# Model Selection

Jared Fisher

Lecture 09b

# Announcements

# Announcements

- Homework 5 is due Friday, April 9 by 11:59pm.
- Midterm 2 is next Thursday on April 15, and will contain two separate parts:
  - Timed multiple choice section on Gradescope (like Midterm 1, but timed at 20 minutes)
  - Written portion that you'll have the full 24 hours to complete, though it's intended to take 30 minutes or so. You'll upload this to Gradescope like a homework assignment. You may hand-write or type (LaTeX, Markdown, etc.)

# Recap

# Definition: Ljung-Box-Pierce test

- Fix a maximum lag $k$ (typically $k = 20$).
- Reject the hypothesis that data $x_1, \ldots, x_n$ was generated from a causal and invertible ARMA(p,q) model if

$$\tilde{Q}(x_1, \ldots, x_n) > q_{1-\alpha},$$

- where $q_{1-\alpha}$ denotes the $(1-\alpha)$-quantile of the $\chi^2$ distribution with $k - p - q$ degrees of freedom.

# What is this Q?

- Assume that the data $X_1, \ldots, X_n$ is generated from an invertible ARMA(p,q) model with parameters $\phi, \theta$.
- By invertibility:
$$X_t = -\sum_{j \geq 0} \pi_j X_{t-j} + W_t$$
- Hence, the best linear prediction of $X_t$ based on$X_{t-1}, X_{t-2}, \ldots$ is given by
$$\hat{X}_t(\phi, \theta) = -\sum_{j \geq 0} \pi_j X_{t-j}.$$
- The residuals $R_t = \hat{X}_t(\phi, \theta) - X_t = W_t$ coincide with the white noise process $\{W_t\}$.

# What is this Q?

- Recall that sample acf of the residuals $R_t$ $(r_1, \ldots, r_k)$ for some maximal lag $k$, are approximately i.i.d. N(0,1/n)
- Thus, we create $Q$ which follows a chi-square distribution with $k$ degrees of freedom:

$$Q = n \sum_{i=1}^{k} r_i^2 \sim \chi_k^2$$

- When the true parameters $\phi, \theta$ are replaced by appropriate estimates $\hat{\phi}, \hat{\theta}$, the respective estimated residuals

$$\hat{R}_t = \hat{X}_t(\hat{\phi}, \hat{\theta}) - X_t$$

should still be approximately white noise.

# What is this Q?

- The *Box-Pierce* test statistic is

$$\hat{Q} = n \sum_{i=1}^{k} \hat{r}_i^2$$

- Under an ARMA(p,q) model, one can show that for $n$ large enough $\hat{Q}$ is approximately chi-square distributed with $k - p - q$ degrees of freedom

$$\hat{Q} \to \chi^2_{k-p-q} \quad \text{for } n \to \infty.$$

- In practice, one often considers a slightly modified version of the statistic $\hat{Q}$, namely

$$\tilde{Q} = n(n+2) \sum_{i=1}^{k} \frac{\hat{r}_i^2}{n-i},$$

which is denoted as the *Ljung-Box-Pierce* test statistic.

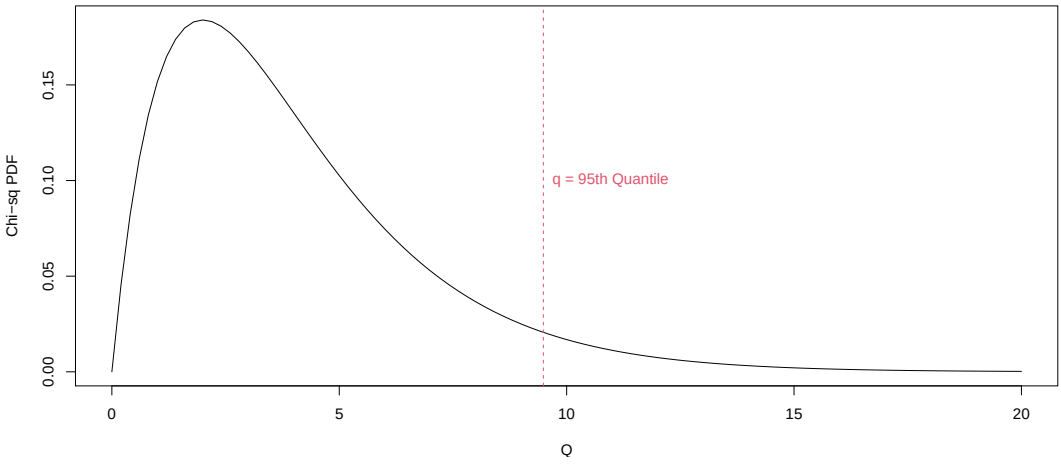
# Definition: Ljung-Box-Pierce test

- Fix a maximum lag $k$ (typically $k = 20$).
- Reject the hypothesis that data $x_1, \ldots, x_n$ was generated from a causal and invertible ARMA(p,q) model if

