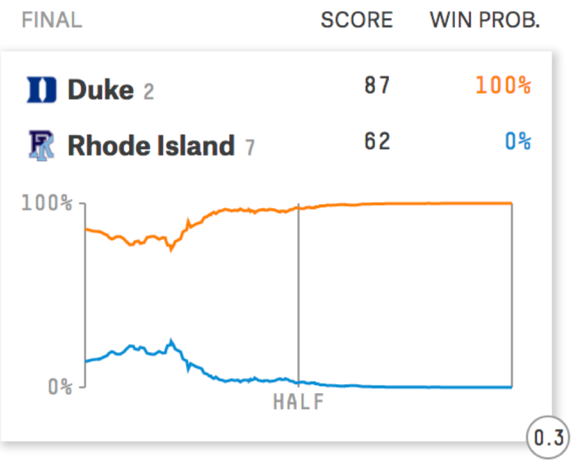
Lecture 19:
Introduction to Time Series

David Carlson

Time Series are Everywhere

- Previously, we assumed that each instance of data that we had was *independent* and that there was no structure defining the relationships between data points.

- However, *time series* are everywhere.
  - Sensor data
  - Medical records
  - Text
  - Audio
  - Migration trajectories
- We need to utilize this structure to understand and utilize the data.



Even basketball games are time series!

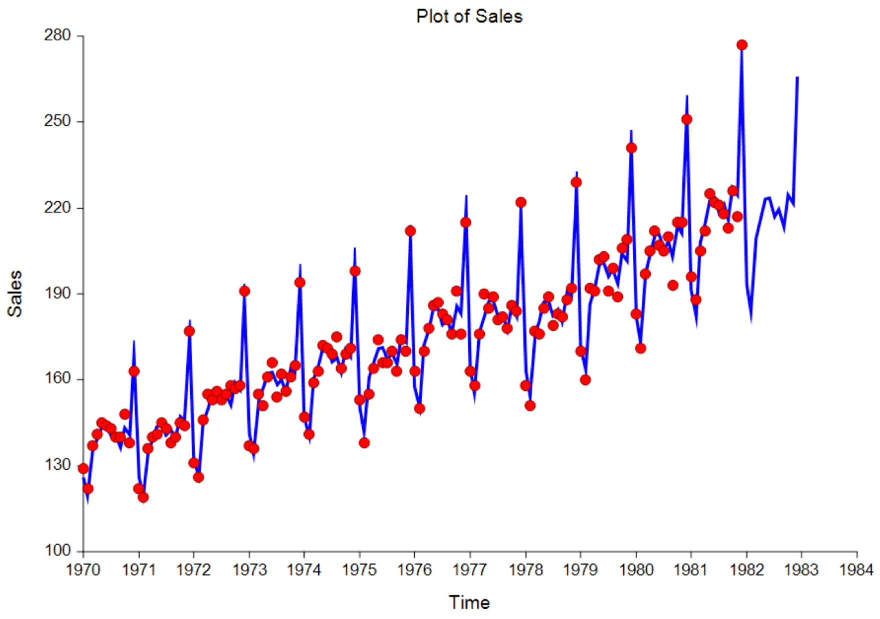
What are some types of questions we can ask?

- There are a variety of possible questions that we could ask, some examples:
  - What will happen tomorrow? (e.g. weather, planning, hospital readmissions)
  - What is the relationship between two time series?
  - What is the instantaneous probability of an event (point processes, survival analysis)
  - What are the underlying features of this dataset?
  - What sounds were just made (speech, songbirds, etc.)?
  - What is the causal relationship between time series?

- We'll about some of these applications.

Forecasting

- There are many different types of forecasting, but essentially, the question is what will happen in the future?
- How can we learn these relationships?
- How can we evaluate the performance of our model?



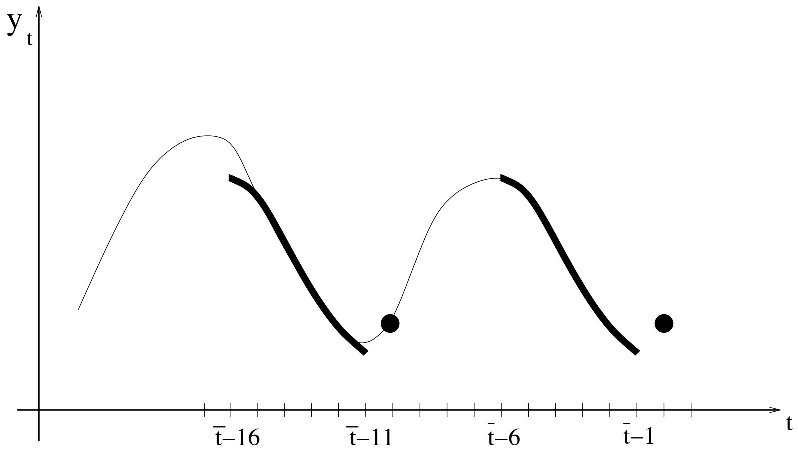
Forecasting as Supervised Learning

- In the past, we've always had the relationship that we were trying to predict an outcome y from covariate/features $\boldsymbol{x}$, where
  - $y \simeq f(\boldsymbol{x})$
- In a time series, we have have one long *discrete* time series that we want to learn to make forecasts from, $\boldsymbol{x}_{1:T}$. We may have other covariates as well, but we will ignore those for now.

- Let's turn this into a prediction task by learning a model (one of our many classification/regression techniques) that predicts:
  - $x_{t+1} \simeq f(x_t)$
- A side comment: we will talk about some specific models for time series as well, but right now will focus on the problem setup.

