

Regularization, Big Data and Text Analysis

John Myles White

June 22, 2011

Data, data, data! I can't make bricks without clay!

According to reputable sources, we're in the Age of Big Data

But what makes data big?

A 4x4 matrix diagram with columns labeled N and D. The matrix is defined by a vertical line labeled N and a horizontal line labeled D.

11	12	91	87
19	21	17	92
84	71	29	35
73	67	53	48

What happens as $N \rightarrow \infty$?

Traditionally, good things:

- ▶ Law of Large Numbers
- ▶ Consistent estimators
- ▶ Asymptotic guarantees

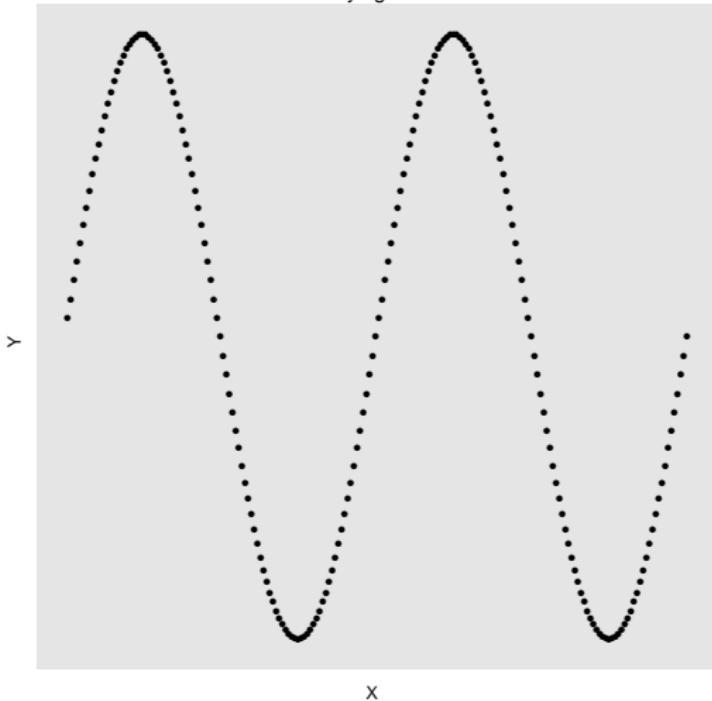
What happens as $D \rightarrow \infty$?

Traditionally, bad things:

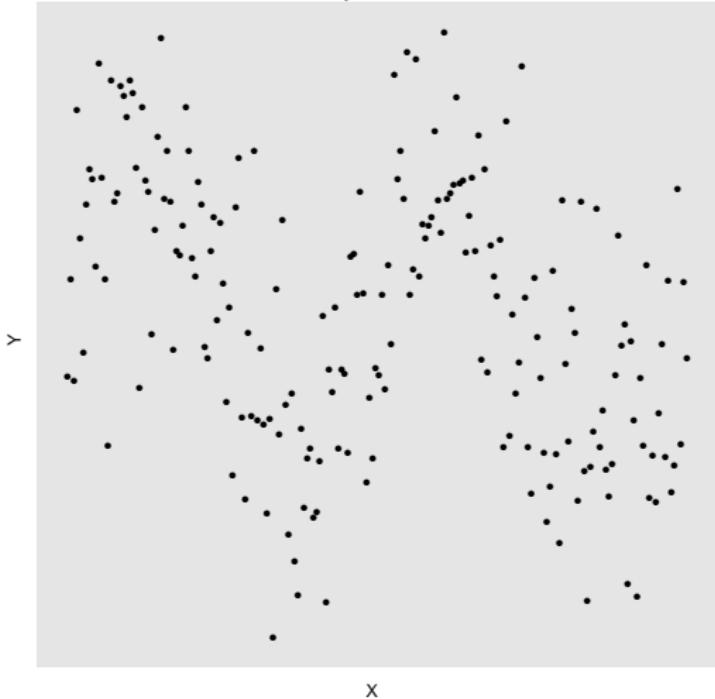
- ▶ Underdetermined systems
- ▶ Many different parameter values fit data equally well
- ▶ Massive overfitting

What leads to overfitting?

Underlying Pattern



Noisy Data Set

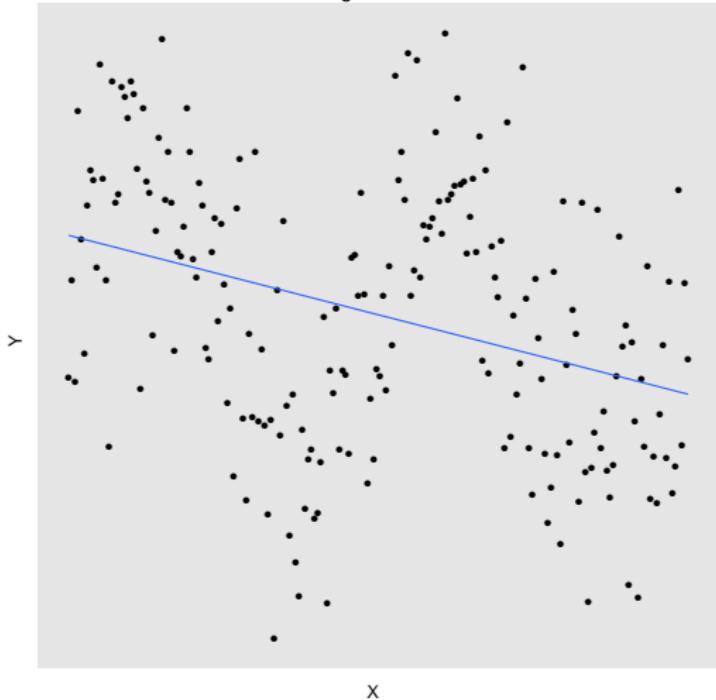


How can we fit a model to this data?

Start with linear regression:

- ▶ $Y = \beta_0 + \beta_1 X$

Linear Regression Results

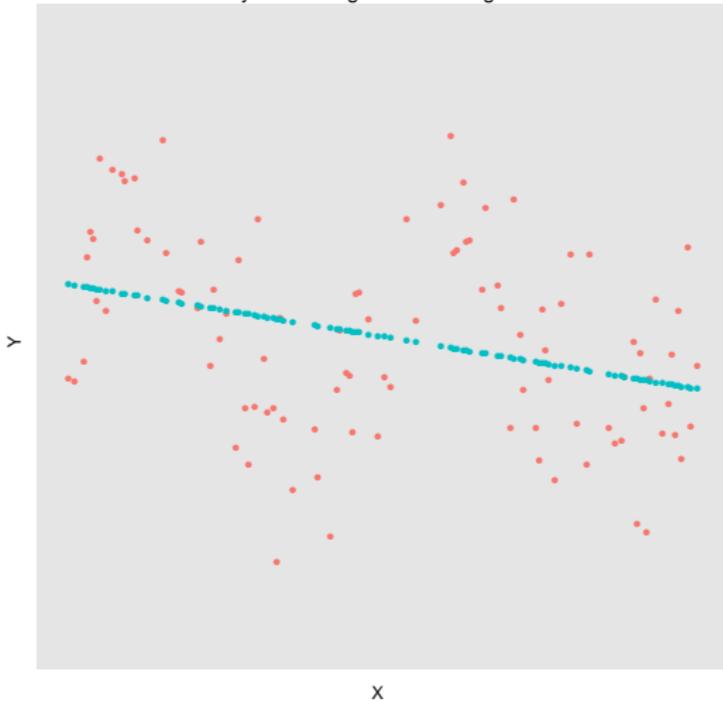


Linear regression isn't very expressive. We need a stronger model

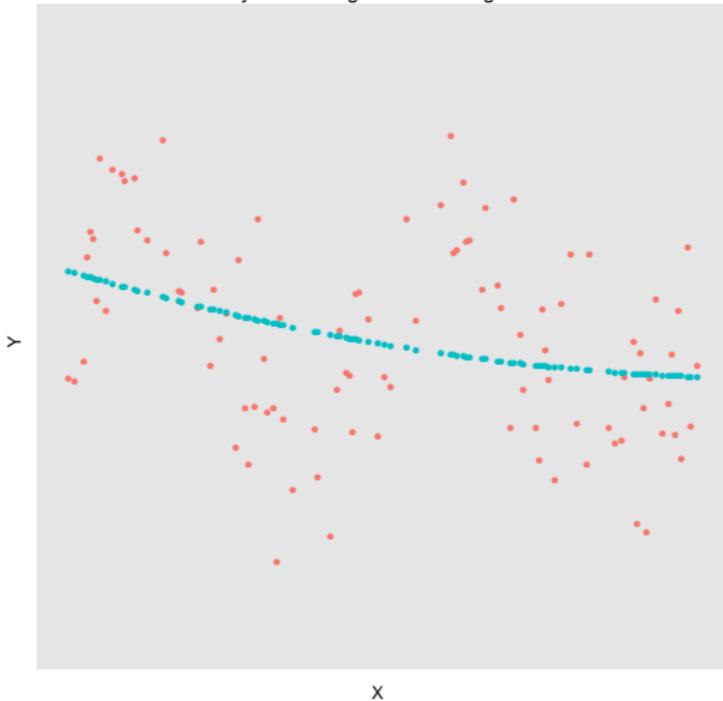
Polynomial regressions:

- ▶ $Y = \beta_0 + \beta_1 X + \beta_2 X^2$
- ▶ $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3$
- ▶ ...
- ▶ $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \dots + \beta_{20} X^{20}$

