MONASH University

MONASH BUSINESS SCHOOL

Tidy Time Series & Forecasting in R

tsibble feasts fable

3. Transformations

robjhyndman.com/workshop2020

Outline

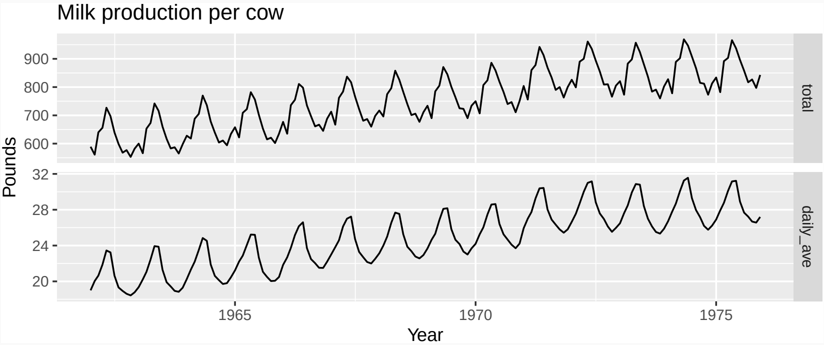
1. Calendar adjustments
2. Per capita adjustments
3. Inflation adjustments
4. Mathematical transformations
5. Lab Session 6

Outline

1. Calendar adjustments
2. Per capita adjustments
3. Inflation adjustments
4. Mathematical transformations
5. Lab Session 6

Calendar adjustments

```
as_tsibble(fma::milk) %>%
  rename(total = value) %>%
  mutate(daily_ave = total / days_in_month(as_date(index))) %>%
  gather(key="Series", value="Milk", factor_key=TRUE) %>%
  ggplot(aes(x=index, y=Milk)) + geom_line() +
    facet_grid(Series ~ ., scales='free') + xlab("Year") +
    ylab("Pounds") +  ggtitle("Milk production per cow")
```

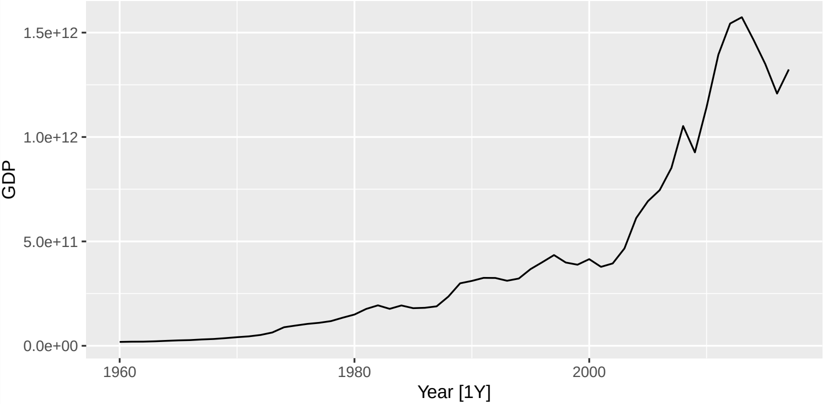


Outline

1. Calendar adjustments
2. Per capita adjustments
3. Inflation adjustments
4. Mathematical transformations
5. Lab Session 6

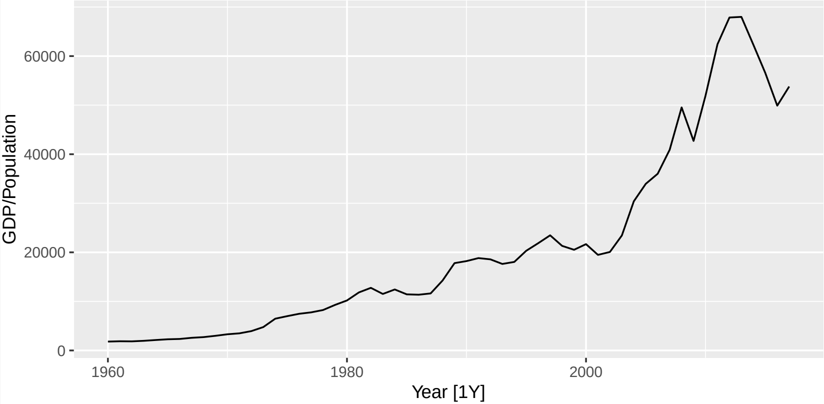
Per capita adjustments

```r
global_economy %>%
  filter(Country == "Australia") %>%
  autoplot(GDP)
```



