

Rob J Hyndman
George Athanasopoulos

FORECASTING PRINCIPLES AND PRACTICE

A comprehensive introduction to the latest forecasting methods using R. Learn to improve your forecast accuracy using dozens of real data examples.



2. Time series graphics

2.8 Autocorrelation

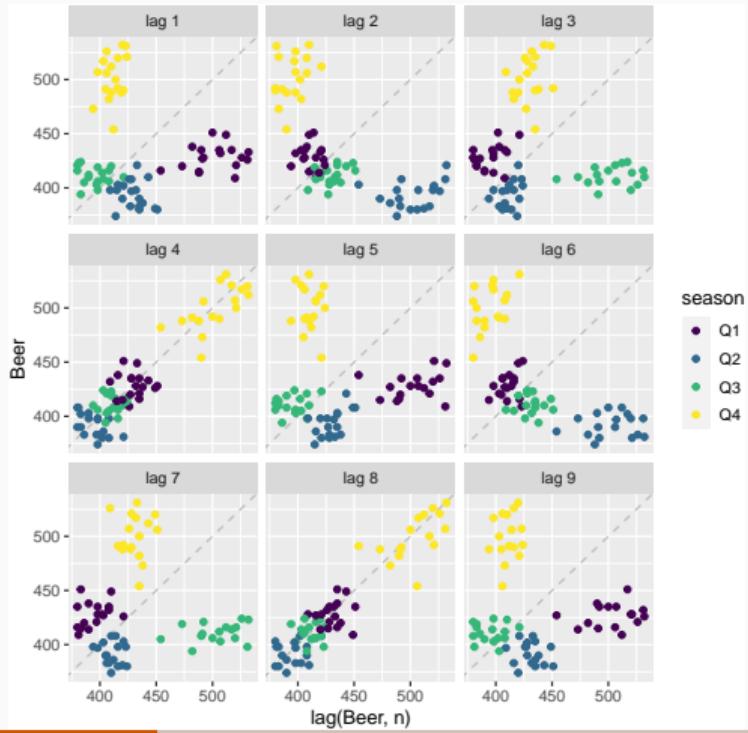
OTexts.org/fpp3/

3RD EDITION

O Texts
ONLINE OPEN-ACCESS TEXTBOOK

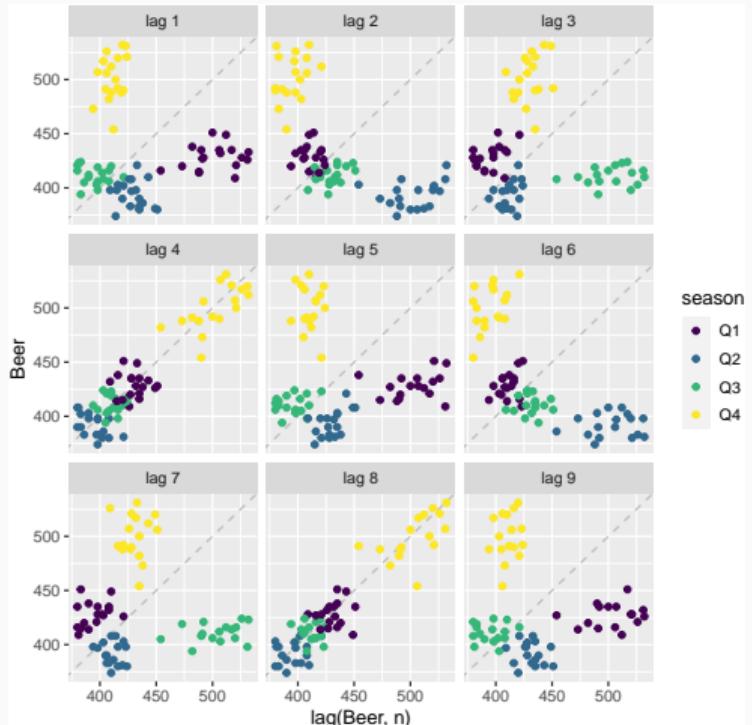
Example: Beer production

```
new_production |> gg_lag(Beer, geom = "point")
```



Example: Beer production

```
new_production |> gg_lag(Beer, geom = "point")
```



- Each graph shows y_t plotted against y_{t-k} for different values of k .
- The autocorrelations are the correlations associated with these scatterplots.
 - $r_1 = \text{Correlation}(y_t, y_{t-1})$
 - $r_2 = \text{Correlation}(y_t, y_{t-2})$
 - $r_3 = \text{Correlation}(y_t, y_{t-3})$
 - \vdots

Autocorrelation

We denote the sample autocovariance at lag k by c_k and the sample autocorrelation at lag k by r_k . Then define

$$c_k = \frac{1}{T} \sum_{t=k+1}^T (y_t - \bar{y})(y_{t-k} - \bar{y})$$

and $r_k = c_k/c_0$

Autocorrelation

We denote the sample autocovariance at lag k by c_k and the sample autocorrelation at lag k by r_k . Then define

$$c_k = \frac{1}{T} \sum_{t=k+1}^T (y_t - \bar{y})(y_{t-k} - \bar{y})$$

and $r_k = c_k/c_0$

- r_1 indicates how successive values of y relate to each other
- r_2 indicates how y values two periods apart relate to each other
- r_k is almost the same as the sample correlation between y_t and y_{t-k} .