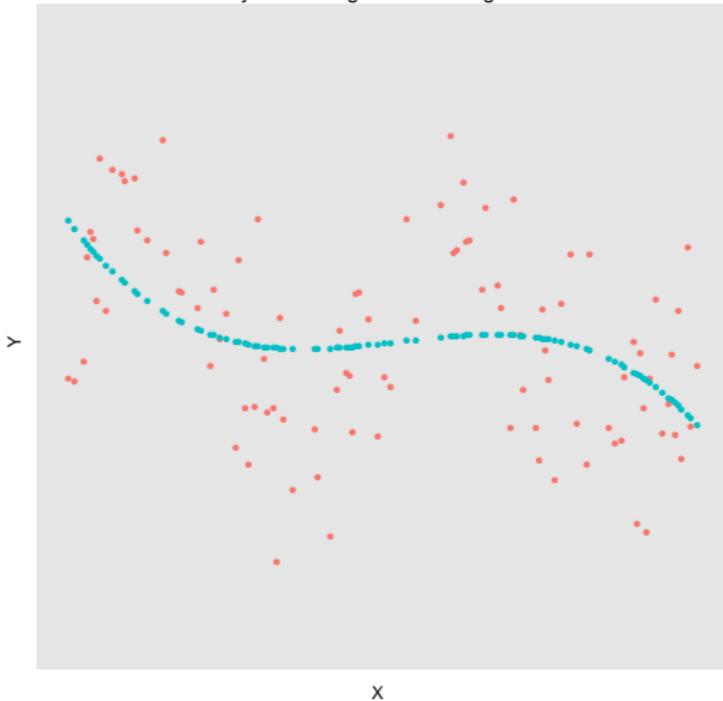
Polynomial Regression of Degree 1



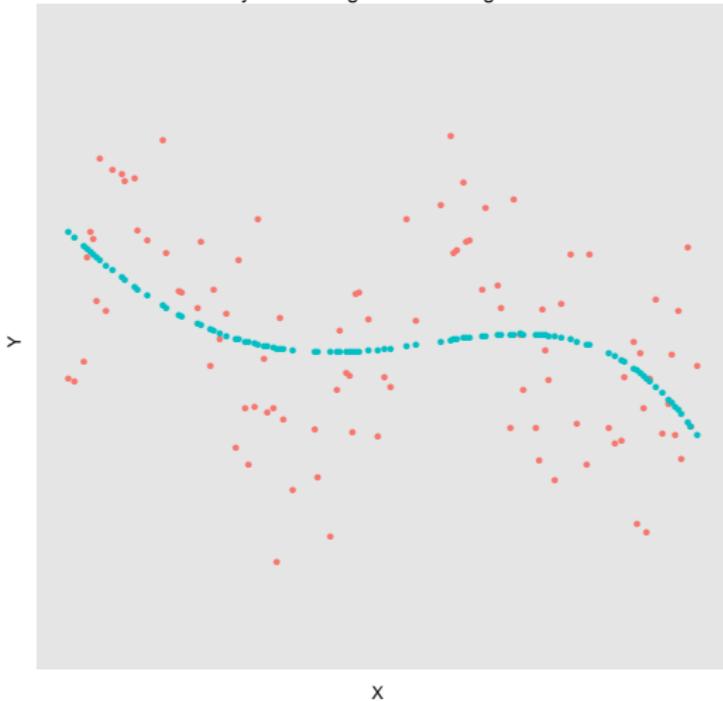
Polynomial Regression of Degree 2



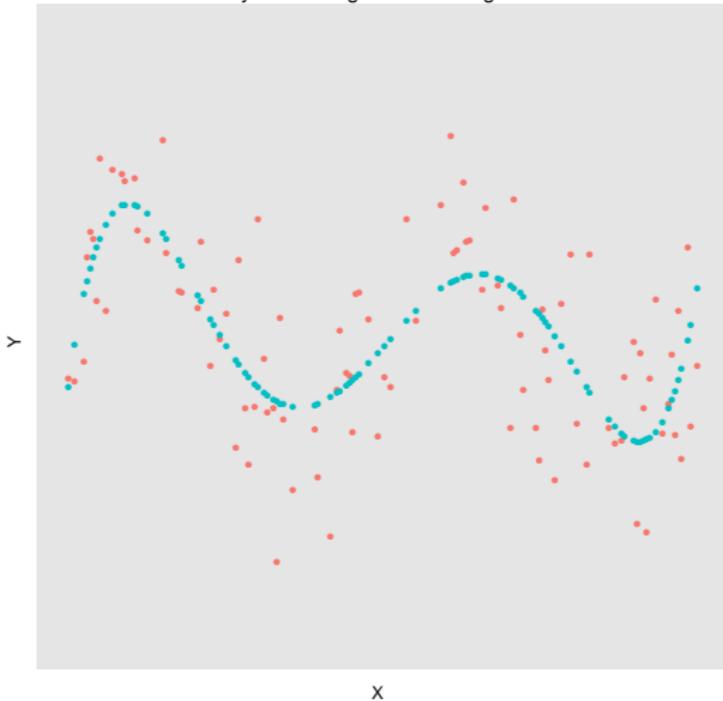
Polynomial Regression of Degree 3



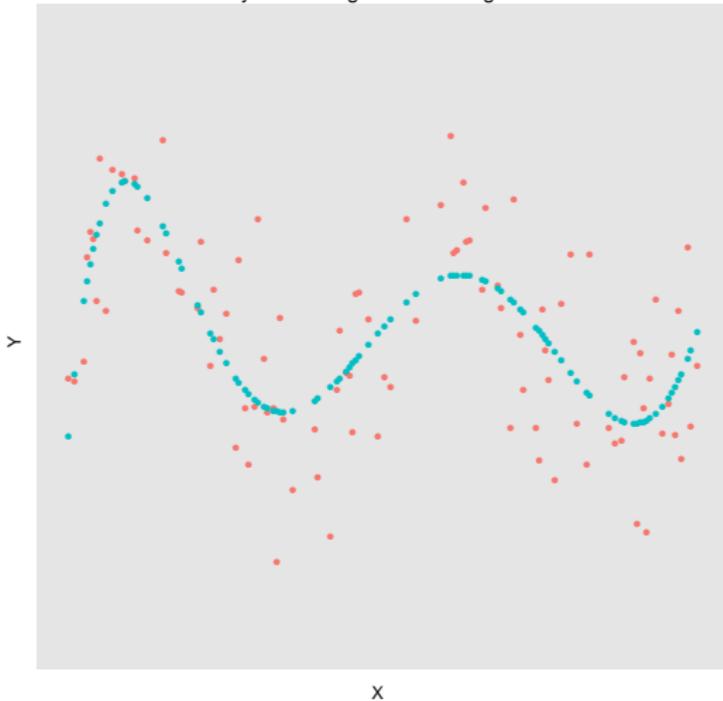
Polynomial Regression of Degree 4



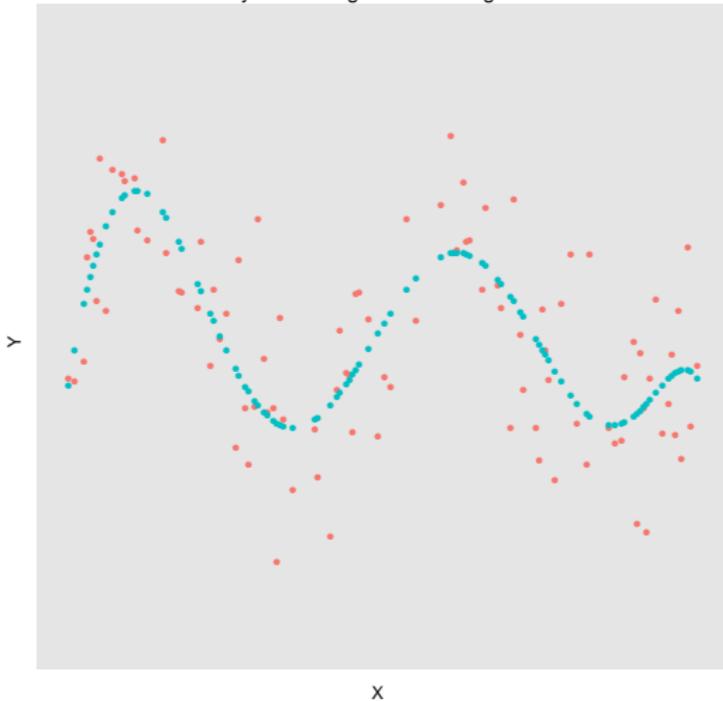
Polynomial Regression of Degree 5



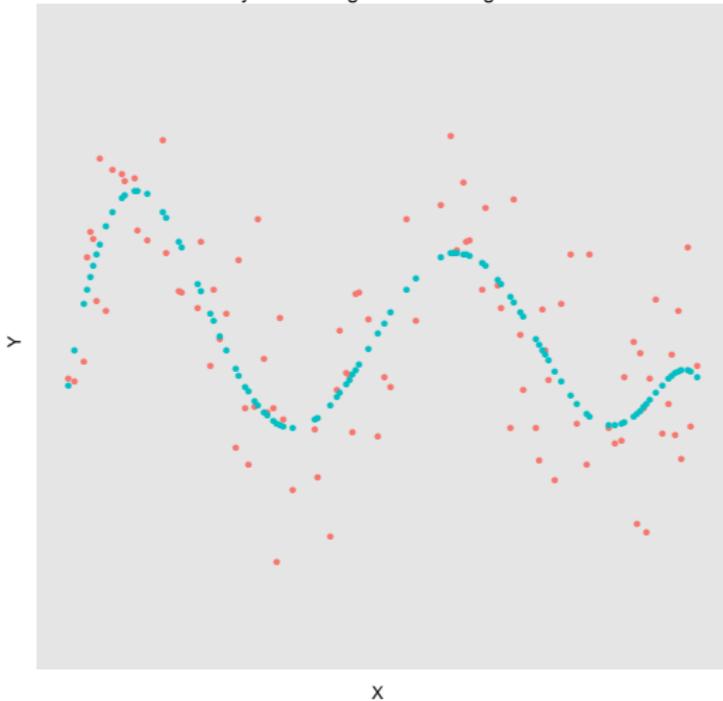
Polynomial Regression of Degree 6



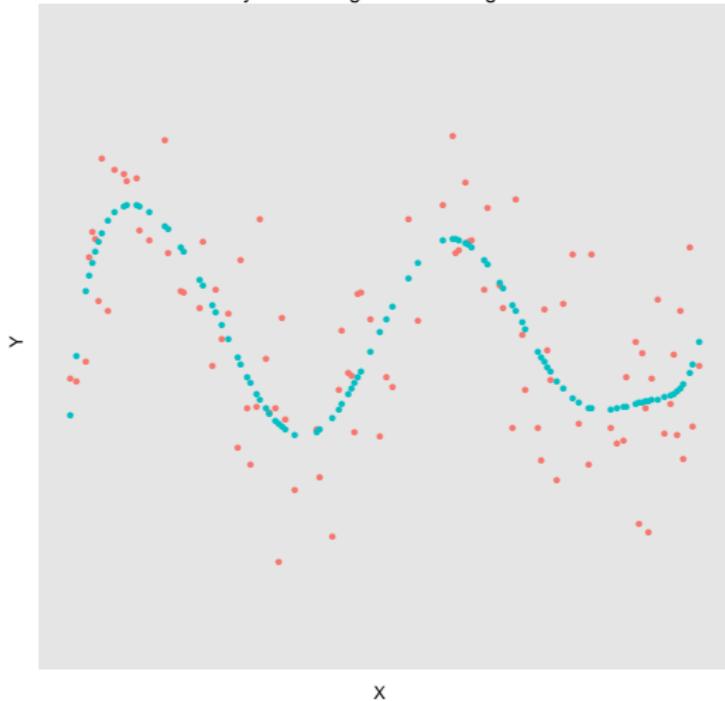
Polynomial Regression of Degree 7



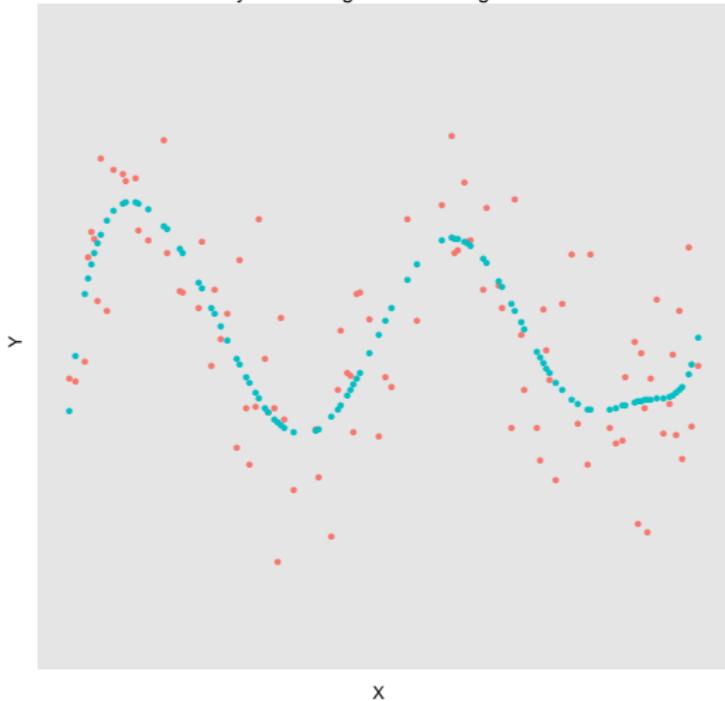
Polynomial Regression of Degree 8



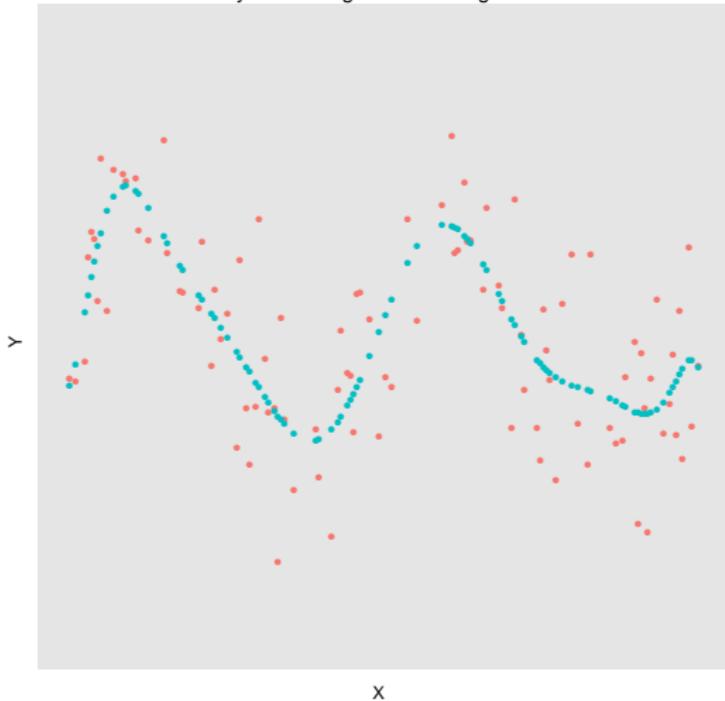
Polynomial Regression of Degree 9



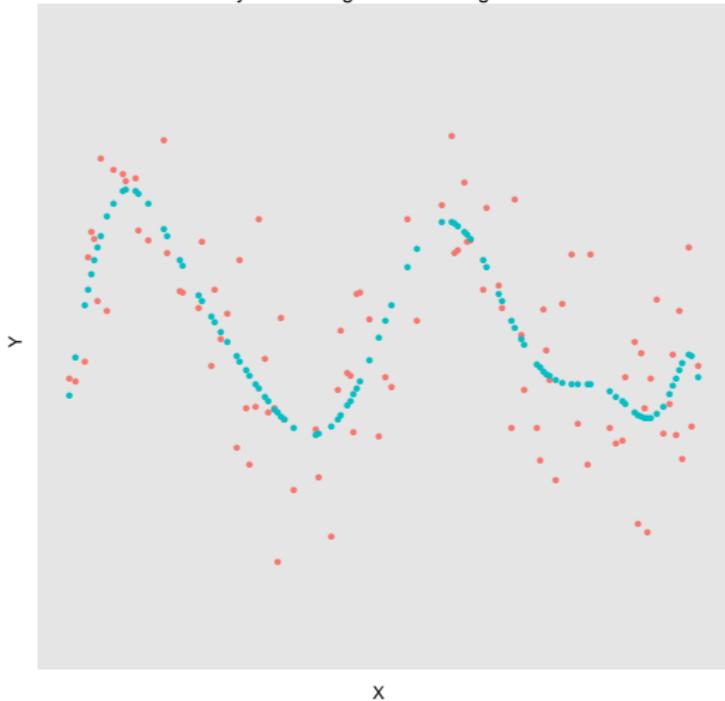
Polynomial Regression of Degree 10



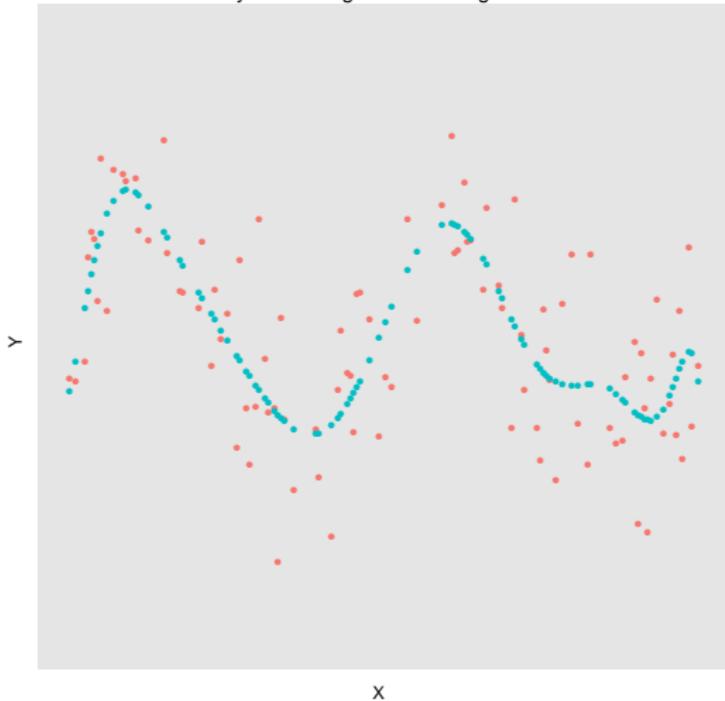
Polynomial Regression of Degree 11



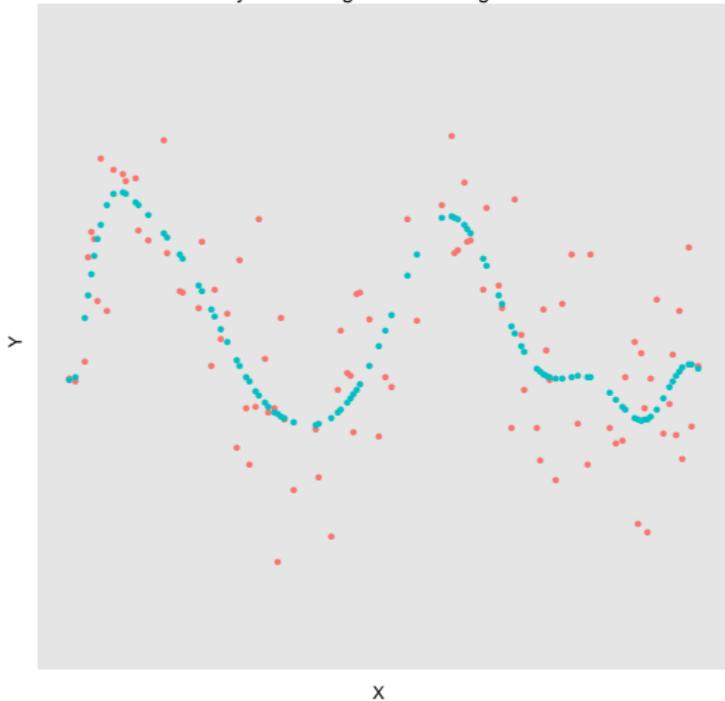
Polynomial Regression of Degree 12



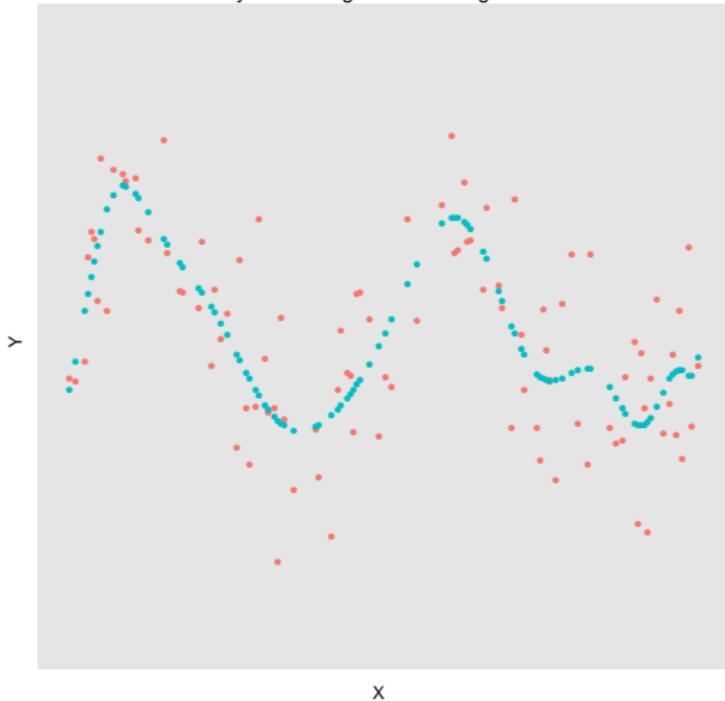
Polynomial Regression of Degree 13



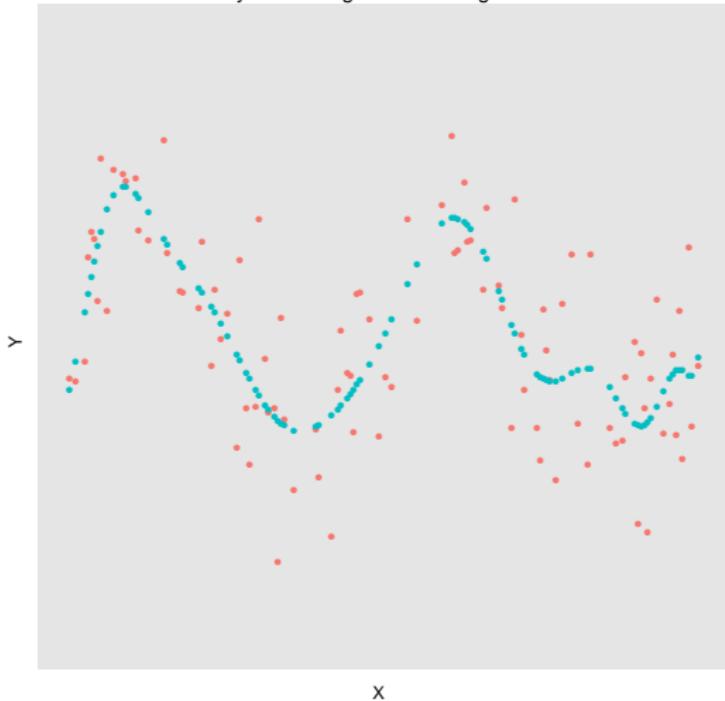