Per capita adjustments

```
global_economy %>%
  filter(Country == "Australia") %>%
  autoplot(GDP / Population)
```



Outline

1 Calendar adjustments

2 Per capita adjustments

3 Inflation adjustments

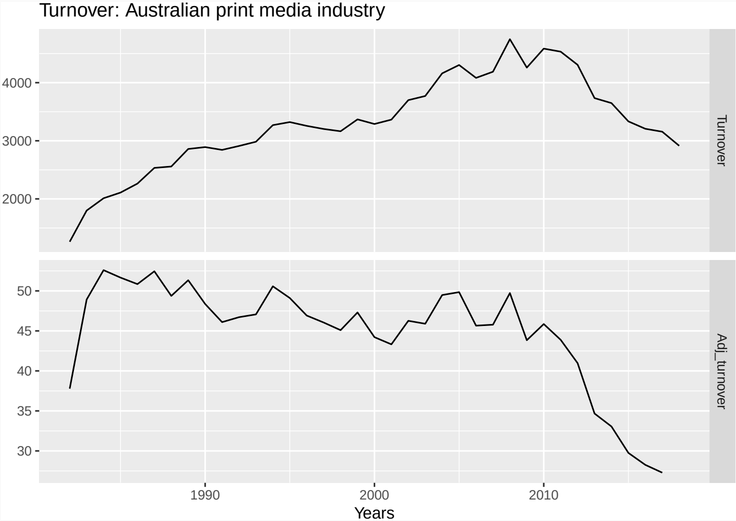
4 Mathematical transformations

5 Lab Session 6

Inflation adjustments

```
print_retail <- aus_retail %>%
  filter(Industry == "Newspaper and book retailing") %>%
  group_by(Industry) %>%
  index_by(Year = year(Month)) %>%
  summarise(Turnover = sum(Turnover))
aus_economy <- filter(global_economy, Code == "AUS")
print_retail %>%
  left_join(aus_economy, by = "Year") %>%
  mutate(Adj_turnover = Turnover / CPI) %>%
  gather("Type", "Turnover", Turnover, Adj_turnover,
         factor_key = TRUE) %>%
  ggplot(aes(x = Year, y = Turnover)) +
    geom_line() +
    facet_grid(vars(Type), scales = "free_y") +
    xlab("Years") + ylab(NULL) +
    ggtitle("Turnover: Australian print media industry")
```

Inflation adjustments



Outline

1. Calendar adjustments
2. Per capita adjustments
3. Inflation adjustments
4. Mathematical transformations
5. Lab Session 6

Variance stabilization

If the data show different variation at different levels of the series, then a transformation can be useful.

Variance stabilization

If the data show different variation at different levels of the series, then a transformation can be useful.

Denote original observations as $y_1, \ldots, y_n$ and transformed observations as $w_1, \ldots, w_n$.

Variance stabilization

If the data show different variation at different levels of the series, then a transformation can be useful.

Denote original observations as $y_1, \dots, y_n$ and transformed observations as $w_1, \dots, w_n$.

Mathematical transformations for stabilizing variation

Square root	$w_t = \sqrt{y_t}$	$\downarrow$
Cube root	$w_t = \sqrt[3]{y_t}$	Increasing
Logarithm	$w_t = \log(y_t)$	strength

Variance stabilization

If the data show different variation at different levels of the series, then a transformation can be useful.

Denote original observations as $y_1, \dots, y_n$ and transformed observations as $w_1, \dots, w_n$.