Often want to use an abundance of history

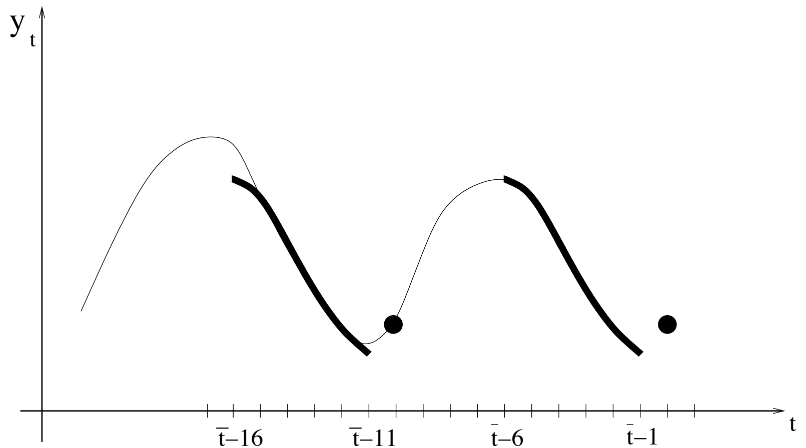
- Predicting the next point is called one-step-ahead prediction in discrete time series.
- Instead of using a single earlier time point to predict the future, we may want to consider a larger history.
- The number of previous time points we're considering is the number of *lags*.
- This is stated as:
- $x_{t+1} \simeq f(\boldsymbol{x}_{t-l:t})$



Nearest Neighbor Forecasting

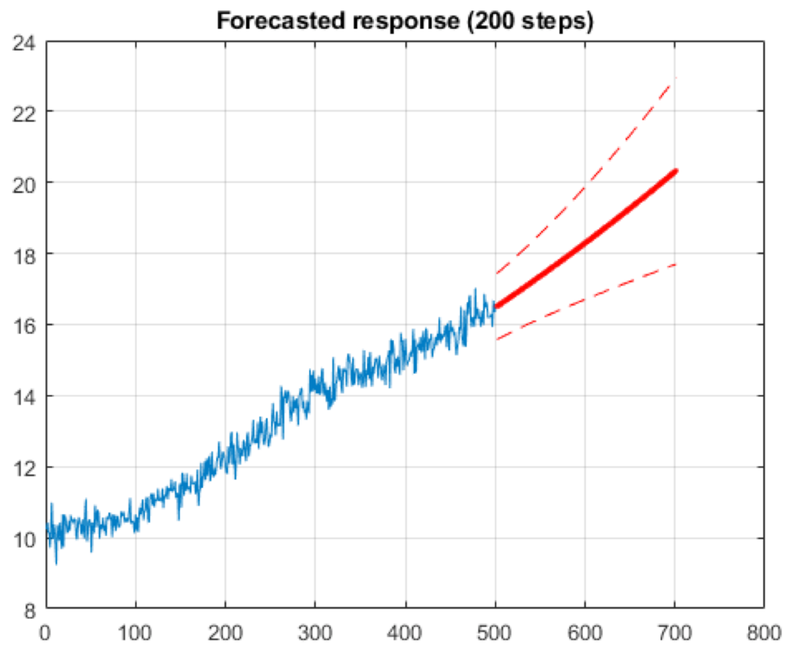
- At this point, we can use whatever method we want for the forecasting, because we have a very similar situation as before.
- In particular,
- $x_{t+1} \simeq f(\boldsymbol{x}_{t-l:t})$
- means we can treat x_{t+1} as the outcome or label and $\boldsymbol{x}_{t-l:t}$ as the features.

- Shown on the right is a nearest neighbor formulation where you search for similar patterns through the history.



Iterated Prediction

- Often, we care about predicting more than 1 step ahead.
- I.E discretization intervals on temperature can be given by minutes, but we want to know the temperature tomorrow.
- Often care about iterated prediction (or a multi-step prediction).
- Many statistical models to maintain uncertainty.
- Can use any supervised approach by just sequentially feeding in predicted values.



Evaluation:

- Since we can treat time series forecasting as a supervised learning problem, can we just run our standard cross-validation and be done?

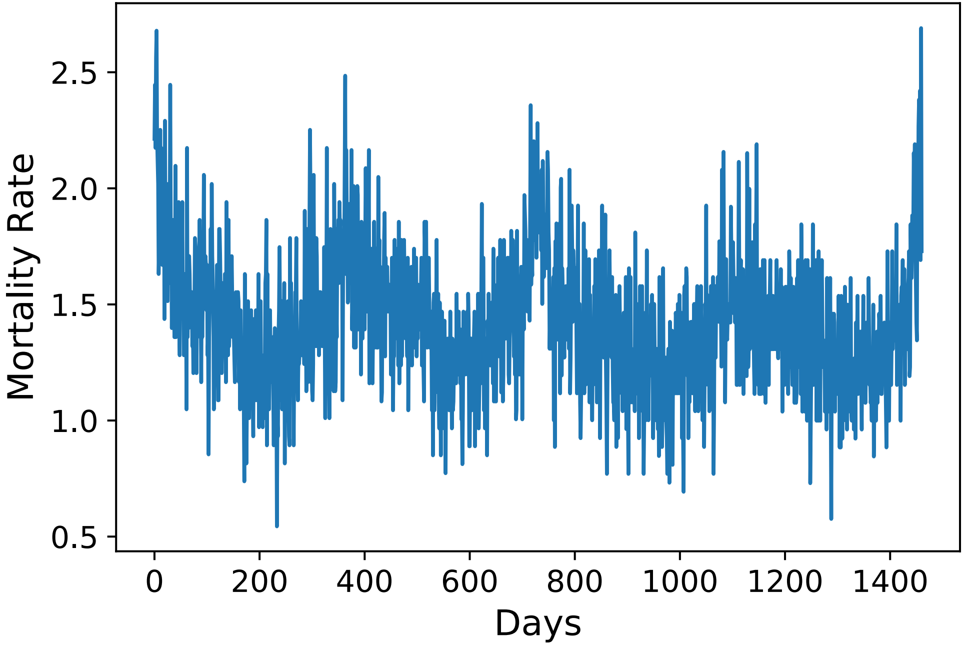
- Answer: No.