Polynomial Regression of Degree 14



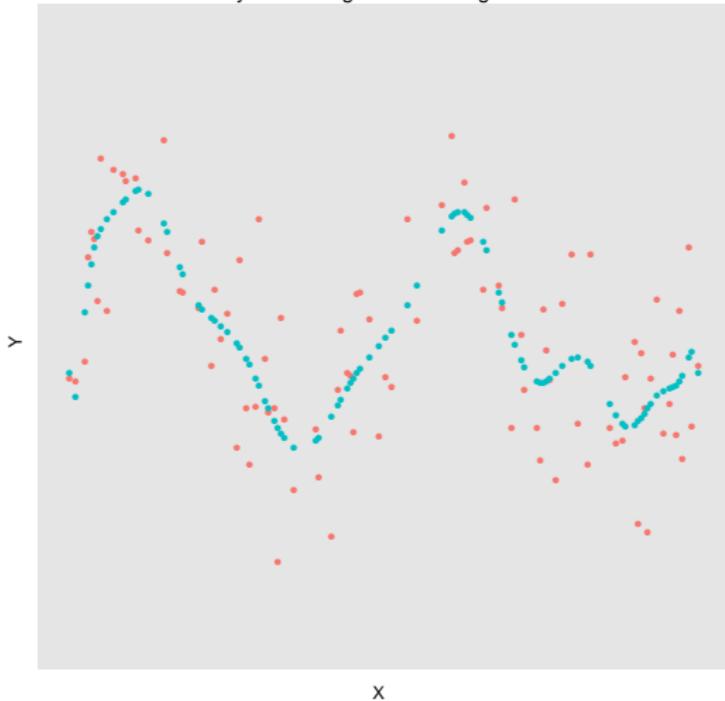
Polynomial Regression of Degree 15



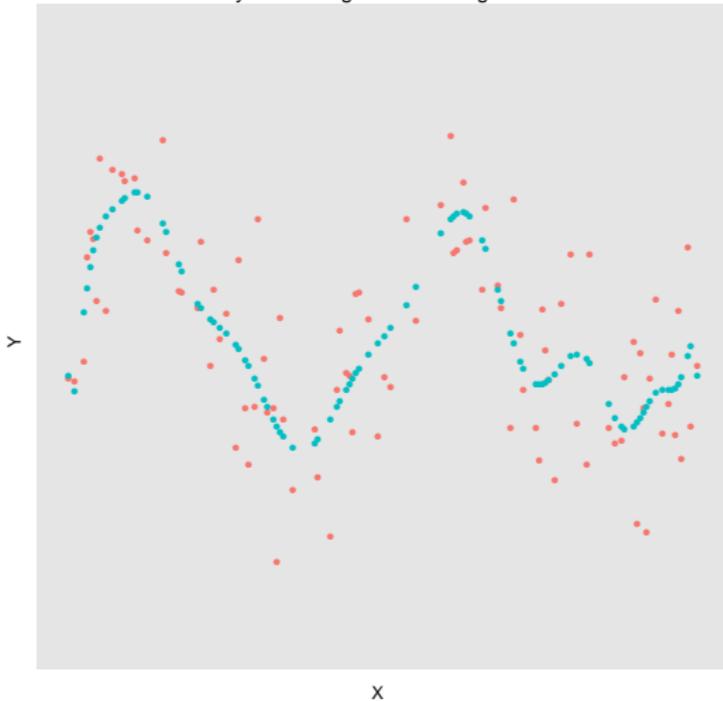
Polynomial Regression of Degree 16



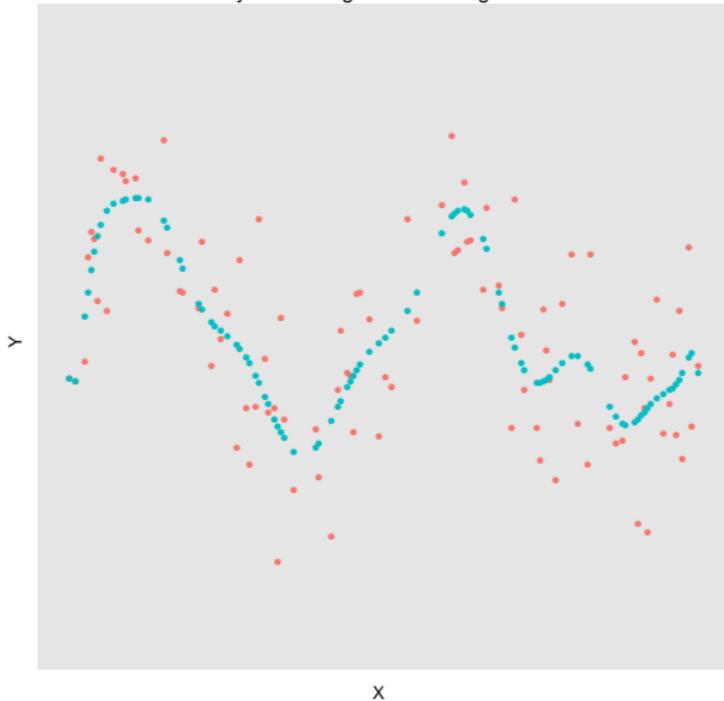
Polynomial Regression of Degree 17



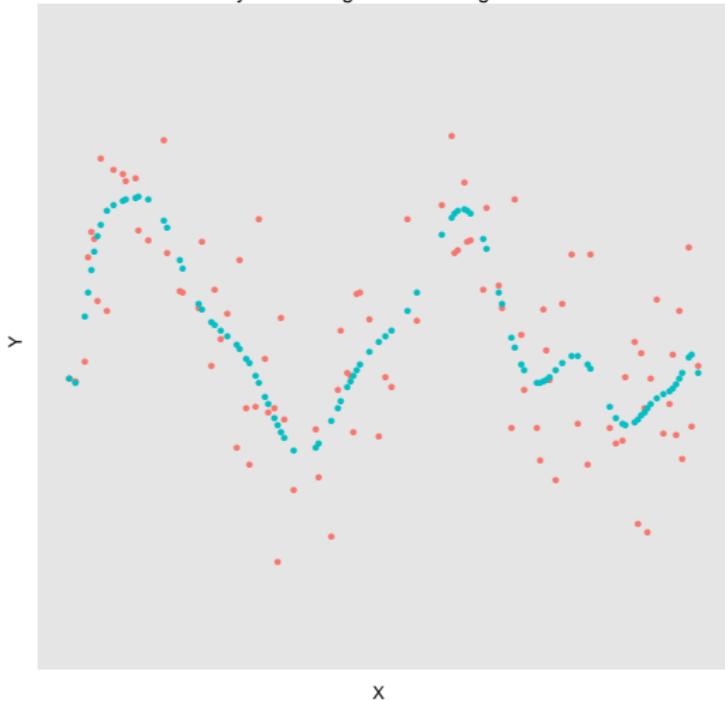
Polynomial Regression of Degree 18



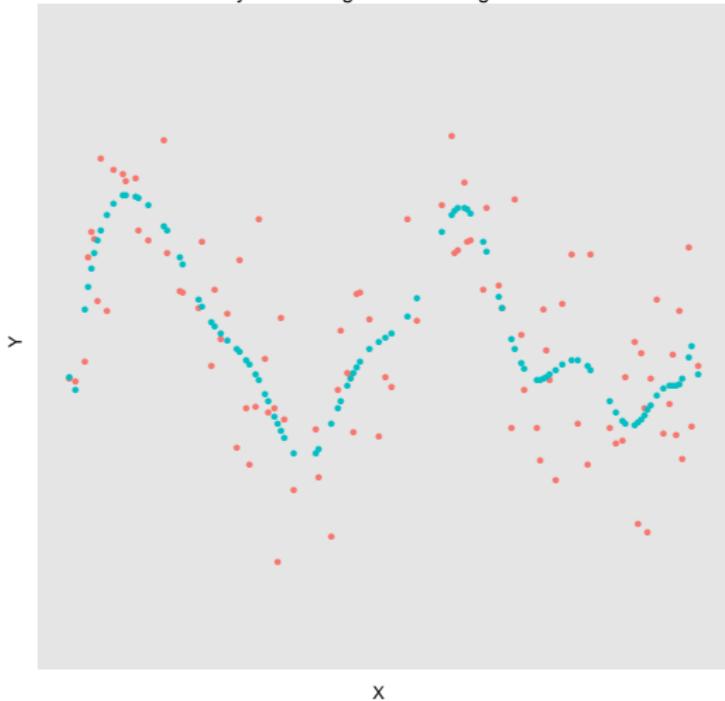
Polynomial Regression of Degree 19



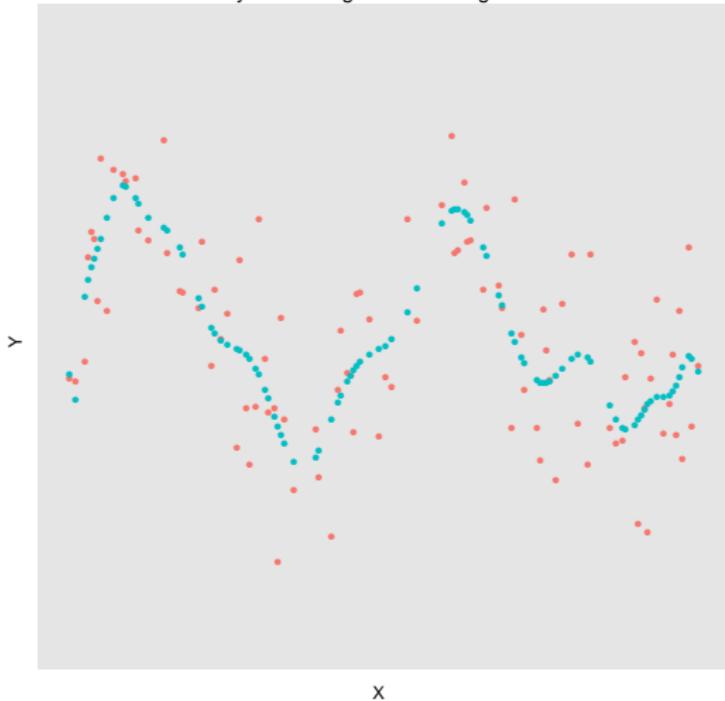
Polynomial Regression of Degree 20



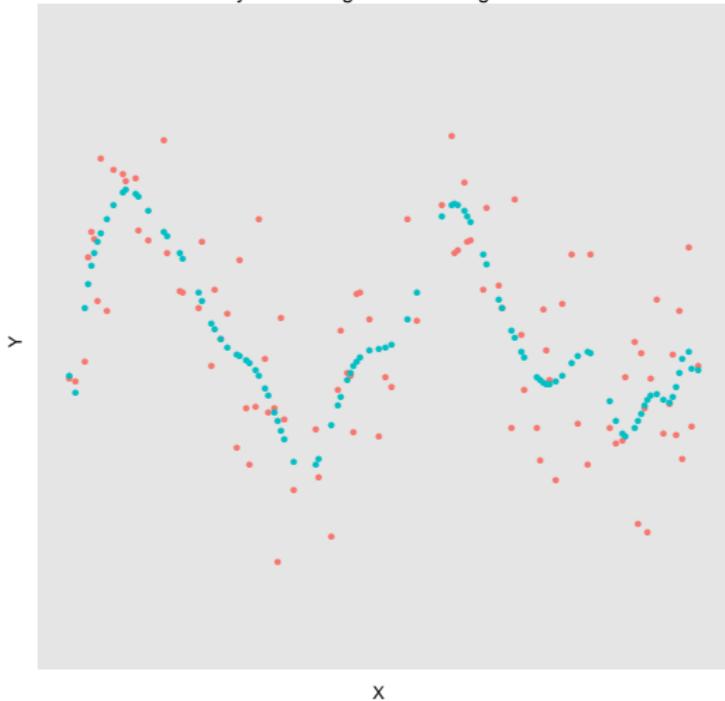
Polynomial Regression of Degree 21



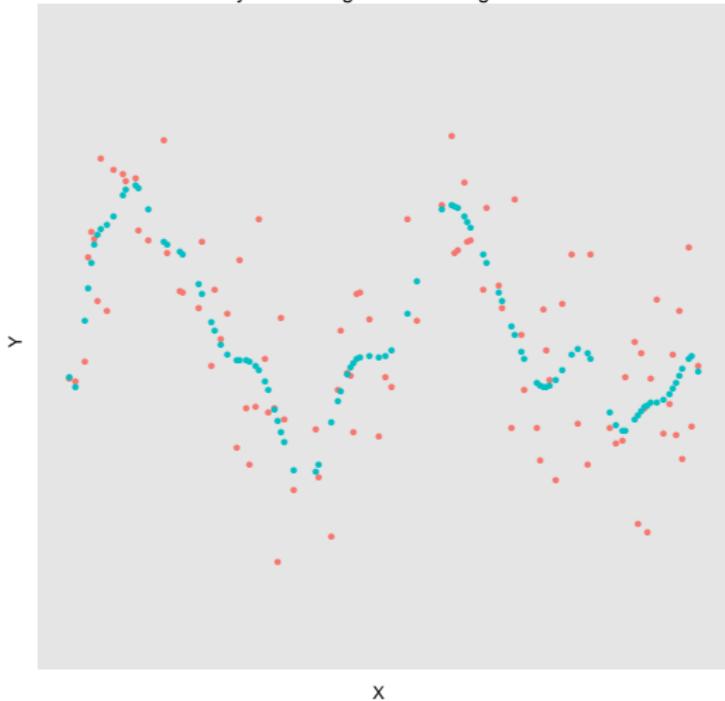
Polynomial Regression of Degree 22



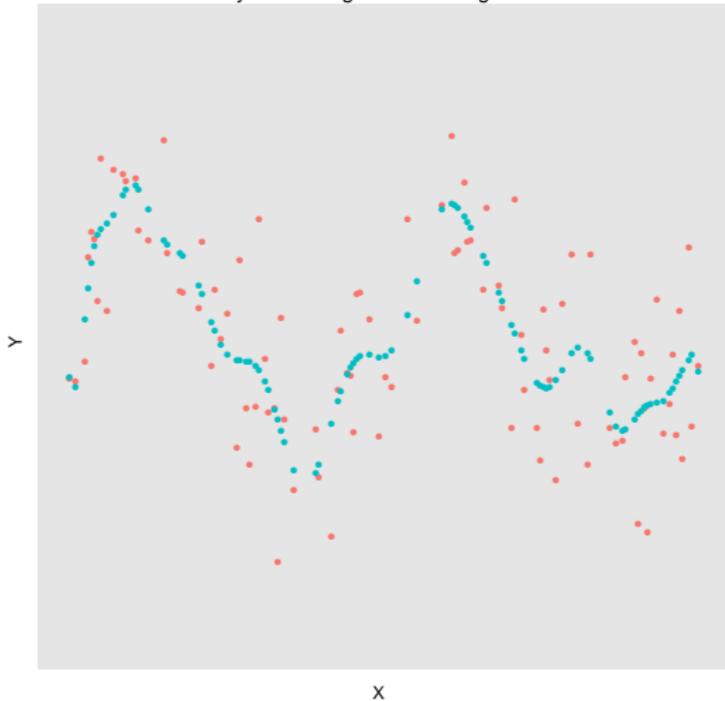
Polynomial Regression of Degree 23



Polynomial Regression of Degree 24



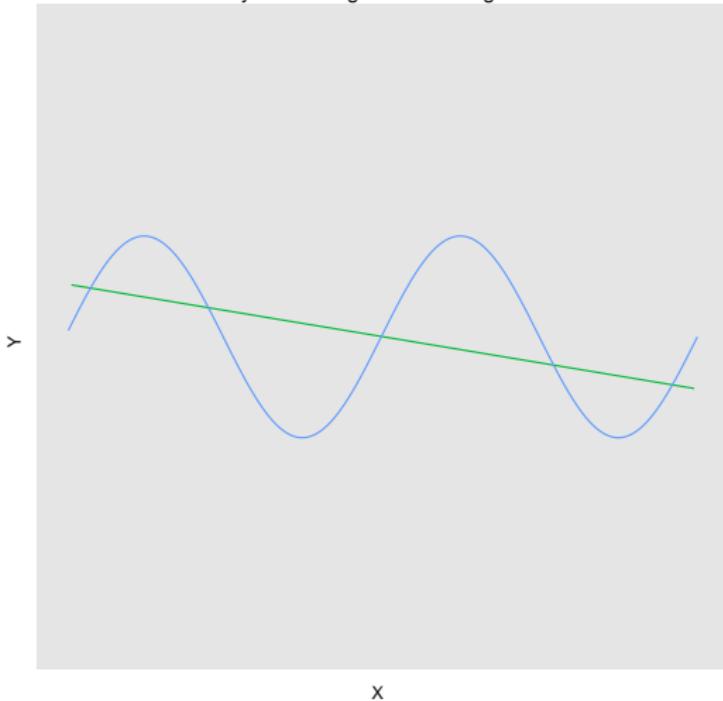
Polynomial Regression of Degree 25



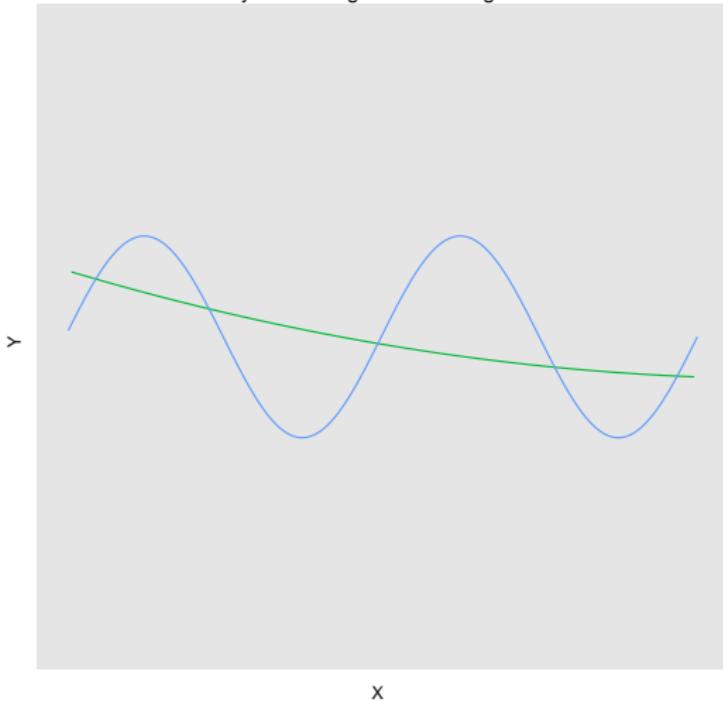
As $D \rightarrow N$, we fit the data better and better

But our model gets further and further from the true pattern...

