# Lag Features

1. Motivation for lag features

* We want to predict future values of the target
* Past values of the target are likely to be predictive
* Past values of a feature could also be predictive (e.g., the sales on a day is related to ad spend on prior days)

1. Lag features

* A lag feature is the value of the target or feature k period(s) in the past
* k is the lag, set by user

A table with numbers and arrows

Description automatically generated with medium confidence

A table with numbers and a few words

Description automatically generated with medium confidence

* We can create multiple lag features with different lags from the target and features
* Problem: which lags to use? How many lag features to create?

1. Lag feature implementation in Pandas

* Pandas.DataFrame.shift

A screenshot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

1. Lag feature implementation in Feature-engine

* Feature\_engine.timeseries.forecasting.LagFeatures

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Summary

* Lag features are a way of using the past to predict the future
* Can lag the target or other features

# How to choose the lags

1. How to choose the lags

* Domain knowledge
* Feature selection and modeling
* Time-series correlation methods

1. Domain knowledge

* If seasonality is known use a lag of the same seasonal order (seasonal lag)
* E.g. retail sales: yearly seasonality -> use lag of 1 year
* E.g. electricity demand: multiple seasonalities such as yearly, weekly, daily -> use lag of 1 year, 1 week, and 1 day
* Most recent values tend to be predictive -> use small lag
* When lagging the features
* Use the subset of features you think are most important in affecting the target
* Only use the value of the feature that is known at predict time to avoid data leakage
* Most recent values tend to be predictive -> use small lags
* Pros:
* Likely to result in fewer additional features as we will pick a small number of lags and features known to be important
* Cons:
* We may not know all the seasonal patterns or which ones are most important
* We may not know which features are important to lag
* Not scalable

1. Feature selection and modeling

* Create a bunch of different lags which are reasonable given the feature and use case
* Use feature selection and/or modeling to best utilize the feature and determine a subset which minimizes forecast error
* Pros:
* Automatic – less hands-on decision making
* May find useful features which you may not have been used otherwise
* Cons:
* Will create very large number of features
* The lags of the same features will be highly correlated to each other
* More complex model than necessary
* Computationally expensive

1. Time series correlation methods

* The main idea
* Measure how corelated the lag features are with the target
* If the lag feature is highly correlated to the target then it might be helpful
* 3 main methods
* Autocorrelation function (ACF)
* Partial autocorrelation function (PACF)
* Cross-correlation function (CCF)
* Pros
* More robust way to find relevant lags
* Can indicate whether there is any predictive info in the historic time series at all
* Can help identify important seasonalities
* Cons:
* Can be difficult to interpret correlation plots
* Time consuming to interpret and read correlation plots -> not scalable to large number of features
* Even if one feature is not highly correlated with the target it could still be predictive in the presence of other features -> not captured in these methods
* These methods only measure linear relationships between variables

1. Summary

* Lags of the target and other features can create predictive features for forecasting
* User must decide which lag features to create
* Domain knowledge, feature selection & modeling, time series correlation methods can help in selection of lag

# Autoregressive (AR) processes

1. Motivation

* We will want to show how tools we introduce later (e.g. lag plots, correlation functions) behave for time series with various properties
* Three properties already covered in the course so far: trend, seasonality, white noise
* New property of a time series
* Autoregressive (AR) property

1. Scope

* AR processes are a large topic covered in the theory of time series analysis and is broader topic than just forecasting.
* In scope:
* Definitions of an AR process
* Intuition about the behavior of AR processes
* Out of scope:
* Mathematical proofs, derivations, theorems about AR processes

1. White noise

* White noise has no predictive info in past values – no correlation at any 2 points in time

1. Autoregressive (AR) processes

* Generate a time series , which is determined only from the previous values , a constant , and some white noise :
* An AR(1) process as it depends only on a lag of 1