Autocorrelation

Results for first 9 lags for beer data:

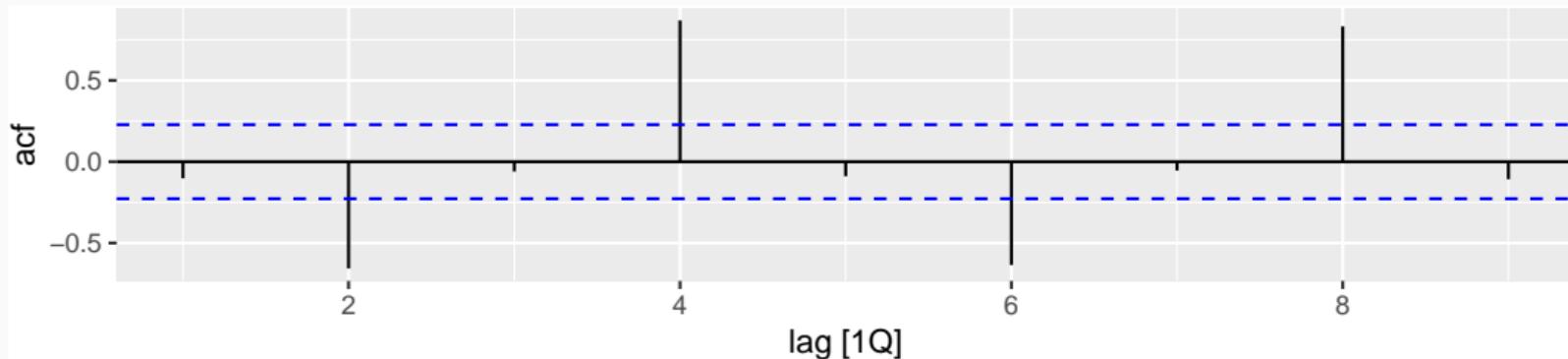
```
new_production |> ACF(Beer, lag_max = 9)
```

```
## # A tsibble: 9 x 2 [1Q]
##       lag      acf
##   <cf_lag>  <dbl>
## 1 1Q -0.102
## 2 2Q -0.657
## 3 3Q -0.0603
## 4 4Q  0.869
## 5 5Q -0.0892
## 6 6Q -0.635
## 7 7Q -0.0542
```

Autocorrelation

Results for first 9 lags for beer data:

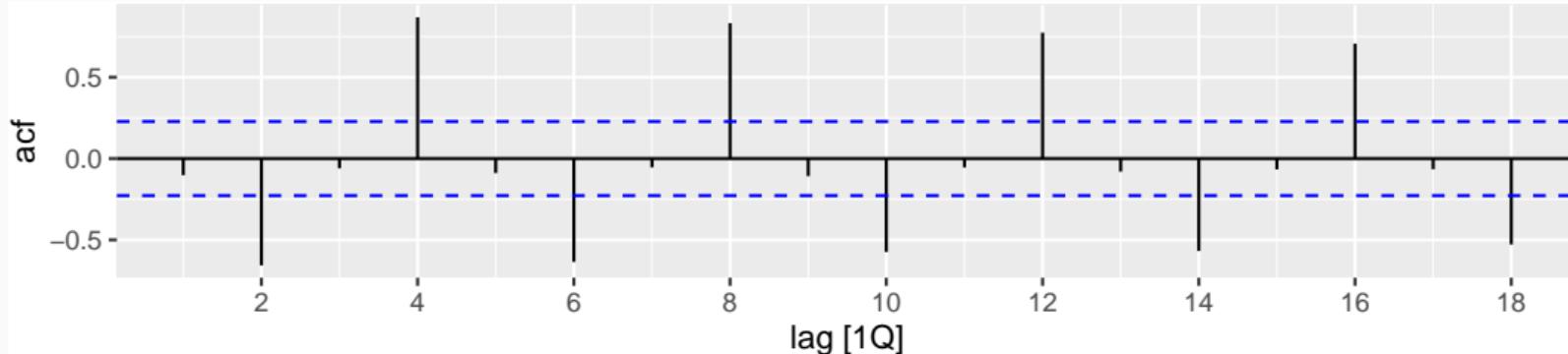
```
new_production |>  
ACF(Beer, lag_max = 9) |>  
autoplot()
```



- Together, the autocorrelations at lags 1, 2, ..., make up the *autocorrelation* or ACF.