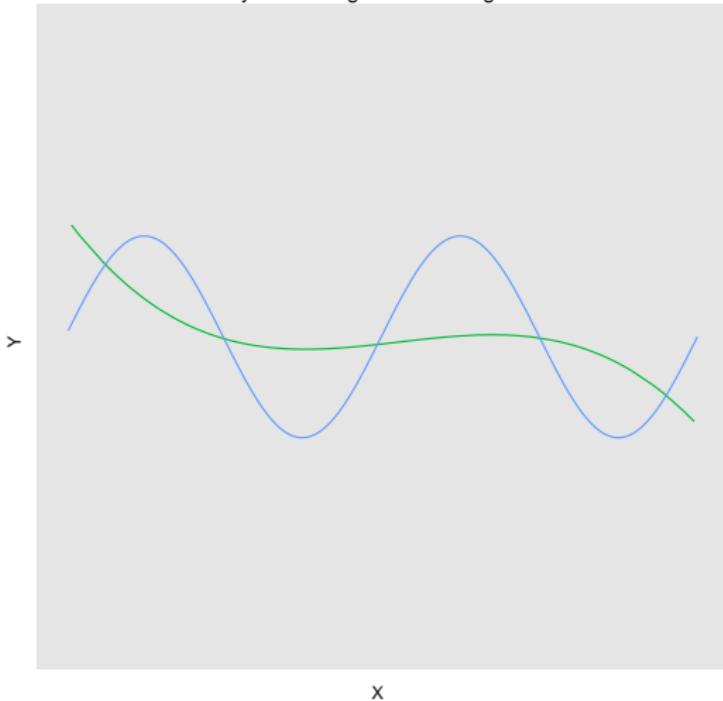
Polynomial Regression of Degree 1



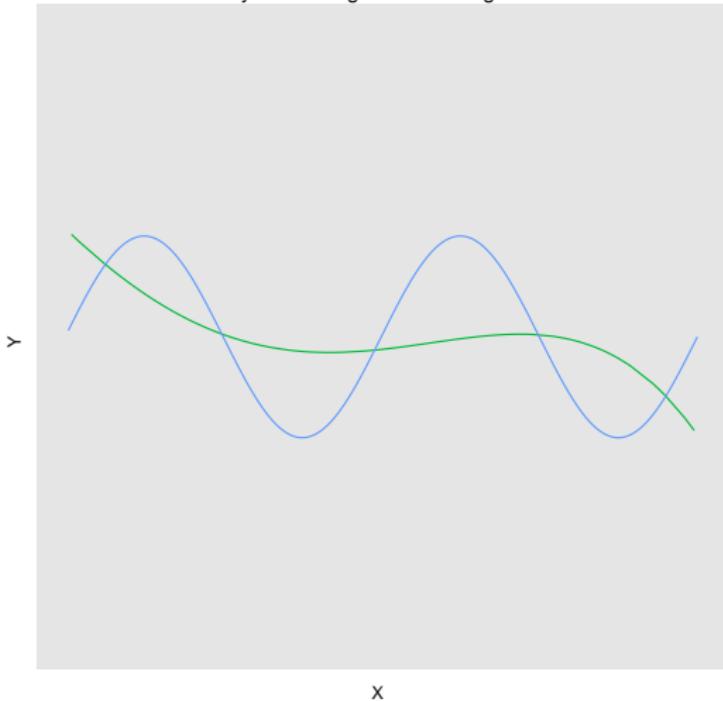
Polynomial Regression of Degree 2



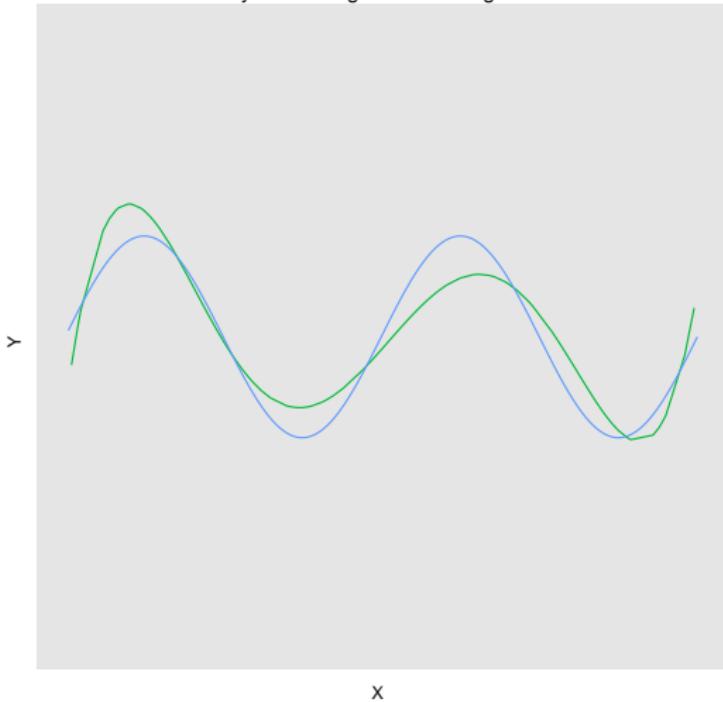
Polynomial Regression of Degree 3



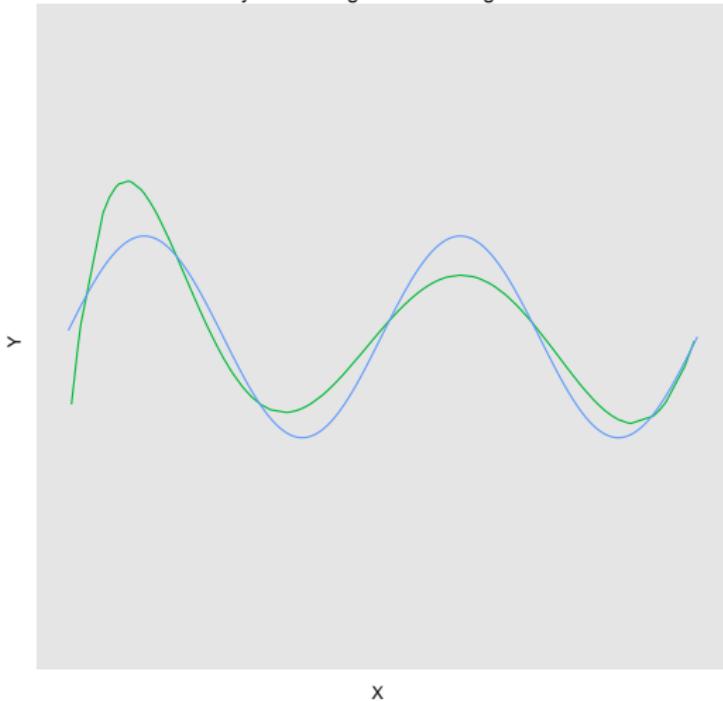
Polynomial Regression of Degree 4



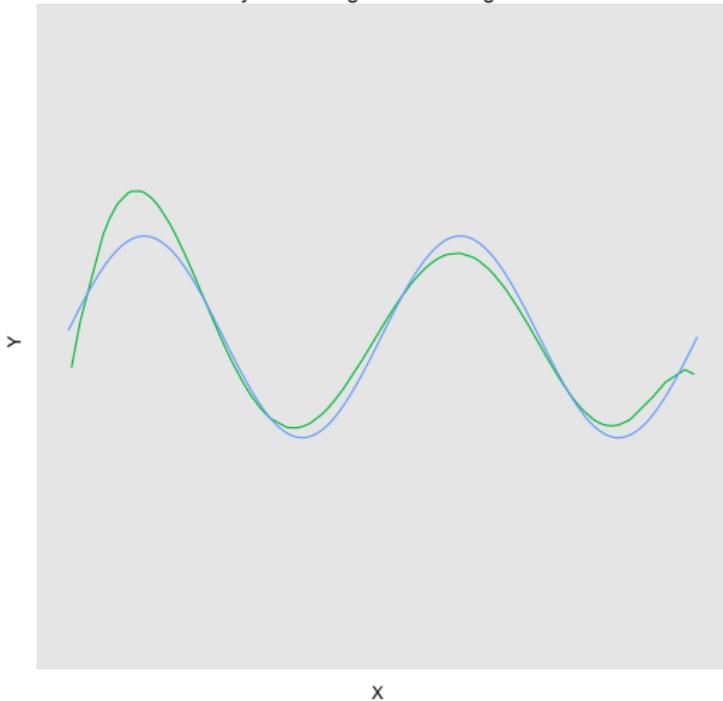
Polynomial Regression of Degree 5



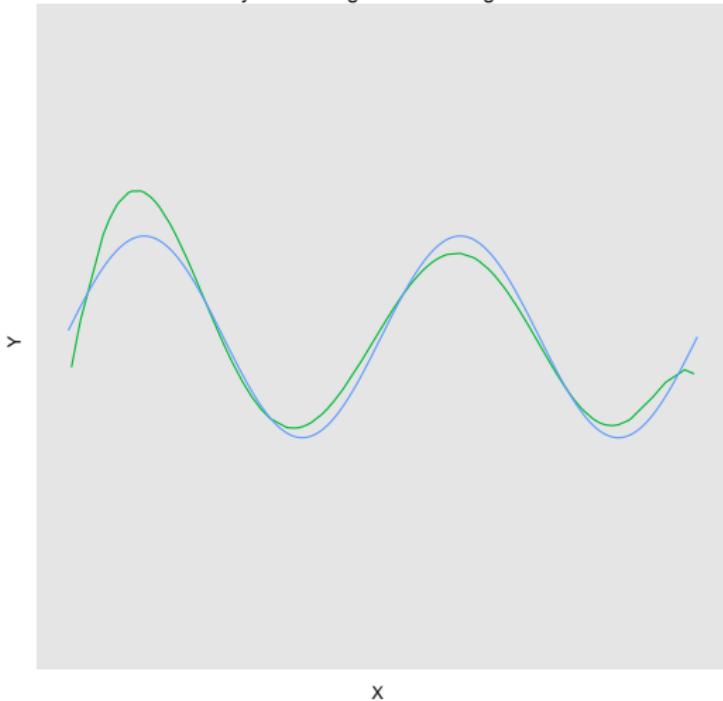
Polynomial Regression of Degree 6



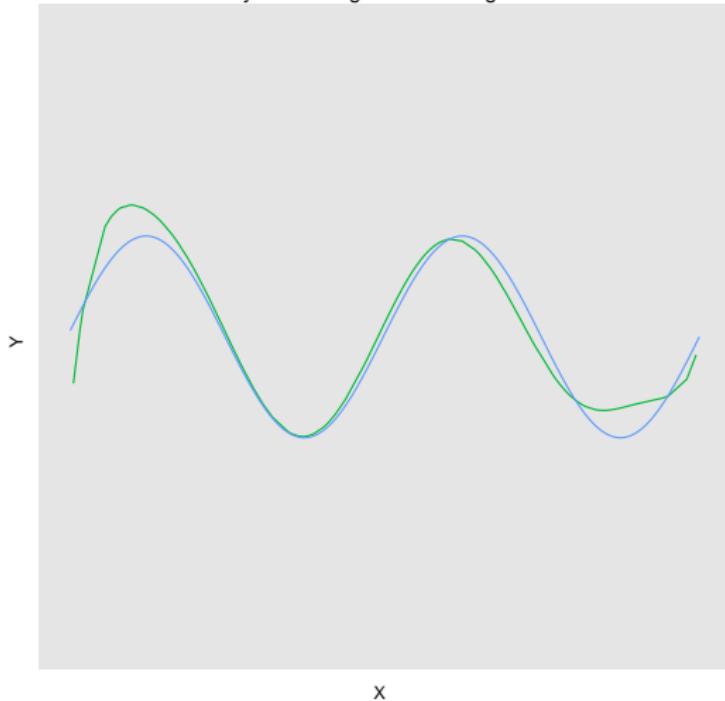
Polynomial Regression of Degree 7



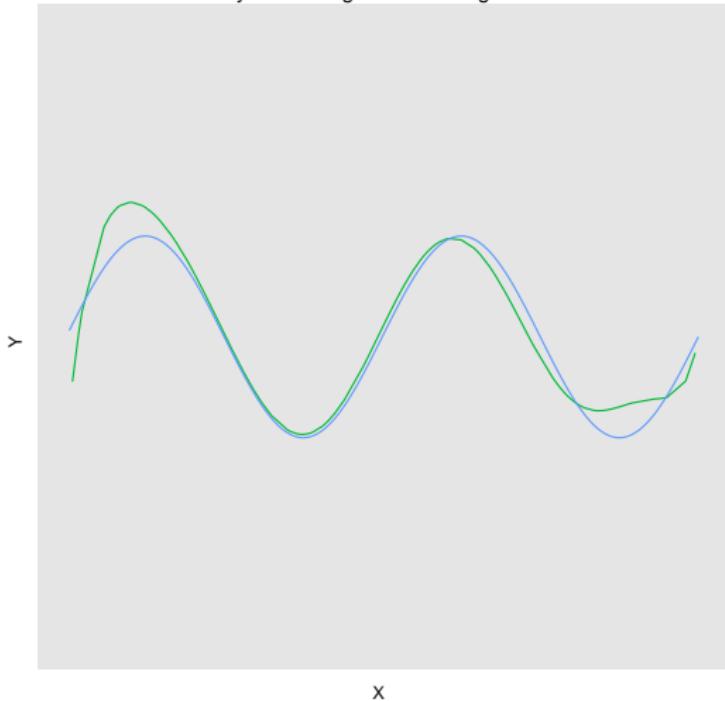
Polynomial Regression of Degree 8



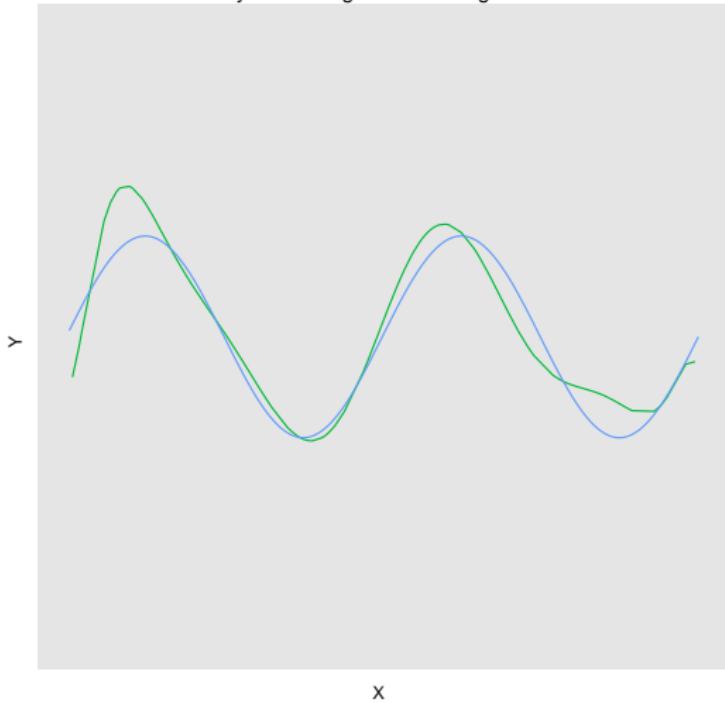
Polynomial Regression of Degree 9



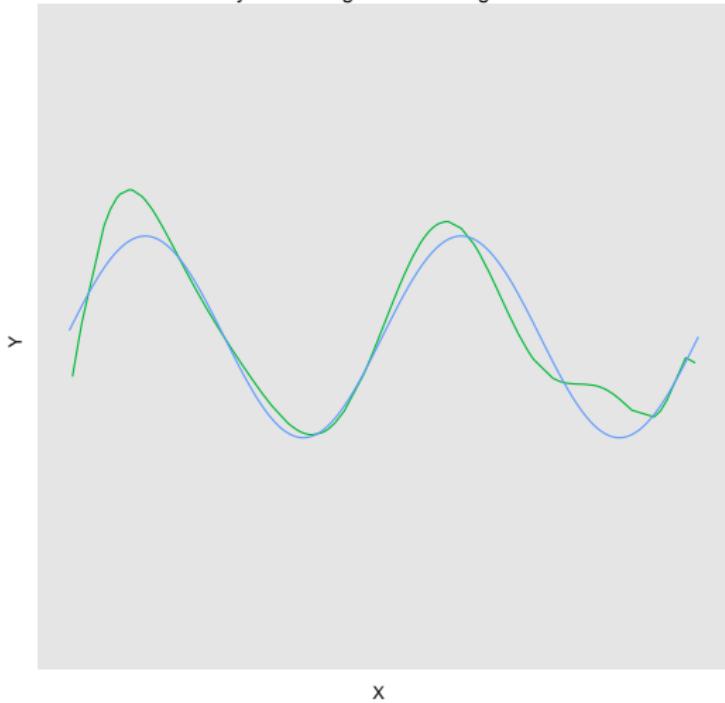
Polynomial Regression of Degree 10



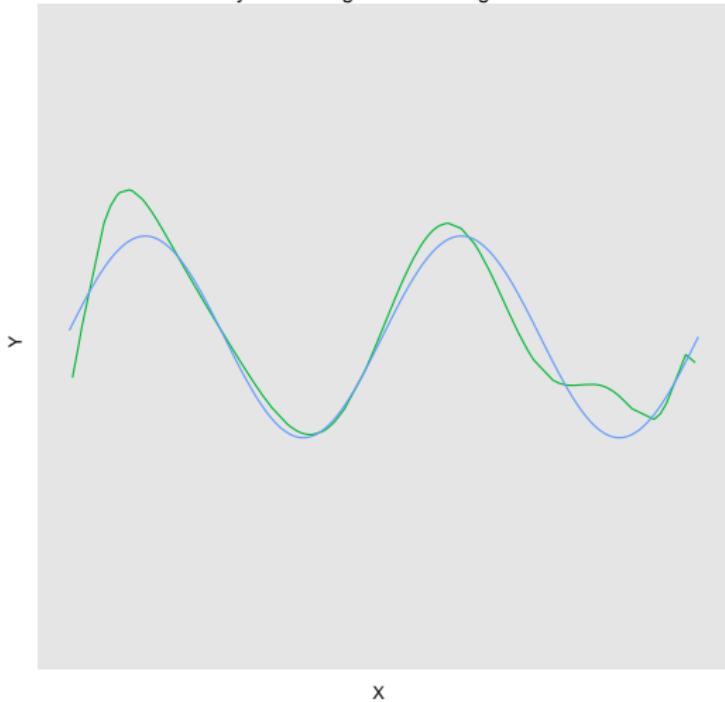
Polynomial Regression of Degree 11



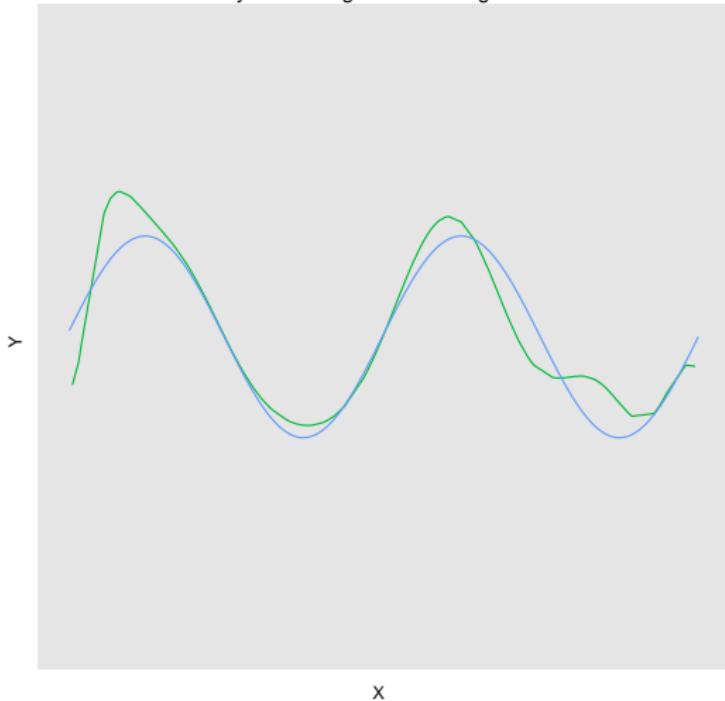
Polynomial Regression of Degree 12



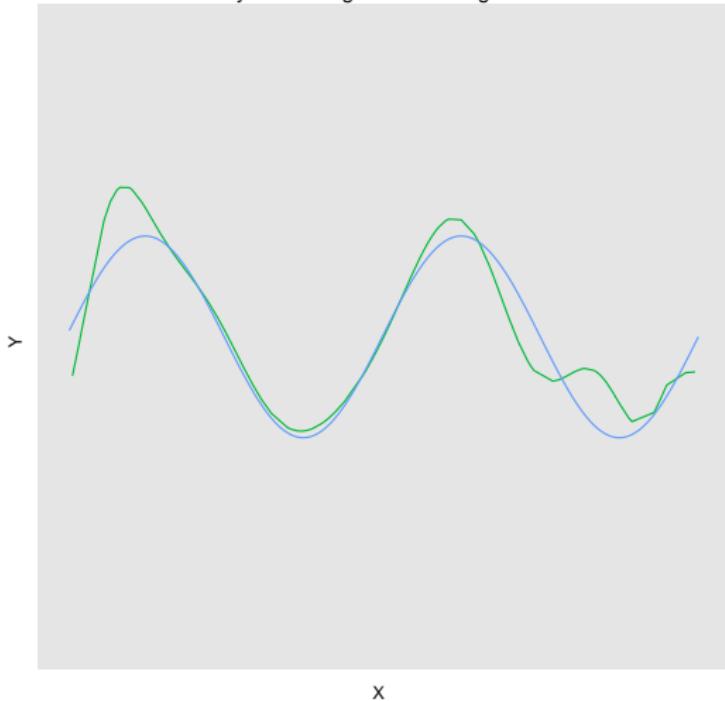
Polynomial Regression of Degree 13



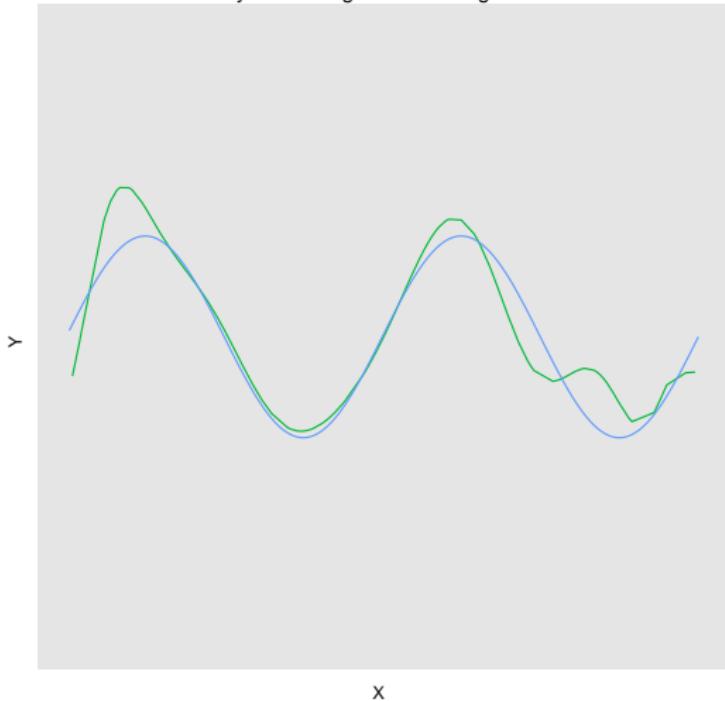
Polynomial Regression of Degree 14