A computer screen shot of a program

Description automatically generated

* **If**

A graph showing the time of a graph

Description automatically generated with medium confidence

A graph showing a graph of time

Description automatically generated with medium confidence

* Lag component coef is now smaller -> noise component makes it look noisier
* **If ,** the time series settle at a new baseline. This follows from the fact that the mean of an AR(1) process when is given by

A graph showing the time and time

Description automatically generated with medium confidence

* **If** , the time series grow exponentially

A graph with numbers and a line

Description automatically generated

* **If** , the time series grow exponentially but oscillates from a negative to positive value over each iteration

A graph with numbers and a line

Description automatically generated

* **If ,** the time series grow linearly – all we are doing is adding a factor and some noise to the previous time step

A graph with a line

Description automatically generated

1. AR(1) summary

* Correlated to the most recent lag (i.e., lag of 1)
* : exponential growth, time series is not stationary
* : linear grow, time series is not stationary
* : varies around a mean value, time series is stationary
* : oscillates between positive and negative values, stationary if
* We can use an AR(1) process to generate time series where future values are correlated to past values. This allows us to test methods which identify whether a lag of 1 is helpful or not

1. AR(p) process

* Time series which depend on more lags
* An AR(p) process depends on p lags and is defined as:
* Recall that an AR(1) process requires to be stationary
* For an AR(p) process there are much more complex requirements on all the coefficients to ensure the process is stationary
* These time series are interesting because by design they depend on multiple lagged values
* Hence, an AR(p) process is a good test case for methods that select which set of lags are important or not.

1. Summary

* An autoregressive (AR) process is a class of time series where future values depend on past values and white noise
* An AR process is determined by prev values and therefore will be correlated to lag values of itself. Hence, lag features should help predict an AR process
* AR processes provide time series which we can use to test methods which identify helpful lags

# Lag plots

1. Scatterplots

* Scatterplots can help identify if 2 variables are related

1. Lag plot

* A scatterplot of a time series against a lagged version of itself

1. What can we learn from a lag plot

* A lag plot is a visual tool which can help show if show a non-random relationship with
* If it does then a lag of k could be a useful feature for forecasting
* We shall look at time series with properties
* White noise
* AR(1) process
* Completely periodic (just seasonality)
* Trend
* Trend and seasonality

1. White noise

* Random time series with no correlation between points
* No predictive info in the historic data
* No strong relationship in the lag plot, as expected from white noise
* This shows us what to look for when determining when a lag may not be useful

A group of squares with blue dots

Description automatically generated

1. AR(1) process

* The time series is determined by the previous lag
* We expect this time series to be correlated to lagged values
* Strong linear correlation between and
* We see the correlation between and its lagged values decay as we look at bigger lags
* a time series which is determined only by a small number of previous lags can generate correlations at multiple lags
* When we discuss PACF, we will see how we can identify that lag 1 is the most important

A group of blue dots in squares

Description automatically generated

1. Seasonality

* A time series which repeats exactly every 12 months
* Any multiple lag of 12 should be the most predictive of future values as the time series is exactly periodic every 12 months

A graph showing the number of seasons

Description automatically generated with medium confidence

* Lag plot:

A graph with blue dots

Description automatically generated

* Strong seasonality at lag 12 indicated by the strong linear correlation
* A lag of 12 could be a helpful feature
* Additional lags:

A group of graphs with numbers and symbols

Description automatically generated with medium confidence

* is exactly periodic with a seasonal period of 12
* There are only 12 unique values that can take
* This causes every lag plot to only have 12 data points in different configurations which repeat every 12 lags
* Important when we discuss the autocorrelation function
* Any seasonal patterns will appear as a strong linear correlation on the lag plot. This occurs at multiplies of the seasonal period (e.g. every 12 lags for yearly seasonality)

1. Trend

* A lot of predictive info in historic data
* When is small so is
* Large positive relationship