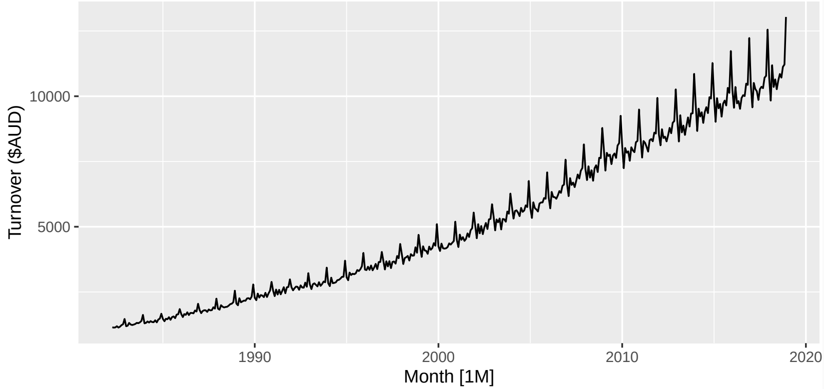
Mathematical transformations for stabilizing variation

Square root	$w_t = \sqrt{y_t}$	$\downarrow$
Cube root	$w_t = \sqrt[3]{y_t}$	Increasing
Logarithm	$w_t = \log(y_t)$	strength

Logarithms, in particular, are useful because they are more interpretable: changes in a log value are **relative (percent) changes on the original scale**.

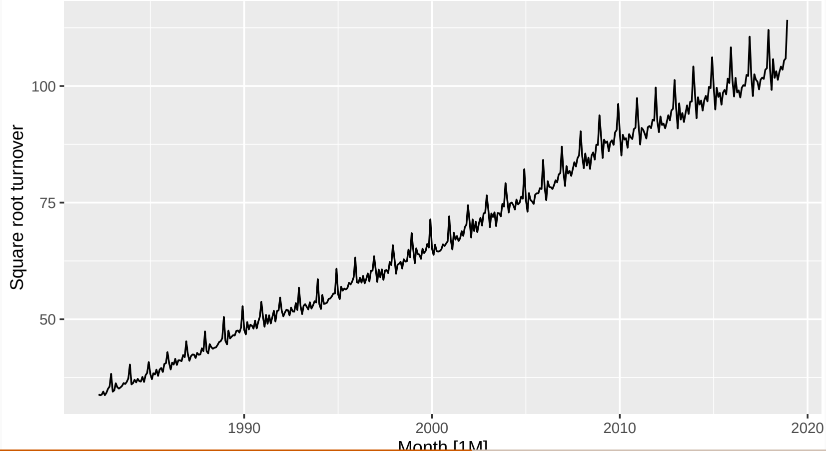
Variance stabilization

```
food <- aus_retail %>%
  filter(Industry == "Food retailing") %>%
  summarise(Turnover = sum(Turnover))
```



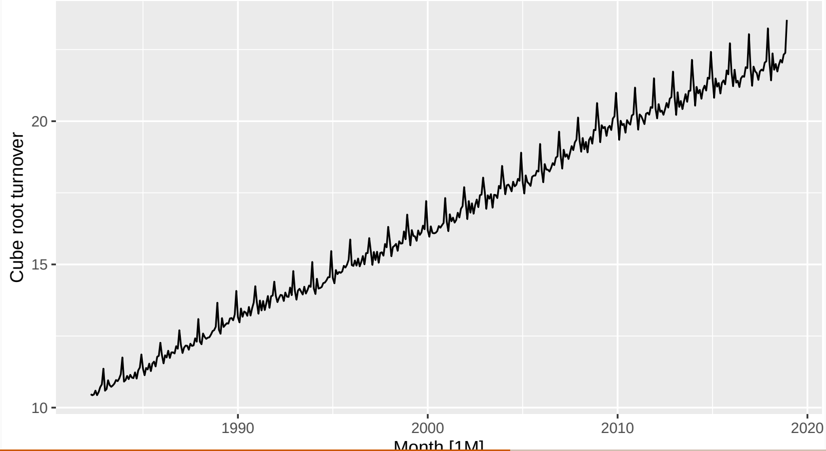
Variance stabilization

```r
food %>% autoplot(sqrt(Turnover)) +
  labs(y = "Square root turnover")
```