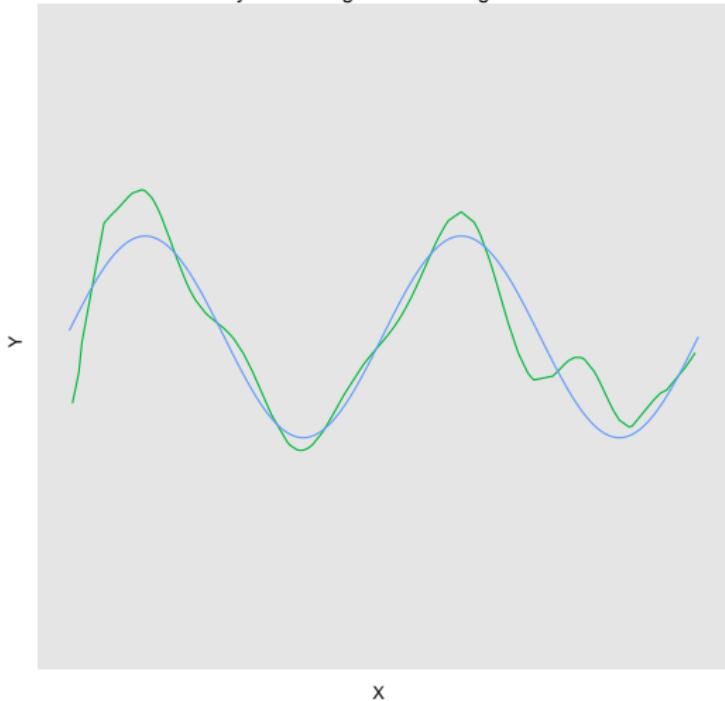
Polynomial Regression of Degree 15



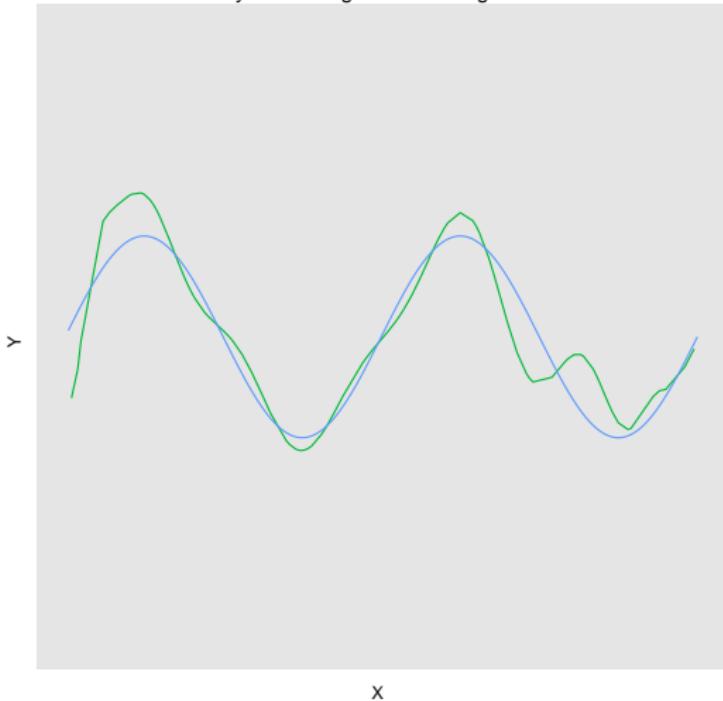
Polynomial Regression of Degree 16



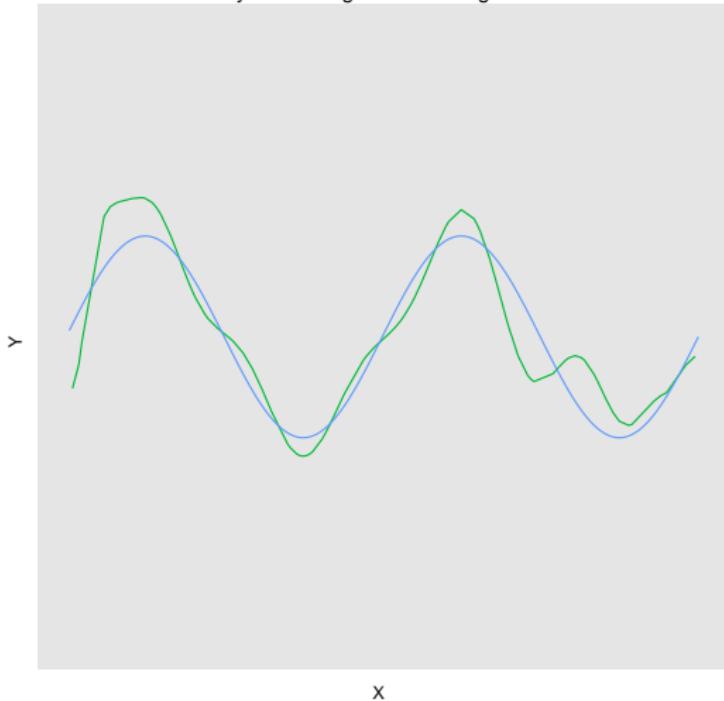
Polynomial Regression of Degree 17



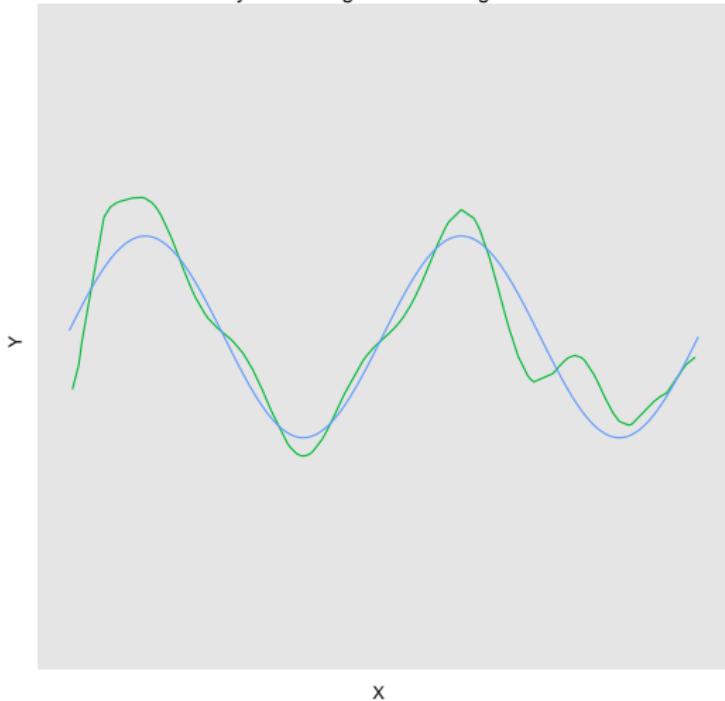
Polynomial Regression of Degree 18



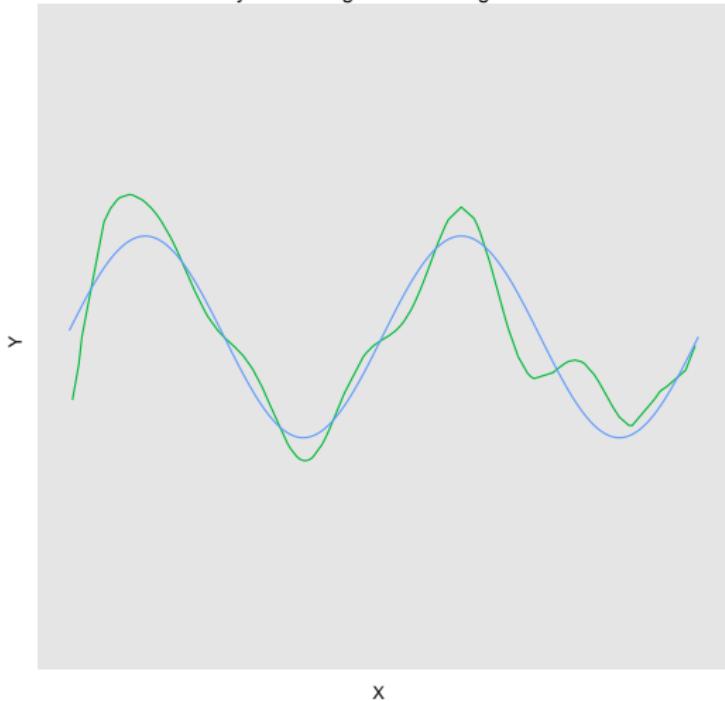
Polynomial Regression of Degree 19



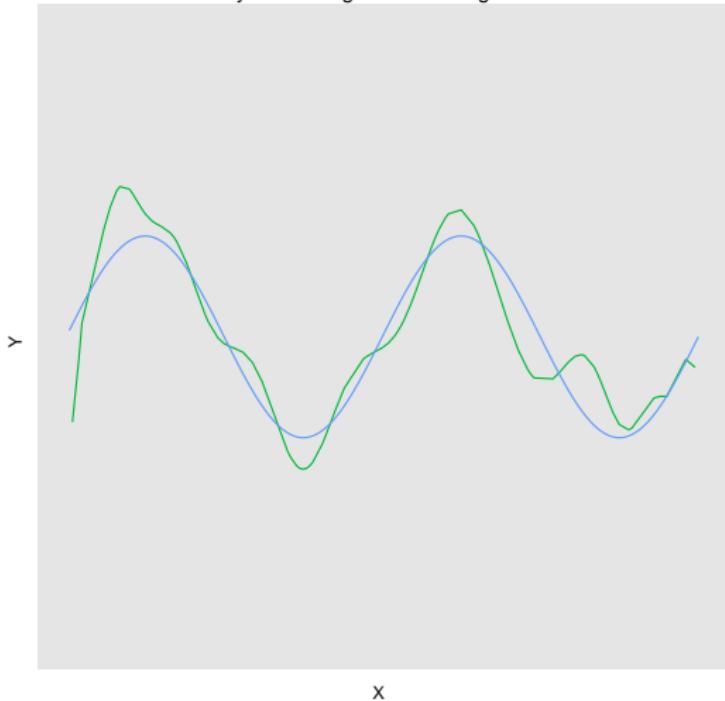
Polynomial Regression of Degree 20



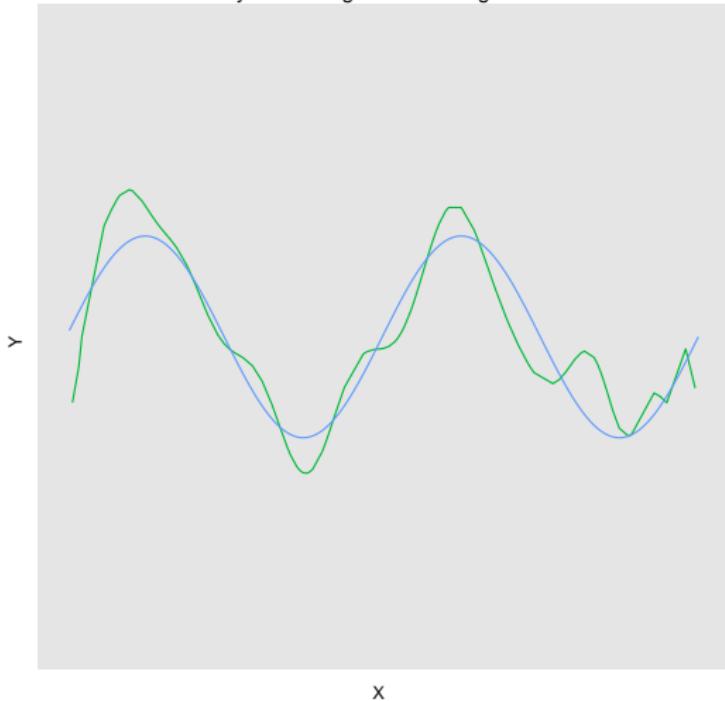
Polynomial Regression of Degree 21



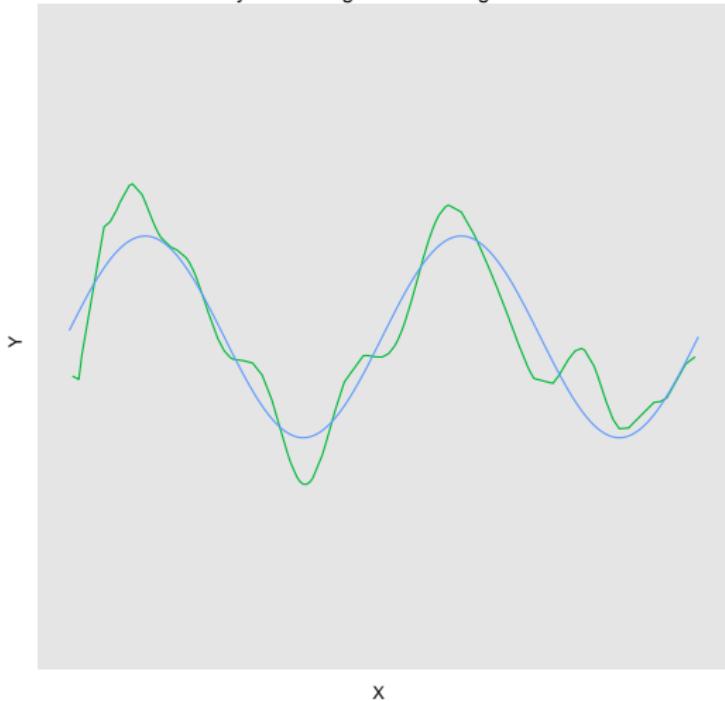
Polynomial Regression of Degree 22



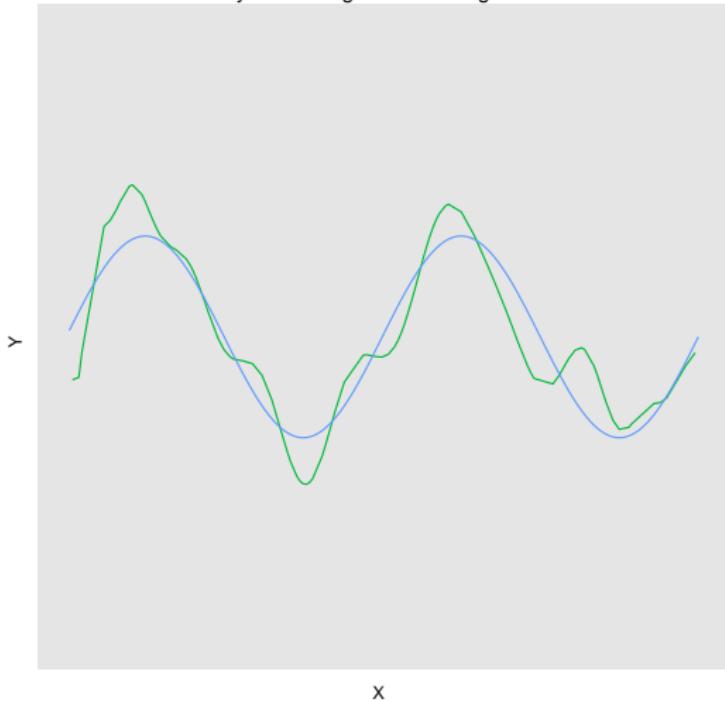
Polynomial Regression of Degree 23



Polynomial Regression of Degree 24



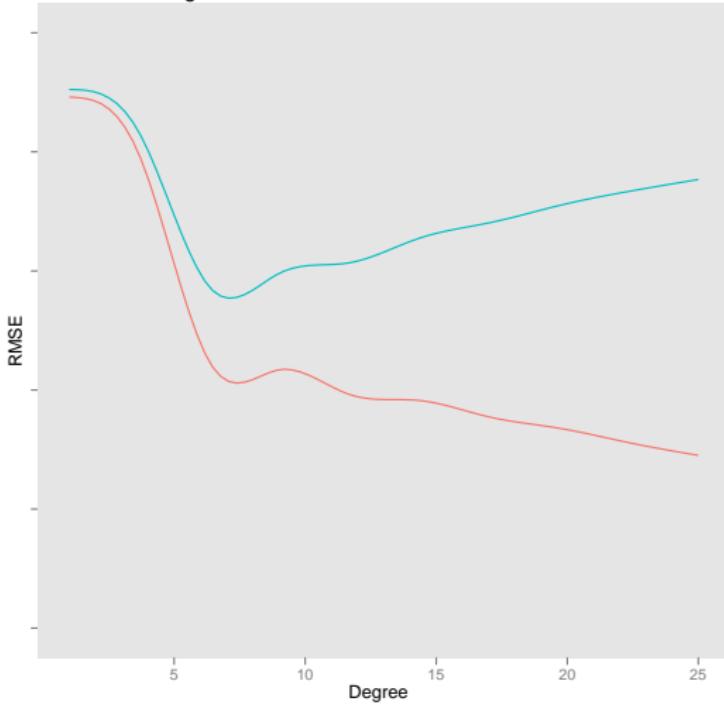
Polynomial Regression of Degree 25



Overfitting occurs when our models become too expressive

We find patterns that aren't real and fit our model to noise

Training Set Performance versus Test Set Performance

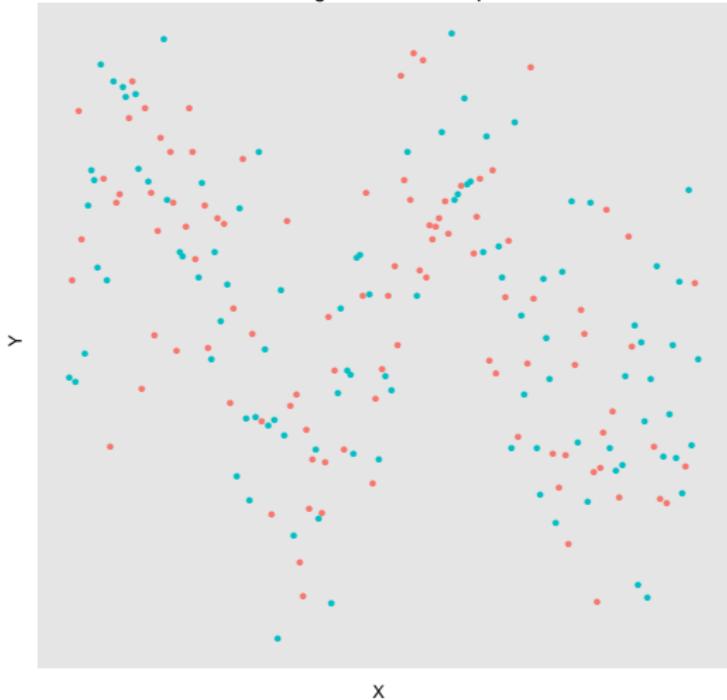


How do we prevent overfitting?

