

INTRO to DATA SCIENCE

LECTURE 4: REGRESSION & REGULARIZATION

Rob Hall

DAT9 SF // August 14, 2014

LAST TIME:

- INTRO TO ML**
- KNN CLASSIFICATION**
- INTRO TO MATPLOTLIB FOR VISUALIZATION**

QUESTIONS?

AGENDA

I. LINEAR REGRESSION

II. POLYNOMIAL REGRESSION

III. REGULARIZATION

LAB:

**IV. IMPLEMENTING MULTIPLE REGRESSION & POLYNOMIAL
REGRESSION IN PYTHON**

I. LINEAR REGRESSION

	<i>continuous</i>	<i>categorical</i>
<i>supervised</i>	???	???
<i>unsupervised</i>	???	???

	<i>continuous</i>	<i>categorical</i>
<i>supervised</i>	<i>regression</i>	<i>classification</i>
<i>unsupervised</i>	<i>dimension reduction</i>	<i>clustering</i>

Q: What is a regression model?

*Q: What is a **regression** model?*

A: A functional relationship between input & response variables.

*Q: What is a **regression** model?*

A: A functional relationship between input & response variables

*The **simple linear regression** model captures a linear relationship between a single input variable x and a response variable y :*

*Q: What is a **regression** model?*

A: A functional relationship between input & response variables

*The **simple linear regression** model captures a linear relationship between a single input variable x and a response variable y :*

$$y = \alpha + \beta x + \varepsilon$$

Q: What do the terms in this model mean?

$$y = \alpha + \beta x + \varepsilon$$

Q: What do the terms in this model mean?

$$y = \alpha + \beta x + \varepsilon$$

*A: y = **response variable** (the one we want to predict)*

Q: What do the terms in this model mean?

$$y = \alpha + \beta x + \varepsilon$$

*A: y = **response variable** (the one we want to predict)*

*x = **input variable** (the one we use to train the model)*

Q: What do the terms in this model mean?

$$y = \alpha + \beta x + \varepsilon$$

*A: y = **response variable** (the one we want to predict)*

*x = **input variable** (the one we use to train the model)*

*α = **intercept** (where the line crosses the y -axis)*

Q: What do the terms in this model mean?

$$y = \alpha + \beta x + \varepsilon$$

*A: y = **response variable** (the one we want to predict)*

*x = **input variable** (the one we use to train the model)*

*α = **intercept** (where the line crosses the y -axis)*

*β = **regression coefficient** (the model “parameter”)*

Q: What do the terms in this model mean?

