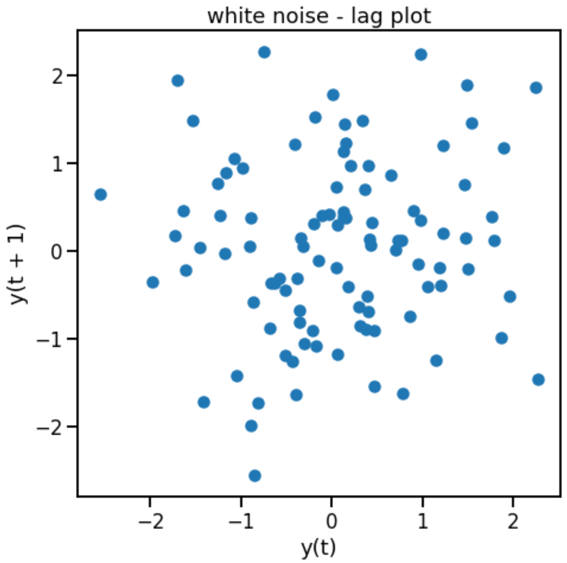
Autocorrelation function

Lag features

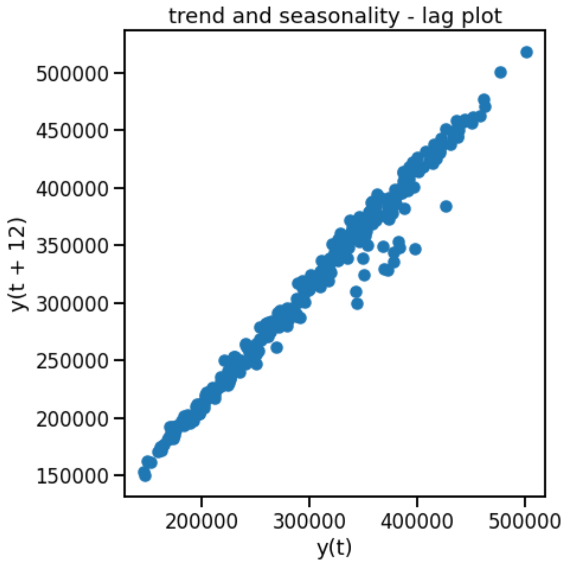
Contents

Lag plot limitations

- Lag plots are a visual tool which can help identify useful lags but are not scalable.
- If we quantify when y_{t-k} is highly correlated with y_t then it would be easier to identify useful lags.
- Autocorrelation is a method to quantify the correlation of a time series with itself.



Low autocorrelation



High autocorrelation

Pearson correlation coefficient

Measures the strength of the **linear** relationship between two variables

x	y
23	26
30	31
35	32
30	29

Mean

$$r_{xy} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

Pearson correlation coefficient

Measures the strength of the **linear** relationship between two variables

- r = 0: No correlation
- r > 0: Positive linear correlation
- r < 0: Negative linear correlation
- -1 ≤ r ≤ 1

$$r_{xy} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

Pearson correlation coefficient

Measures the strength of the **linear** relationship between two variables

- r = 0: No correlation
- r > 0: Positive linear correlation
- r < 0: Negative linear correlation
- -1 ≤ r ≤ 1

$$r_{xy} = \frac{\overbrace{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}^{\text{Covariance}}}{\underbrace{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}}_{\text{Standard deviation}}\ \underbrace{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2}}_{\text{Standard deviation}}}$$

Pearson correlation coefficient

Measures the strength of the **linear** relationship between two variables

- r = 0: No correlation
- r > 0: Positive linear correlation
- r < 0: Negative linear correlation
- -1 ≤ r ≤ 1

$$r_{xy} = \frac{\overbrace{cov(x,y)}^{\text{Covariance}}}{\sigma_x \, \sigma_y}$$

Standard deviation

Pearson correlation coefficient

Measures the strength of the **linear** relationship between two variables

- r = 0: No correlation
- r > 0: Positive linear correlation
- r < 0: Negative linear correlation
- -1 ≤ r ≤ 1

$$corr(x,y) = \frac{\overbrace{cov(x,y)}^{\text{Covariance}}}{\sigma_x \, \sigma_y}$$

Standard deviation

Pearson correlation coefficient

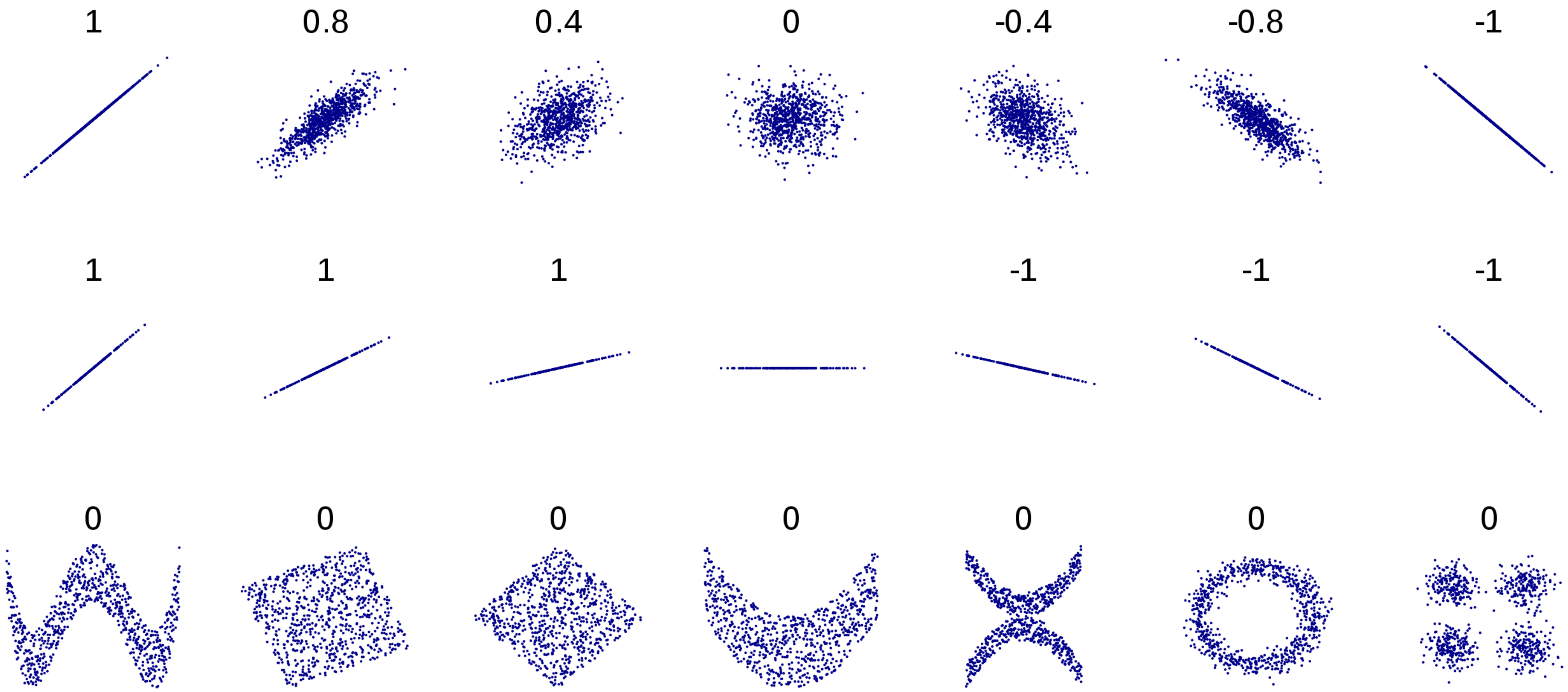


Image from: https://commons.wikimedia.org/wiki/File:Correlation_examples2.svg

Autocorrelation function (ACF)

The ACF is correlation of a time series y_t with a lagged version of itself y_{t-k}.

y	y Lag 2
23	NaN
30	NaN
35	23
30	30

$$corr(x,y) = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

Autocorrelation function (ACF)

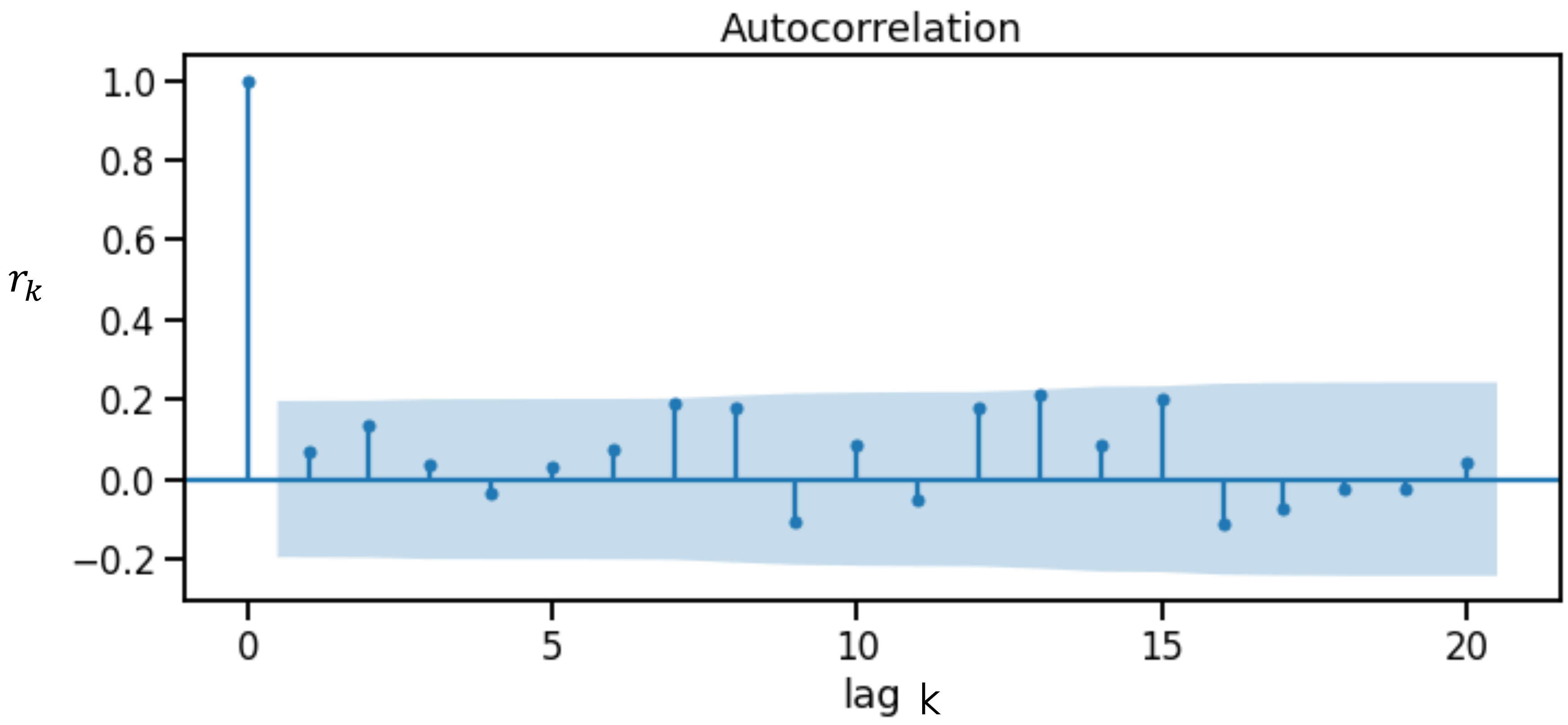
The ACF is correlation of a time series y_t with a lagged version of itself y_{t-k}.

If the autocorrelation at lag k is large then it might be helpful in forecasting.

y	y Lag 2
23	NaN
30	NaN
35	23
30	30

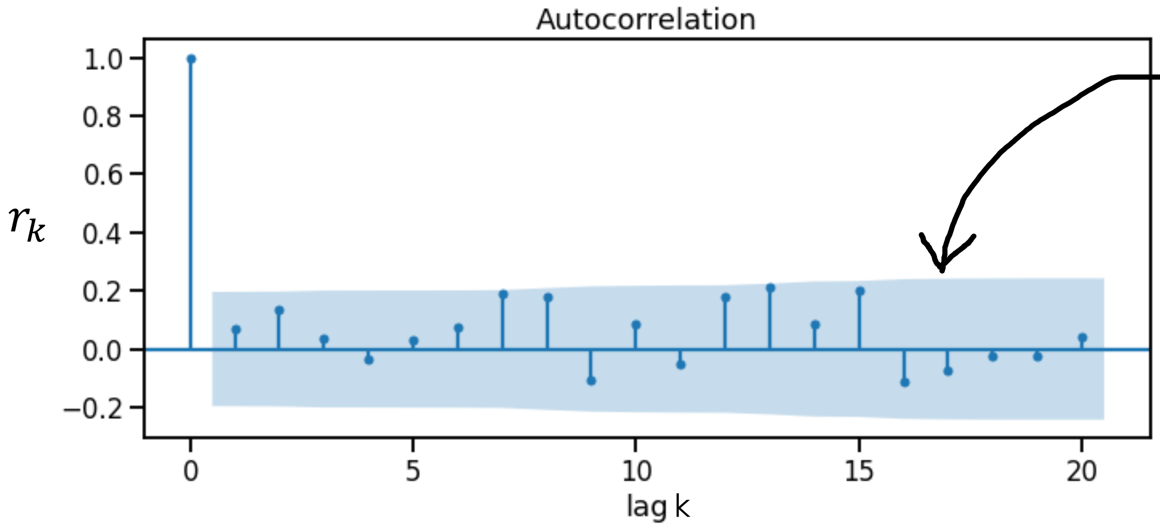
$$corr(y_t, y_{t-k}) = r_k = \frac{\sum_{t=k+1}^{N}(y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^{N}(y_t - \bar{y})^2}$$

Autcorrelogram



ACF: Confidence Intervals

- Is the r_k that is *estimated* from the data significantly different from zero?
- We can compute the confidence interval (CI) of r_k if it were generated by a random process.
- Bartlett's formula [1] provides a confidence interval (typically 95% CI used).
- If the r_k is outside of this interval we can conclude that r_k is significant.