A graph with blue and orange lines

Description automatically generated

* Lag plot

A group of blue lines with numbers and letters

Description automatically generated with medium confidence

* Strong linear relationship in the lag plot, as expected
* This is seen across multiple lags because of the overall shape of the original time series means that when is relatively large so is
* The trend causes correlations at many lags – this can make it difficult to identify patterns (e.g. seasonality) which appear as strong correlations only at specific lags
* The lag plot does not provide much info about whether a specific lag will be helpful

1. Trend and seasonality

* A lot of predictive info in historic data
* Will get a combination of effects from the trend, seasonality, and noise in the lag plots
* Lag plot
* The strong trend results in linear relationships across many lags
* The seasonal component is seen in the seasonal lag as a much stronger relationship (lag 12 in this example)

A screenshot of a graph

Description automatically generated

1. Lag plot implementation in Pandas

A screenshot of a computer program

Description automatically generated

1. Limitations

* Lag plots are a visual tool which can help identify useful lags but are not scalable
* If we quantify when is highly correlated with – easier to identify useful lags
* Autocorrelation is a method to quantify the correlation of a time series with itself and can be used to understand properties of a time series including useful lags

1. Summary

* Scatterplots can help identify if 2 variables are related
* Lag plot is a scatterplot of a time series against a lagged version of itself
* Lag plots can identify lags which are strongly related to the original time series
* Trend, seasonality, autocorrelation, and noise leave their own signatures on a lag plot

# Autocorrelation functions

1. Lag plot limitations

* Lag plots are a visual tool which can help identify useful lags >< not scalable
* If we quantify when is highly correlated with then it would be easier to identify useful lags
* Autocorrelation is a method to quantify the correlation of a time series with itself

1. Pearson correlation coefficient

* Measures the strength of linear relationships between 2 variables
* Does not care about the gradient

A group of blue shapes

Description automatically generated with medium confidence

1. Autocorrelation function (ACF)

* The ACF is correlation of a timeseries with a lagged version of itself
* If the autocorrelation at lag k is large then it might be helpful in forecasting
* Autocorrelogram:

A graph of a graph with blue lines

Description automatically generated

1. ACF: Confidence intervals

* Is the that is estimated from the data significantly different from 0?
* We can compute the confidence interval (CI) of if it were generated by a random process
* Bartlett’s formula provides a confidence interval (typically 95% CI used – statsmodels)
* If the is outside of this interval we can conclude that is significant

1. ACF for different timeseries

* White noise

A graph with blue lines and white text

Description automatically generated

* No significant autocorrelation at any lag > 0
* The sinusoidal fluctuations are a result of the finite sample size and would shrink as sample size increases
* Autoregressive processes
* AR(1) process:

A graph of a graph with blue lines

Description automatically generated with medium confidence

* For an AR(1) process we see significant autocorrelation for multiple lags. The ACF decays exponentially (this will contrast with the ACF of a trend component which decays more slowly)
* Seasonality (e.g. yearly seasonality – seasonal period = 12)
* 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

A graph with blue lines and dots

Description automatically generated

* Very significant autocorrelation at multiples of the seasonal period
* Trend
* Expect to see ACF being large at many lags

A graph of a line graph

Description automatically generated with medium confidence

* The ACF decays slowly and 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
* Trend and seasonality
* Expect to see a peak in the ACF at the seasonal lag and a long decay

A graph showing sales and sales

Description automatically generated with medium confidence

* Elements of both the trend and seasonality present in the ACF
* ACF decays slowly due to strong trend. Large autocorrelations for many lags
* Spikes at multiples of the seasonal lag – seasonality
* De-trending the time series (e.g., using LOWESS) can make it easier to see the signatures of periodic behavior and other lags in the ACF
* De-trending the data

A graph showing the results of a graph

Description automatically generated with medium confidence

* ACF: After de-trending

A graph with blue lines and arrows

Description automatically generated

* What happens if we also remove the seasonality (e.g. using STL)? Will there still be autocorrelation left in the data?
* De-trending and de-seasonalizing the data
* The residual component is equivalent to y – trend – seasonality
* The residual component is equivalent to de-trending and de-seasonalizing the data
* ACF: After de-trending and de-seasonalizing

