Tidy Time Series & Forecasting in R

3. Transformations

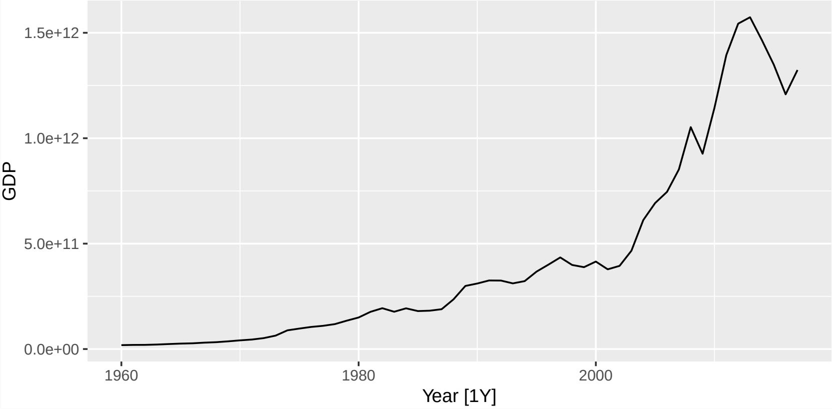
bit.ly/fable202

Outline

Outline

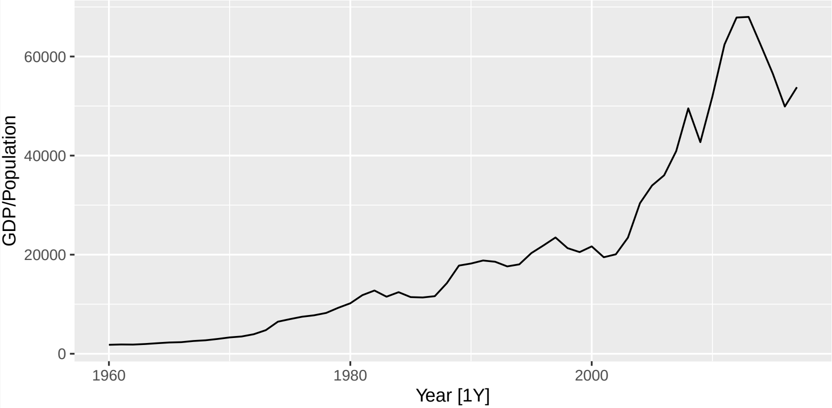
Per capita adjustments

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



Per capita adjustments

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



Outline

Lab Session 6

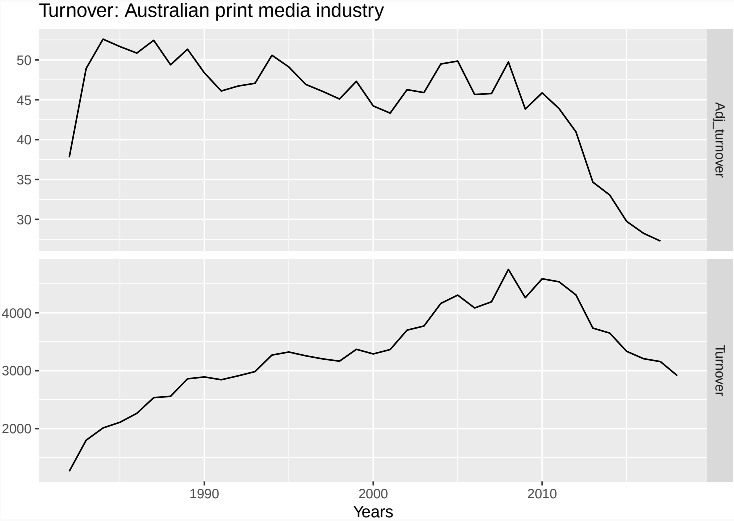
Consider the GDP information in `global_economy`. Plot the GDP per capita for each country over time. Which country has the highest GDP per capita? How has this changed over time?

Outline

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) %>%
  pivot_longer(c(Turnover, Adj_turnover),
    names_to = "Type", values_to = "Turnover"
  ) %>%
  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

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 $w_1, \dots, w_n$.