Autocorrelation

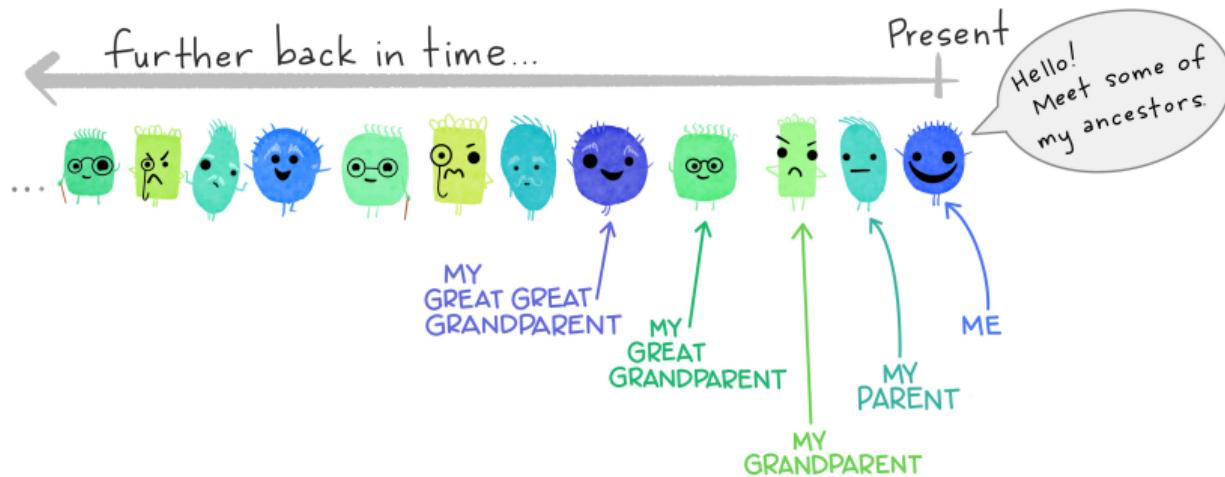
```
new_production |>  
  ACF(Beer) |>  
  autoplot()
```



- r_4 higher than for the other lags due to **the seasonal pattern in the data**: peaks tend to be **4 quarters** apart and troughs tend to be **4 quarters** apart.
- r_2 is more negative than for the other lags because troughs tend to

Autocorrelation functions

intro to the
autocorrelation function (ACF)

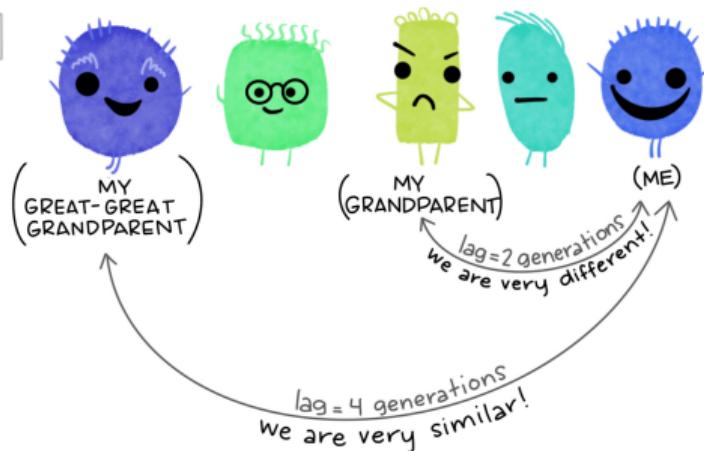


Autocorrelation functions

in our family MONSTERS tend to be...

- A little similar to their parent and great-grandparent
- Very different from their grandparent
- Very similar to their great-great grandparent

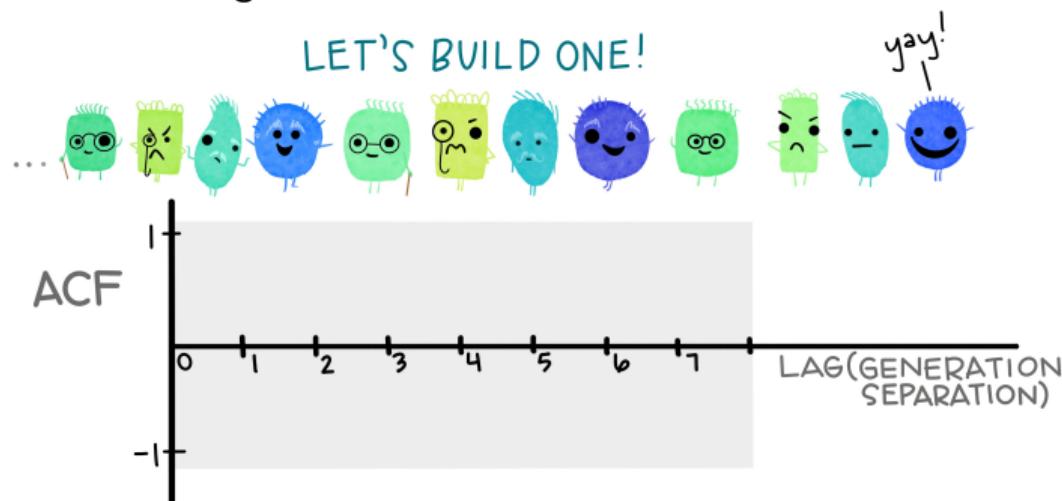
FOR EXAMPLE:



Autocorrelation functions

THE autocorrelation function (ACF)

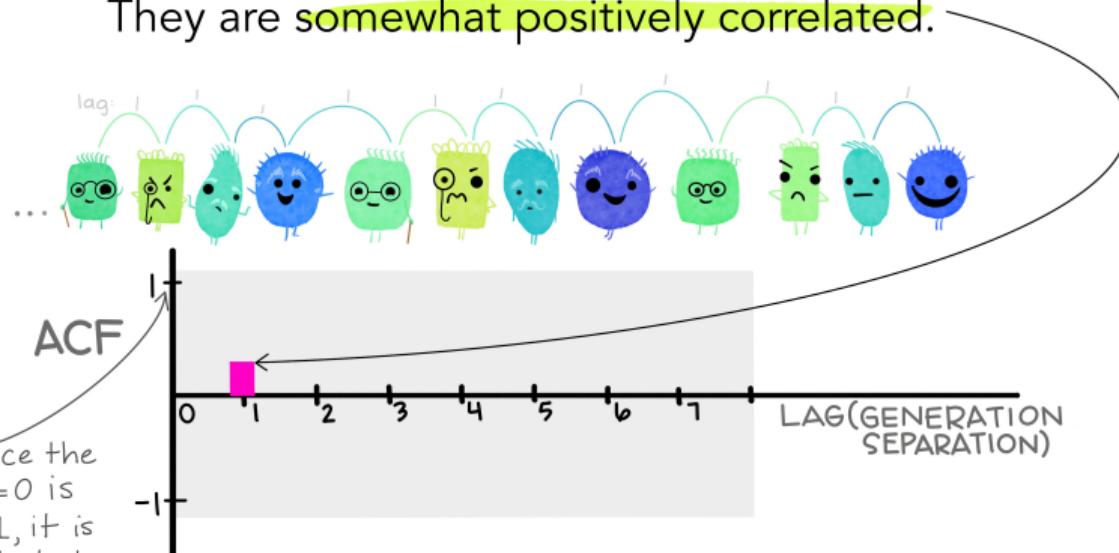
The ACF is a plot of autocorrelation between a variable and itself separated by specified lags (in our case, generations)



Autocorrelation functions

At lag = 1, we find the correlation between
monsters and their **parent**.

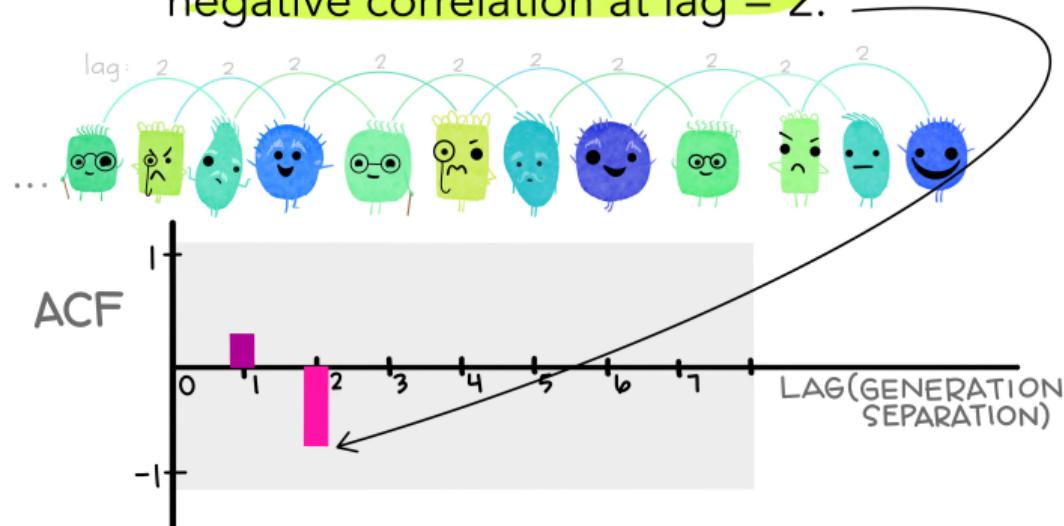
They are **somewhat positively correlated**.



Autocorrelation functions

At lag = 2, we find the correlation between
monsters and their **grandparent**.