$$y = \alpha + \beta x + \varepsilon$$

*A: y = **response variable** (the one we want to predict)*

*x = **input variable** (the one we use to train the model)*

*α = **intercept** (where the line crosses the y -axis)*

*β = **regression coefficient** (the model “parameter”)*

*ε = **residual** (the prediction error)*

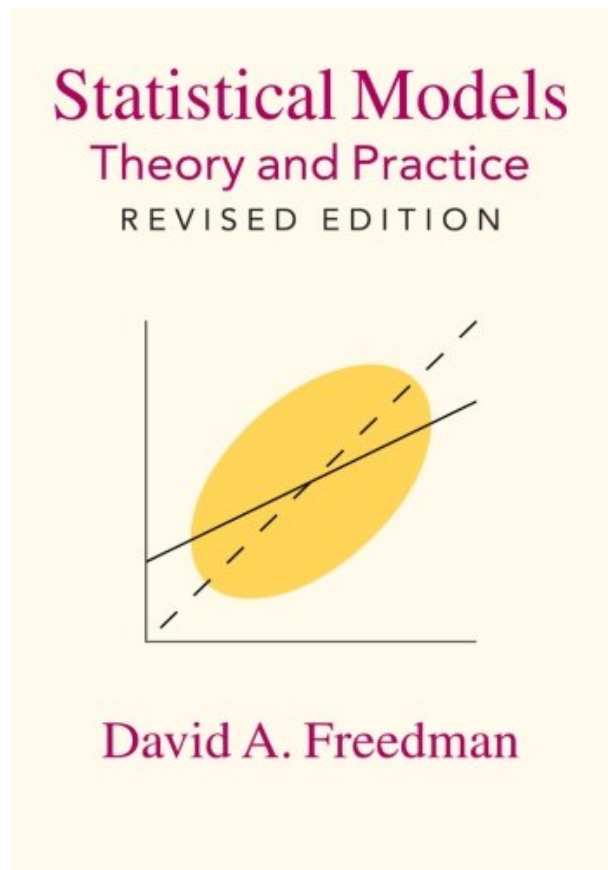
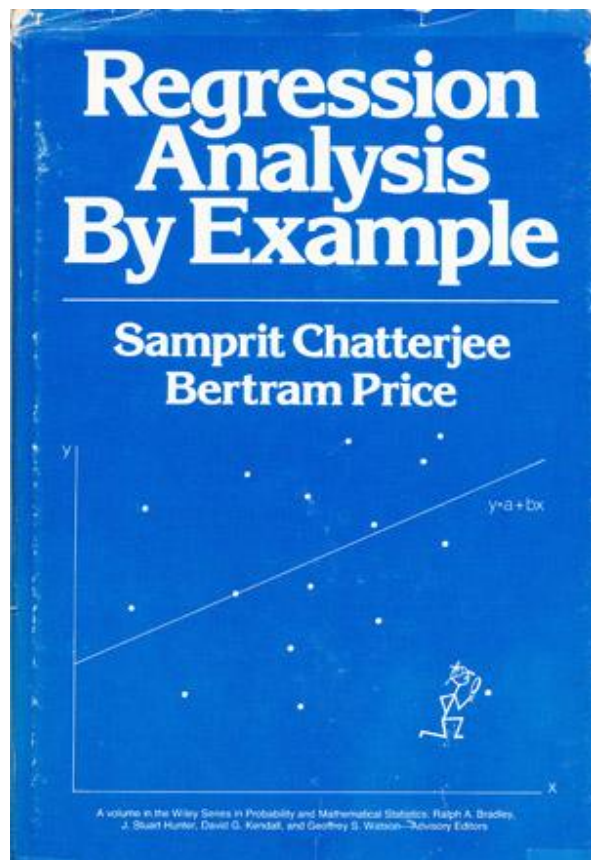
We can extend this model to several input variables, giving us the
multiple linear regression model:

We can extend this model to several input variables, giving us the
multiple linear regression model:

$$y = \alpha + \beta_1 X_1 + \dots + \beta_n X_n + \varepsilon$$

Linear regression involves several technical assumptions and is often presented with lots of mathematical formality.

The math is not very important for our purposes, but you should check it out if you get serious about solving regression problems.



Q: How do we fit a regression model to a dataset?

Q: How do we fit a regression model to a dataset?

A: In theory, minimize the sum of the squared residuals (OLS).

Q: How do we fit a regression model to a dataset?

A: In theory, minimize the sum of the squared residuals (OLS).

In practice, any respectable piece of software will do this for you.

Q: How do we fit a regression model to a dataset?

A: In theory, minimize the sum of the squared residuals (OLS).

In practice, any respectable piece of software will do this for you.

But again, if you get serious about regression, you should learn how this works!

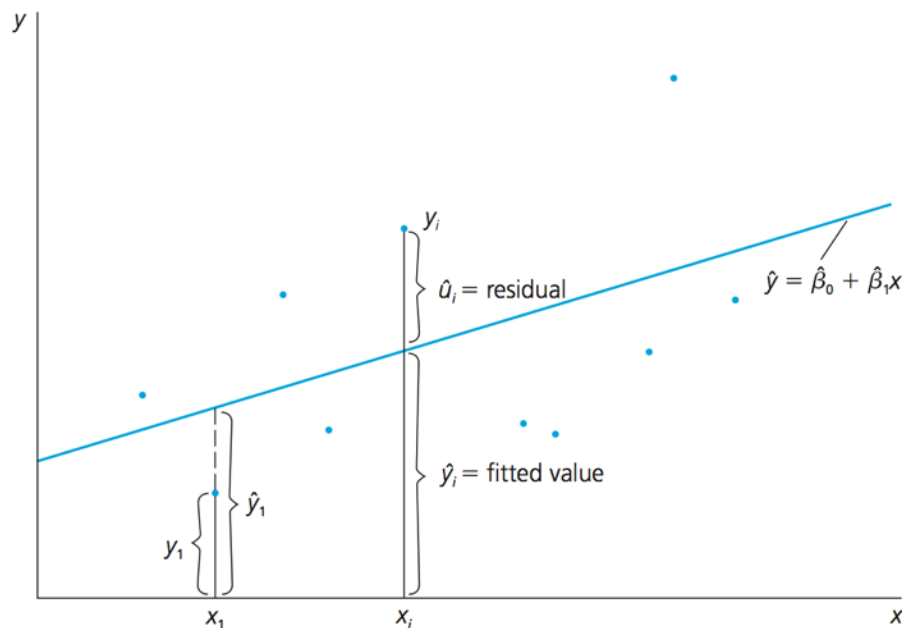
Q: How do we fit a regression model to a dataset?