One approach:

- ▶ Split data into training set and test set
- ▶ Fit model to training set
- ▶ Assess performance on test set
- ▶ Pick model that does best on test set

Training Set / Test Set Split



This may underestimate the strongest model we can safely use

Another approach:

- ▶ Regularize our model

Unregularized models minimize prediction error:

- ▶ OLS regression

OLS regression:

$$\beta^* = \arg \min_{\beta} [(Y - X\beta)^2]$$

Regularized models minimize prediction error and model size:

- ▶ Ridge regression
- ▶ Lasso regression

Ridge regression:

$$\beta^* = \arg \min_{\beta} [(Y - X\beta)^2 + \lambda\beta^2]$$

Lasso regression:

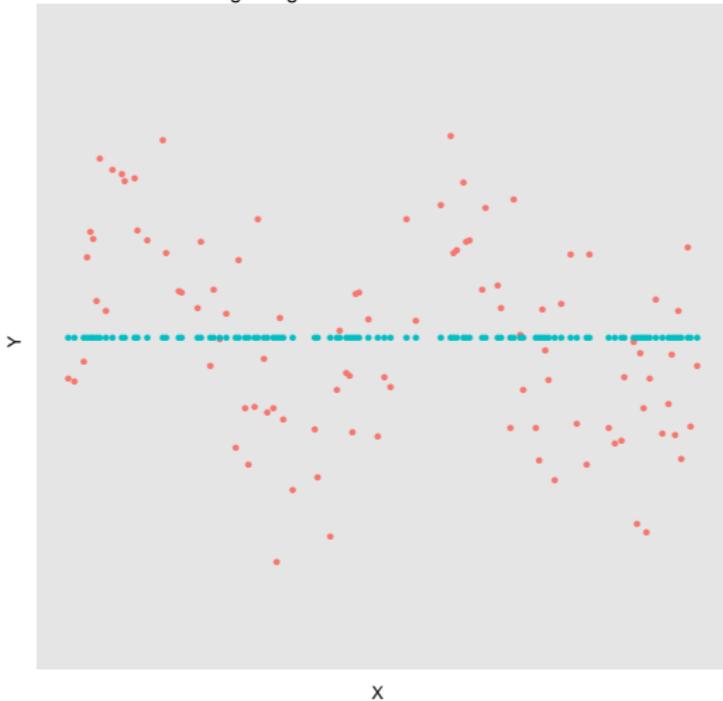
$$\beta^* = \arg \min_{\beta} [(Y - X\beta)^2 + \lambda|\beta|]$$

The hyperparameter λ controls the amount of regularization

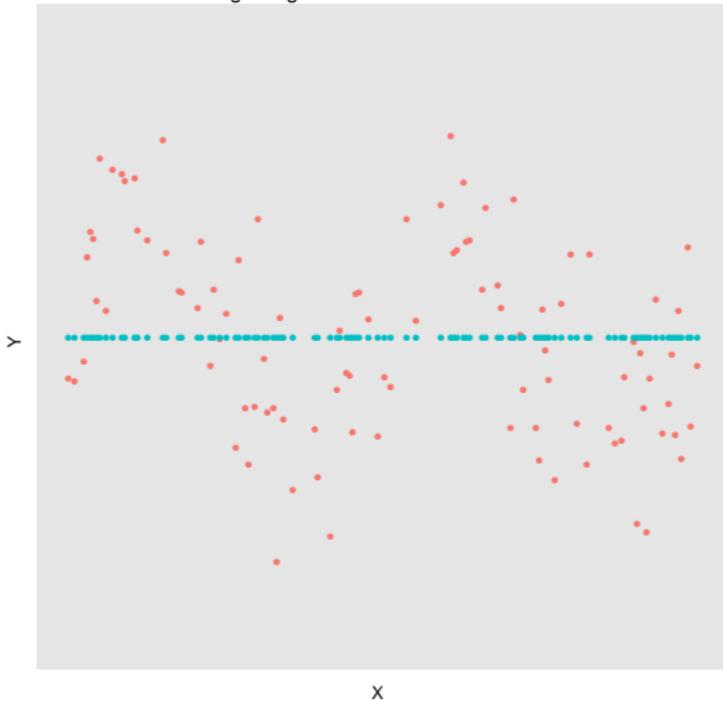
How does λ affect ridge regression?