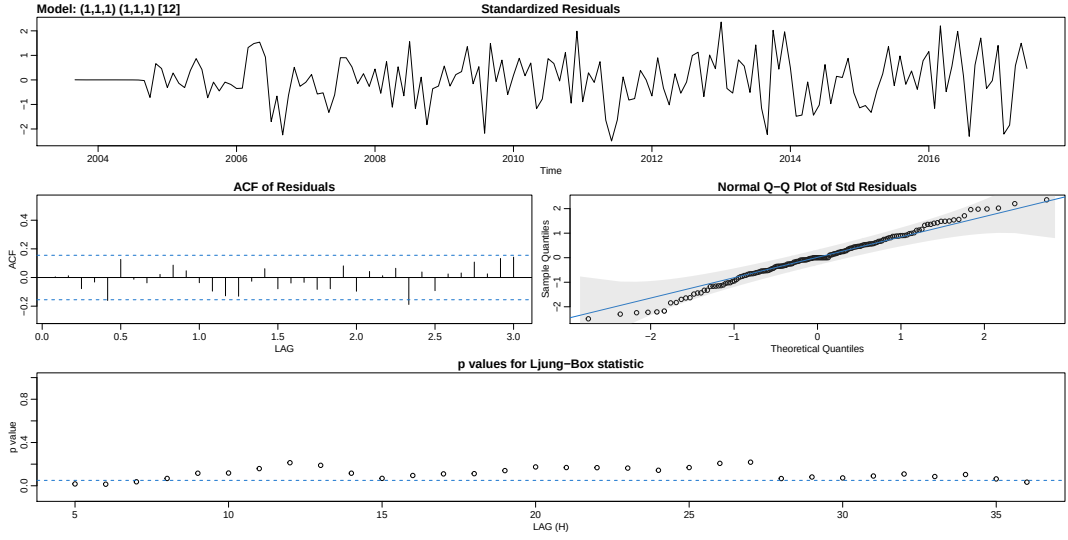
$$\tilde{Q}(x_1, \ldots, x_n) > q_{1-\alpha},$$

- where $q_{1-\alpha}$ denotes the $(1-\alpha)$-quantile of the $\chi^2$ distribution with $k - p - q$ degrees of freedom.

# Higher Q = Smaller P-value = Evidence Against ARMA(p,q)



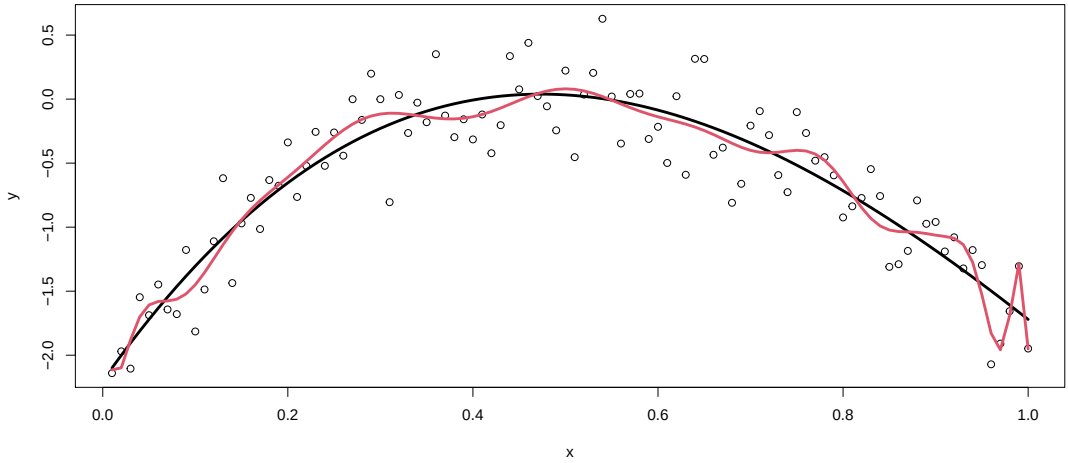
# Ljung-Box in sarima() diagnostics



# Which model fits best in-sample?

- The Ljung-Box-Pierce test provides a strategy to evaluate, for given $p, q$, whether or not an ARMA(p,q) model is appropriate for data $x_1, \ldots, x_n$. So just choose large p,q right?
- Clearly every ARMA(p,q) model can be arbitrarily-well approximated by an ARMA($p'$, $q'$) model with $p\prime > p$ and $q' > q$.
- Like with a polynomial in linear regression, the larger chosen degree of the polynomial, the better the fit.