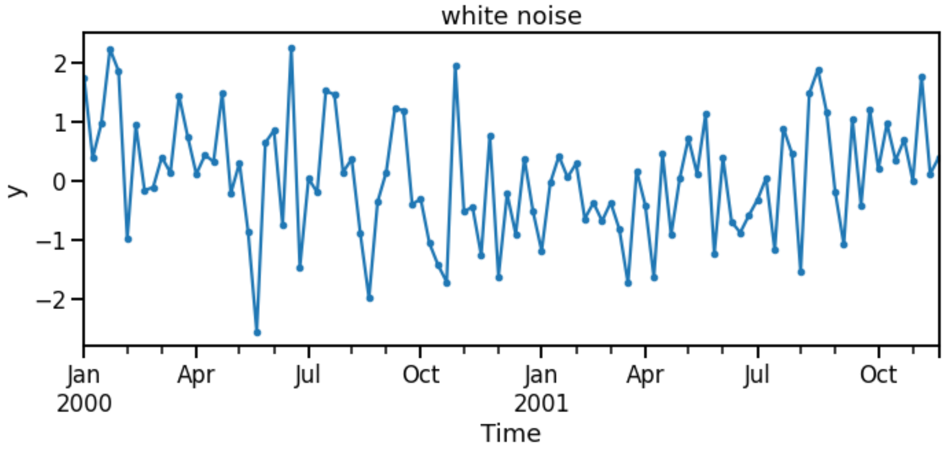


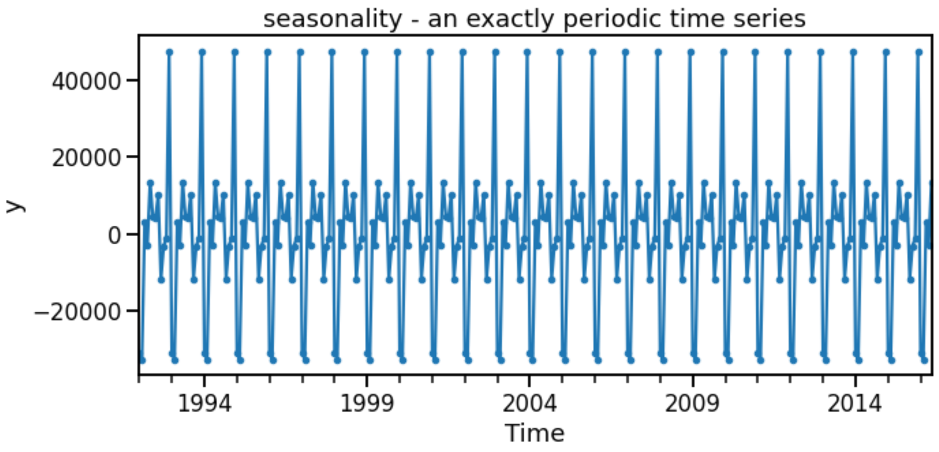
95% Confidence interval from Bartlett's formula used by default in Statsmodels shown by blue shaded region.

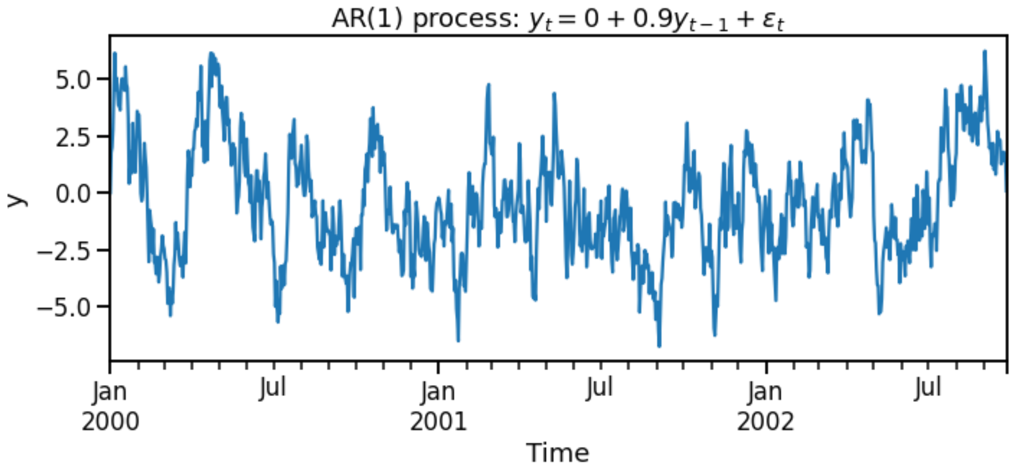
It is normally used as a guide to help analysis.

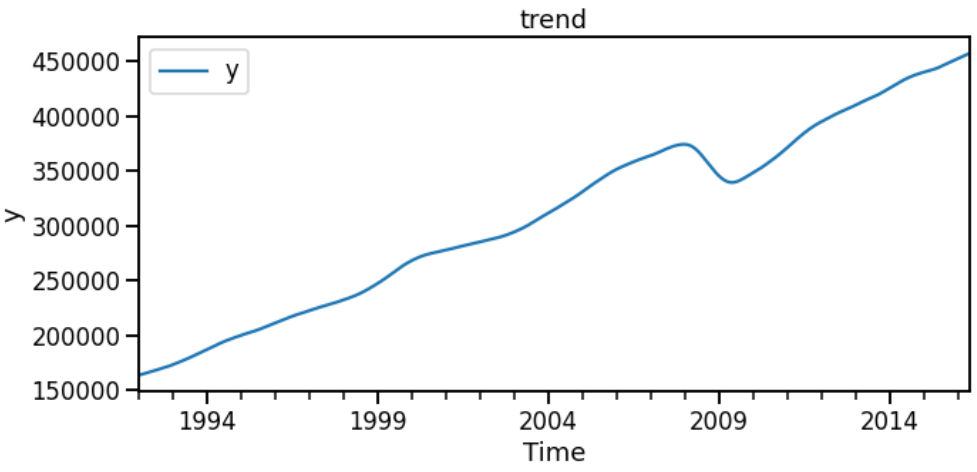
[1]- Brockwell and Davis, 2010. Introduction to Time Series and Forecasting, 2nd edition.