Start with degree 25 polynomial and weaken regularization

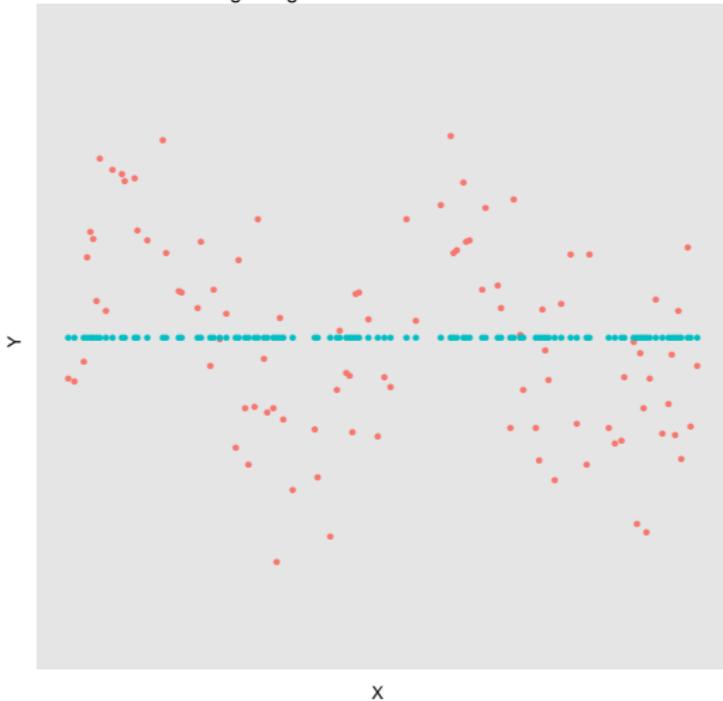
Ridge Regression with Lambda = 524



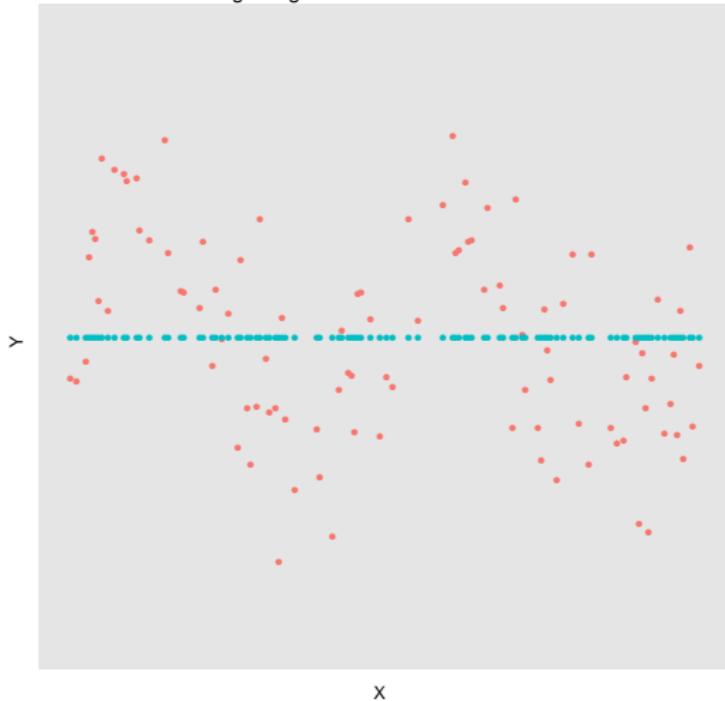
Ridge Regression with Lambda = 478



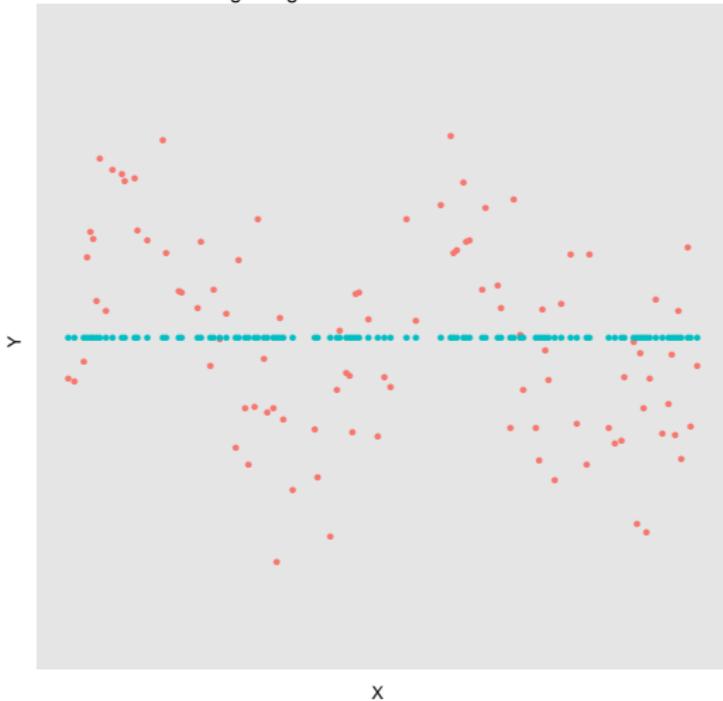
Ridge Regression with Lambda = 435



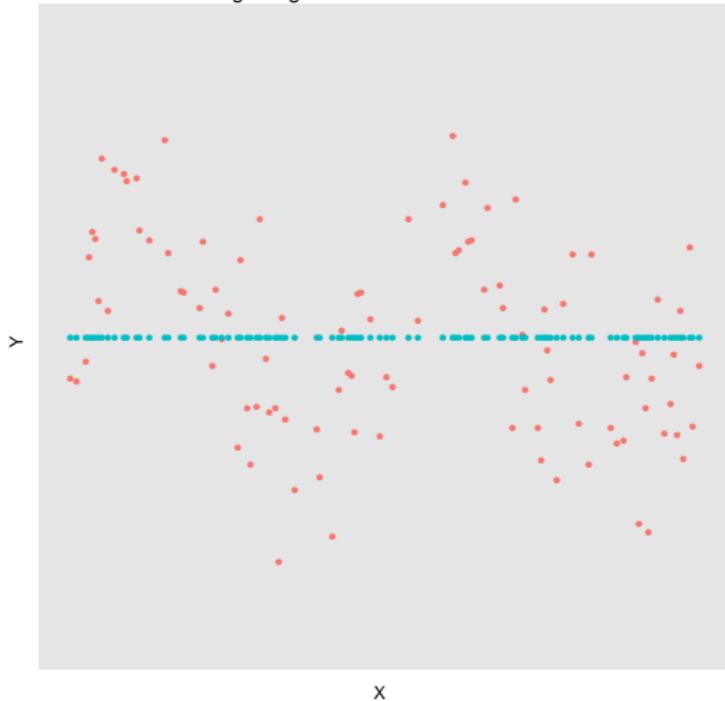
Ridge Regression with Lambda = 396



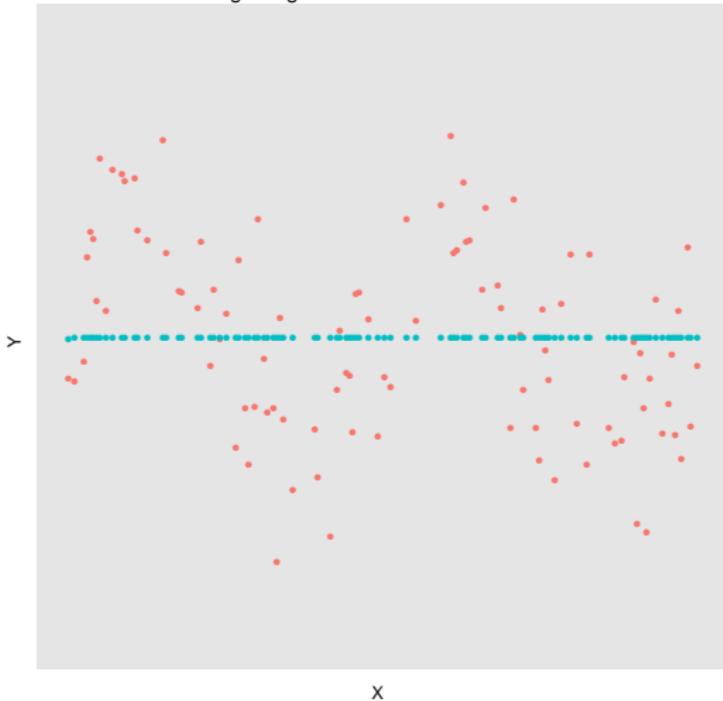
Ridge Regression with Lambda = 361



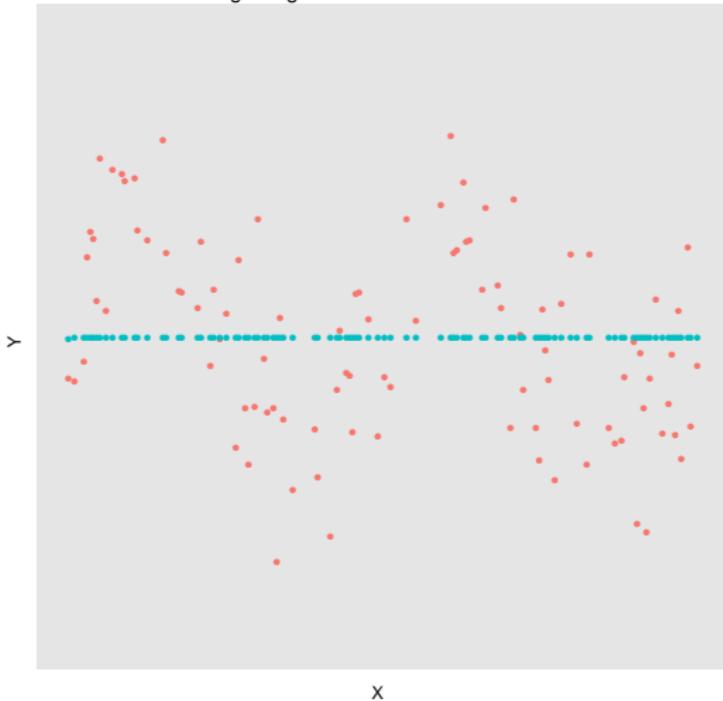
Ridge Regression with Lambda = 329



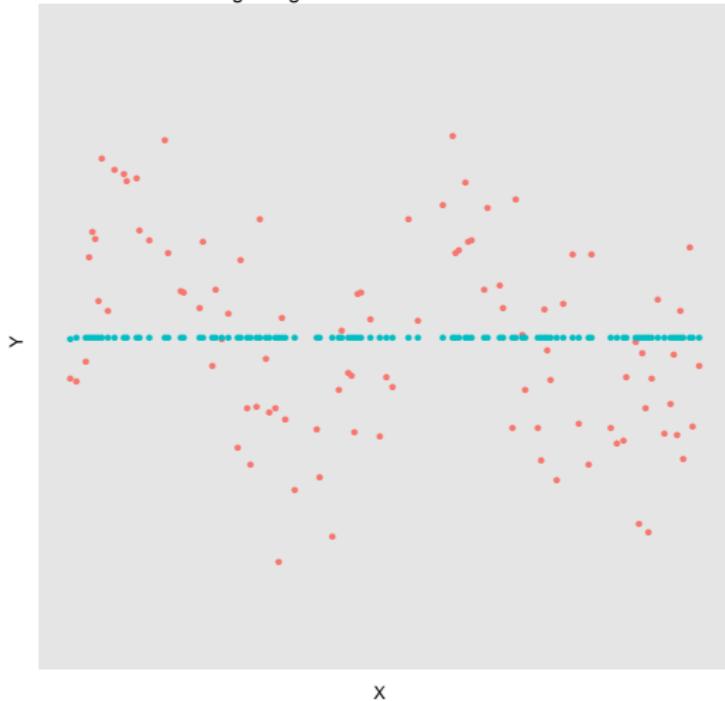
Ridge Regression with Lambda = 300



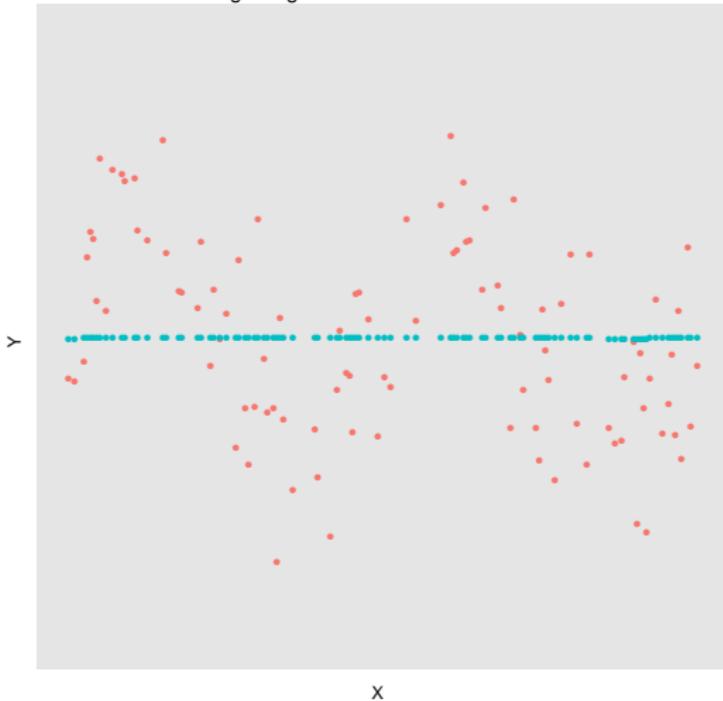
Ridge Regression with Lambda = 273



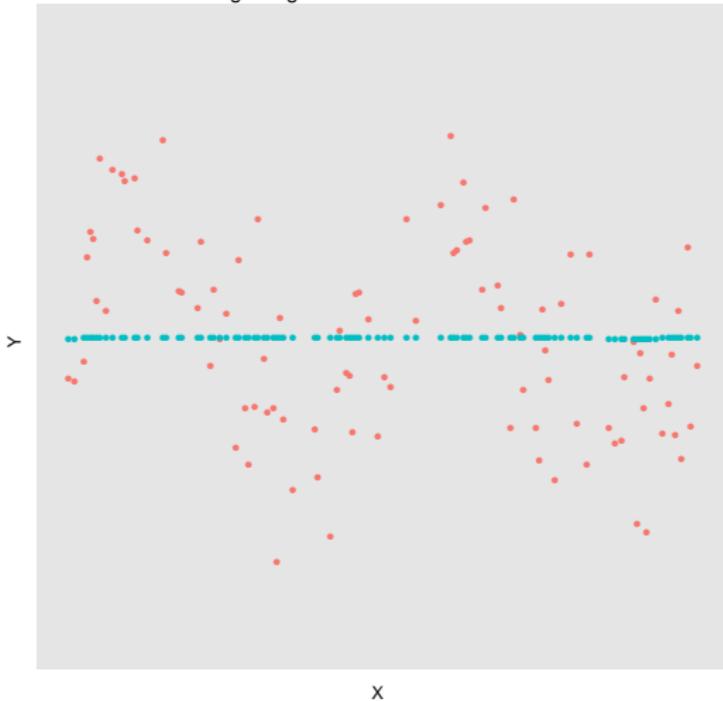
Ridge Regression with Lambda = 249



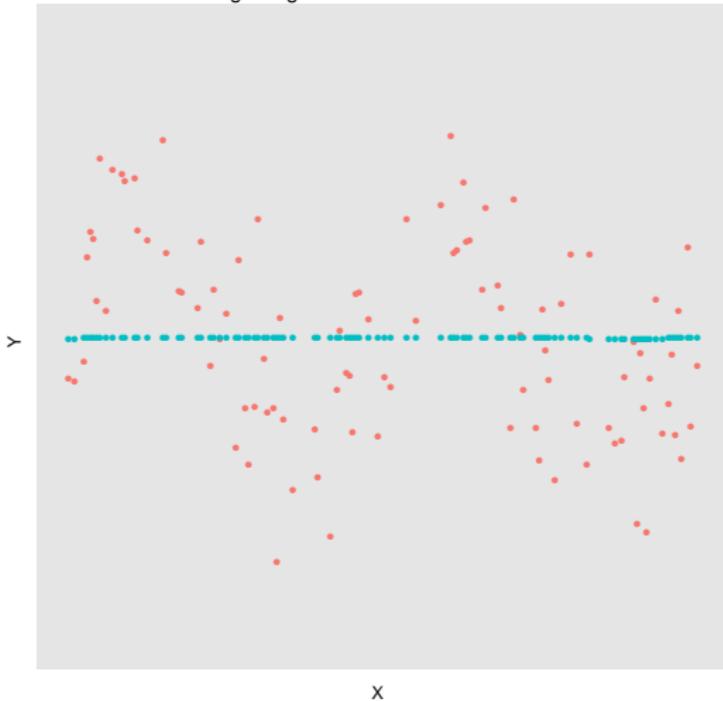
Ridge Regression with Lambda = 227



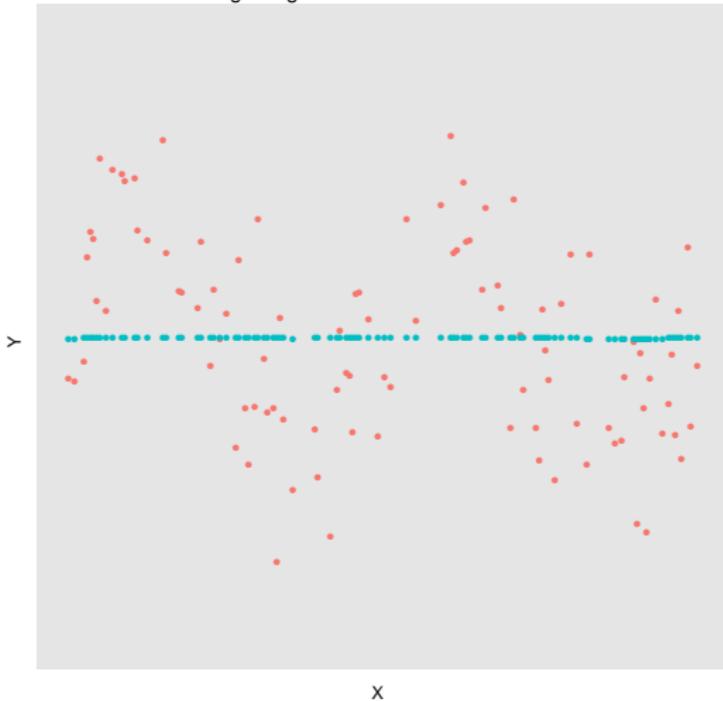
Ridge Regression with Lambda = 207



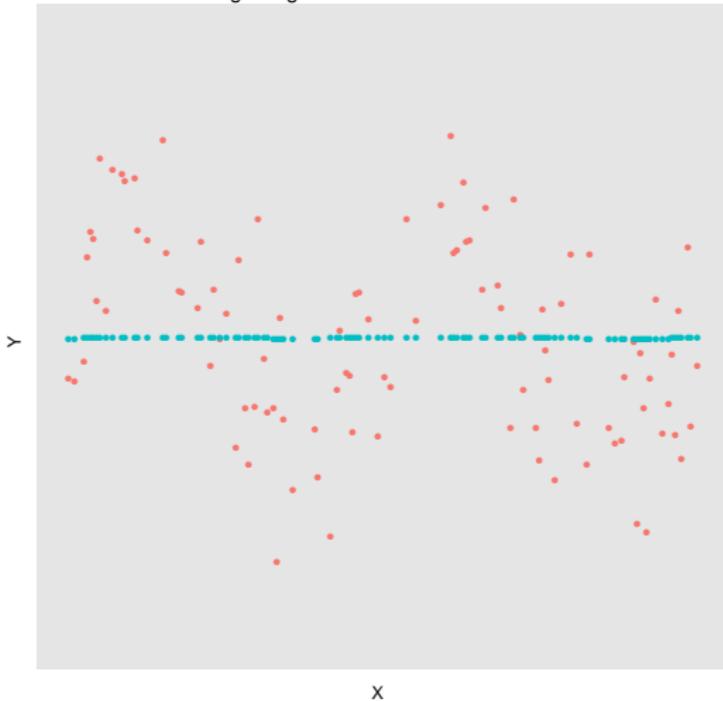
Ridge Regression with Lambda = 188



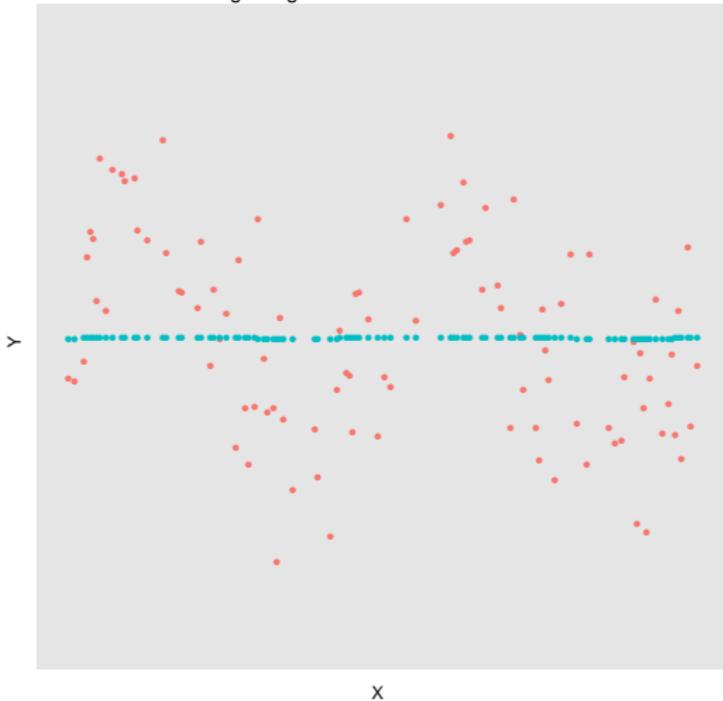
Ridge Regression with Lambda = 172



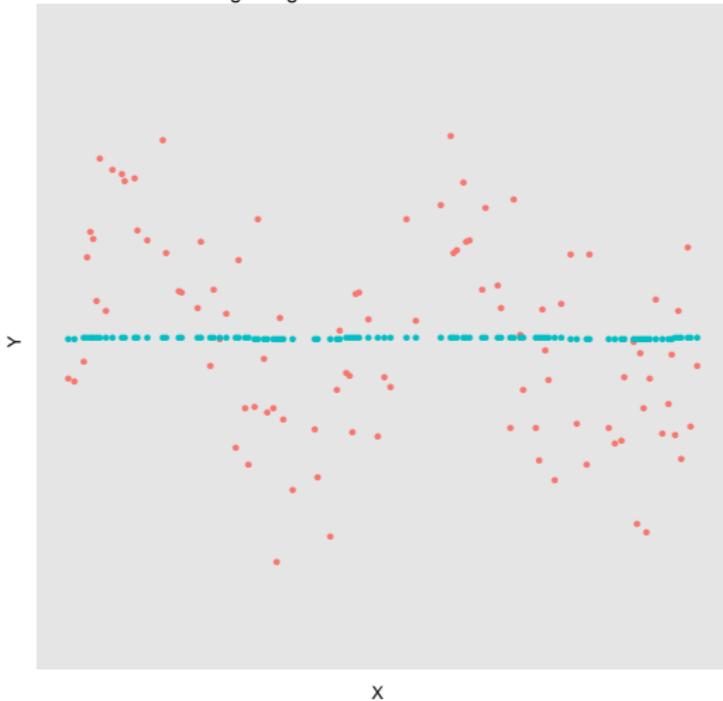
Ridge Regression with Lambda = 156



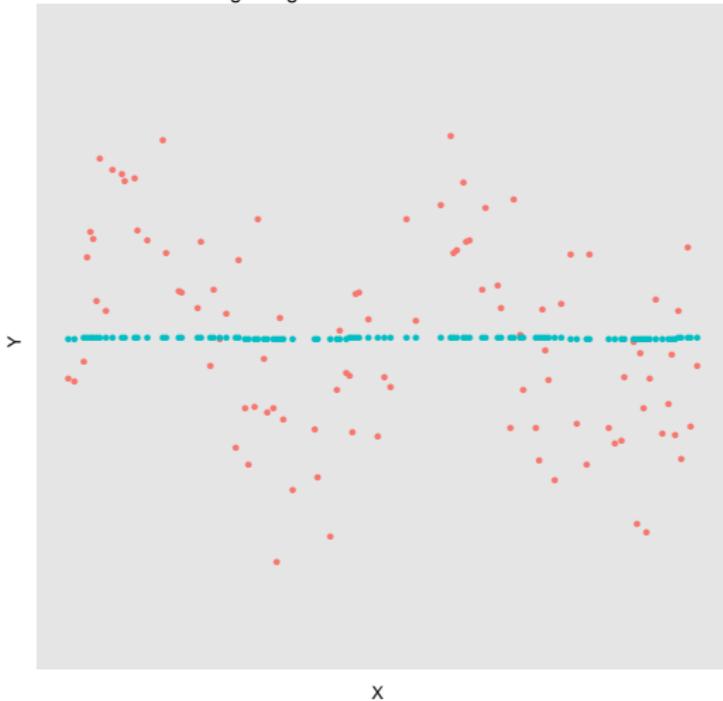