# Overfit



# Overfit

- In the extreme case, when we have a 100 observations and fit a polynomial of degree 99 the fit will be perfect.
- However, such a model is likely overfitting the data and might be useless to predict future values.
- Model selection: we want the number of model parameters to be large enough, so that it can fit the data well. At the same time the number of model parameters should not be too large, which would result in overfitting: fitting the data and not the true underlying process.
- A solution: measure in-sample fit and penalize for model size/complexity.

# But First: Likelihood

- The likelihood is a function of the parameters, the same function as the density of the data

$$L(\textit{parameters}) = f(\textit{data}|\textit{parameters})$$

- For our models this typically looks like

$$L(\mu, \phi, \theta, \sigma^2) = f(x_1, ..., x_n|\mu, \phi, \theta, \sigma^2)$$

# Information Criterion

- $AIC = -2\log(\text{likelihood}) + 2k$
- $AICc = AIC + \frac{2k(k+1)}{n-k-1}$
- $BIC = -2\log(\text{likelihood}) + k\log n$
- Why have different IC's?

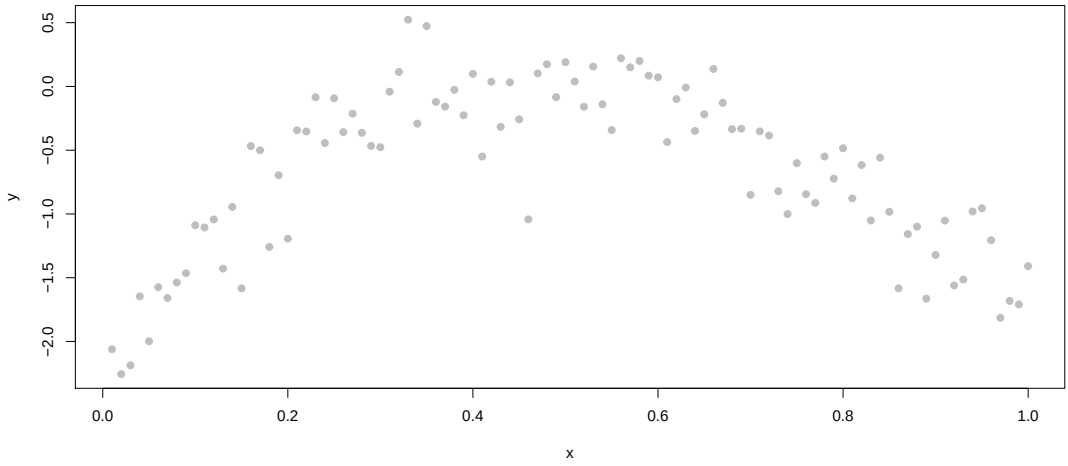
How do we compare models out-of-sample?

# Cross Validation

- Very popular technique for model selection and choice of tuning parameters, in general.
- General idea: we want to know how the model will perform out-of-sample, so let's reserve some of our data to check this out!
- Basic Idea:
  - Divide dataset into "training" and "testing" subsets
  - Fit all candidate models with the training data
  - Evaluate the performance of each model on the testing data
  - Repeat as needed
  - Select the model that performed the best