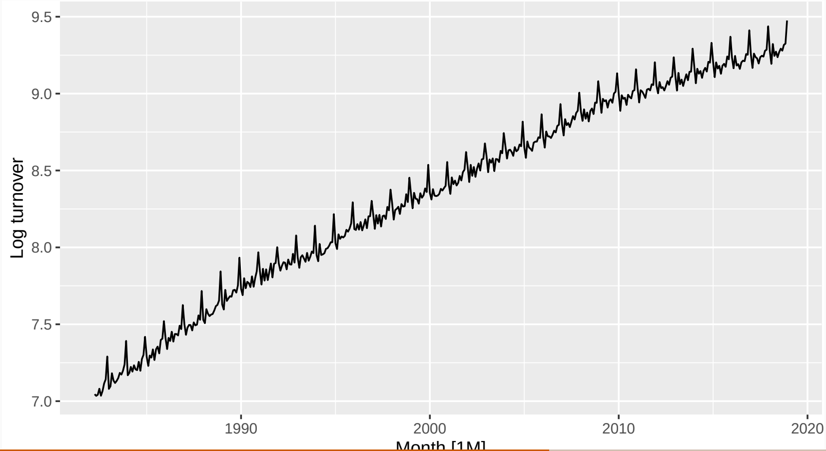
Variance stabilization

```
food %>% autoplot(Turnover^(1/3)) +
  labs(y = "Cube root turnover")
```



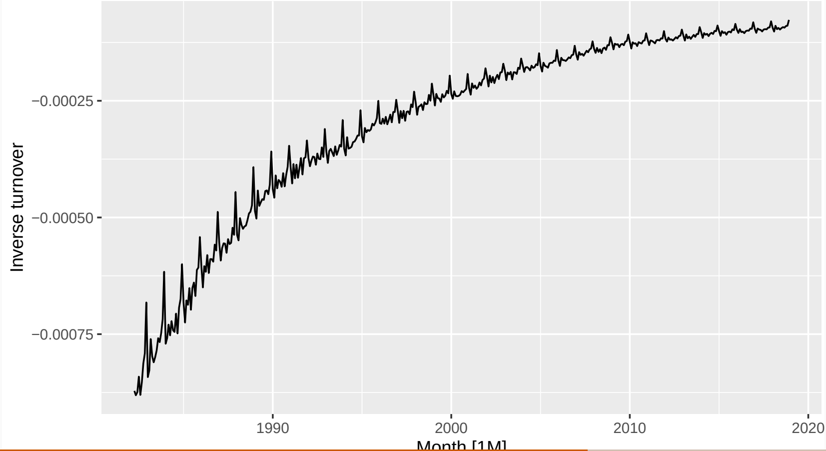
Variance stabilization

```r
food %>% autoplot(log(Turnover)) +
  labs(y = "Log turnover")
```



Variance stabilization

```
food %>% autoplot(-1/Turnover) +
  labs(y = "Inverse turnover")
```



Box-Cox transformations

Each of these transformations is close to a member of the family of **Box-Cox transformations**:

$$w_t = \begin{cases} \log(y_t), & \lambda = 0; \\ (y_t^\lambda - 1)/\lambda, & \lambda \neq 0. \end{cases}$$

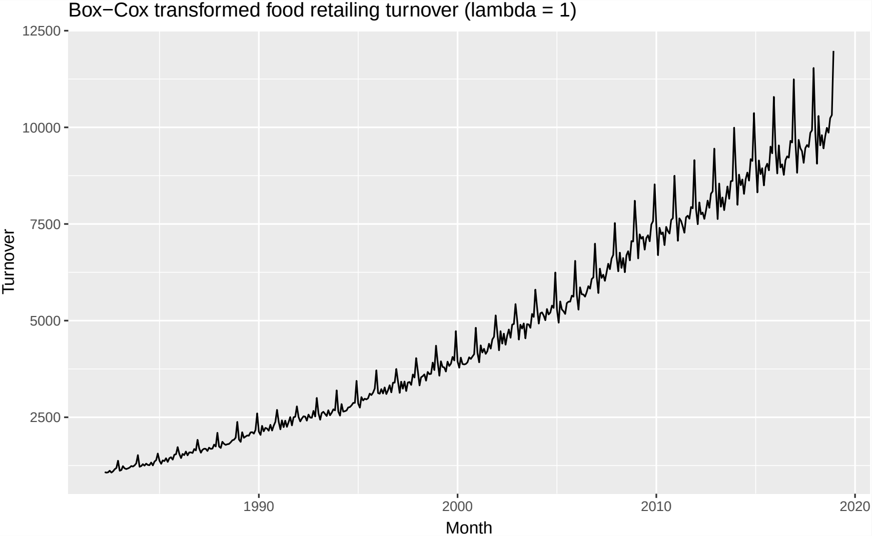
Box-Cox transformations

Each of these transformations is close to a member of the family of **Box-Cox transformations**:

$$w_t = \begin{cases} \log(y_t), & \lambda = 0; \\ (y_t^\lambda - 1)/\lambda, & \lambda \neq 0. \end{cases}$$

