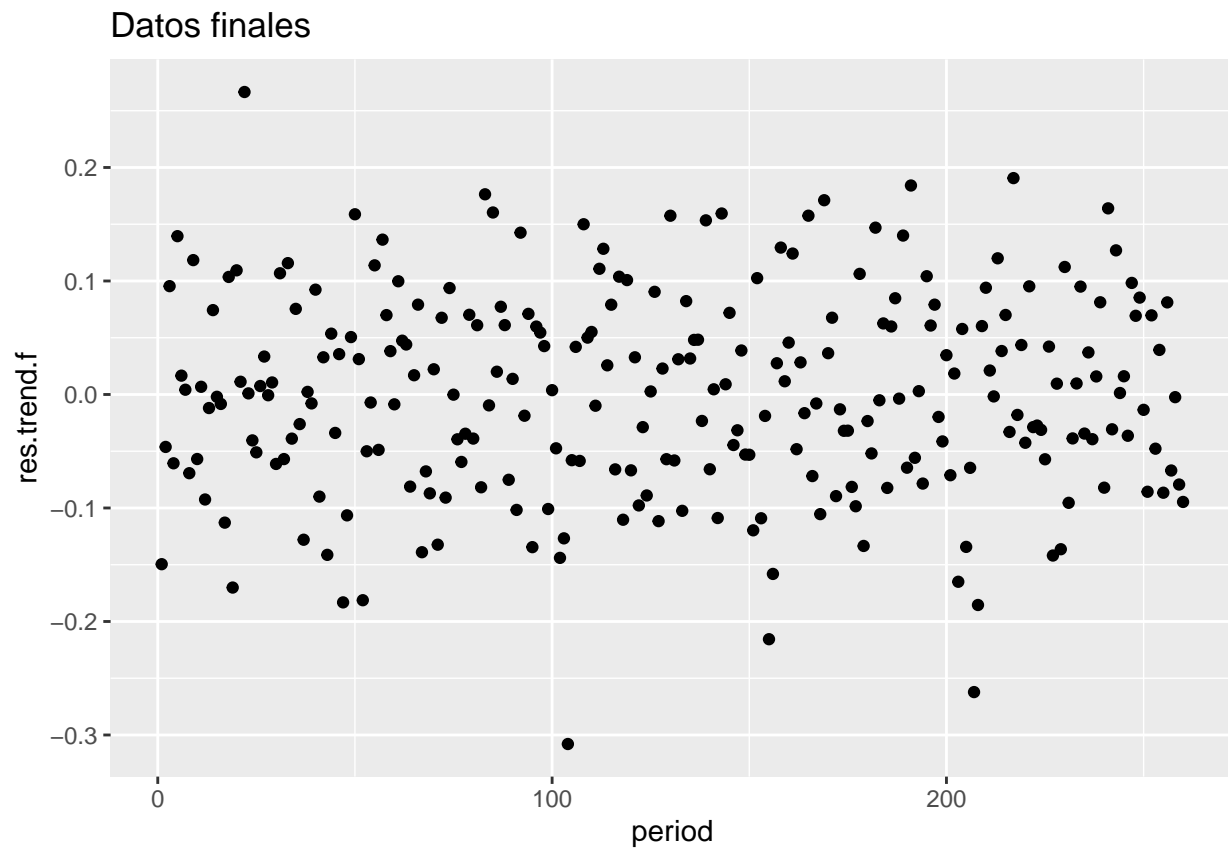


```
print(g3)
```

### Datos sin tendencia y sin estacionalidad



```
print(g4)
```



7. Genera una gráfica de cuantiles para los residuales.

```
## Gráfica de los residuales
ggplot(
  ventas_vf,
  aes(
    sample = res.trend.f
  )
) +
  geom_qq(
    distribution = stats::qunif
  ) +
  ylab("Ventas (residual)") +
  xlab("")
```