- Why not?

VALIDATION IN TIME SERIES

Revisiting Cross-Validation

- In cross-validation, there is a critical assumption that the data samples are *independent.* Are the responses in time series independent variables?



What does it mean to be independent?

- Statistically, independence means that there is no information in the joint model, meaning that:
    - $p(x_t, x_{t+1}) = p(x_t)p(x_{t+1})$
- This implies that there is no information about each other in these values (after the model is fit, at least).

- The very fact that we think we can predict the future from the *historical time series* means that we think that this assumption is untrue.
- Does this matter?

We need to adjust cross-validation

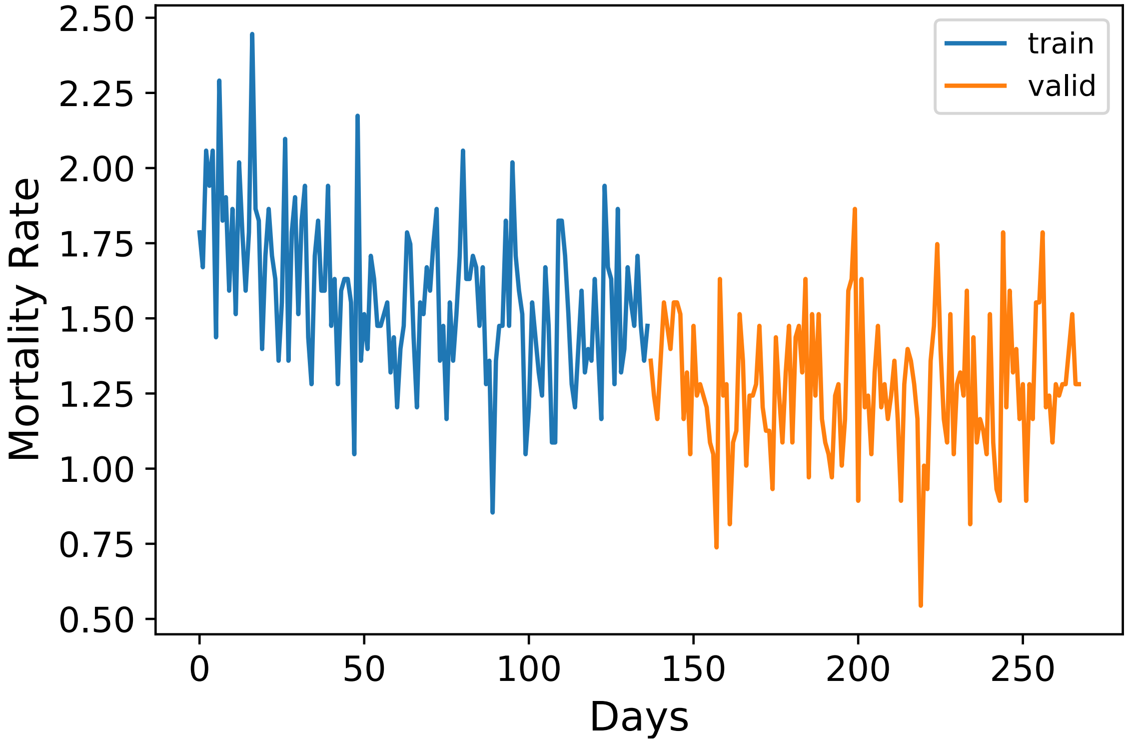
- We'll be talking about two different strategies:
  - Forward prediction
  - Creating pseudo-independent blocks

First strategy: only predict forward

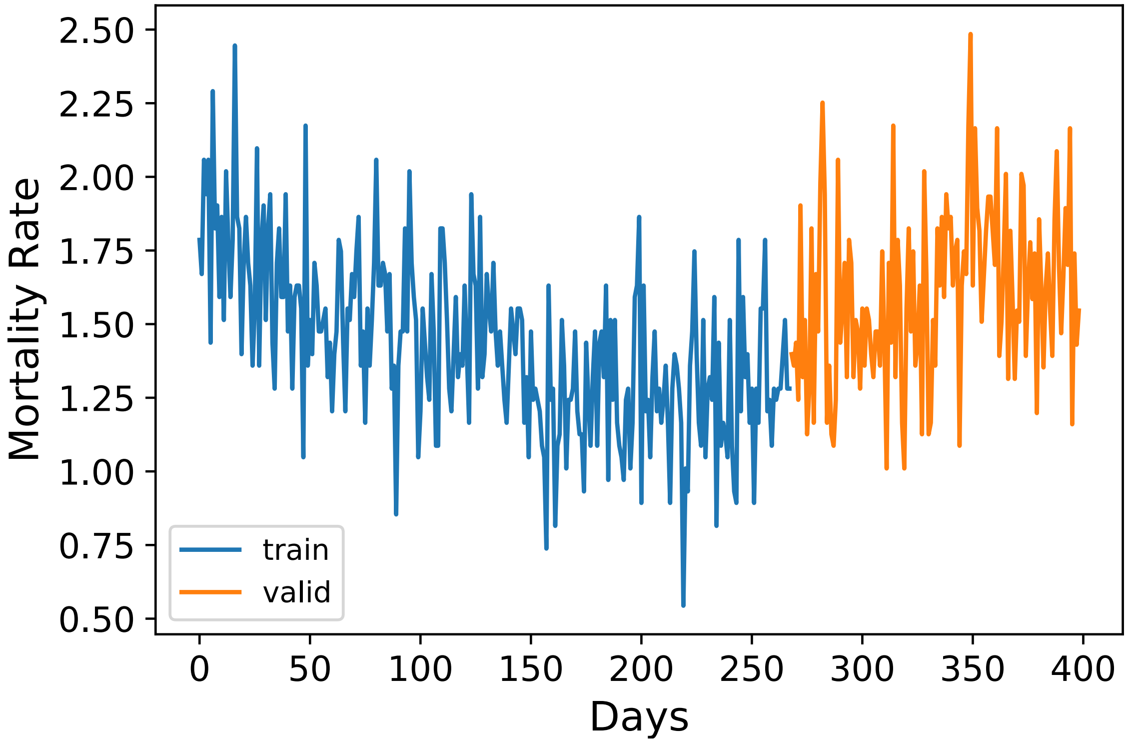
- The first strategy is roughly as follows:
  - Break the time series into K contiguous chunks
  - For $k = 1$ to $K-1$:
    - Train on the first k sets of data
    - Evaluate on the $(k+1)th$ set of data