Let's look at the ACF for different time series

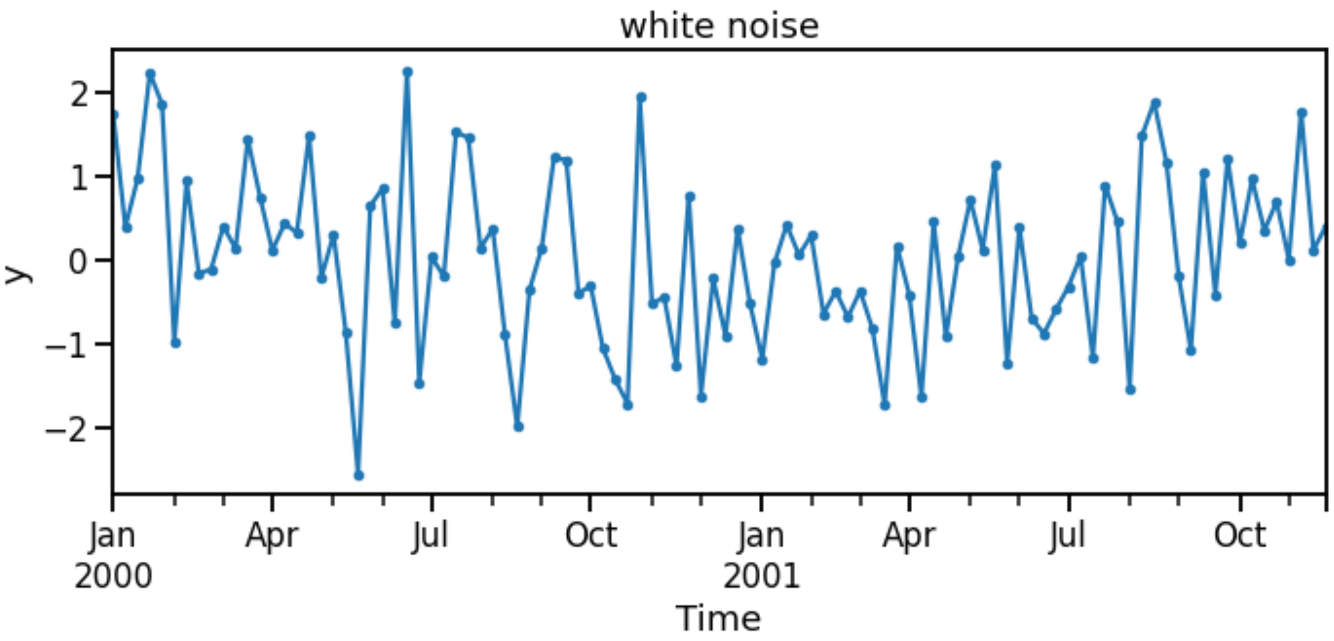


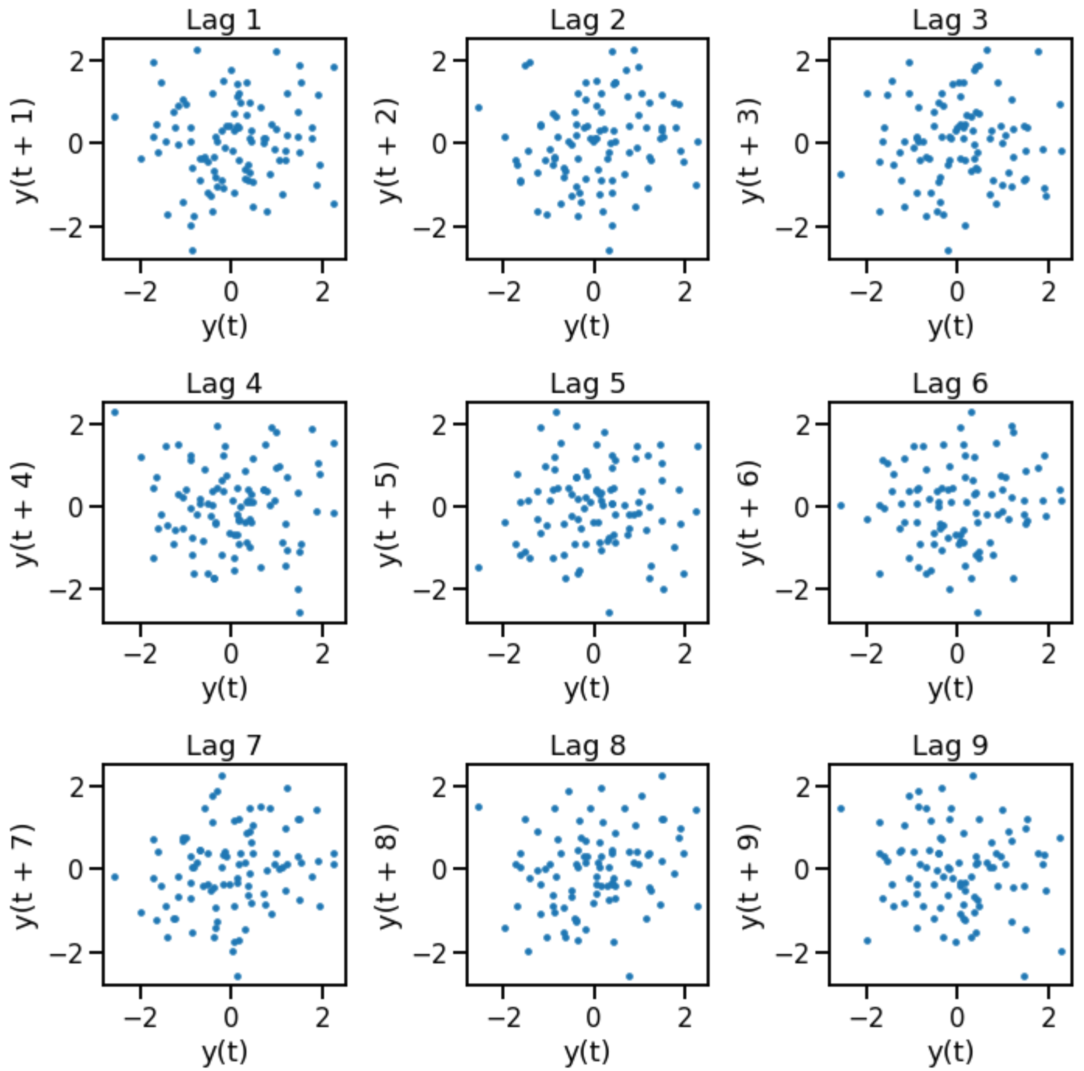






White noise



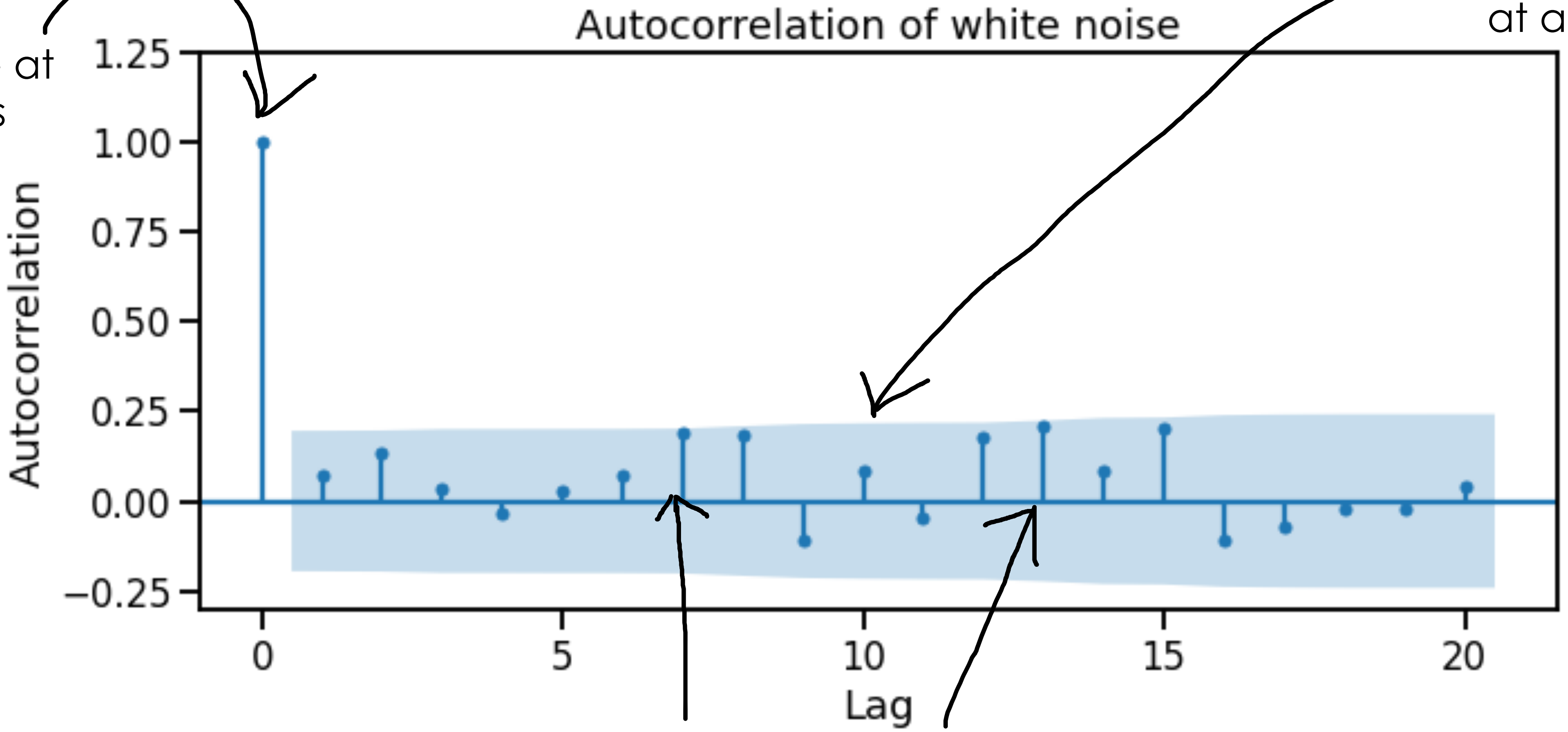


- $y_t = \epsilon_t$ where $\epsilon_t \sim N(0,1)$
- No correlation between points

ACF: white noise

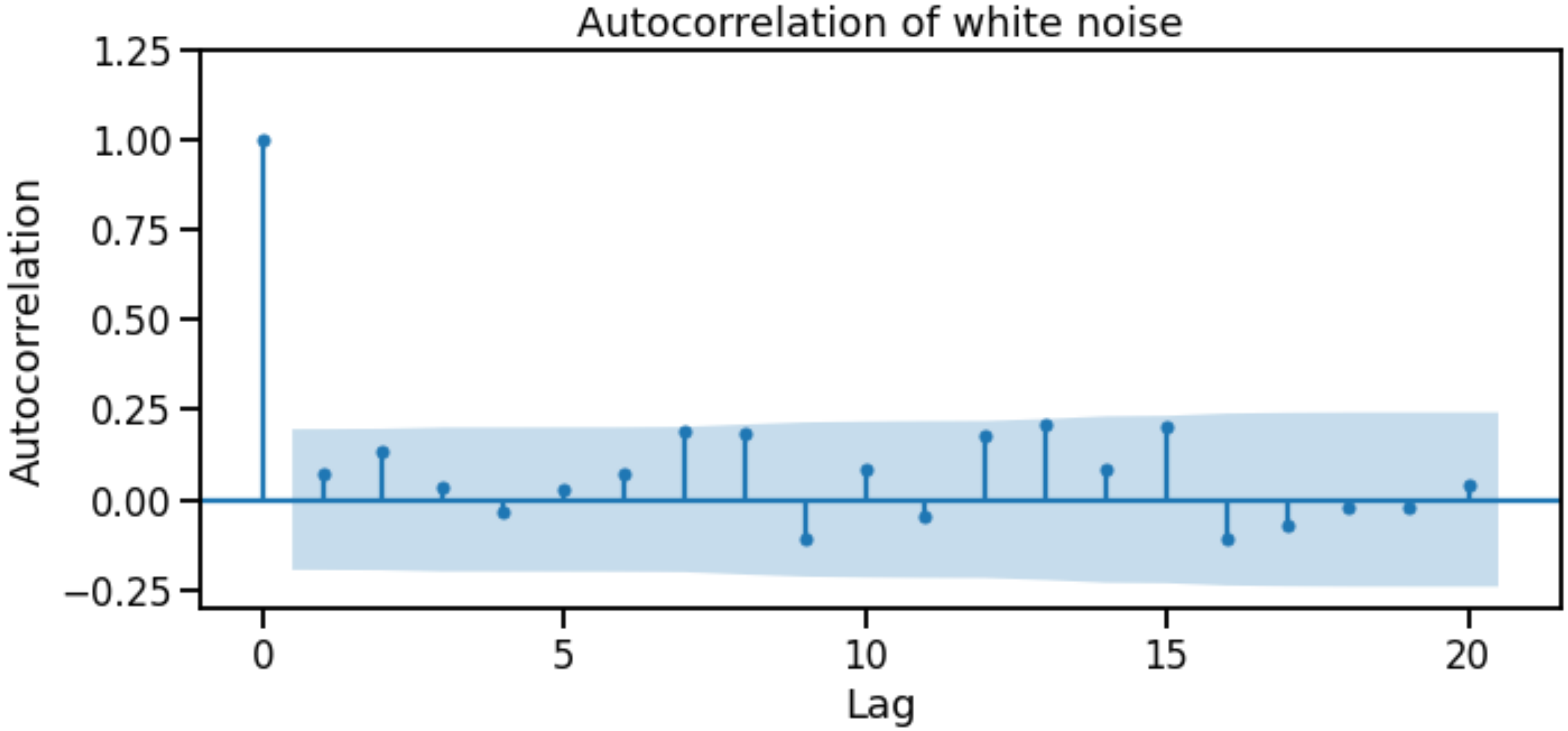
ACF is one at lag zero as expected

No significant autocorrelation at any other lag



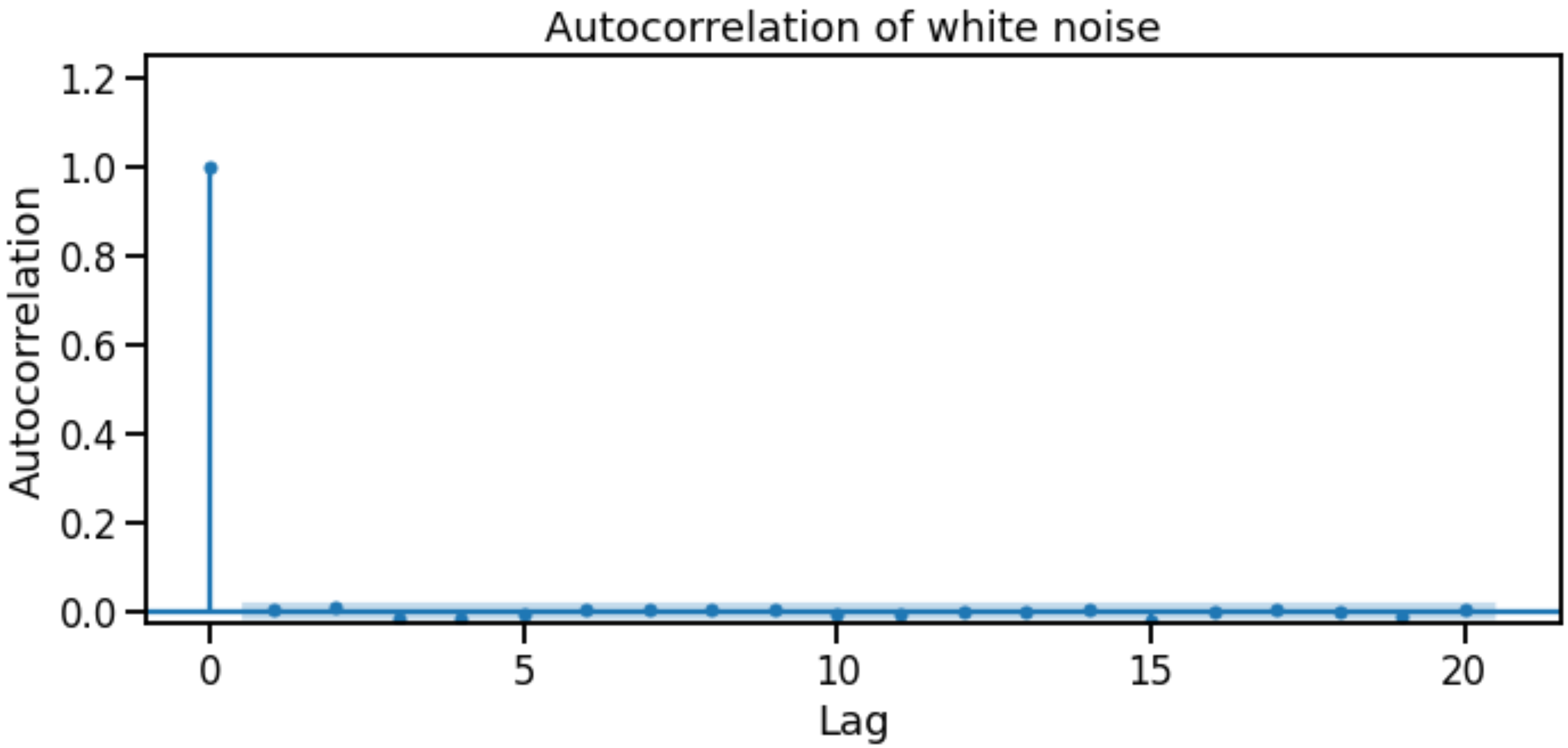
The sinusoidal fluctuations are a result of the finite sample size and would shrink as sample size (i.e., length of time series) increases.

ACF: white noise



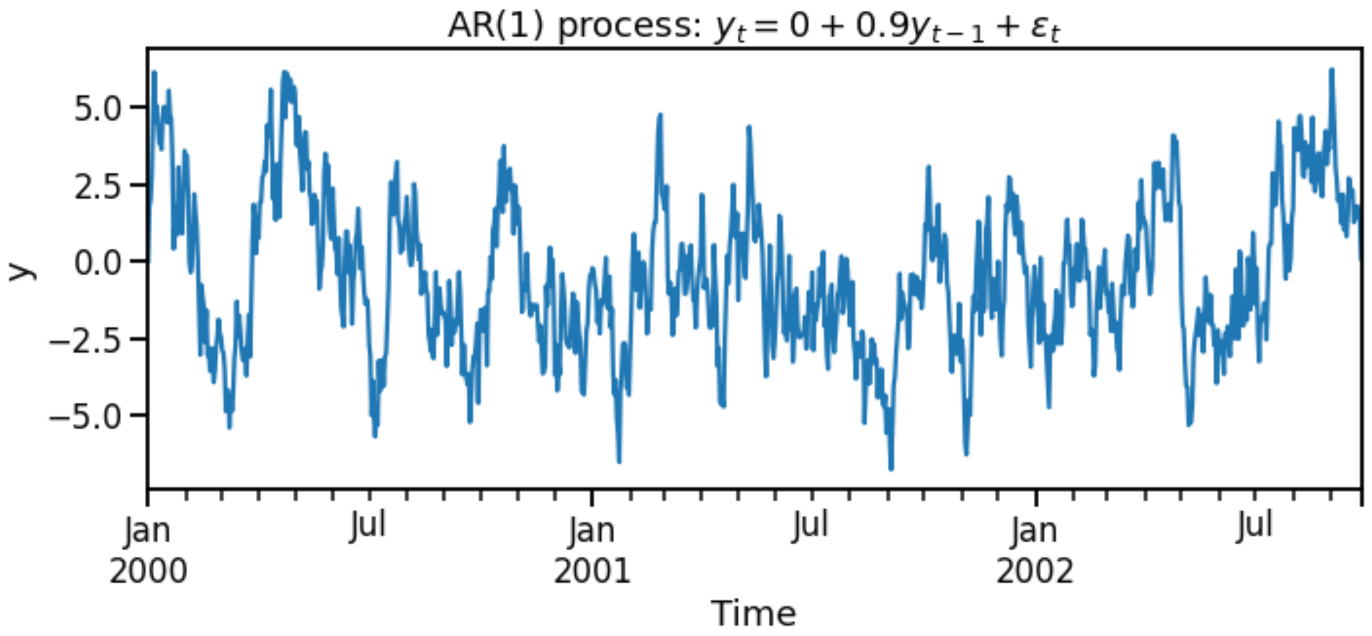
Time series length: 100 → large fluctuations