A: In theory, minimize the sum of the squared residuals (OLS).

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i.$$

$$\hat{u}_i = y_i - \hat{y}_i = y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i.$$

$$\sum_{i=1}^n \hat{u}_i^2 = \sum_{i=1}^n (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2,$$



II: POLYNOMIAL REGRESSION

*Consider the following **polynomial regression model**:*

$$y = \alpha + \beta_1 x + \beta_2 x^2 + \varepsilon$$

*Consider the following **polynomial regression model**:*

$$y = \alpha + \beta_1 x + \beta_2 x^2 + \varepsilon$$

Q: This represents a nonlinear relationship. Is it still a linear model?

*Consider the following **polynomial regression model**:*

$$y = \alpha + \beta_1 x + \beta_2 x^2 + \varepsilon$$

Q: This represents a nonlinear relationship. Is it still a linear model?

A: Yes, because it's linear in the β 's!

*Consider the following **polynomial regression model**:*

$$y = \alpha + \beta_1 x + \beta_2 x^2 + \varepsilon$$

Q: This represents a nonlinear relationship. Is it still a linear model?

A: Yes, because it's linear in the β 's!

“Although polynomial regression fits a nonlinear model to the data, as a statistical estimation problem it is linear, in the sense that the regression function $E(y|x)$ is linear in the unknown parameters that are estimated from the data. For this reason, polynomial regression is considered to be a special case of multiple linear regression.” -- Wikipedia

Polynomial regression allows us to fit very complex curves to data.

$$y = \alpha + \beta_1 x + \beta_2 x^2 + \dots + \beta_n x^n + \varepsilon$$

Polynomial regression allows us to fit very complex curves to data.

$$y = \alpha + \beta_1 x + \beta_2 x^2 + \dots + \beta_n x^n + \varepsilon$$

But there is one problem with the model we've written down so far.

Polynomial regression allows us to fit very complex curves to data.

$$y = \alpha + \beta_1 x + \beta_2 x^2 + \dots + \beta_n x^n + \varepsilon$$

But there is one problem with the model we've written down so far.

Q: Does anyone know what it is?

Polynomial regression allows us to fit very complex curves to data.

$$y = \alpha + \beta_1 x + \beta_2 x^2 + \dots + \beta_n x^n + \varepsilon$$

But there is one problem with the model we've written down so far.

Q: Does anyone know what it is?

A: This model violates one of the assumptions of linear regression!



This model displays multicollinearity, which means the predictor variables are highly correlated with each other.

$$y = \alpha + \beta_1 x + \beta_2 x^2 + \dots + \beta_n x^n + \varepsilon$$

```
> x <- seq(1, 10, 0.1)
> cor(x^9, x^10)
[1] 0.9987608
```

*This model displays **multicollinearity**, which means the predictor variables are highly correlated with each other.*

$$y = \alpha + \beta_1 x + \beta_2 x^2 + \dots + \beta_n x^n + \varepsilon$$

Multicollinearity causes the linear regression model to break down, because it can't tell the predictor variables apart.

Q: What can we do about this?

Q: What can we do about this?

A: Replace the correlated predictors with uncorrelated predictors.

Q: What can we do about this?

A: Replace the correlated predictors with uncorrelated predictors.

$$y = \alpha + \beta_1 f_1(x) + \beta_2 f_2(x^2) + \dots + \beta_n f_n(x^n) + \varepsilon$$

Q: What can we do about this?

A: Replace the correlated predictors with uncorrelated predictors.

$$y = \alpha + \beta_1 f_1(x) + \beta_2 f_2(x^2) + \dots + \beta_n f_n(x^n) + \varepsilon$$

OPTIONAL NOTE

These polynomial functions form an *orthogonal basis* of the function space.