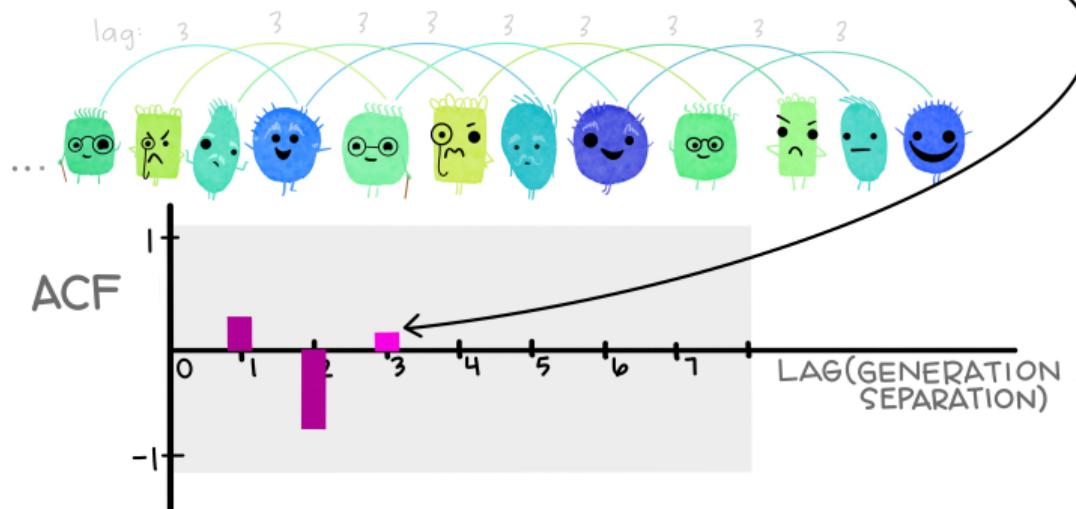
Since they tend to be very different, we find a
negative correlation at lag = 2.



Autocorrelation functions

At lag = 3, we find the correlation between
monsters and their **great-grandparent**.

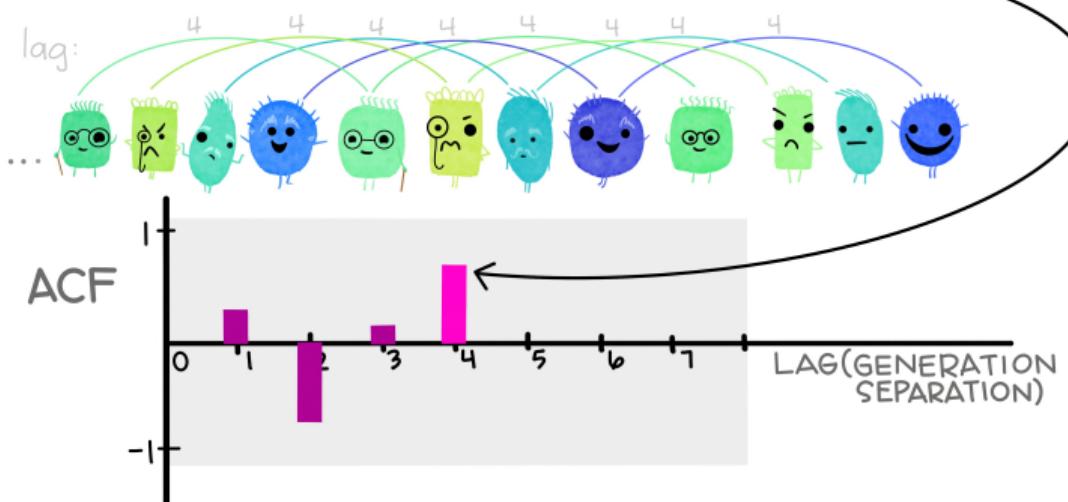
They are slightly positively correlated.



Autocorrelation functions

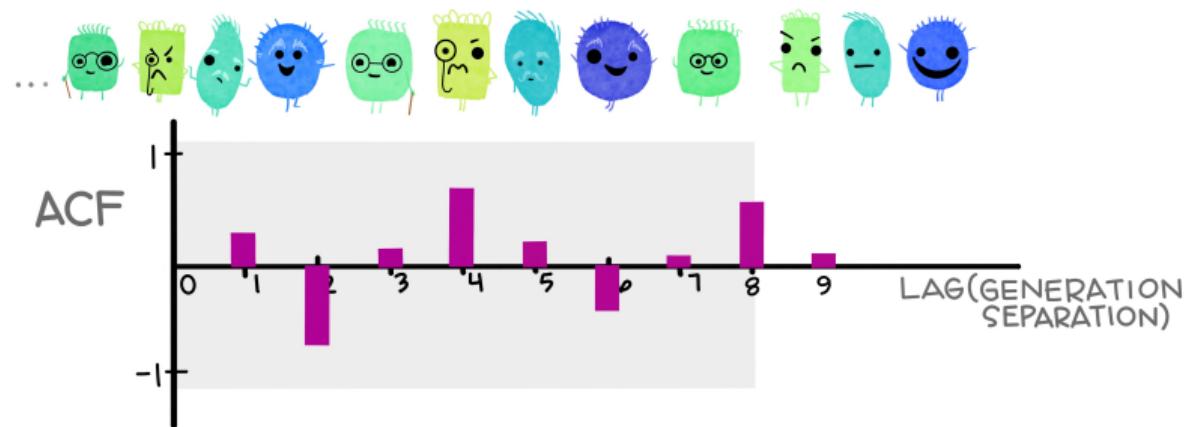
At lag = 4, we find the correlation between **monsters** and their **great-great grandparent**.