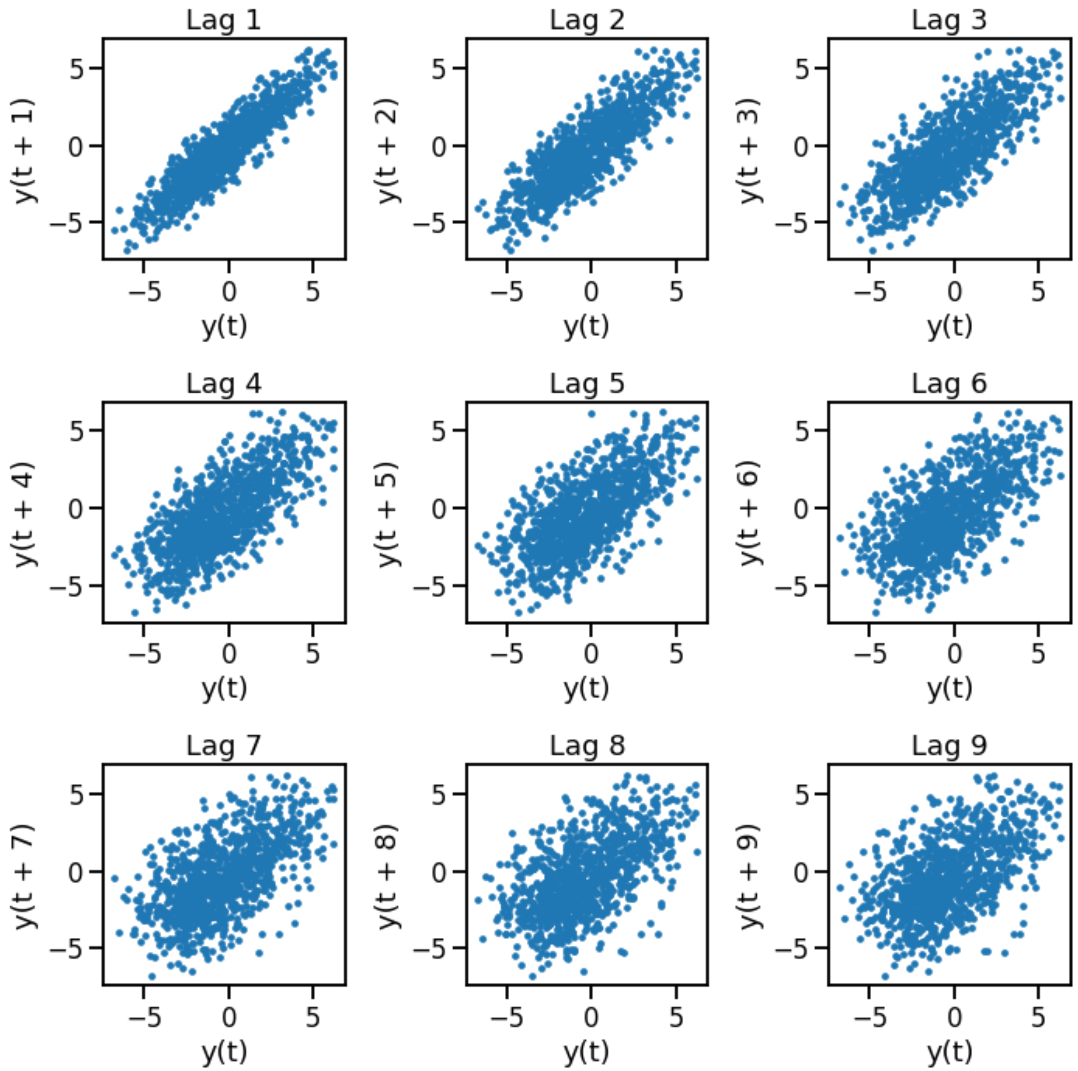
ACF: white noise



Time series length: 10,000 → small fluctuations

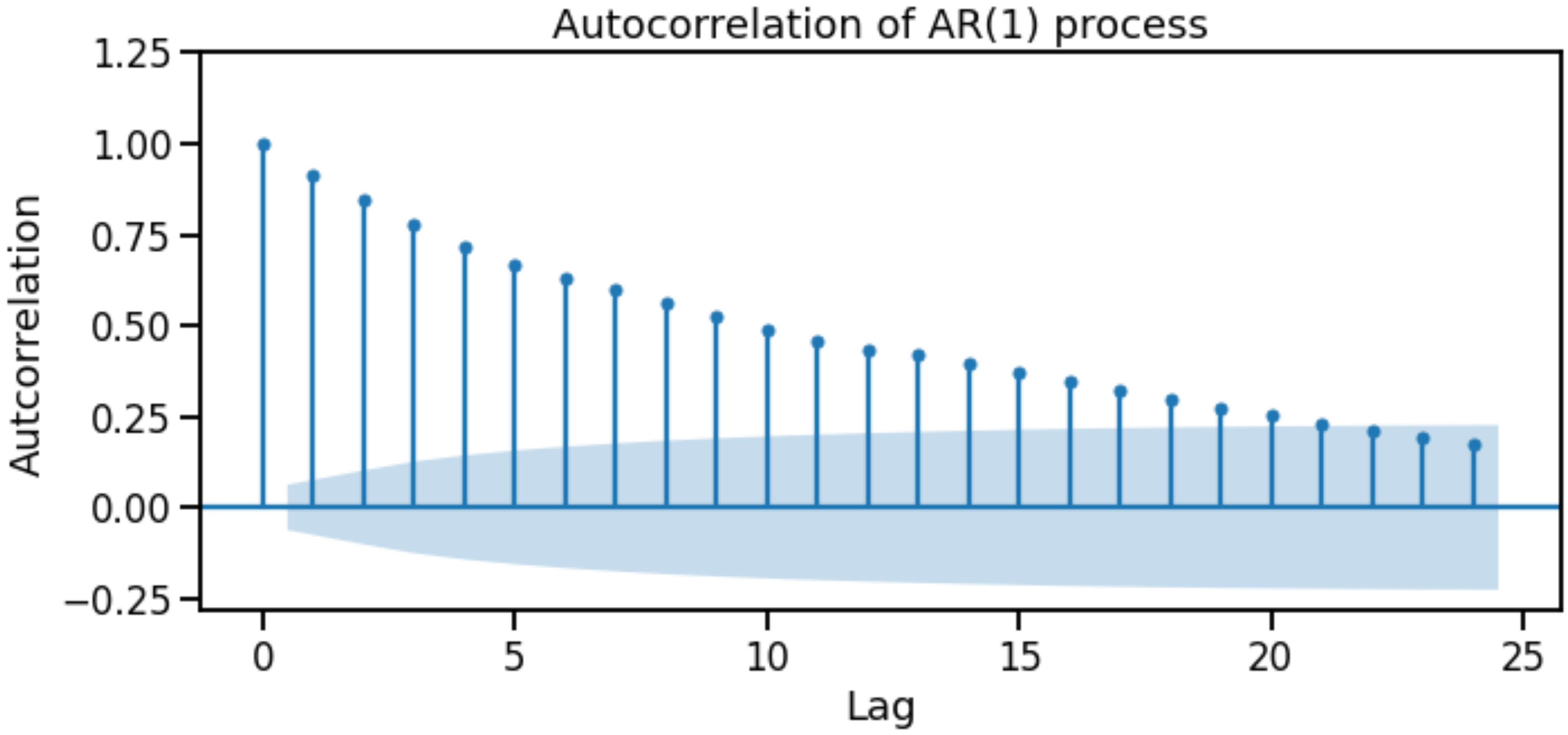
Autoregressive processes





- $y_t = c + \phi_1 y_{t-1} + \epsilon_t$ where $\epsilon_t \sim N(0,1)$
- We expect this time series to be correlated to lagged values.

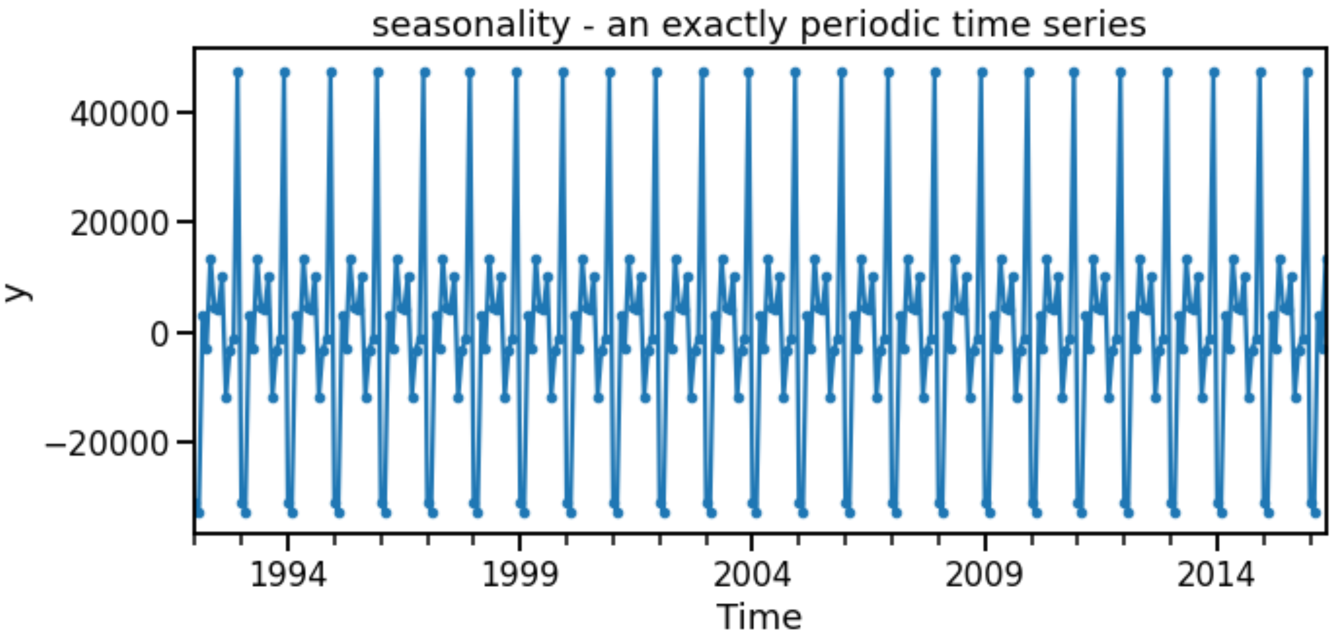
ACF: Autoregressive processes

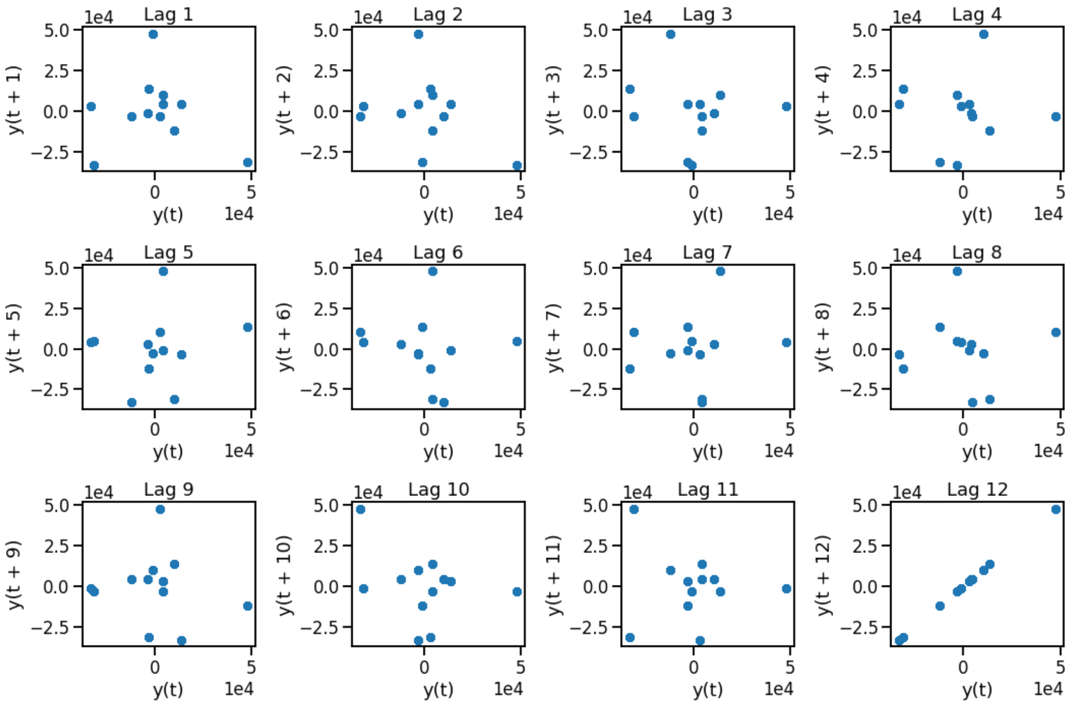


For an AR(1) process we see significant autocorrelation for multiple lags. The ACF decays exponentially [1] (this will contrast with the ACF of a trend component which decays more slowly).

[1] - George, E. P. "Box. Time series analysis: forecasting and control." (1970).