So far, we've seen how polynomial regression allows us to fit complex nonlinear relationships, and even to avoid multicollinearity (by using basis functions).

So far, we've seen how polynomial regression allows us to fit complex nonlinear relationships, and even to avoid multicollinearity (by using basis functions).

Q: Can a regression model be too complex?

III: REGULARIZATION

*Recall our earlier discussion of **overfitting**.*

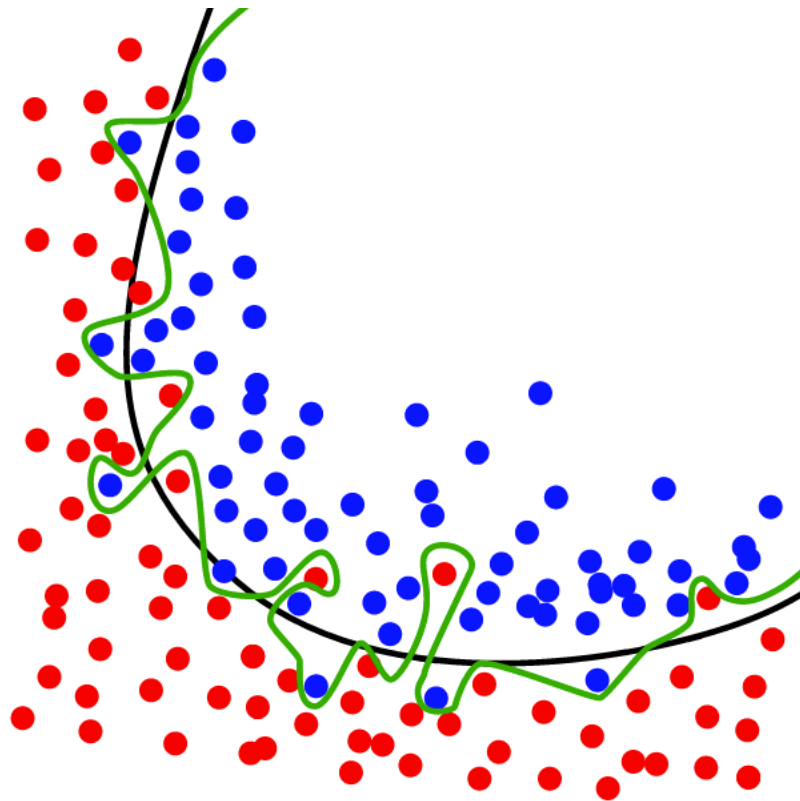
*Recall our earlier discussion of **overfitting**.*

When we talked about this in the context of classification, we said that it was a result of matching the training set too closely.

*Recall our earlier discussion of **overfitting**.*

When we talked about this in the context of classification, we said that it was a result of matching the training set too closely.

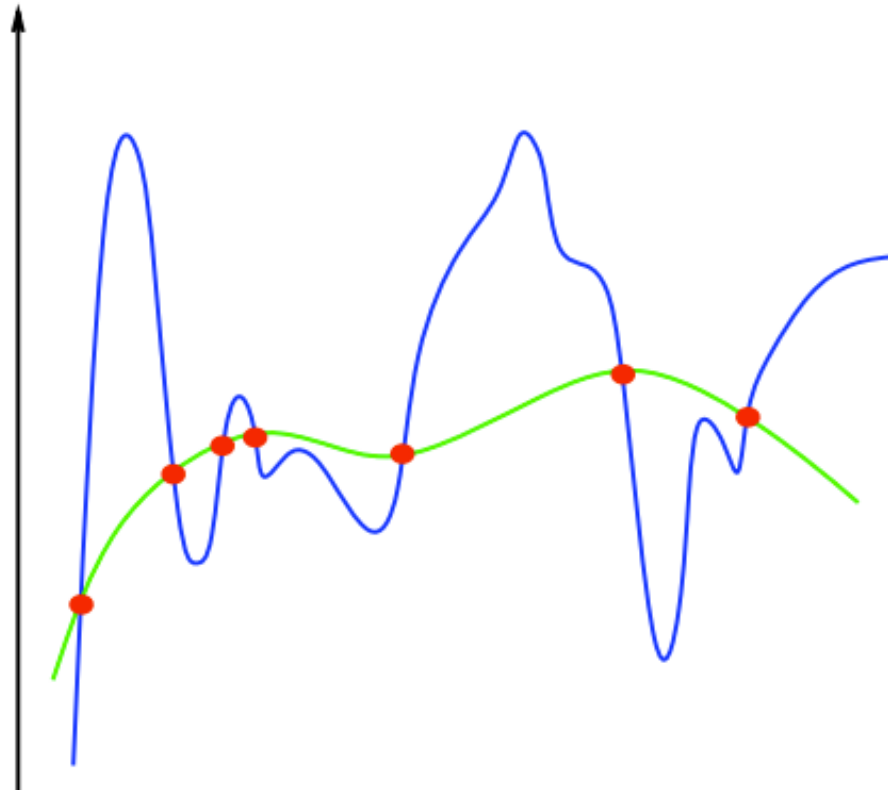
*In other words, an overfit model matches the **noise** in the dataset instead of the **signal**.*