They tend to be very similar
(there is a positive correlation).



Autocorrelation functions

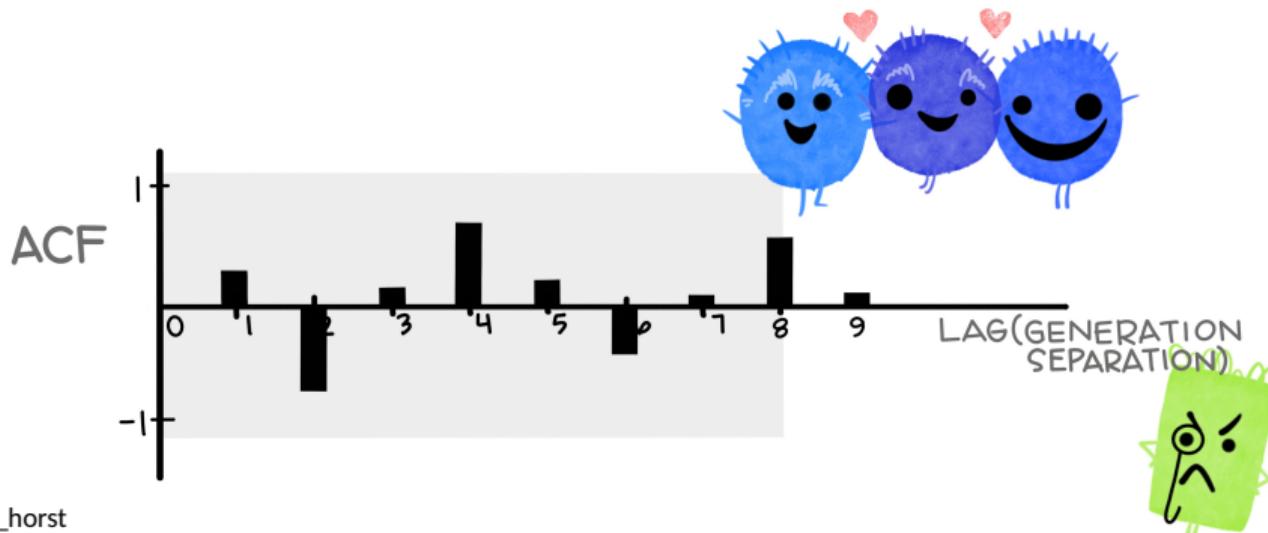
...and we continue finding the correlations as we increase the lag (generations) between the monsters...



Autocorrelation functions

in summary:

The autocorrelation function (ACF) tells us the correlation between observations and those that came before them, separated by different lags (here, monster generations)!