Seasonality



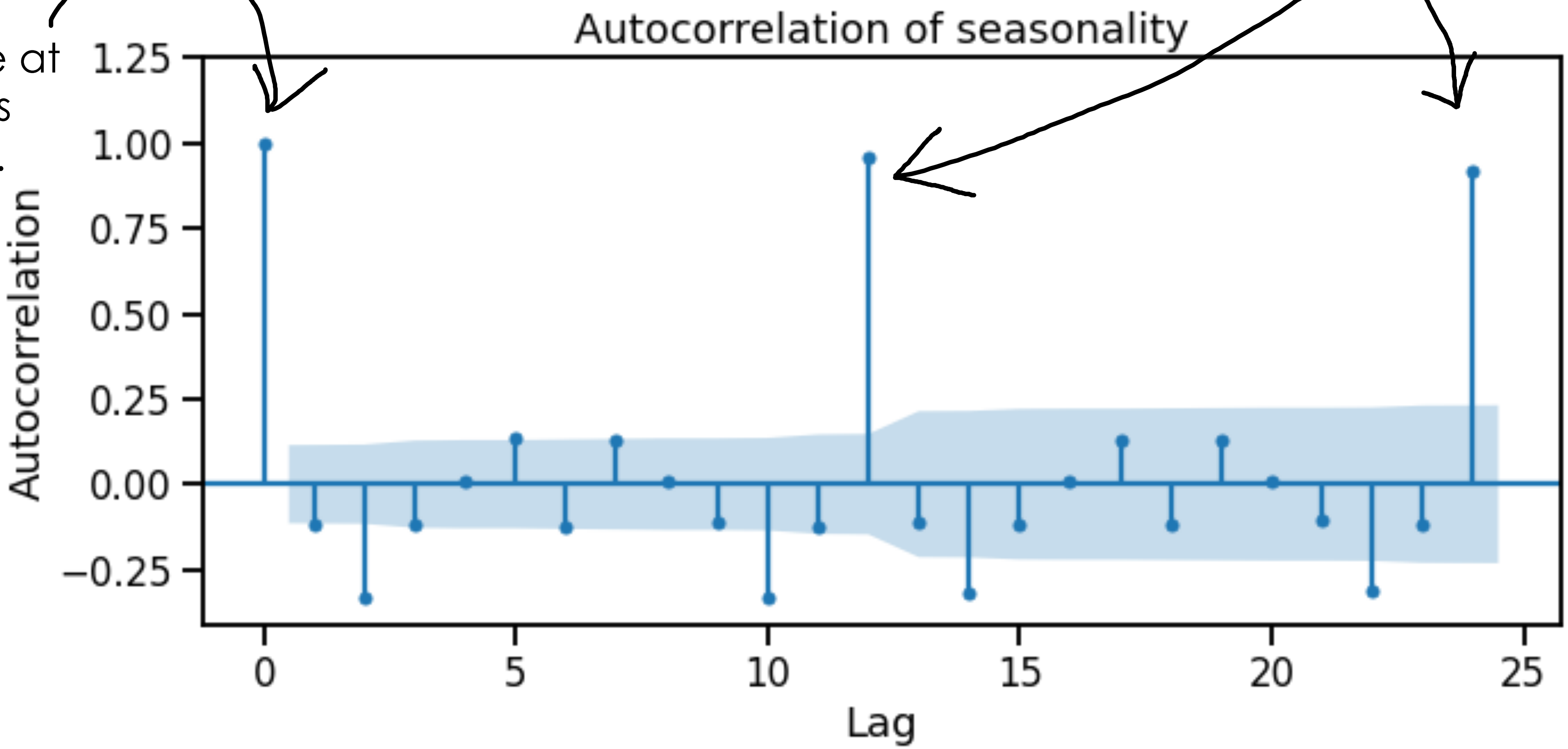


As the time series only has 12 distinct values the lag plots repeat themselves as different configurations of 12 points. This repeating pattern will be reflected in the ACF.

ACF: seasonality

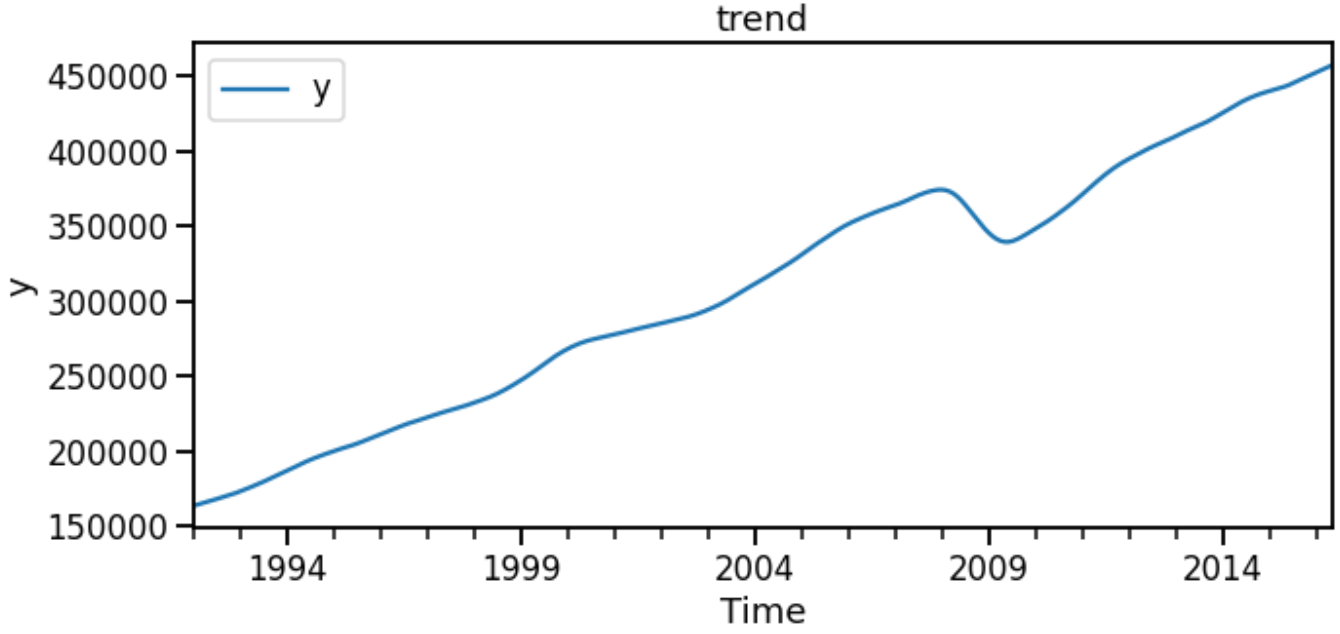
ACF is one at lag zero as expected.

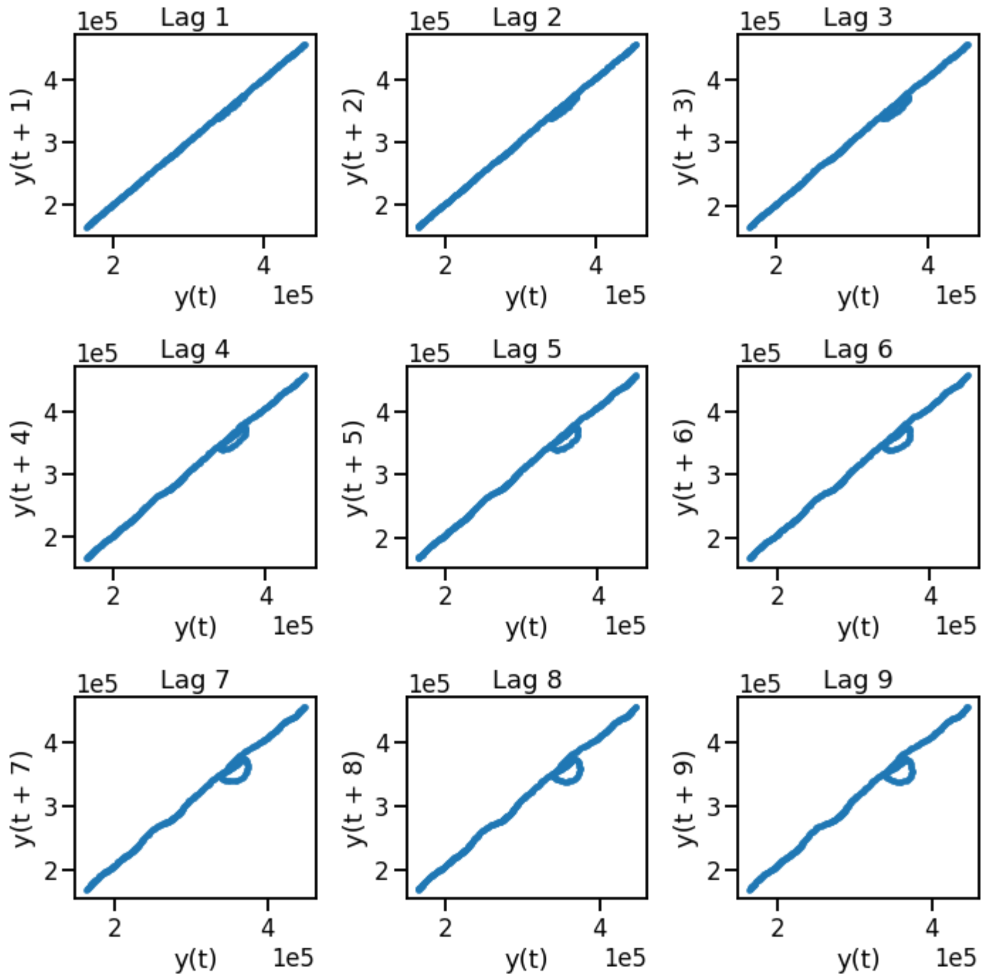
Very significant autocorrelation at multiples of the seasonal period.



As the time series only has 12 distinct values the lag plots repeat themselves as different configurations of 12 points. This repeating pattern will be reflected in the ACF.

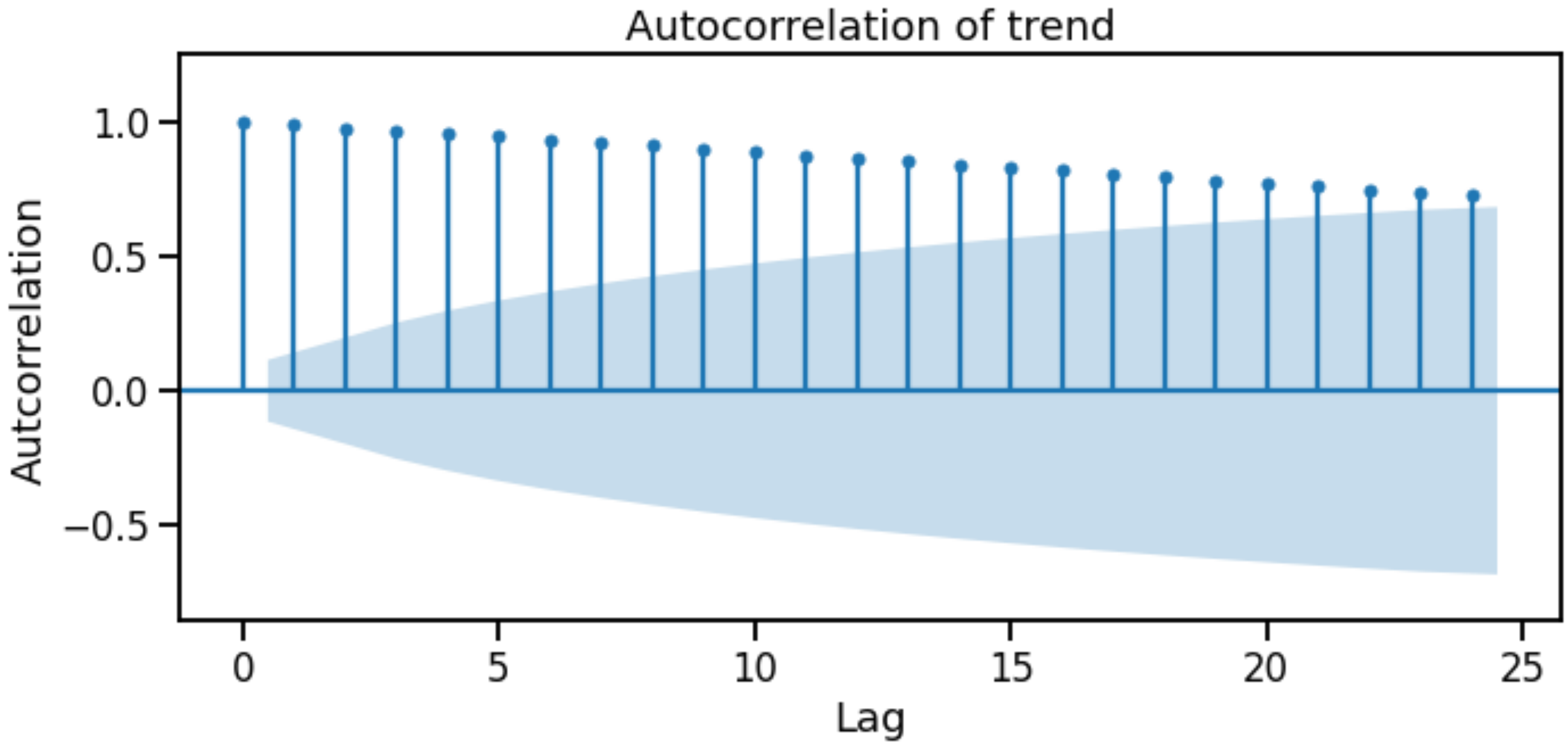
Trend





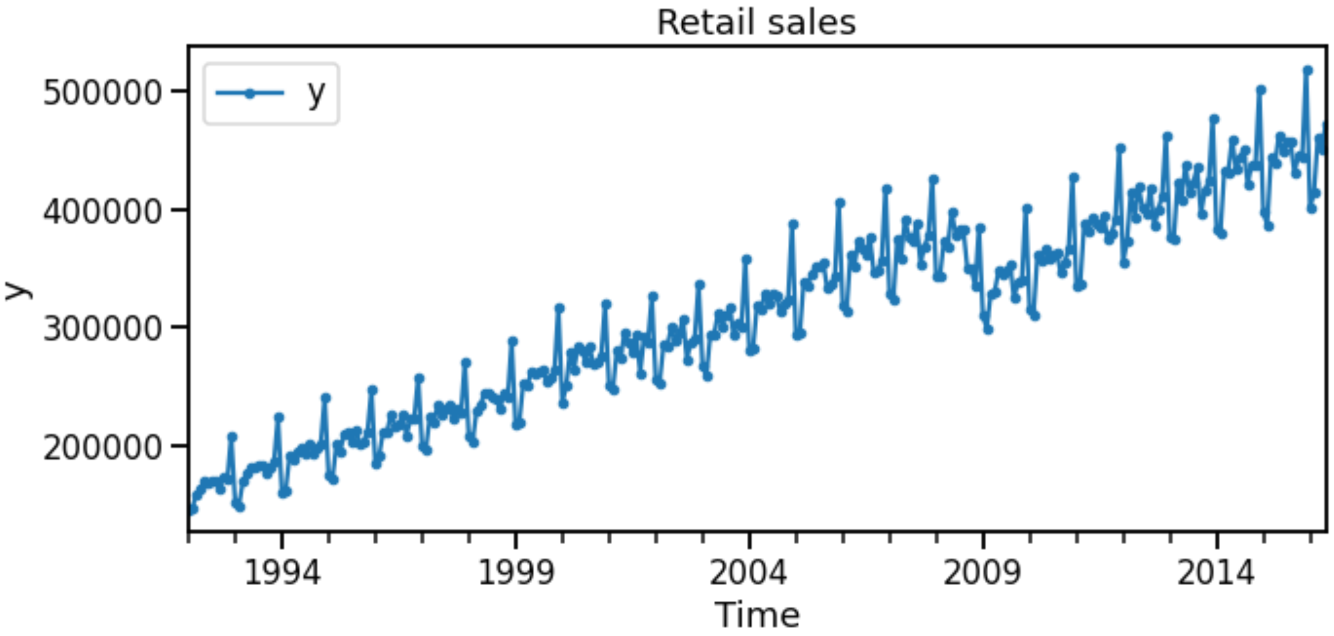
Expect to see the ACF being large at many lags.