- Let's go through this visually to understand what's happening, and then return to the rationale.

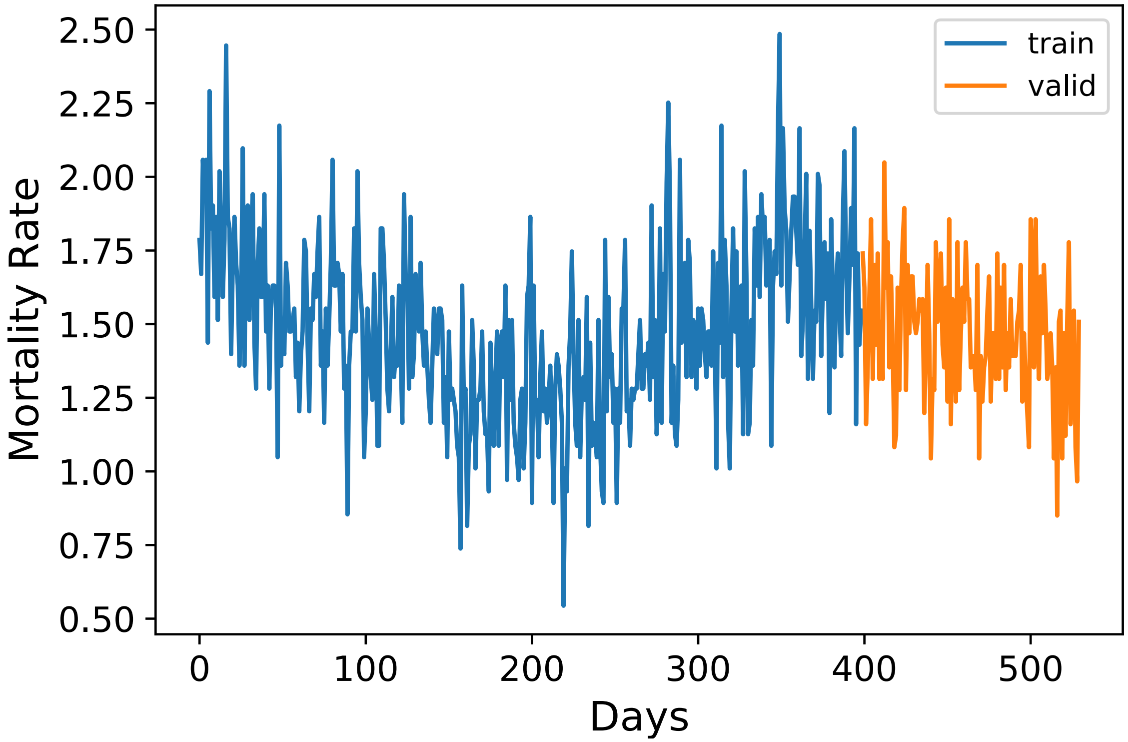
First Training Set



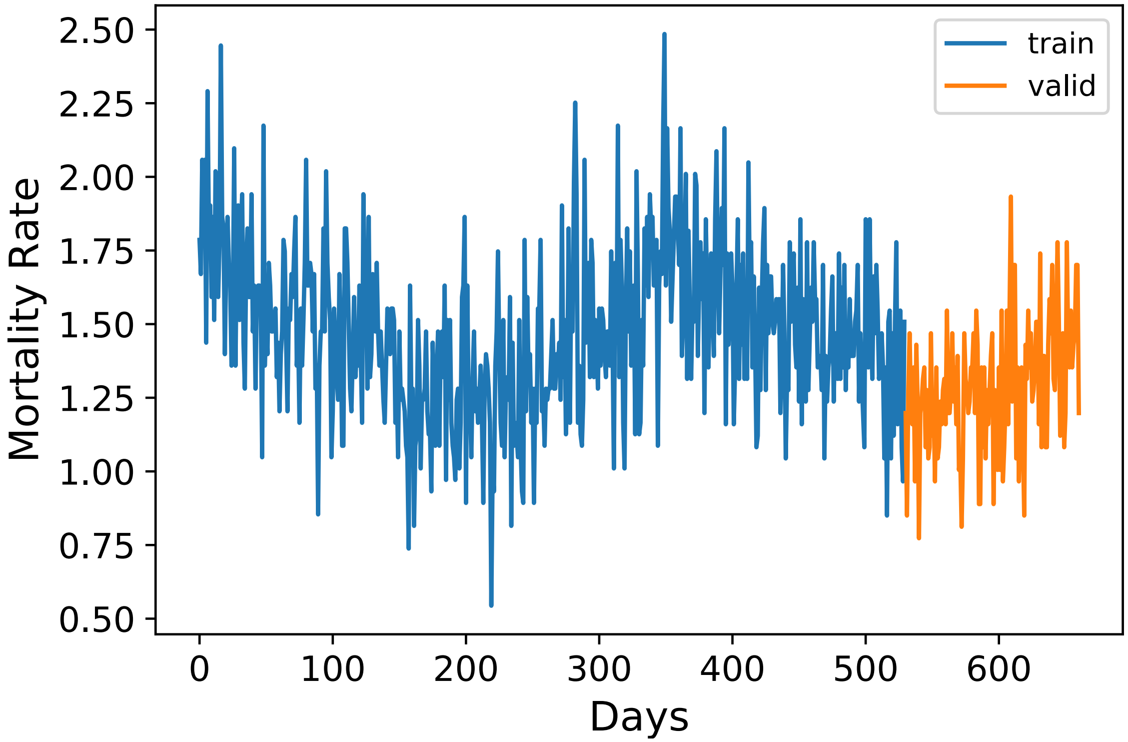
Second Training Set



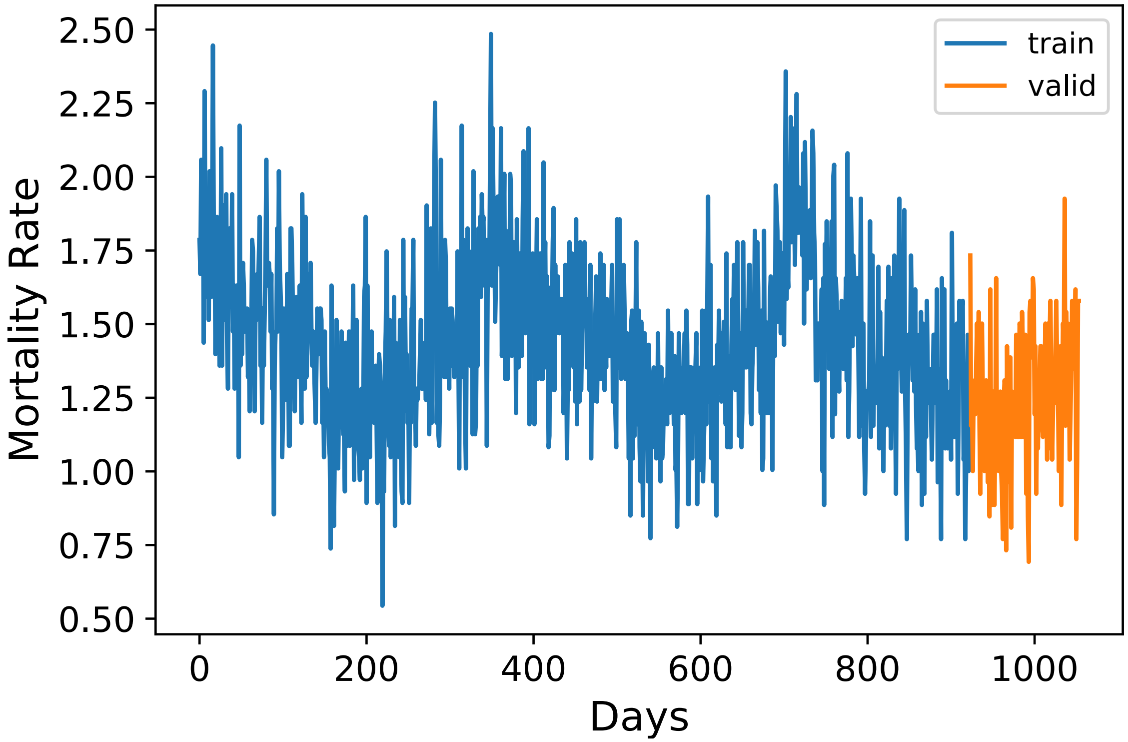
Third Training Set



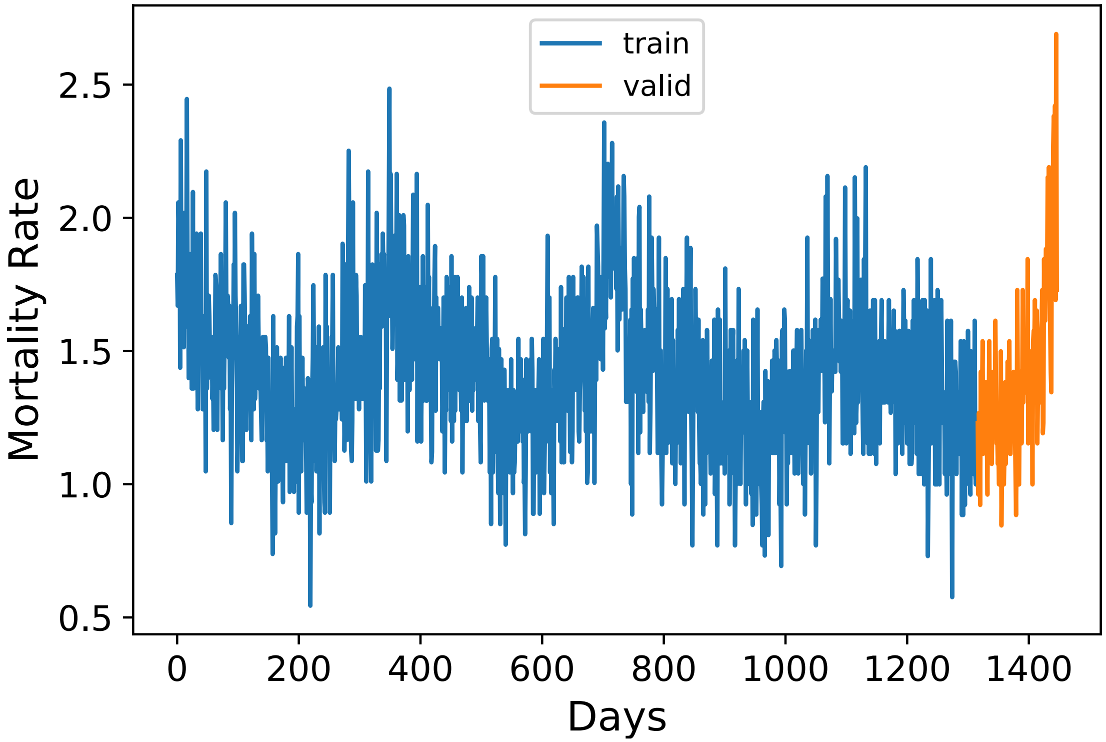
Fourth Training Set



Seventh Training Set



Tenth (and last) Training Set



Pro/Cons of this Approach

- Cons:
  - Doesn't use all of the data
  - Train sizes are variable; best performing model may depend on the amount of data given
  - In a lot of ways, the last evaluation split is the most indicative of real-world performance (because it's what happens when you have all the training data)
- Pros:
  - Simple to implement, theoretically sound
  - Mimics how the real world works. If we're interested in predicting the future, then this mimics only having historical data.
  - Realistic of real learning process
  - Learning can be rigorous analyzed both from a statistical point of view and from an "online learning" point of view

Big Pro of this Approach

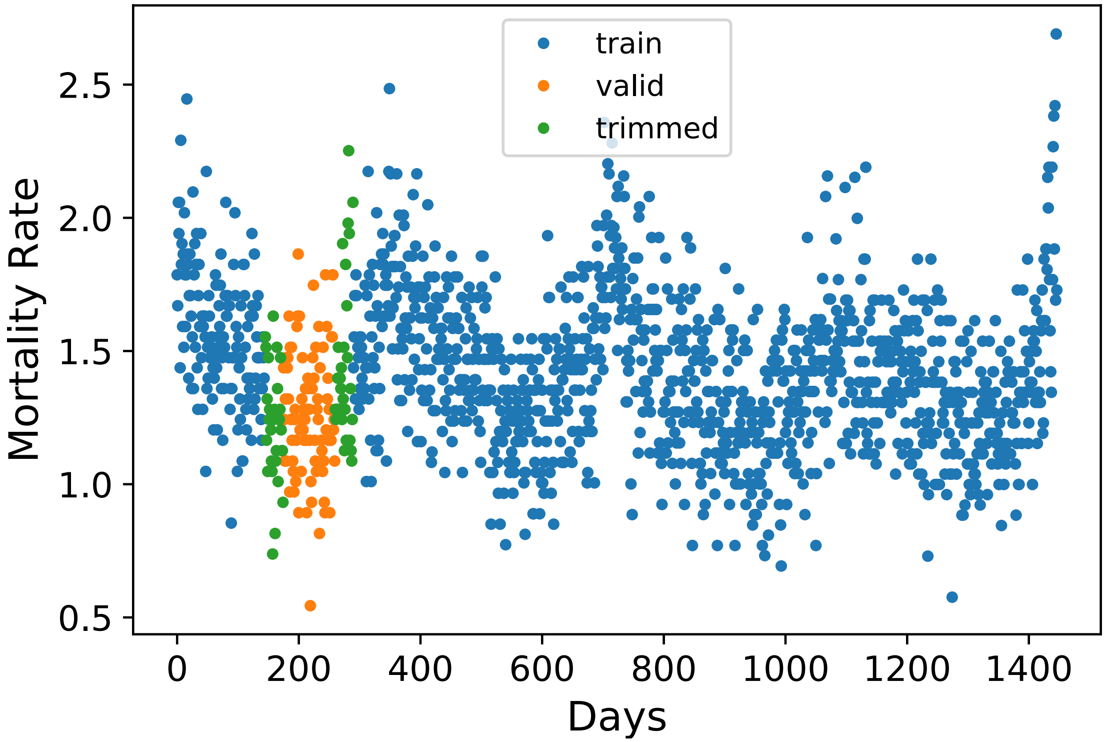
- Can implement in practice by invoking the "TimeSeriesSplit" method in python (https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.TimeSeriesSplit.html)
- One line change to include in our cross-validation grid-search methods…