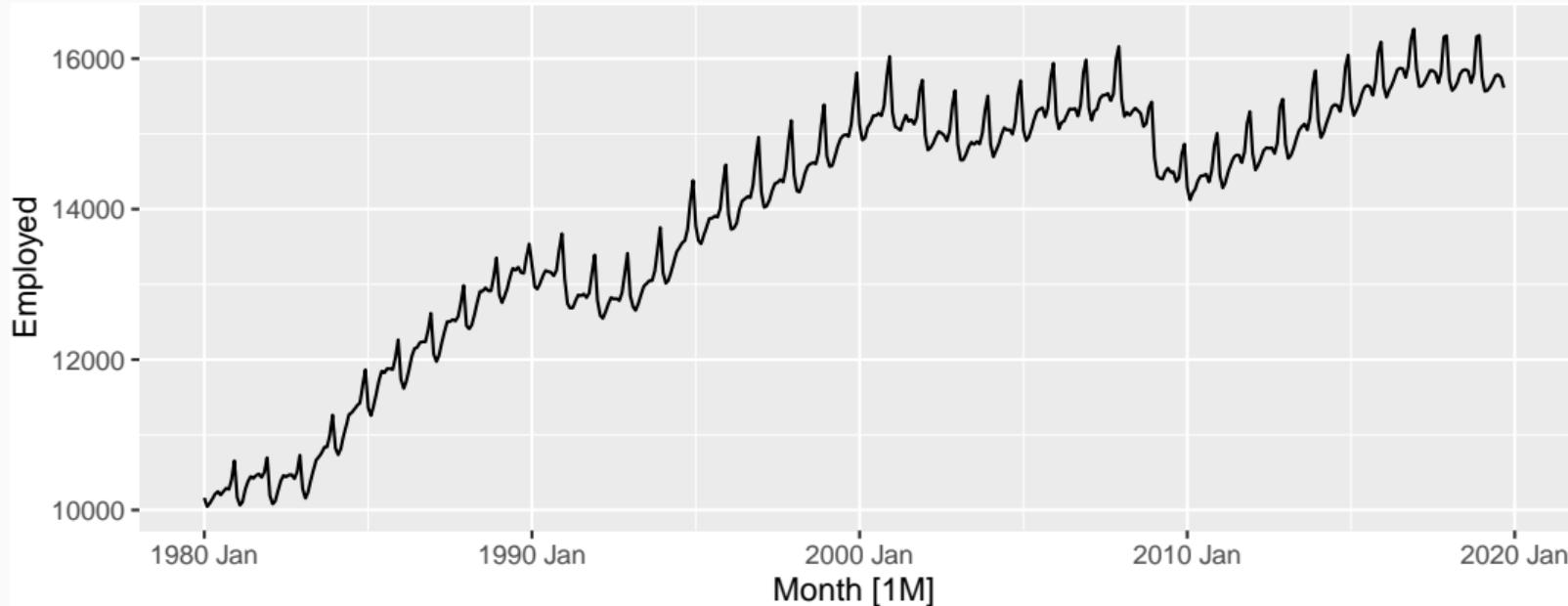


Trend and seasonality in ACF plots

- When data have a trend, the autocorrelations for small lags tend to be large and positive.
- When data are seasonal, the autocorrelations will be larger at the seasonal lags (i.e., at multiples of the seasonal frequency)
- When data are trended and seasonal, you see a combination of these effects.

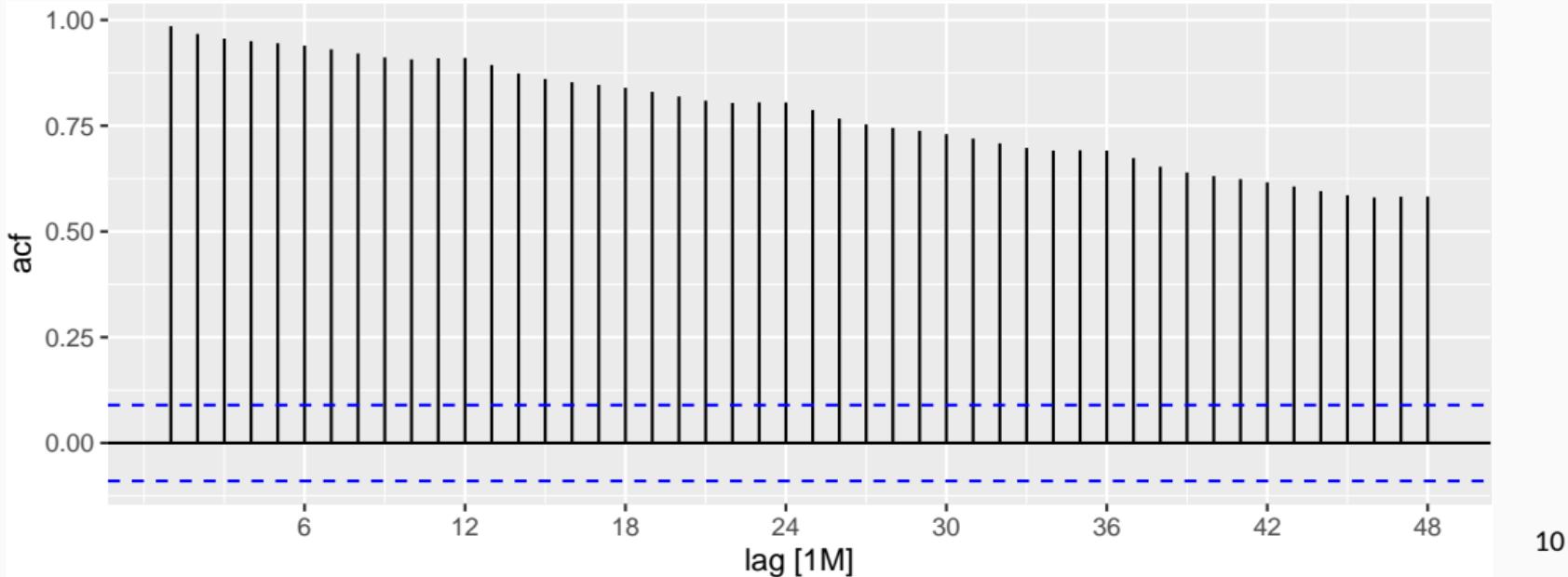
US retail trade employment

```
retail <- us_employment |>  
  filter>Title == "Retail Trade", year(Month) >= 1980  
retail |> autoplot(Employed)
```