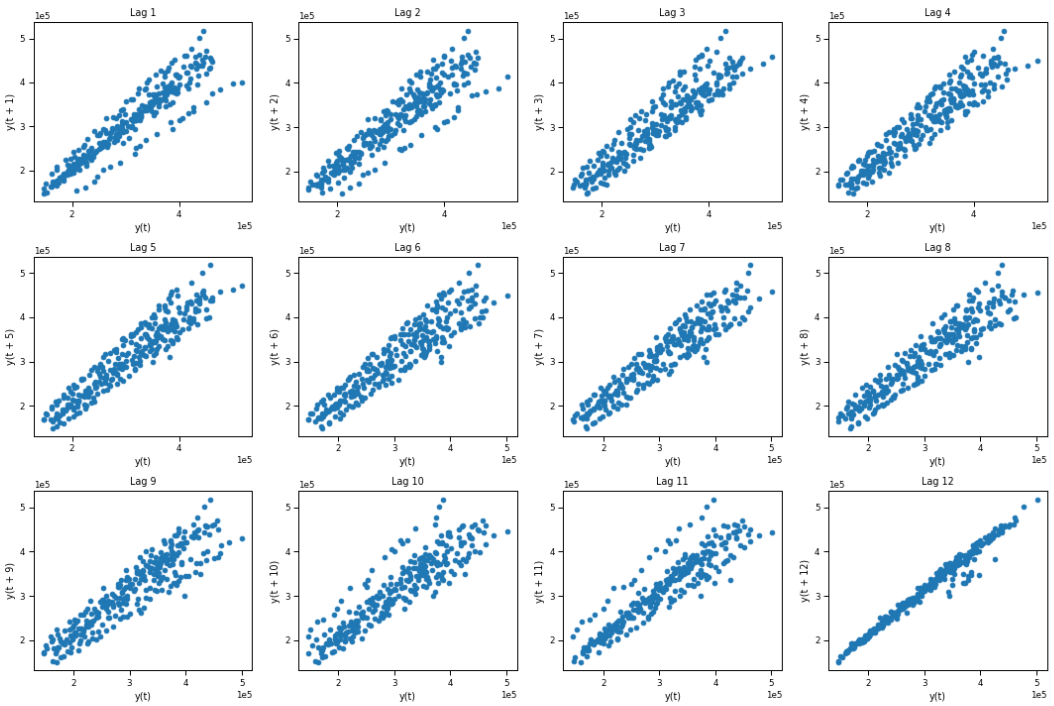
ACF: trend



The ACF decays slowly and we see large autocorrelations for multiple lags. Hence, the ACF is not as useful in identifying a specific lag to use when there is a strong trend component.

ACF: trend and seasonality



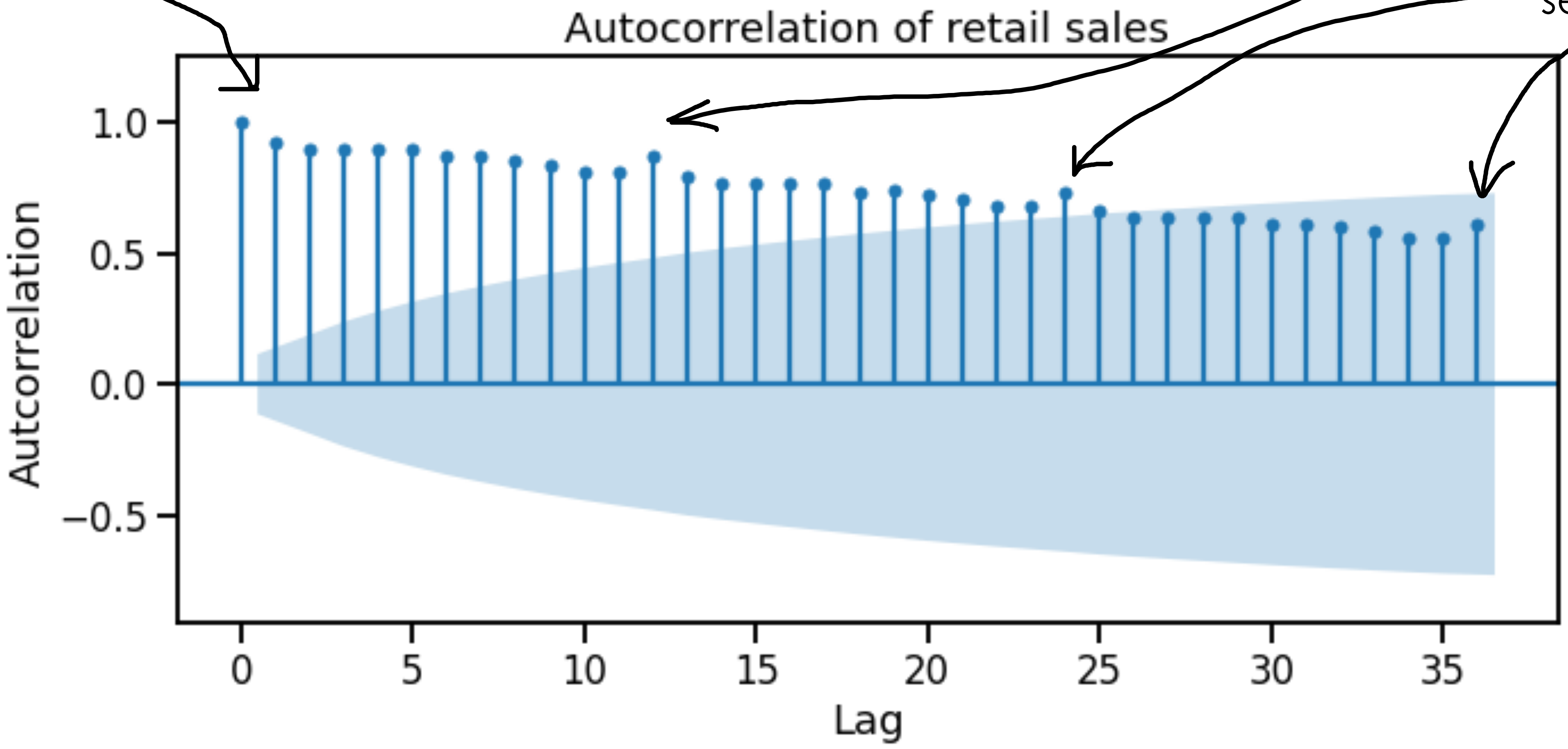


Expect to see a peak in the ACF at the seasonal lag and a long decay.

ACF: trend and seasonality

The ACF decays slowly due to the strong trend.

We see spikes at multiples of the seasonal lag.

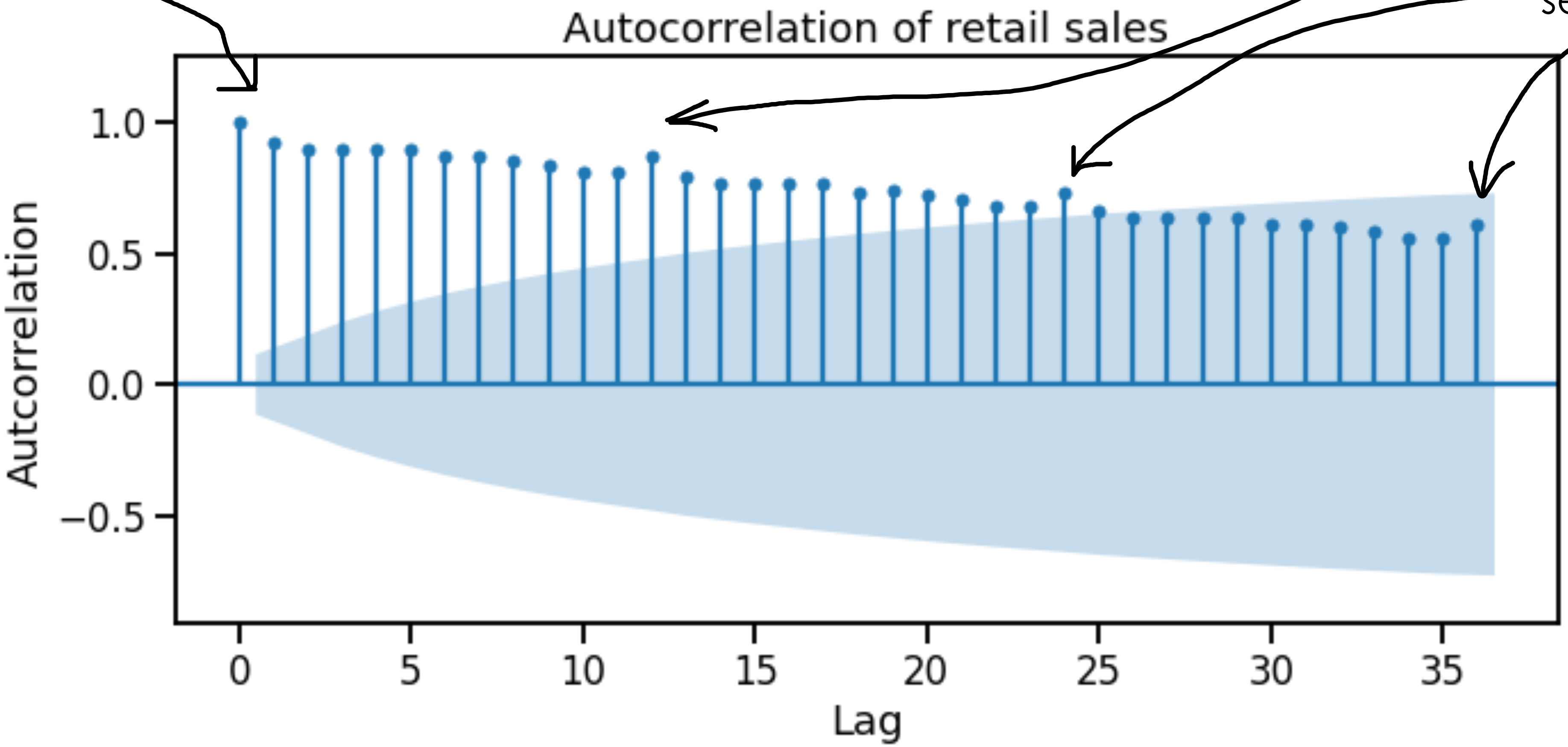


We see elements of both the trend and seasonality in the ACF.

ACF: trend and seasonality

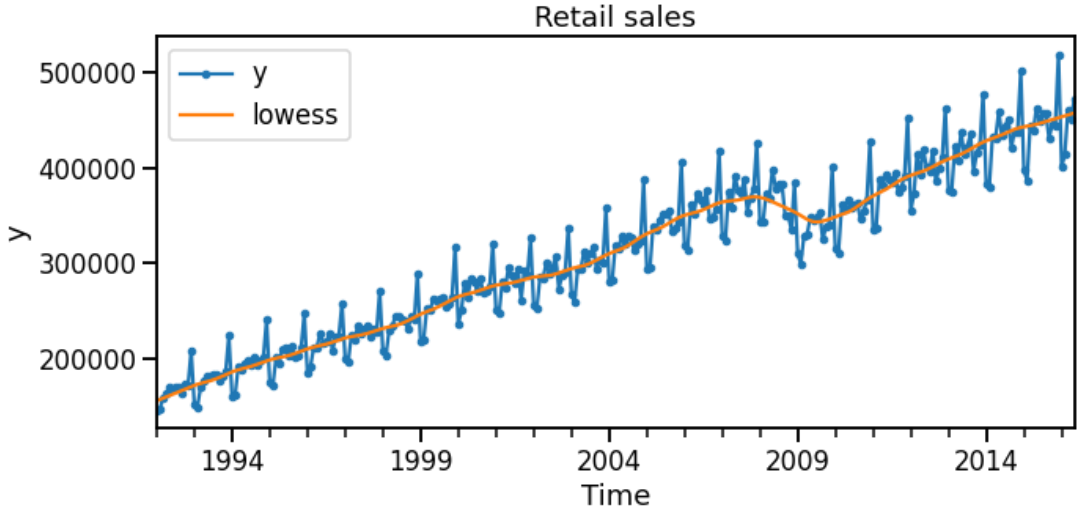
The ACF decays slowly due to the strong trend.