A graph of sales

Description automatically generated

* Difficult to interpret as the small lags are only just significant. Despite this lag 1 would be worth using in any case as the recent past tends to be predictive
* There still appears to be some seasonality left in the data as the autocorrelation is significant for lag 12 and 24
* 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

1. ACF implementation in Statsmodels

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Summary

* ACF measures how correlated a timeseries is with itself at various lags
* The CI of 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 modeling

# Partial autocorrelation function

1. Motivation

* We want to know which lags are the most predictive
* Imagine a process where only the previous lag matters (e.g., AR(1) process)
* will influence which influences

A close-up of a graph

Description automatically generated

* This means that the autocorrelation between and will be non-zero even though only a lag of 1 matters

1. Main idea

* PACF measures the correlation between and after removing the correlation introduced by intermediate lags on
* The PACF at lag 1 is the correlation between
* The PACF at lag 2 is the correlation between after removing the effects of
* The PACF at lag 3 is the correlation between after removing the effects of
* The correlation at lag k is only high if it adds additional info that is not already accounted for by all the lags prior to it

1. Calculation of PACF

* Compute a correlation between which accounts for the correlation introduced from intermediate lags
* How do we remove the effects of intermediate lags? By subtracting the linear impact of the intermediate lags on as given by a linear regression
* The PACF is given by:

Where

* In practice, software packages implement more efficient ways of calculating the PACF
* Assumption: the time series is stationary which means the following should not change with time:
* Mean: try de-trending the data if needed
* Variance: log transform the data to stabilize variance if needed
* Autocorrelation: the correlation between should not depend on t. There are no simple transforms of the data to try to enforce this
* We can still get some info from the PACF even when these assumptions are not met exactly >< difficult to interpret the PACF

1. PACF for different time series

* White noise

A comparison of a graph

Description automatically generated with medium confidence

* The PACF for white noise shows no large significant partial autocorrelations at any lag as we would expect
* Seasonality
* AR(1) process

A graph of different sizes and shapes

Description automatically generated with medium confidence

* PACF is large for k=1 and not significant for k > 2 as we wanted
* AR(2) process

A graph of different sizes and shapes

Description automatically generated with medium confidence

* PACF has significant lags at k = 1 and k = 2 as expected
* Trend (& seasonality)
* Note: not stationary – trend means that mean changes in time

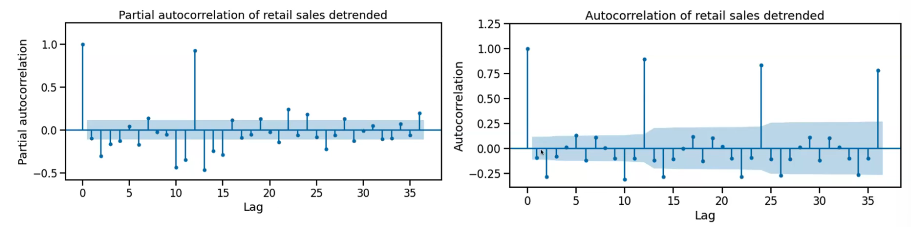
A graph of different sizes and shapes

Description automatically generated with medium confidence

* Despite the lack of stationarity we can still see a strong peak at k = 12 and smaller lags. There are also other lags which are more difficult to interpret (k = 10, k = 13)
* We know that the PACF assumes the time series is stationary. Let’s try detrending the data

A graph of a graph and a graph of a graph

Description automatically generated with medium confidence



* As we have seen previously, the ACF shows a clear signal of seasonality
* Note: variance still changes with time -> not stationary -> address this by also de-seasonalizing
* The PACF shows a strong lag at 12 but not at multiples of 12 afterwards.
* Most of the predictive info from the seasonal lags is captured at lag 12 as expected
* It’s harder to interpret whether other significant lags should be included. In practice this is enough evidence to try some of the smaller lags (k = 1, 2, 3) and measure the impact in modeling
* The other lags such as lag 10, 11, 13 are more difficult to interpret. One reason for not including them is that you do not see peaks at multiples of 10, 11, or 13 in the PACF. Also from domain knowledge, it is highly unlikely that lags much beyond 12 months will be relevant
* Detrending and de-seasonalizing the data
* Residual component = detrending and de-seasonalizing the data
* More stationary-looking data