NON-FORECASTING VALIDATION

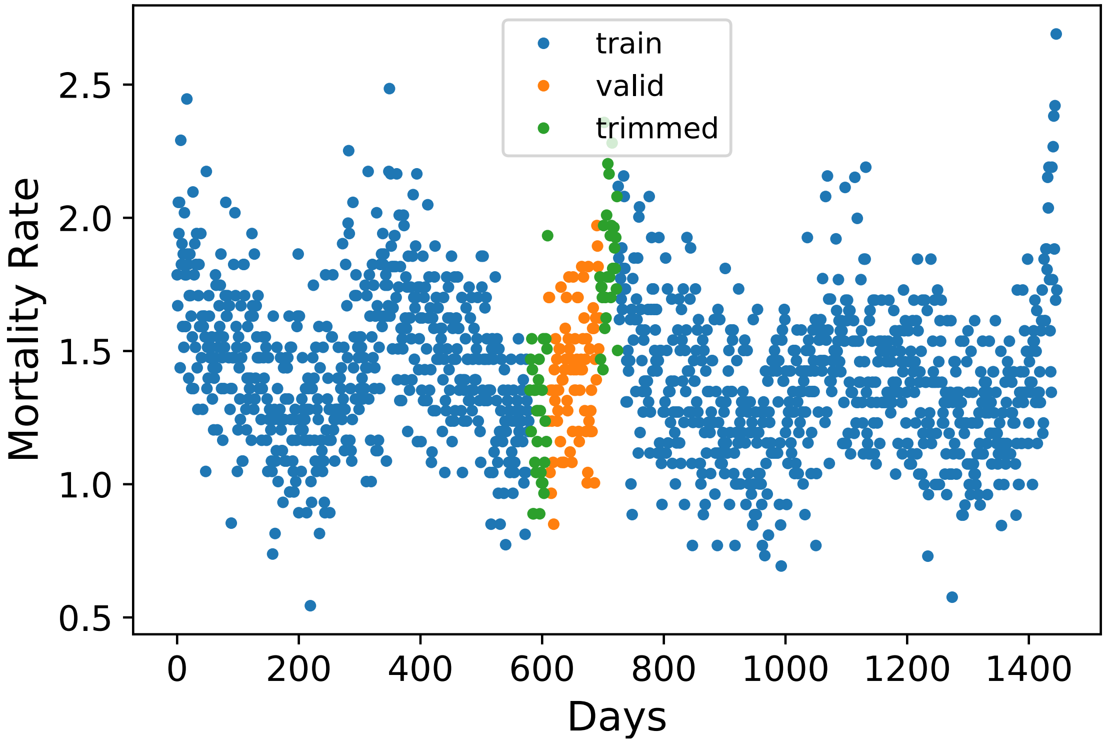
A second approach

- Make pseudo-independent cross-validation sets using the whole data. This is appropriate when the question is about the relationship between covariates and the time-series outcome instead of strictly forecasting.
- The fundamental idea:
  - y_t and $y_{t+\tau}$ are independent as τ goes large
- Essentially, then the approach is
  - Break the time series into *K* contiguous chunks
  - For each chunk:
    - Remove the edges of the contiguous period (i.e. get rid of τ samples near the borders)
    - Train on the rest of the time series
    - Evaluate on this trimmed chunk
- Big question: how big does the gap have to be? We'll come back to this.

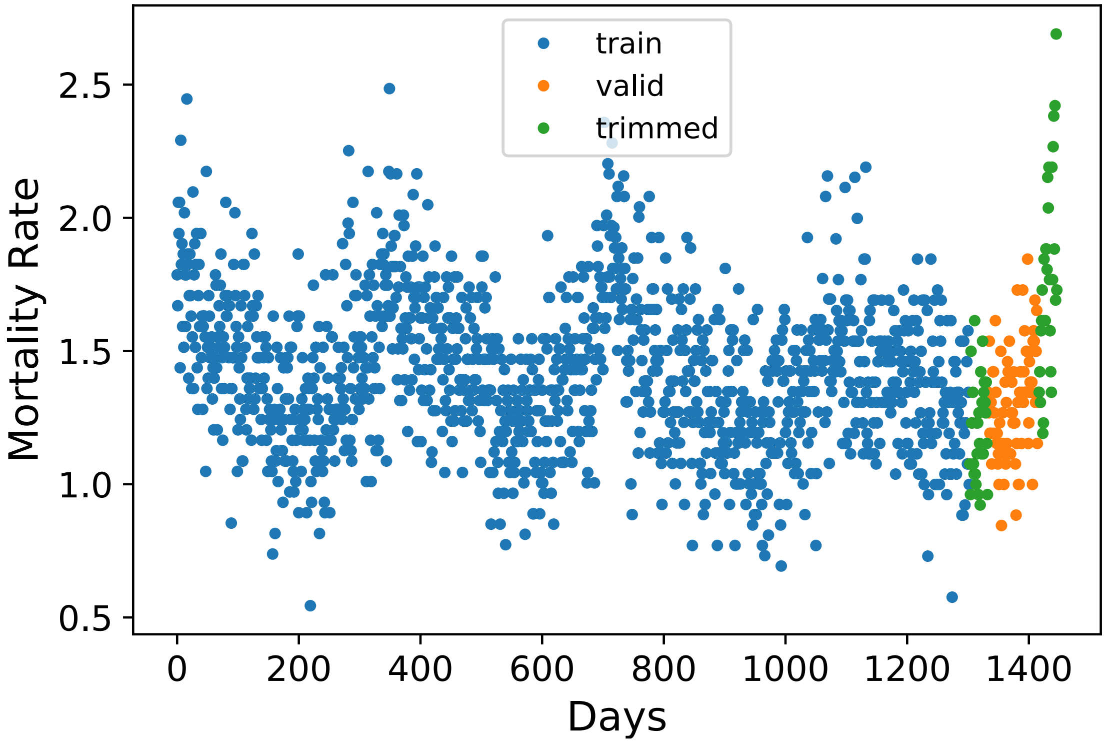
Trimming (Pseudo-Independent)



Trimming (Pseudo-Independent)

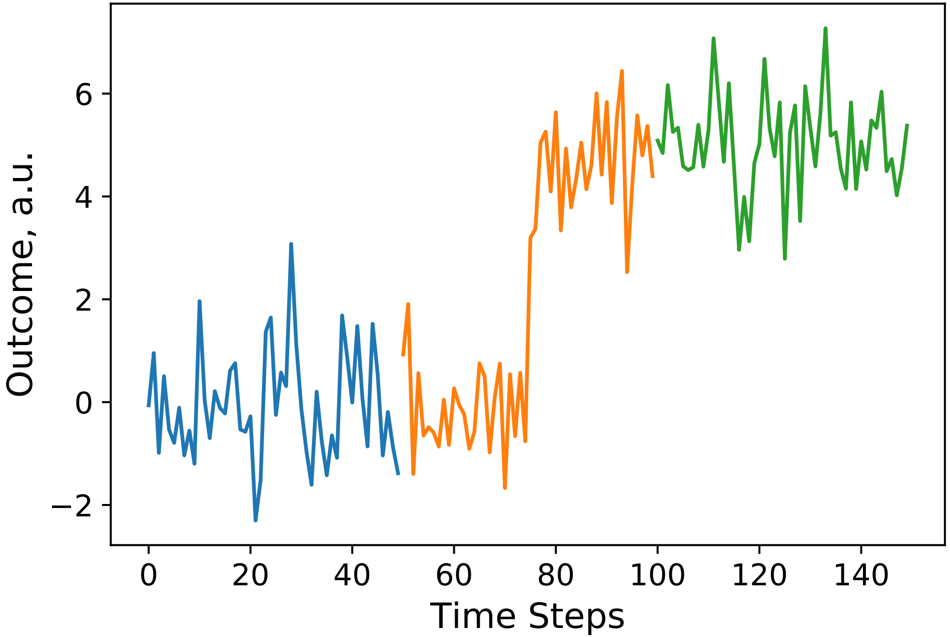


Trimming (Pseudo-Independent)



Is this approach better?