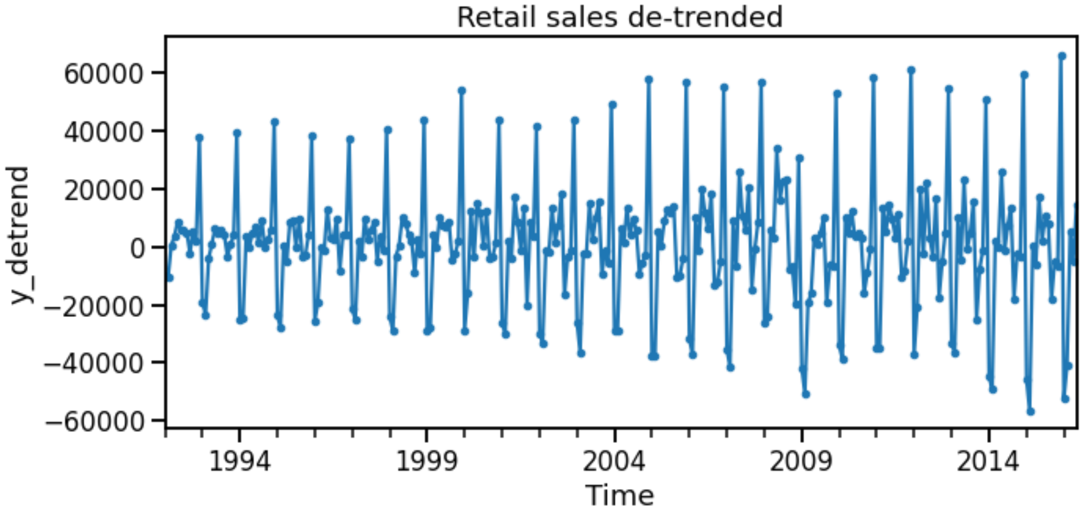
We see spikes at multiples of the seasonal lag.



De-trending the time series (e.g., using LOWESS) can make it easier to see signatures of periodic behaviour and other lags in the ACF.

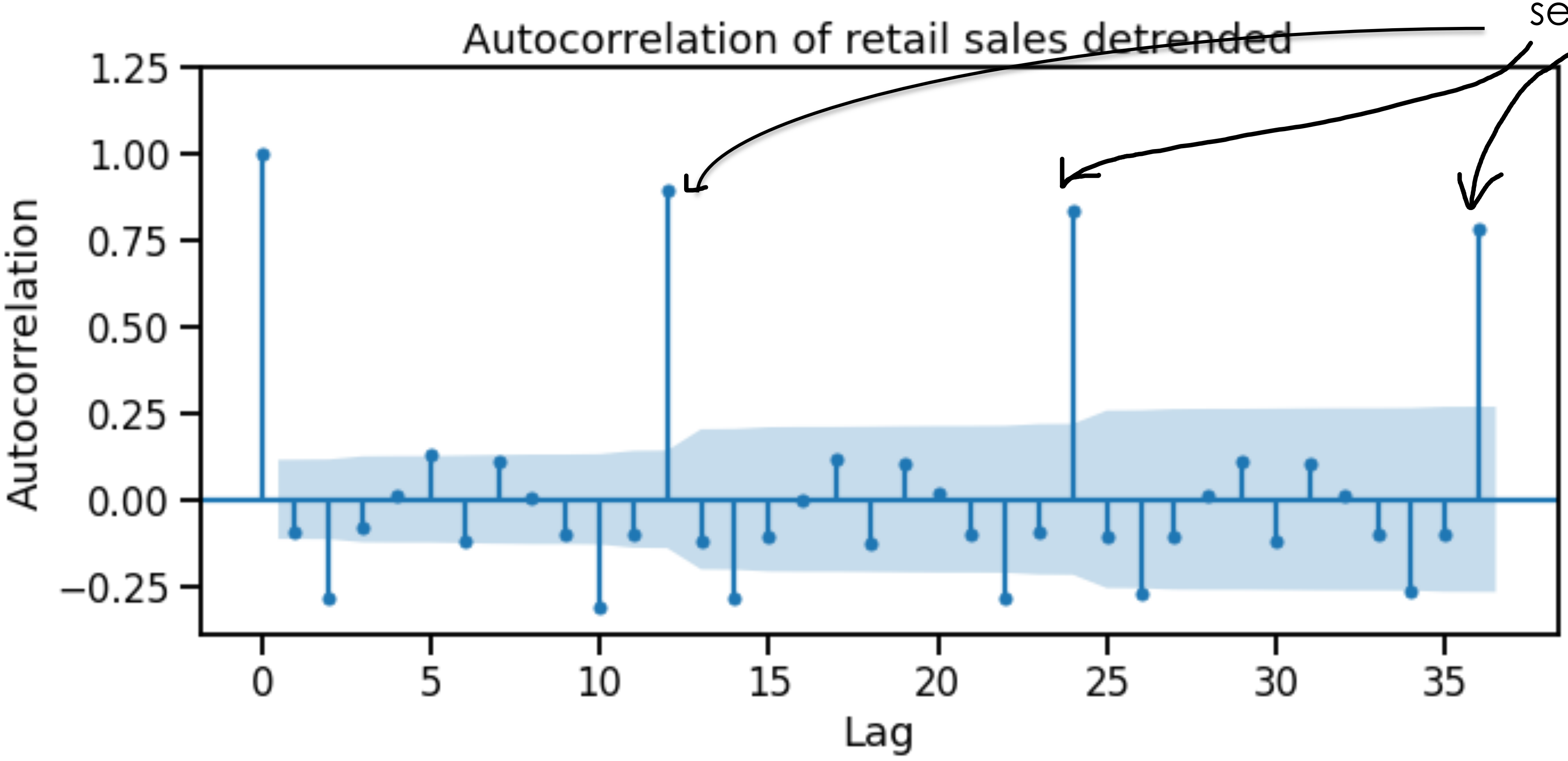
De-trending the data





ACF: After de-trending

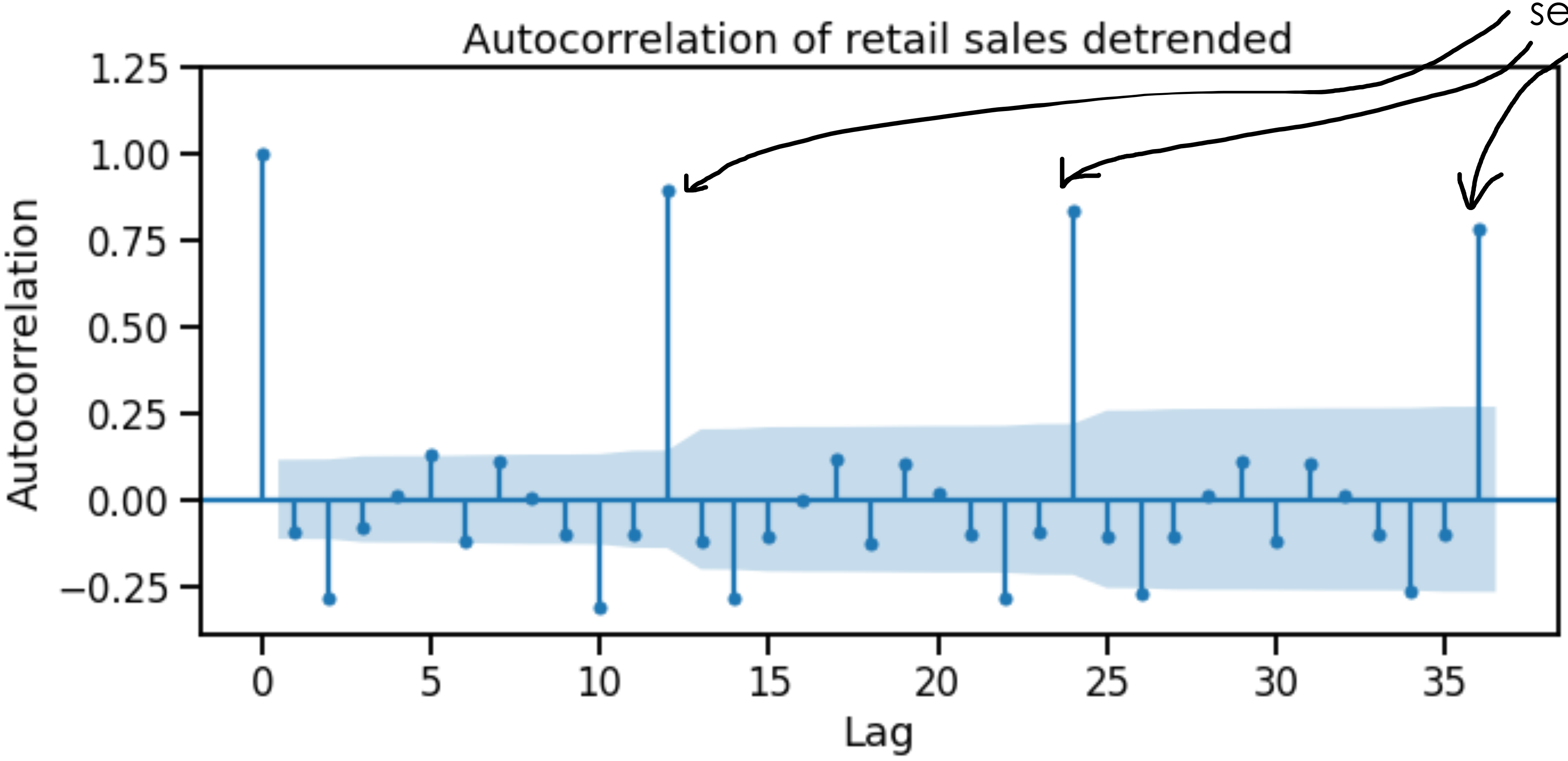
We see spikes at multiples of the seasonal lag



De-trending the time series (e.g., using LOWESS) can make it easier to see signatures of periodic behaviour and other lags in the ACF. The ACF suggests that a lag of 12 would be useful here.

ACF: After de-trending

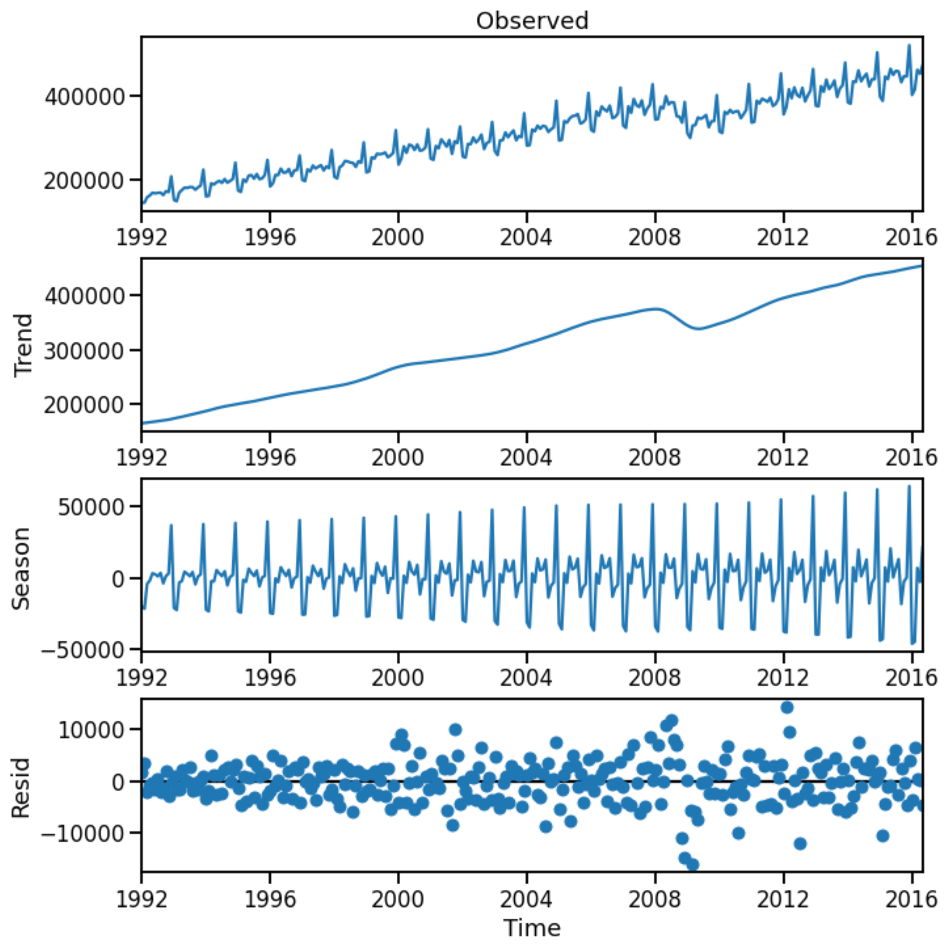
We see spikes at multiples of the seasonal lag