The same thing can happen in regression.

It's possible to design a regression model that matches the noise in the data instead of the signal.

This happens when our model becomes too complex for the data to support.



*Q: How do we define the **complexity** of a regression model?*

*Q: How do we define the **complexity** of a regression model?*

A: One method is to define complexity as a function of the size of the coefficients.

*Q: How do we define the **complexity** of a regression model?*

A: One method is to define complexity as a function of the size of the coefficients.

Ex 1: $\sum |\beta_i|$

Ex 2: $\sum \beta_i^2$

*Q: How do we define the **complexity** of a regression model?*

A: One method is to define complexity as a function of the size of the coefficients.

*Ex 1: $\sum |\beta_i|$ this is called the **L1-norm***

*Ex 2: $\sum \beta_i^2$ this is called the **L2-norm***

*These measures of complexity lead to the following **regularization techniques**:*

*These measures of complexity lead to the following **regularization techniques**:*

L1 regularization: $y = \sum \beta_j x_j + \varepsilon \quad \text{st.} \quad \sum |\beta_j| < s$

*These measures of complexity lead to the following **regularization techniques**:*

L1 regularization: $y = \sum \beta_i x_i + \varepsilon \quad \text{st.} \quad \sum |\beta_i| < s$

L2 regularization: $y = \sum \beta_i x_i + \varepsilon \quad \text{st.} \quad \sum \beta_i^2 < s$

*These measures of complexity lead to the following **regularization techniques**:*

L1 regularization: $y = \sum \beta_i x_i + \varepsilon \quad \text{st.} \quad \sum |\beta_i| < s$

L2 regularization: $y = \sum \beta_i x_i + \varepsilon \quad \text{st.} \quad \sum \beta_i^2 < s$

Regularization *refers to the method of preventing overfitting by explicitly controlling model complexity.*

These regularization problems can also be expressed as:

L1 regularization: $\min(\|y - x\beta\|^2 + \lambda\|\beta\|)$

L2 regularization: $\min(\|y - x\beta\|^2 + \lambda\|\beta\|^2)$

These regularization problems can also be expressed as:

L1 regularization (Lasso): $\min(\|y - x\beta\|^2 + \lambda\|x\|)$

L2 regularization (Ridge): $\min(\|y - x\beta\|^2 + \lambda\|x\|^2)$

This (Lagrangian) formulation reflects the fact that there is a cost associated with regularization.

Q: What are bias and variance?

Q: What are bias and variance?

A: Bias refers to predictions that are systematically inaccurate.

Q: What are bias and variance?

*A: Bias refers to predictions that are systematically inaccurate.
Variance refers to predictions that are generally inaccurate.*