A comparison of a graph

Description automatically generated

* There aren’t very large significant lags other than potentially k = 4, 10, 12, and 24
* There are significant lags at multiples of 12 which suggests seasonal component is still in the residuals and was not perfectly extracted by STL.
* May need to tune STL seasonal parameter
* Practically speaking from looking at this plot there wouldn’t be an additional lag beyond 1 or 2 that we would want to add for feature engineering purposes

1. PACF implementation in Statsmodels

A screenshot of a computer program

Description automatically generated

A screenshot of a computer screen

Description automatically generated

1. Summary

* PACF measures how corelated a time series is with itself at lags after removing the effect of intermediate lags
* The PACF for an AR(p) process has non-zero lags up to p and 0 afterwards
* Assume the time series is stationary
* The PACF can help identify lags which may be helpful for forecasting

# Cross-correlation function

1. Cross-correlation function

* So far we have shown the correlation between and a lagged value of itself
* Now we show the **correlation between and the lagged value of a feature**
* The cross-correlation function between is defined as:
* This is equivalent to
* We want to know which lags are most predictive of
* We can look at the cross-correlation function between to do this
* When the correlation between and does not depend on time we can write
* Equivalent to:
* Extended equation
* Disclaimer: This only captures the linear relationship between and . If a non-linear relationship exists, it wont necessarily be picked up by the CCF
* Need to transform the feature (e.g., power transform, Box-Cox transform) before plugging it into the CCF

1. CCF for different time series

* White noise – 2 time series

A graph of sound waves

Description automatically generated

A diagram of a graph

Description automatically generated

* CCF shows very small correlations as expected for 2 uncorrelated time series. We expect some significant lags just by chance so these are not anything to be concerned with
* The 95% confidence interval (blue shaded area) for the cross-correlation is given by: where n is the length of the time series. The null hypothesis is that there is no correlation
* AR(1) process – 2 correlated time series

A graph of a graph of a graph

Description automatically generated with medium confidence

* Lag plots: we see that the largest cross-correlation occurs at a lag of 10 as expected

A group of blue dots

Description automatically generated

* CCF plot

A graph with blue dots and red line

Description automatically generated

* We see many significant lags even though only one is important (lag 10). Nevertheless, we see that the CCF peaks at the lag of 10 allowing us to determine that it is an important lag
* 2 time series with trend and seasonality

A graph of sales and a few other sales

Description automatically generated with medium confidence

* Lag plot
  + Due to the trend we see that when one time series is small so is the other, when one is large, so is the other
  + As a result the time series are correlated at many different lags even though we know these 2 time series are not causally linked
  + **In fact any 2 time series with a trend would be cross-correlated. Therefore, the CCF is not useful for these cases**
* CCF

**A graph of a graph with blue dots

Description automatically generated with medium confidence**

* + we see many significant lags in the CCF due to the trend. We also see oscillations in the CCF as both time series have seasonality
  + Takeaway: De-trend and de-seasonalize (i.e., try to make stationary) data with strong trend and seasonality for the purpose of measuring the CCF between 2 time series
* De-seasonalize and de-trend the data

**A graph of air passengers and retailing

Description automatically generated**

* The CCF shows much less correlation now. There are no large significant lags.
* There is not much info in one of these time series to predict the other, as we would expect

1. CCF implementation in Statsmodels

A screenshot of a computer code

Description automatically generated

1. Summary

* CCF measures how correlated is with another variable at some lag
* Large values in the CCF can help identify useful lags of a feature to use for forecasting
* Trend and seasonality can create spurious correlations. Try to ensure the data is stationary before using the CCF

# Creating good lag features demo