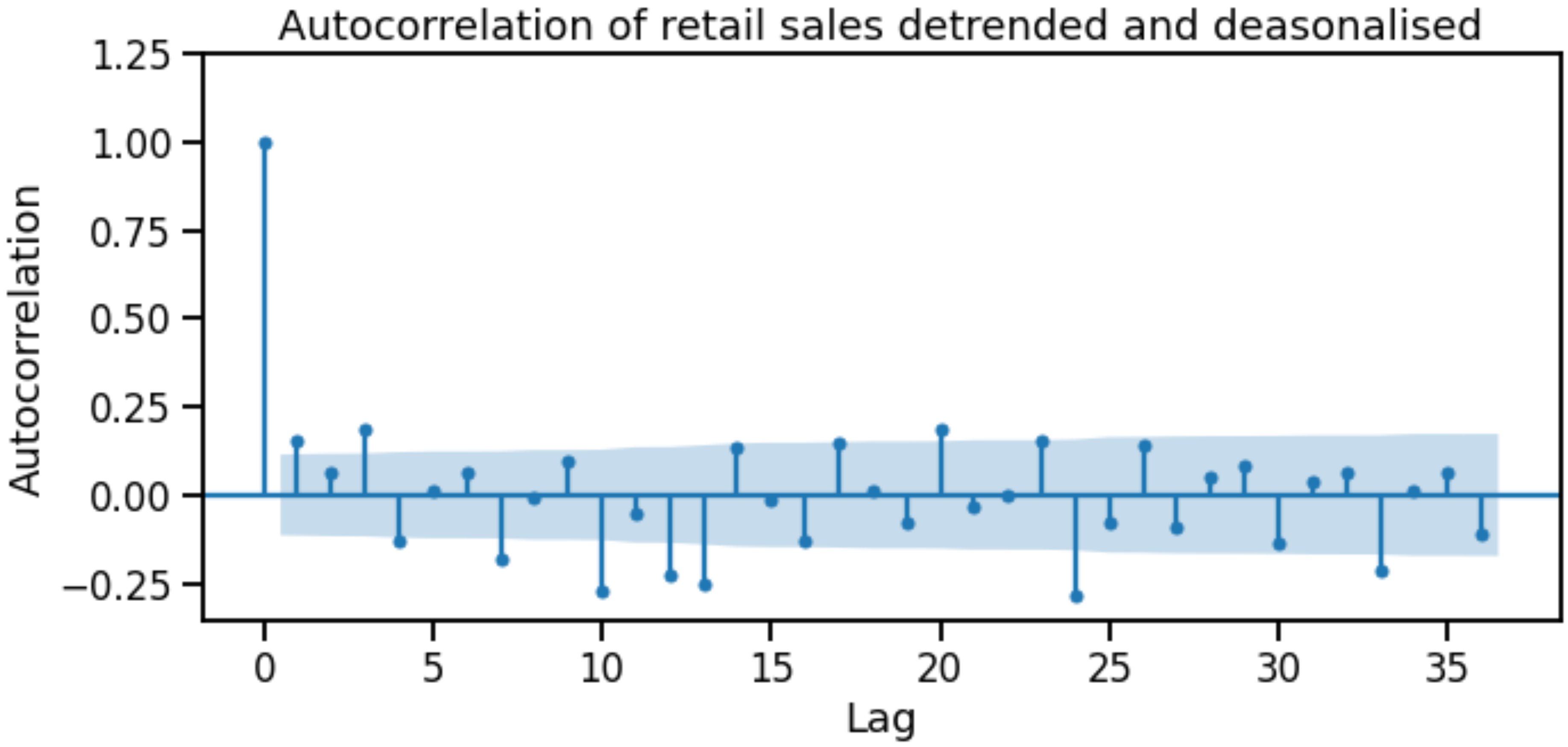
Let's remove the seasonality as well to see if there is any autocorrelation left in the data. We shall use STL to extract the seasonality and trend and remove them from the data.

De-trending & de-seasonalising the data

- This is the STL decomposition of the retail sales dataset.
- The residual component is equivalent to y – trend – seasonality.
- This means the residual component is equivalent to de-trending and de-seasonalizing the data.

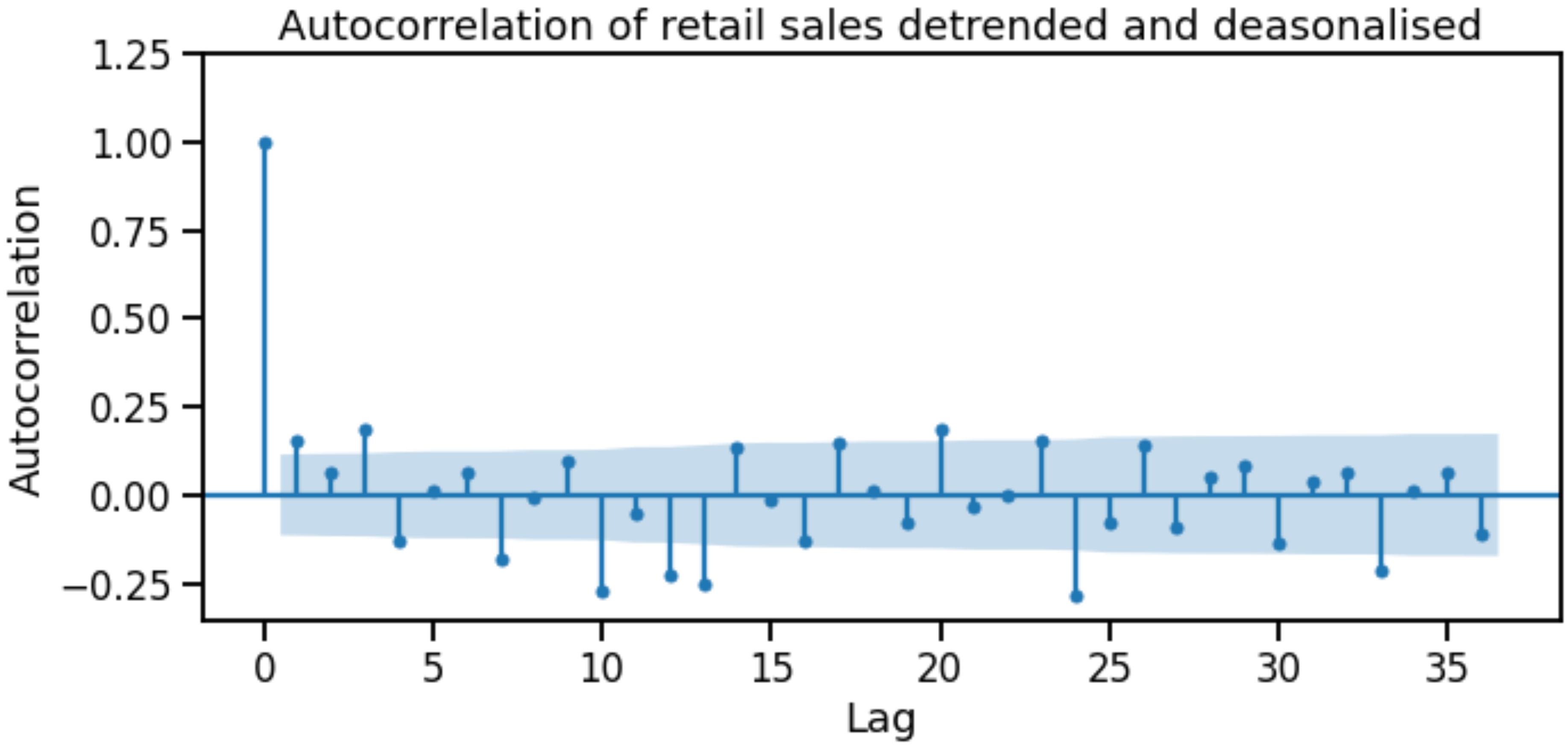


ACF: After de-trending and de-seaosonalising



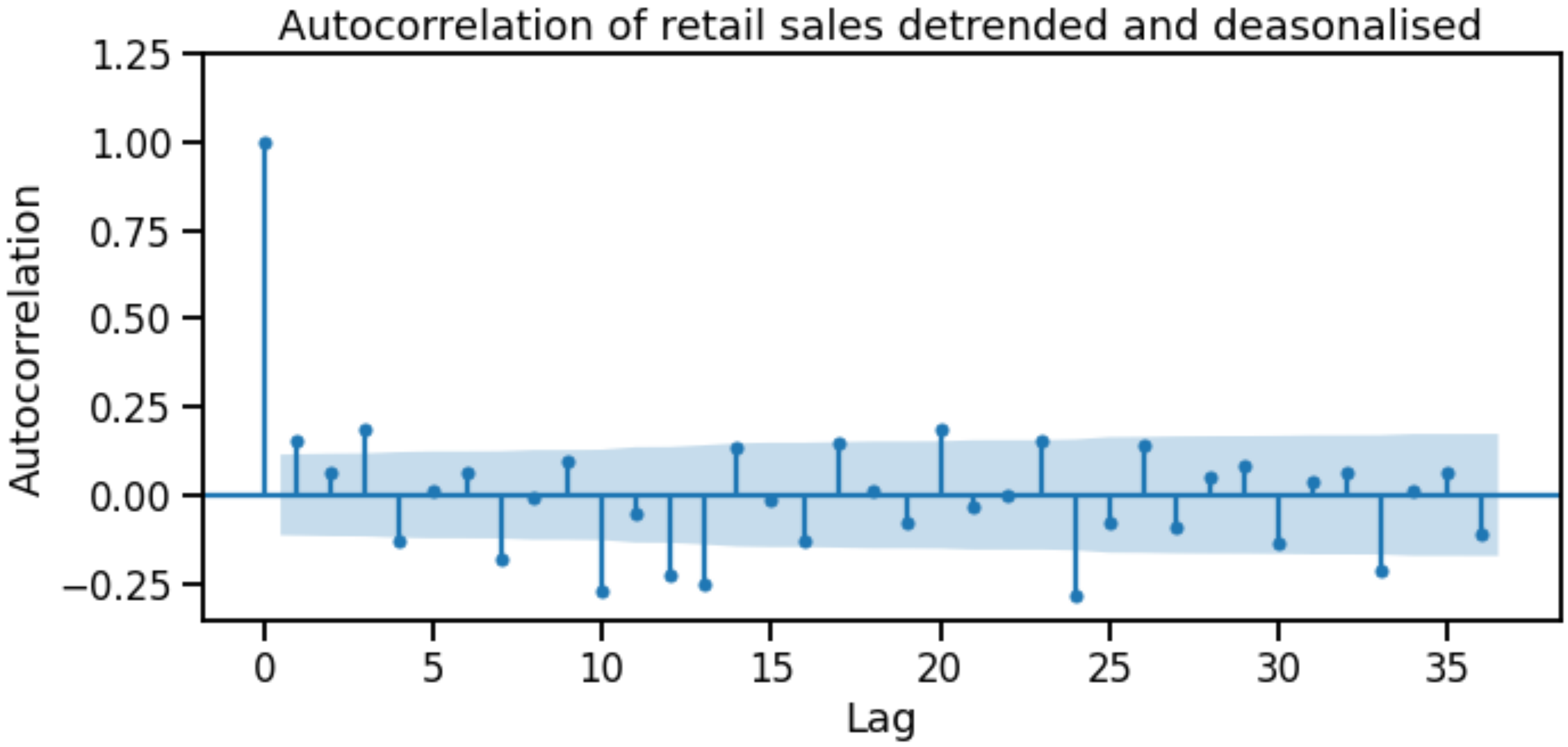
This is difficult to interpret as the small lags (e.g., 1 and 3) are only just significant. Despite this lag 1 would be worth using in any case as the recent past tends to be predictive.

ACF: After de-trending and de-seaosonalising



There still appears to be some seasonality left in the data as the autocorrelation is significant for lag 12 and 24.

ACF: After de-trending and de-seaosonalising



It is difficult to determine whether to use lag 7, 10, or 13. Given the context (retail sales) it is unlikely that they would be helpful. In this case they could be used and evaluated using LASSO.

ACF implementation in Statsmodels

statsmodels.tsa.stattools.acf

statsmodels.tsa.stattools.acf(*x*, *adjusted=False*, *nlags=None*, *qstat=False*, *fft=True*, *alpha=None*, *bartlett_confint=True*, *missing='none'*)[source]

Calculate the autocorrelation function.

Parameters

x : numpy:array_like

The time series data.

adjusted : bool, `default` `False`

If True, then denominators for autocovariance are n-k, otherwise n.

nlags : `int`, `optional`

Number of lags to return autocorrelation for. If not provided, uses min(10 * np.log10(nobs), nobs - 1). The returned value includes lag 0 (ie., 1) so size of the acf vector is (nlags + 1,).

qstat : bool, `default` `False`

```
acf(x=df['y'], nlags=12)
```

```
array([1.        , 0.9199895 , 0.89711646, 0.89897886, 0.89725296,
       0.89779829, 0.86710923, 0.87124837, 0.85433604, 0.83759833,
       0.81262566, 0.8136589 , 0.86701028])
```

ACF implementation in Statsmodels

statsmodels.graphics.tsaplots.plot_acf

statsmodels.graphics.tsaplots.plot_acf(*x*, *ax=None*, *lags=None*, ***, *alpha=0.05*, *use_vlines=True*, *adjusted=False*, *fft=False*, *missing='none'*, *title='Autocorrelation'*, *zero=True*, *auto_ylims=False*, *bartlett_confint=True*, *vlines_kwargs=None*, ***kwargs*)[source]

Plot the autocorrelation function

Plots lags on the horizontal and the correlations on vertical axis.

Parameters

x : numpy:array_like

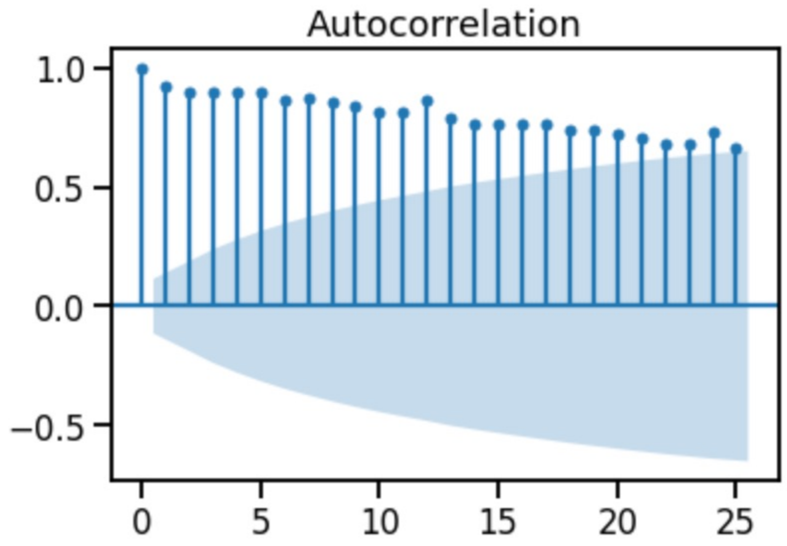
Array of time-series values

ax : `AxesSubplot`, `optional`

If given, this subplot is used to plot in instead of a new figure being created.

lags : {`int`, numpy:array_like}, `optional`

```
plot_acf(df['y']);
```



Summary

Autocorrelation function (ACF) measures how correlated a time series is with itself at various lags.

The confidence interval of the ACF at a given lag can be given by the Bartlett formula which helps determine if the autocorrelation is significant.

Noise, autoregression, trend, and seasonality all leave different signatures on the ACF which can be used to pick a relevant lag for modelling.