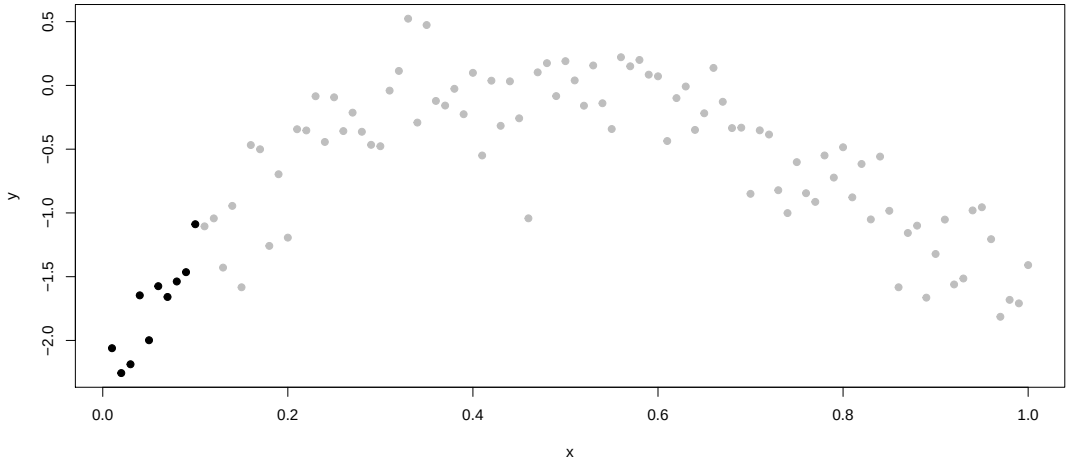
# Time Series Cross Validation, Visually

Gray points are data points that we treat as “unseen”



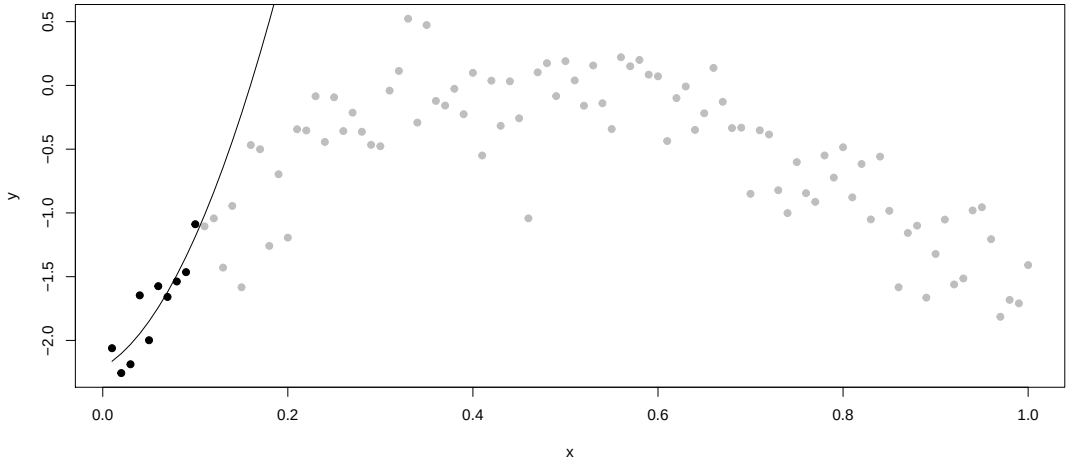
# Time Series Cross Validation, Visually

Black points are our training set (first 10 points)



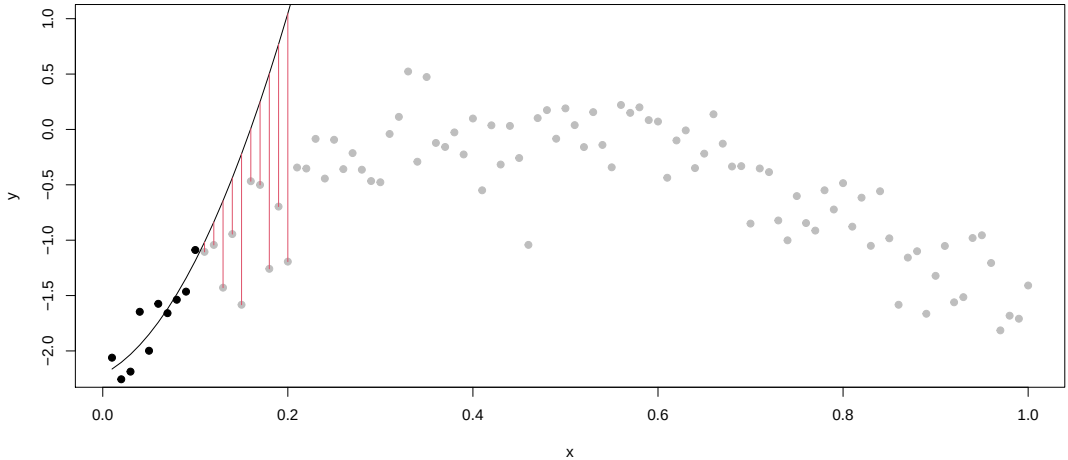
# Time Series Cross Validation, Visually