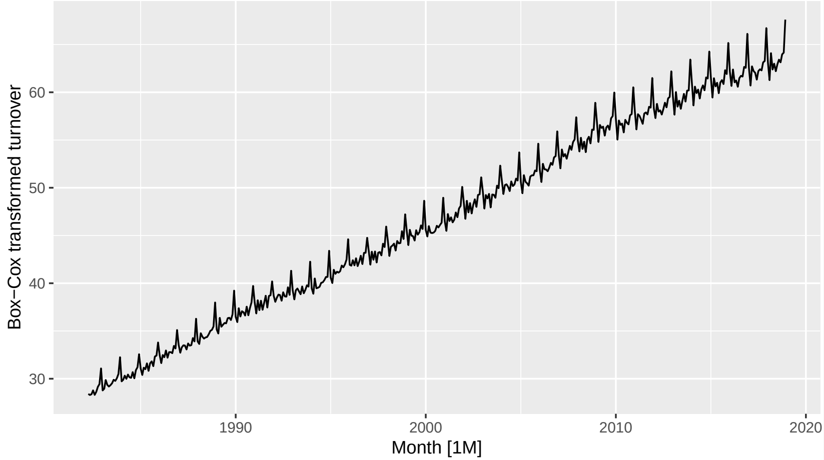
- $\lambda = 1$: (No substantive transformation)
- $\lambda = \frac{1}{2}$: (Square root plus linear transformation)
- $\lambda = 0$: (Natural logarithm)
- $\lambda = -1$: (Inverse plus 1)

Box-Cox transformations



Box-Cox transformations

```
food %>% autoplot(box_cox(Turnover, 1/3)) +
  labs(y = "Box-Cox transformed turnover")
```



Box-Cox transformations

- y_t^λ for λ close to zero behaves like logs.
- If some $y_t = 0$, then must have $\lambda > 0$
- if some $y_t < 0$, no power transformation is possible unless all y_t adjusted by **adding a constant to all values**.
- Simple values of λ are easier to explain.
- Results are relatively insensitive to λ.
- Often no transformation ($\lambda = 1$) needed.
- Transformation can have very large effect on PI.
- Choosing $\lambda = 0$ is a simple way to force forecasts to be positive

Box-Cox transformations

```
food %>%
  features(Turnover, features = guerrero)
```

```
## # A tibble: 1 x 1
##   lambda_guerrero
##             <dbl>
## 1          0.0524
```

Box-Cox transformations

```r
food %>%
  features(Turnover, features = guerrero)
```

```
## # A tibble: 1 x 1
##   lambda_guerrero
##             <dbl>
## 1          0.0524
```

- This attempts to balance the seasonal fluctuations and random variation across the series.
- Always check the results.
- A low value of λ can give extremely large prediction intervals.

Back-transformation

We must reverse the transformation (or *back-transform*) to obtain forecasts on the original scale. The reverse Box-Cox transformations are given by

$$y_t = \begin{cases} \exp(w_t), & \lambda = 0; \\ (\lambda W_t + 1)^{1/\lambda}, & \lambda \neq 0. \end{cases}$$

Outline

1 Calendar adjustments

2 Per capita adjustments

3 Inflation adjustments

4 Mathematical transformations

5 Lab Session 6

Lab Session 6

1. For the following series, find an appropriate Box-Cox transformation in order to stabilise the variance.
    - United States GDP from `global_economy`
    - Slaughter of Victorian “Bulls, bullocks and steers” in `aus_livestock`
    - Gas production from `aus_production`
2. Why is a Box-Cox transformation unhelpful for the `expsmooth::cangas` data?
3. For each of the following series, make a graph of the data. If transforming seems appropriate, do so and describe the effect. Tobacco from `aus_production`, Economy class passengers between Melbourne and Sydney from `ansett`, and Victorian Electricity Demand from `vic_elec`.