US retail trade employment

```
retail |>  
  ACF(Employed, lag_max = 48) |>  
  autoplot()
```



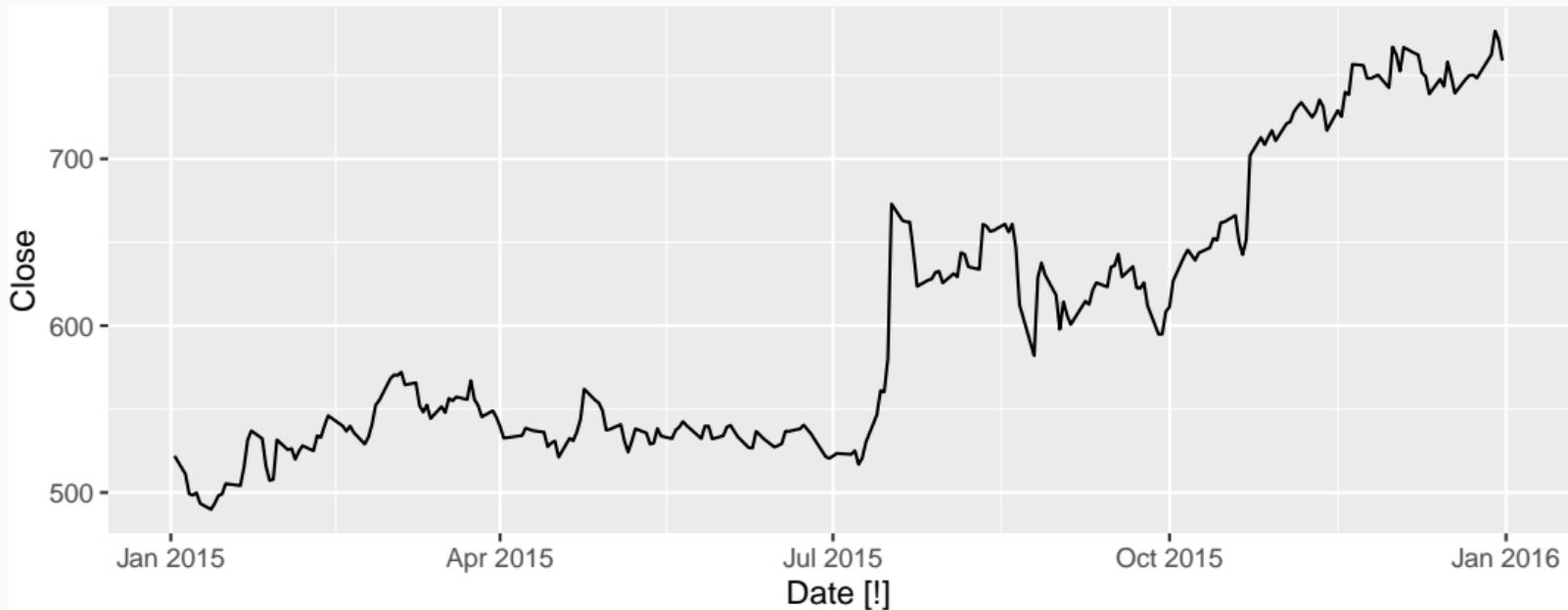
Google stock price

```
google_2015 <- gafa_stock |>
  filter(Symbol == "GOOG", year(Date) == 2015) |>
  select(Date, Close)
google_2015
```

```
## # A tsibble: 252 x 2 [!]
##   Date      Close
##   <date>    <dbl>
## 1 2015-01-02  522.
## 2 2015-01-05  511.
## 3 2015-01-06  499.
## 4 2015-01-07  498.
## 5 2015-01-08  500.
## 6 2015-01-09  493.
## 7 2015-01-12  490.
## 8 2015-01-13  493.
## 9 2015-01-14  498.
## 10 2015-01-15 499.
```

Google stock price

```
google_2015 |> autoplot(Close)
```



Google stock price

```
google_2015 |>  
  ACF(Close, lag_max = 100)
```

```
## # A tsibble: 100 x 2 [1]  
##       lag   acf  
##     <cf_lag> <dbl>  
## 1      1  0.982  
## 2      2  0.959  
## 3      3  0.937  
## 4      4  0.918  
## 5      5  0.901  
## 6      6  0.883  
## 7      7  0.865  
## 8      8  0.849  
## 9      9  0.834  
## 10    10  0.818  
## # ... with 90 more rows
```

Google stock price

```
google_2015 |>  
  ACF(Close, lag_max = 100) |>  
  autoplot()
```