We have the black curve/model created from the training set



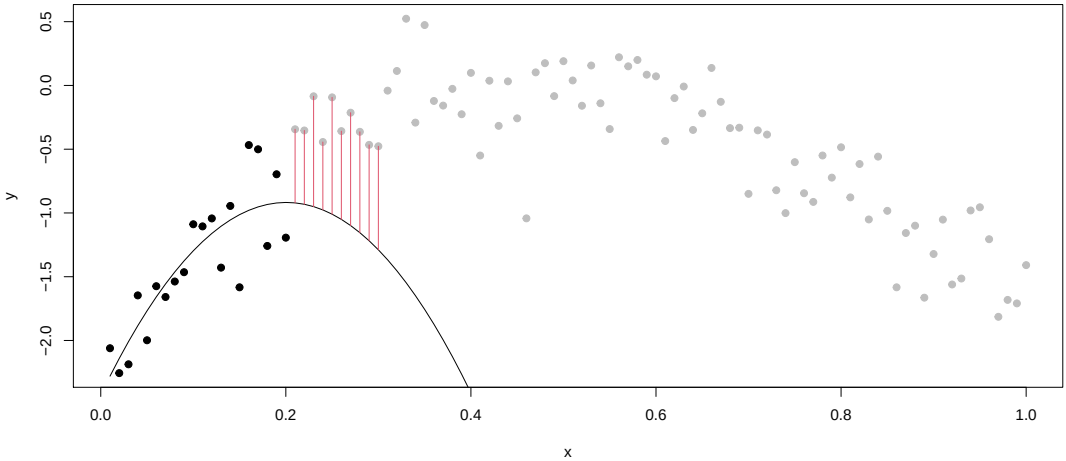
# Time Series Cross Validation, Visually

The residuals for the testing set (next 10 points), record this SSE



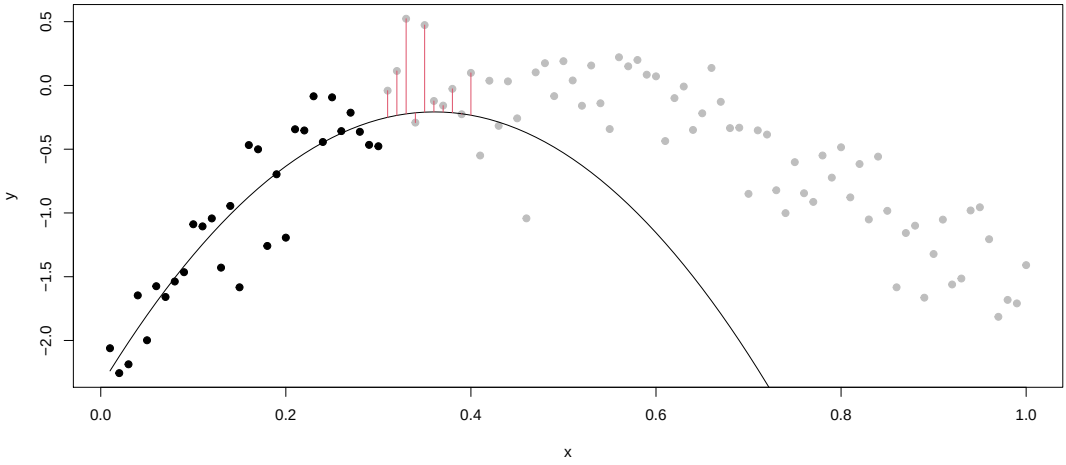
# Time Series Cross Validation, Visually

Roll forward 10 years:



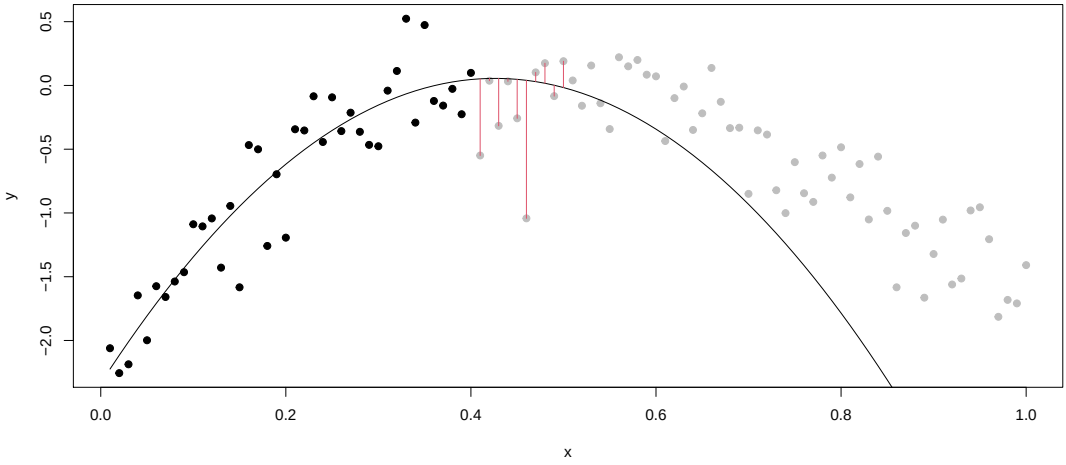
# Time Series Cross Validation, Visually

Roll forward 10 years:



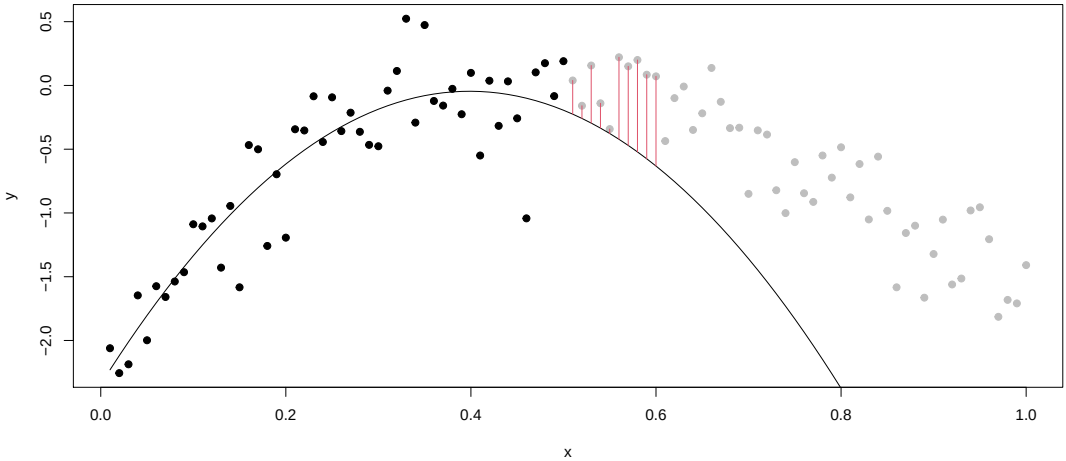
# Time Series Cross Validation, Visually

Roll forward 10 years:



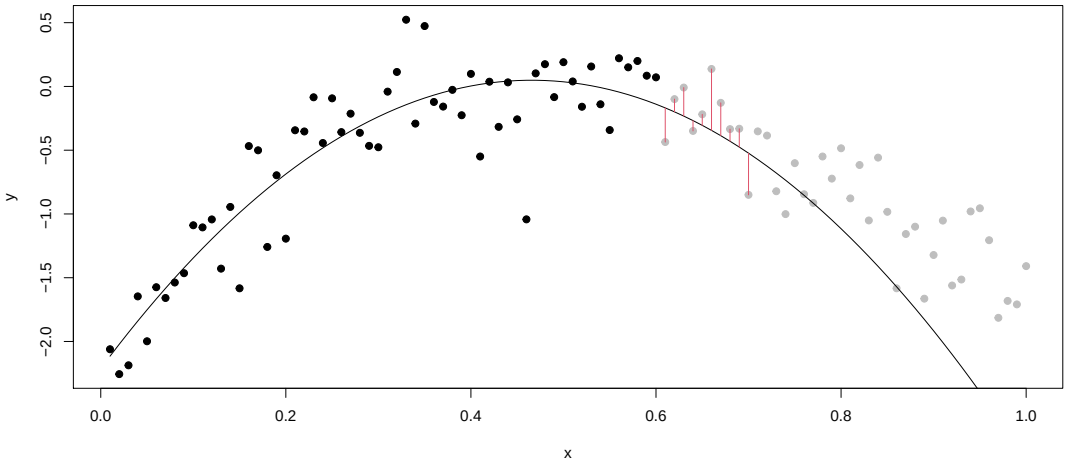
# Time Series Cross Validation, Visually

Roll forward 10 years:



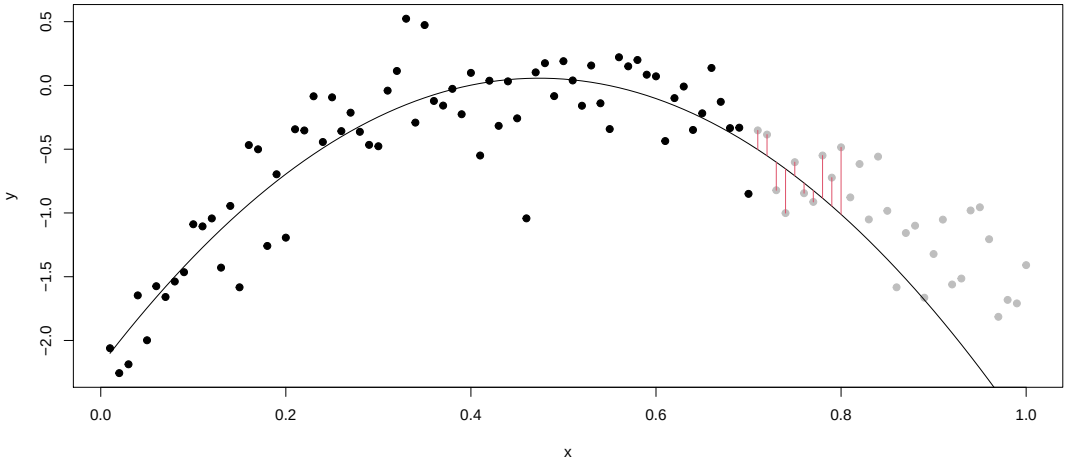
# Time Series Cross Validation, Visually

Roll forward 10 years:



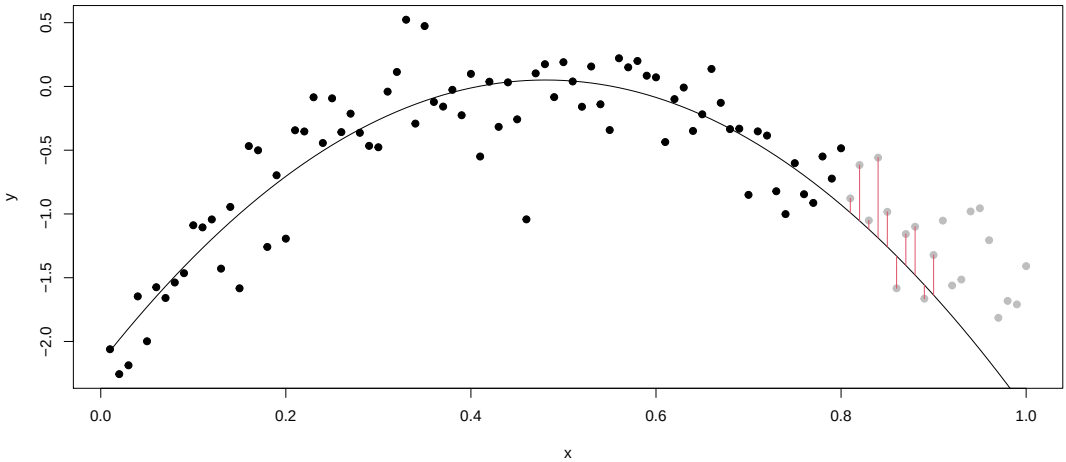
# Time Series Cross Validation, Visually

Roll forward 10 years:



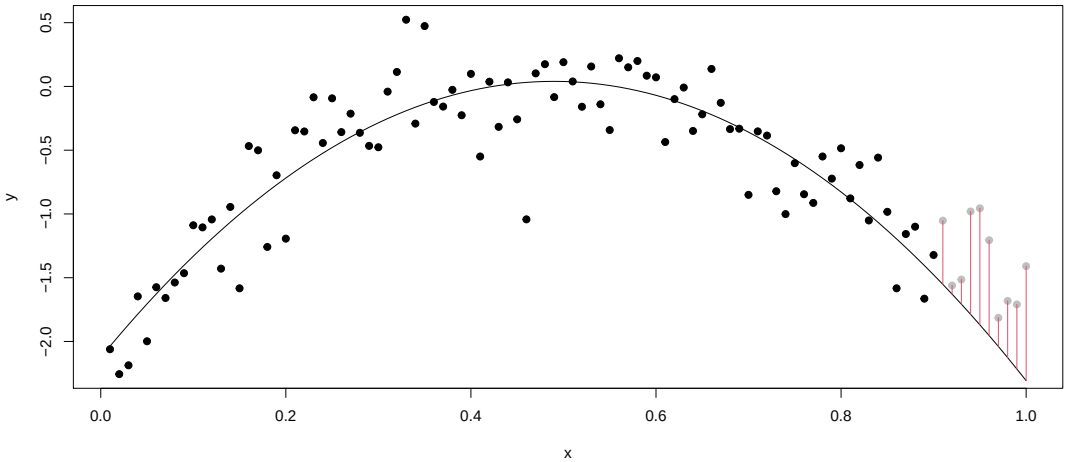
# Time Series Cross Validation, Visually