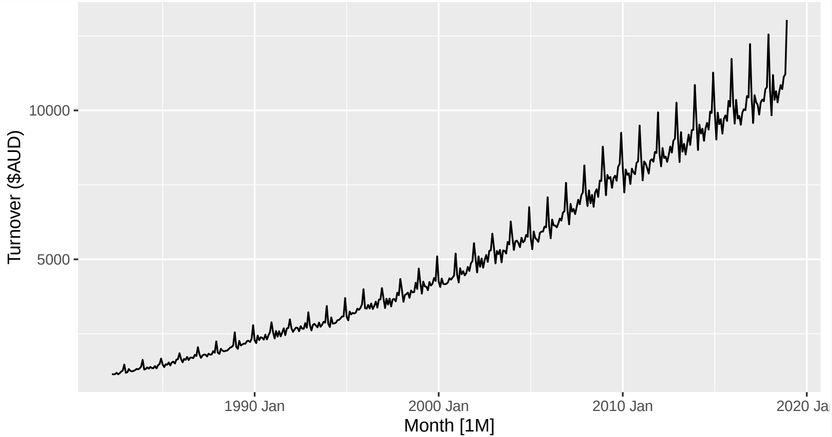
Mathematical transformations for stabilizing variation

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**.

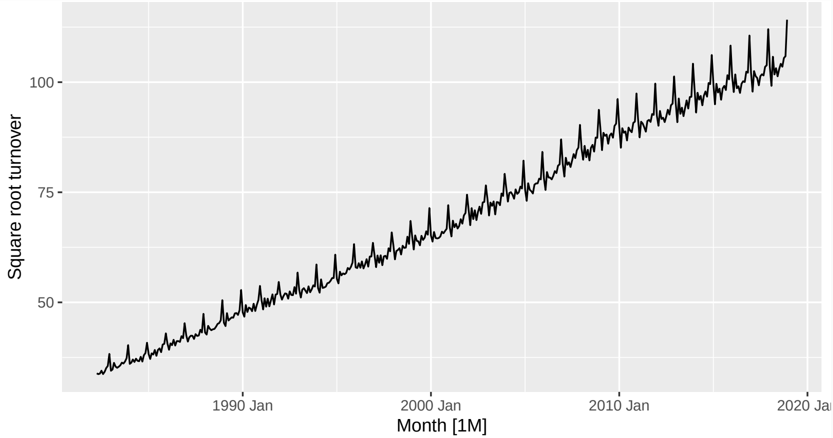
Variance stabilization

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



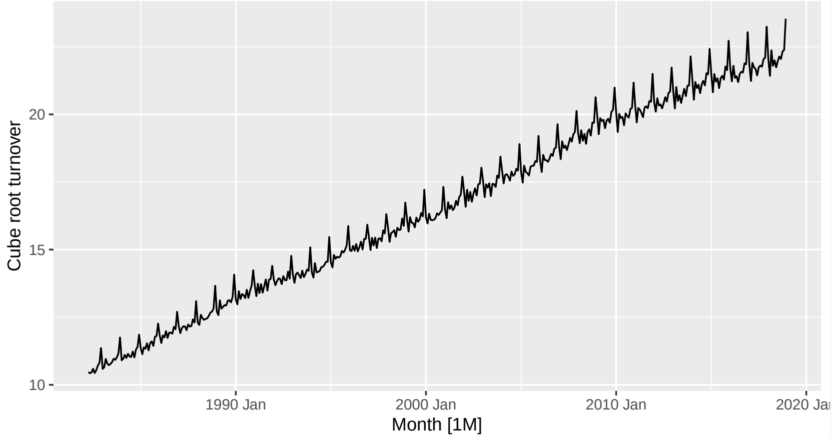
Variance stabilization

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



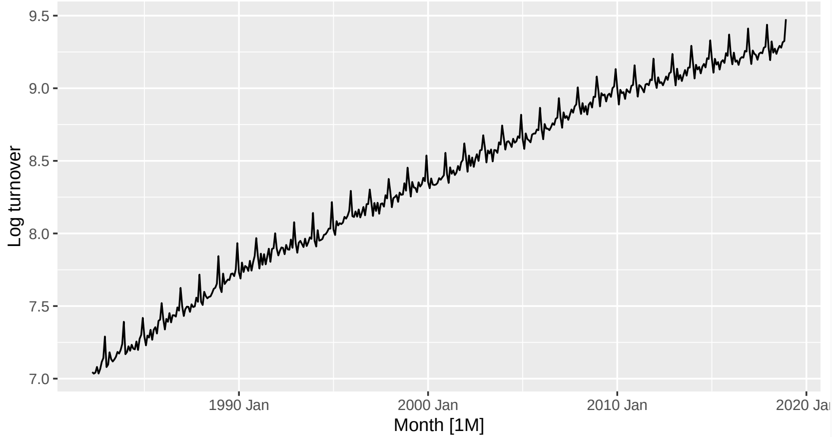
Variance stabilization

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



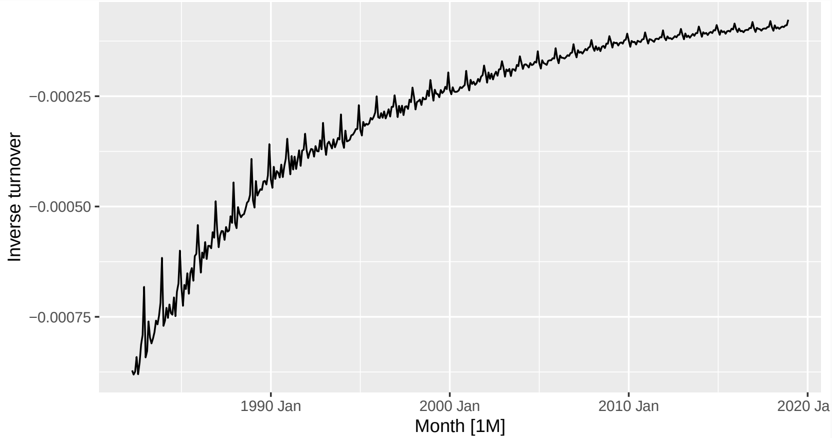
Variance stabilization

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



Variance stabilization

```r
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**:

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

- $\lambda = 1$: (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 %>%
  features(Turnover, features = guerrero)
```

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

Box-Cox transformations

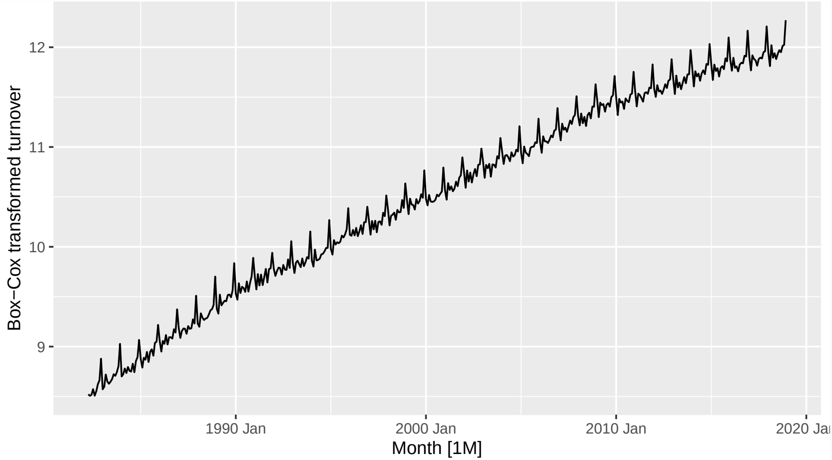
```
food %>%
  features(Turnover, features = guerrero)
```

```
## # 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.

Box-Cox transformations

```
food %>% autoplot(box_cox(Turnover, 0.0524)) +
  labs(y = "Box-Cox transformed turnover")
```



Transformations

- Often no transformation needed.
- Simple transformations are easier to explain and work well enough.
- Transformations can have very large effect on PI.
- If the data contains zeros, then don't take logs.
- `logp1()` can be useful for data with zeros.
- If some data are negative, no power transformation is possible unless a constant is added to all values.
- Choosing logs is a simple way to force forecasts to be positive
- Transformations must be reversed to obtain forecasts on the original scale. (Handled

Outline

Lab Session 7

1. For the following series, find an appropriate transformation in order to stabilise the variance.
    - United States GDP from `global_economy`
    - Slaughter of Victorian "Bulls, bullocks and steers" in `aus_livestock`
    - Victorian Electricity Demand from `vic_elec`.
    - Gas production from `aus_production`
2. Why is a Box-Cox transformation unhelpful for the `canadian_gas` data?