- Only for *some* questions.
- This using all the data and can be appropriate for asking questions about relationships.
- Not appropriate for asking questions about forecasting ability. Consider the case on the right.



How much data to trim to get independence?

- This is an extremely difficult question to answer; often a heuristic is to take a number of samples to get below the threshold in autocorrelation.
- Autocorrelation assumes that we have a *stationary process*, which essentially means that $p(x_i) = p(x_j)$ for any points in time, or that the distribution doesn't shift when we move through time.
- Autocovariance at lag *t* is defined as:
- $E_t[(x_t - \mu)(x_{t+\tau} - \mu)]$
- Which can be empirically estimated as:
- $E_t[(x_t - \mu)(x_{t+\tau} - \mu)] \simeq \frac{1}{T-\tau}\sum_{t=1}^{T-\tau}(x_t - \bar{x})(x_{t+\tau} - \bar{x})$

DATA SEASONALITY

Does this explain all that we need?

Autocorrelation shown at the right reveals that all the samples are correlated!

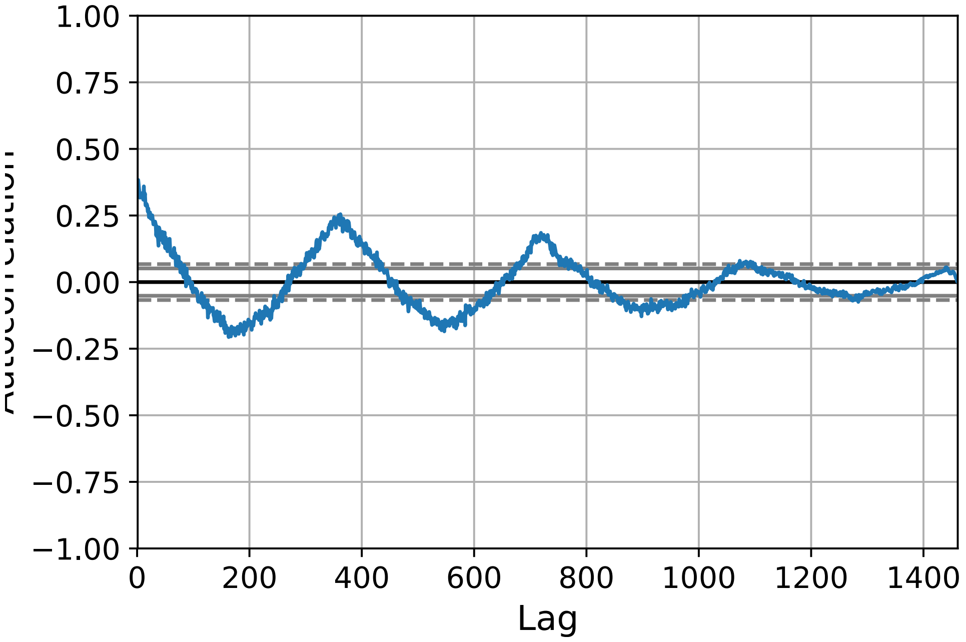
Think about the fact that there are clear *seasonal* trends that make this analysis more complicated.

There are several types of trends that we'll want to deal with:

Linear

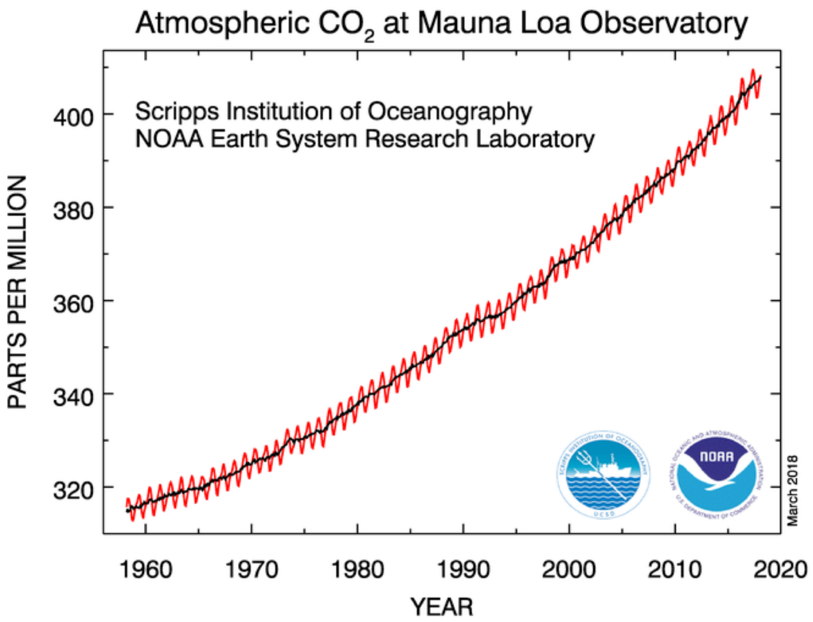
Non-linear

Periodic (this case)



We often want to remove periodic trends

- On the right is data from the Mauna Loa observatory.
- In that case, we wanted to evaluate the overall trend in the data, and the periodic trend is distracting.
- Also, not modeling it removes statistical power.
- How can we remove such features?
- Two main approaches:
  - Autoregressive models in statistics
  - Filtering/decompositions in signal processing



Conclusions from Today

- We can treat time series forecasting as a supervised learning problem, and incorporate many of our previous techniques.
- Evaluation requires some careful analysis, but can be done.
- Reminder: forecasting is different from extrapolation
- We want to be able to clean our data and remove certain trends. Our analysis will be much stronger if we do that.