Roll forward 10 years:



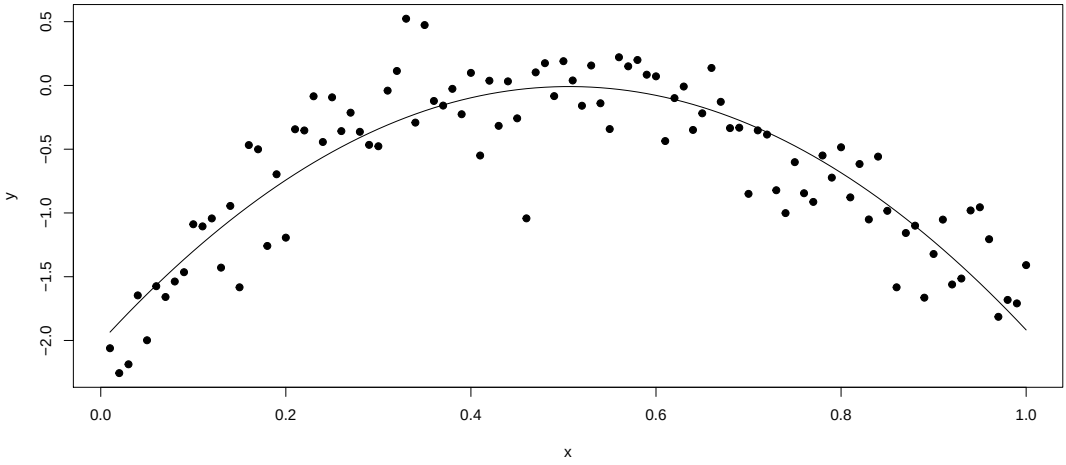
# Time Series Cross Validation, Visually

Roll forward 10 years:



# Time Series Cross Validation, Visually

The model for ACTUAL prediction uses all of the past data:



# Notes about this example

- Can use metric other than SSE/MSE
- Probably don't want to start with so little data.
- Note, this is also why we said that a parametric form can be poor at forecasting!

# Cross Validation Example Steps

- Disclaimer: there are many ways to do cross validation for time series. This is one.
- Suppose we have monthly data for $m$ years $x_1, \ldots, x_n$ where $n = 12m$ and the objective is to predict the data for the next year.
- Suppose we have $\ell$ competing models $M_1, \ldots, M_\ell$ for the dataset. We can use cross-validation in order to pick one of these models in the following way:

# Cross Validation Example Steps

1. Fix a model $M_i$. Fix $k < m$.
2. Fit the model $M_i$ to the data from the first $k$ years.
3. Using the fitted model, predict the data for the $(k+1)$st year.
4. Calculate the sum of squares of errors of prediction for the $(k+1)$st year.
5. Repeat these steps for $k = k_0, \ldots, m-1$ where $k_0$ is an arbitrary value of your choice.
6. Average the sum of squares of errors of prediction over $k = k_0, \ldots, m-1$. Denote this value by $CV_i$ and call it the Cross Validation score of model $M_i$.
7. Calculate $CV_i$ for each $i = 1, \ldots, \ell$ and choose the model with the smallest Cross-Validation score.

# Cross Validation Example Steps - Specific

For monthly stock data from 2001 to present, with your model and my model. The psuedo code for this:

```
for(M in Models){
  for(k in 2011:2018){
    fitted.model = model(M, data = 2001 to year (k-1))
    predictions = predict year k using fitted.model
    accuracy_k = sum(([year k data] - predictions)^2)
  }
  CV_M = sum(accuracy_k)/8
}
```

Then, choose the model with smallest CV_M

# To the code!

Now let's finish going through the lecture 9a code, i.e. the cross-validation section.