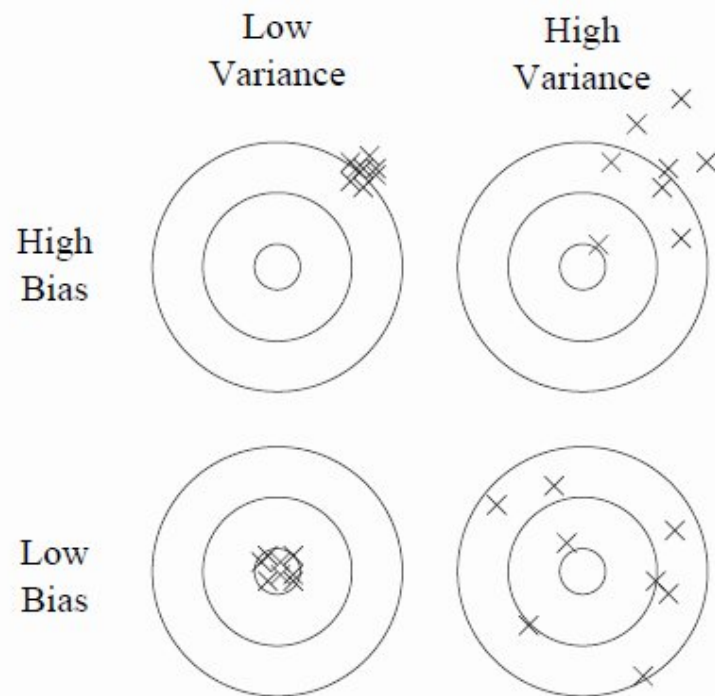


Figure 1: Bias and variance in dart-throwing.

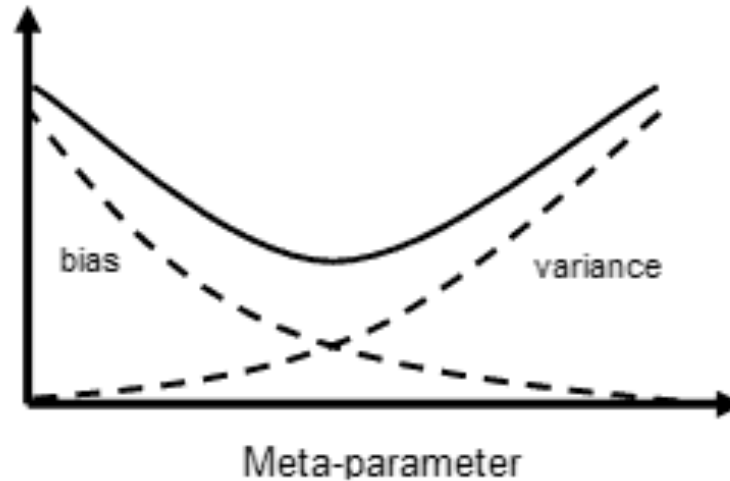
Q: What are bias and variance?

A: Bias refers to predictions that are systematically inaccurate.

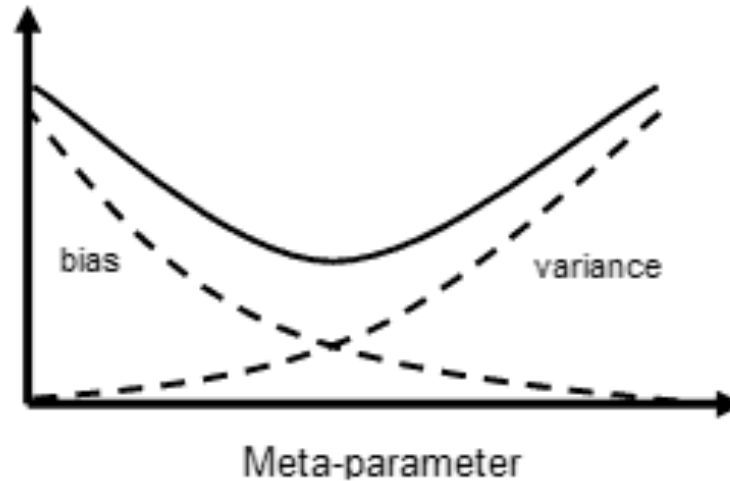
Variance refers to predictions that are generally inaccurate.

It turns out (after some math) that the generalization error in our model can be decomposed into a bias component and variance component.

*This is another example of the **bias-variance tradeoff**.*



*This is another example of the **bias-variance tradeoff**.*

**NOTE**

The “meta-parameter” here is the λ we saw above.

A more typical term is “hyperparameter”.

*This tradeoff is regulated by a **hyperparameter** λ , which we've already seen:*

L1 regularization: $y = \sum \beta_i x_i + \varepsilon \quad \text{st.} \quad \sum |\beta_i| < \lambda$

L2 regularization: $y = \sum \beta_i x_i + \varepsilon \quad \text{st.} \quad \sum \beta_i^2 < \lambda$

So regularization represents a method to trade away some variance for a little bias in our model, thus achieving a better overall fit.

LAB: POLYNOMIAL REGRESSION & REGULARIZATION