Ridge Regression with Lambda = 142



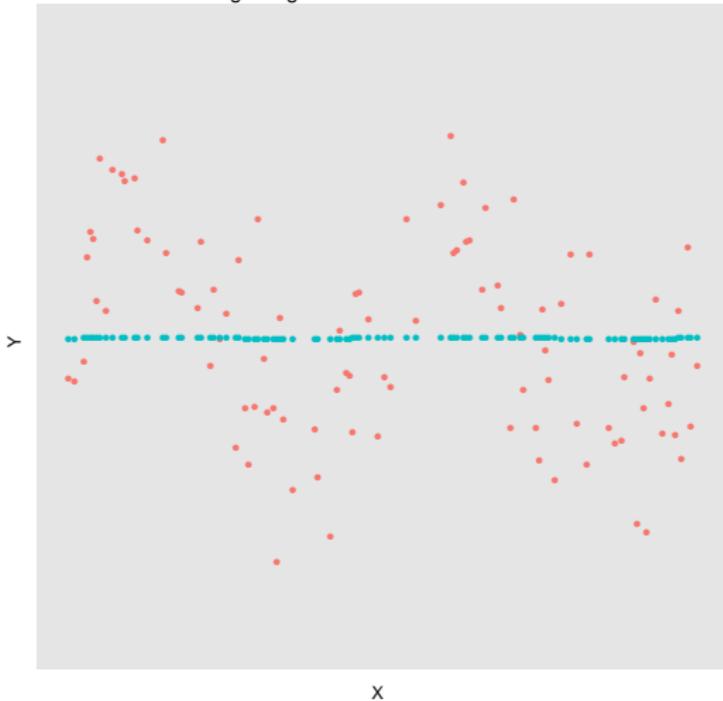
Ridge Regression with Lambda = 130



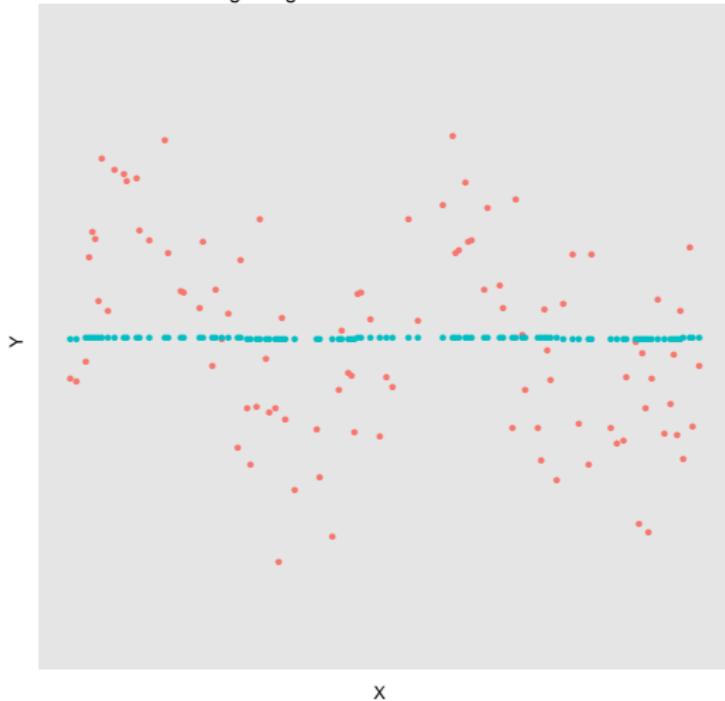
Ridge Regression with Lambda = 118



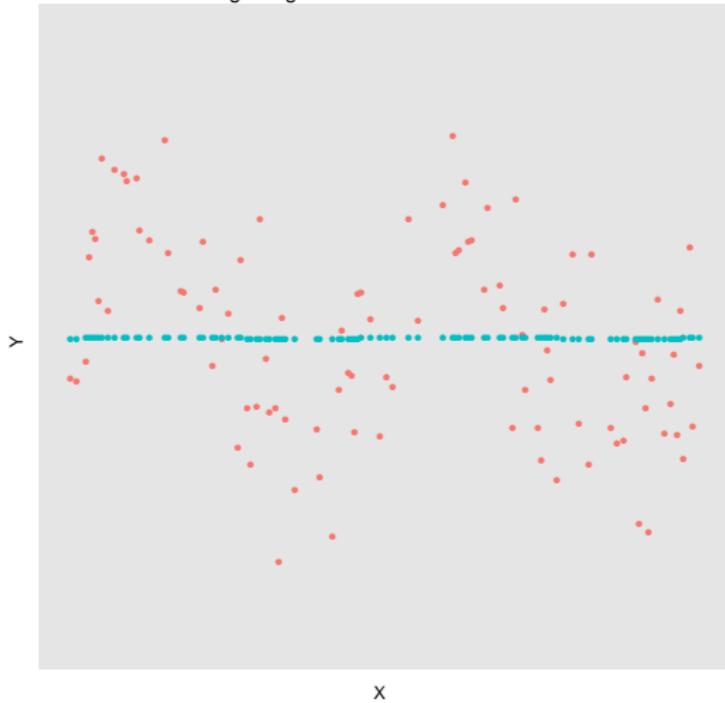
Ridge Regression with Lambda = 108



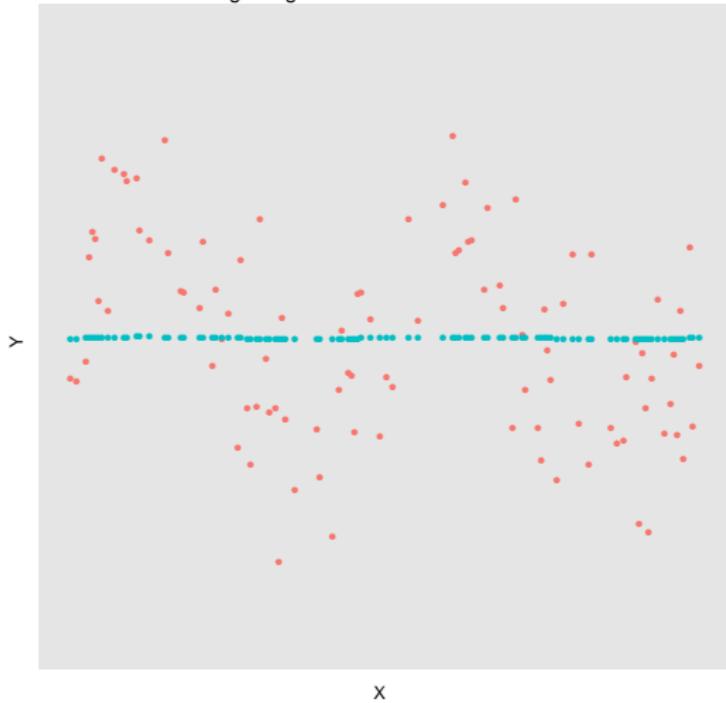
Ridge Regression with Lambda = 98.2



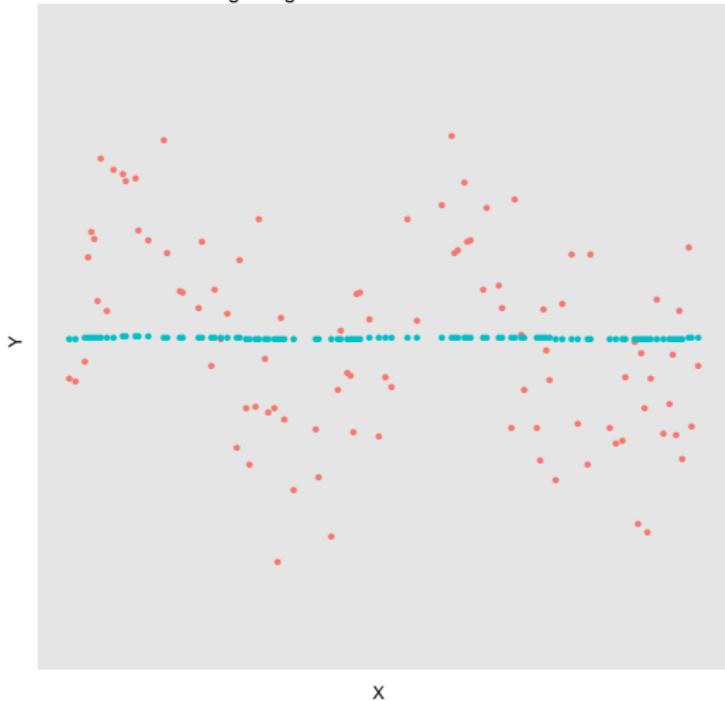
Ridge Regression with Lambda = 89.5



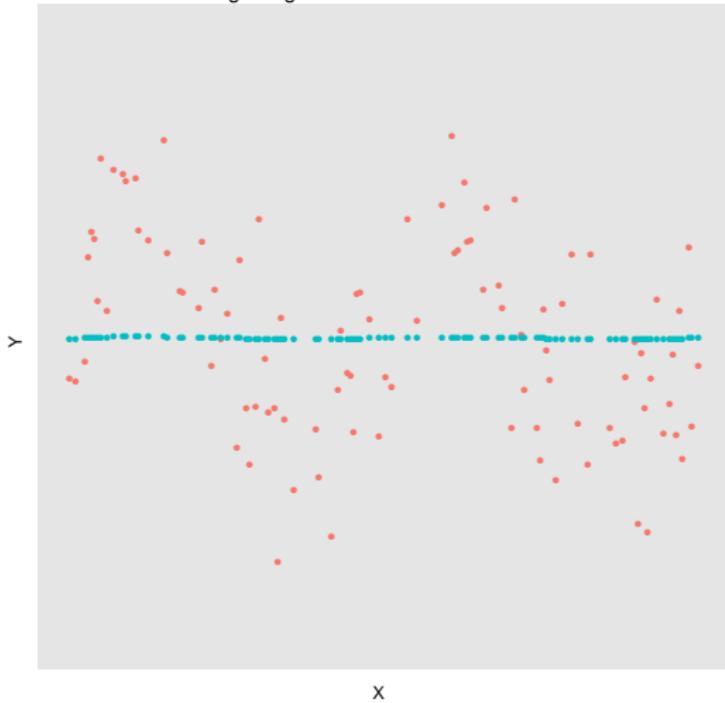
Ridge Regression with Lambda = 81.5



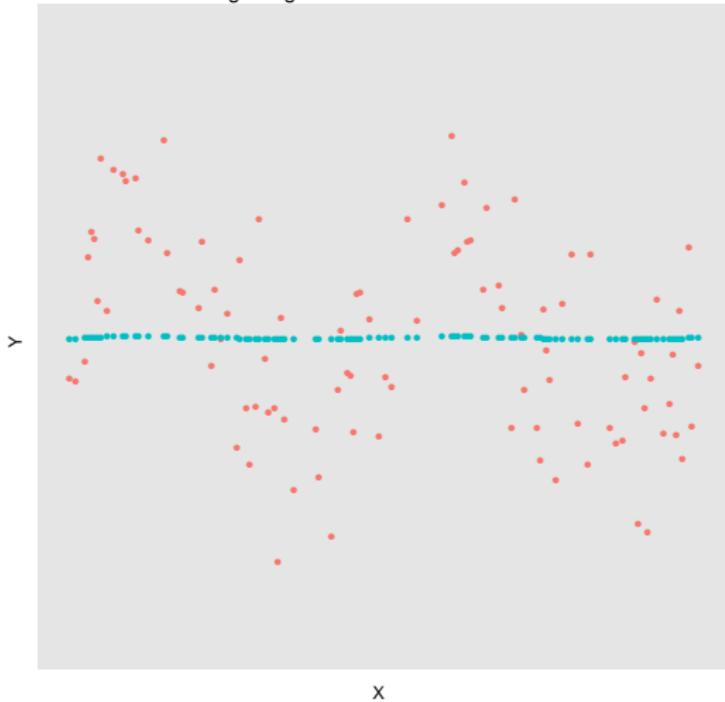
Ridge Regression with Lambda = 74.3



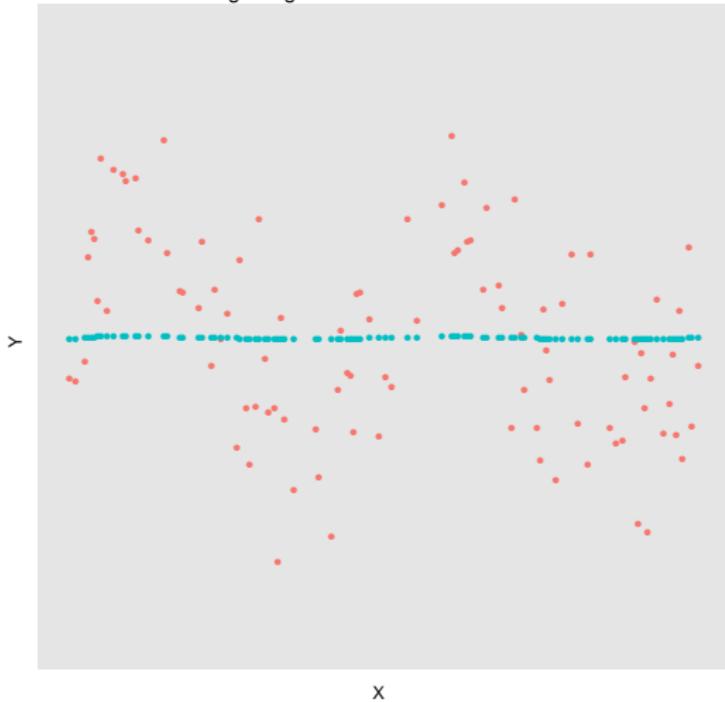
Ridge Regression with Lambda = 67.7



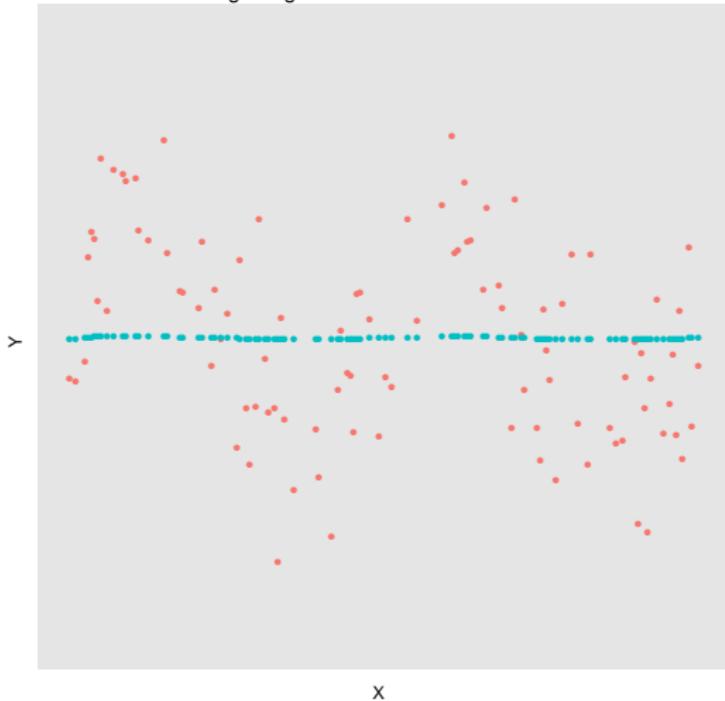
Ridge Regression with Lambda = 61.7



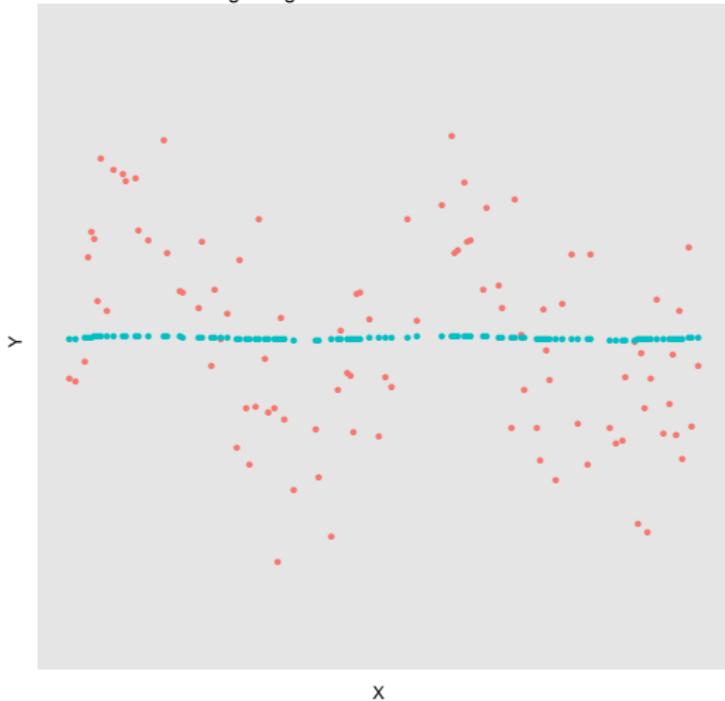
Ridge Regression with Lambda = 56.2



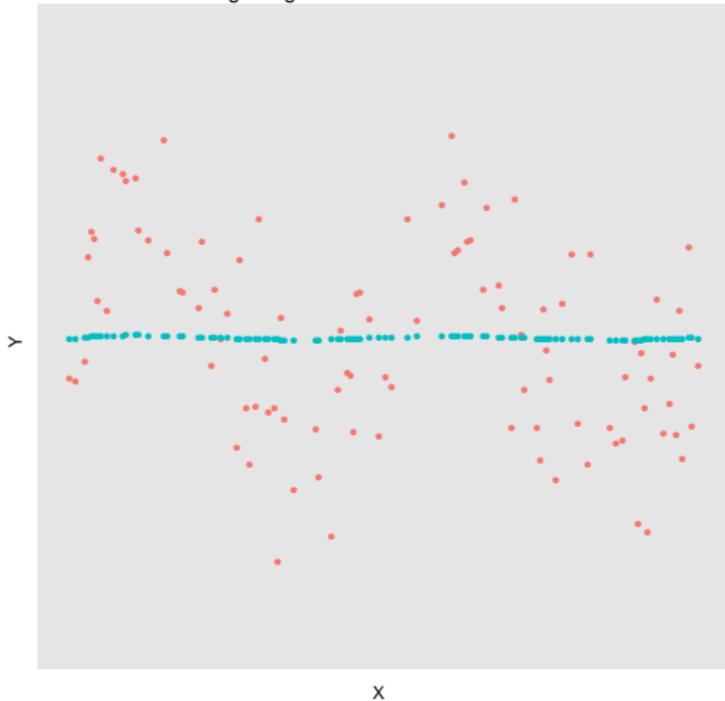
Ridge Regression with Lambda = 51.2



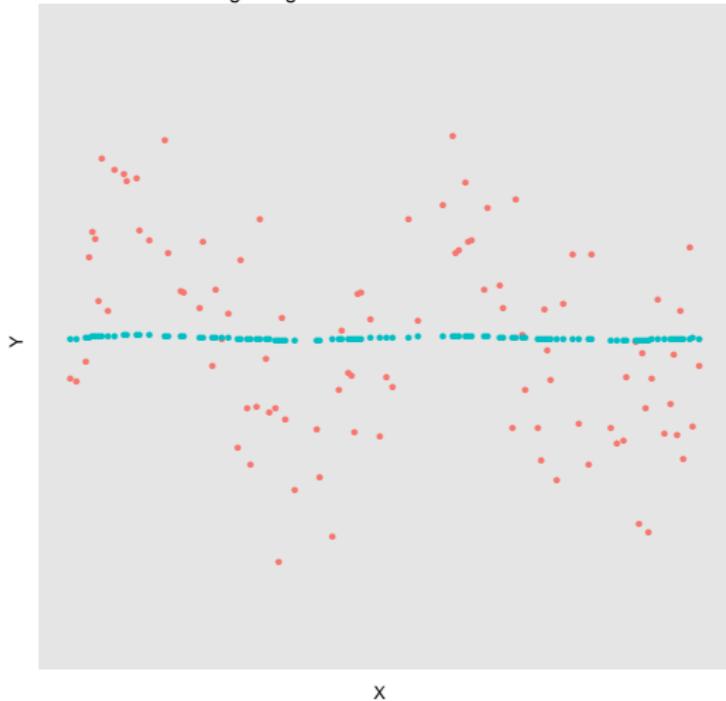
Ridge Regression with Lambda = 46.7



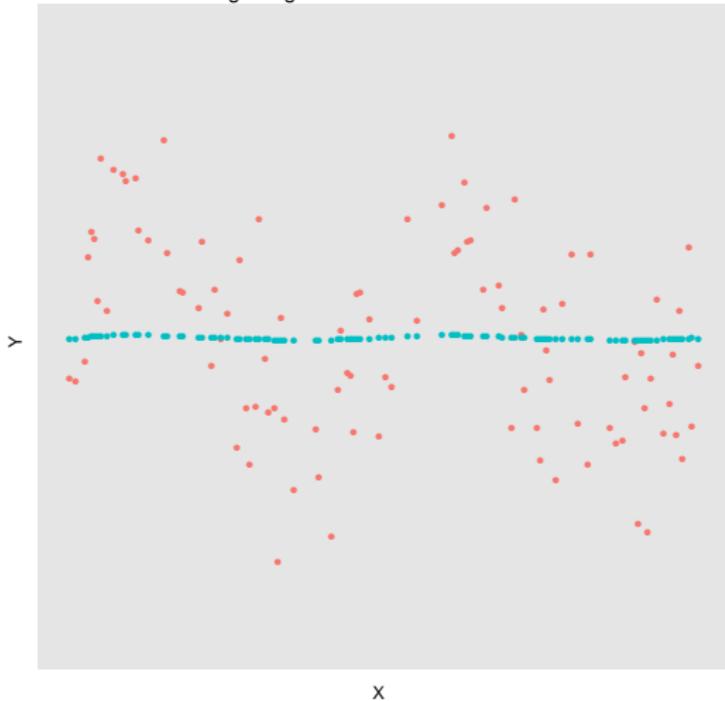
Ridge Regression with Lambda = 42.5



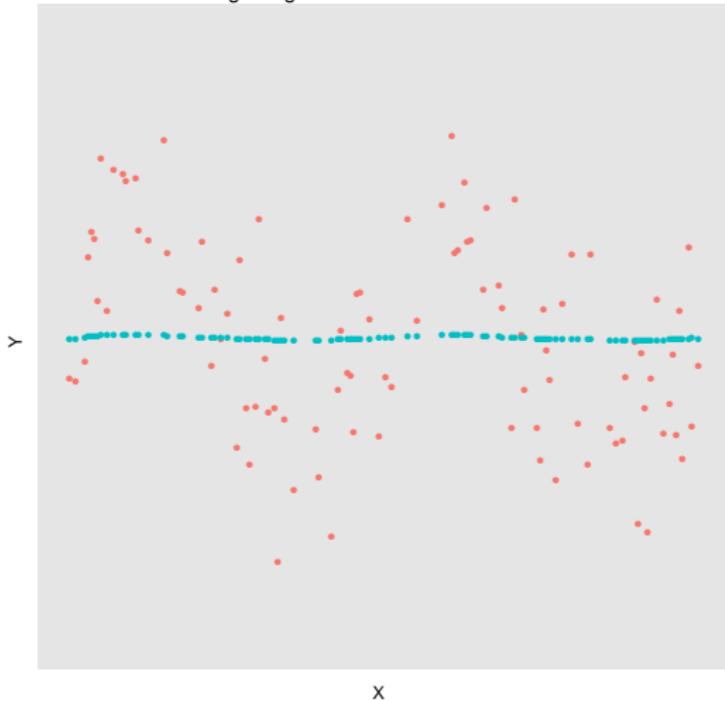
Ridge Regression with Lambda = 38.7



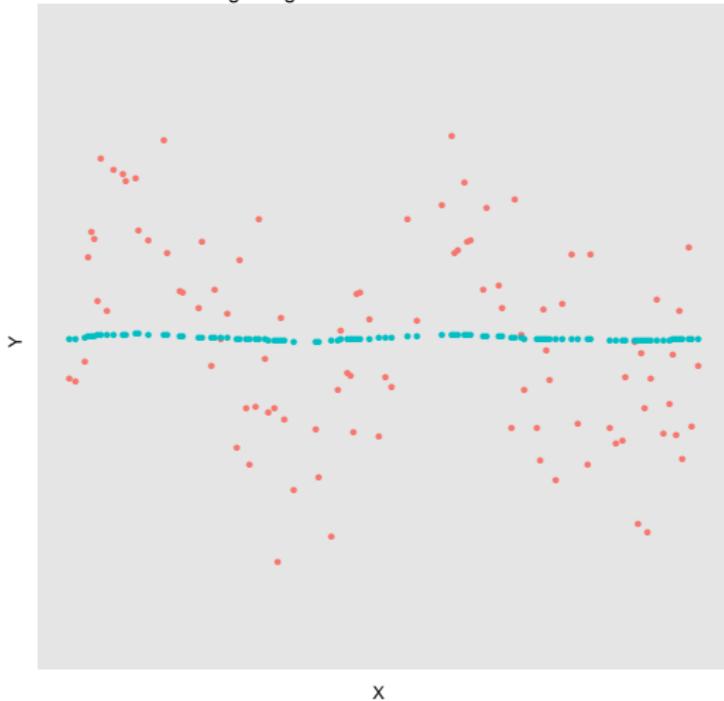
Ridge Regression with Lambda = 35.3



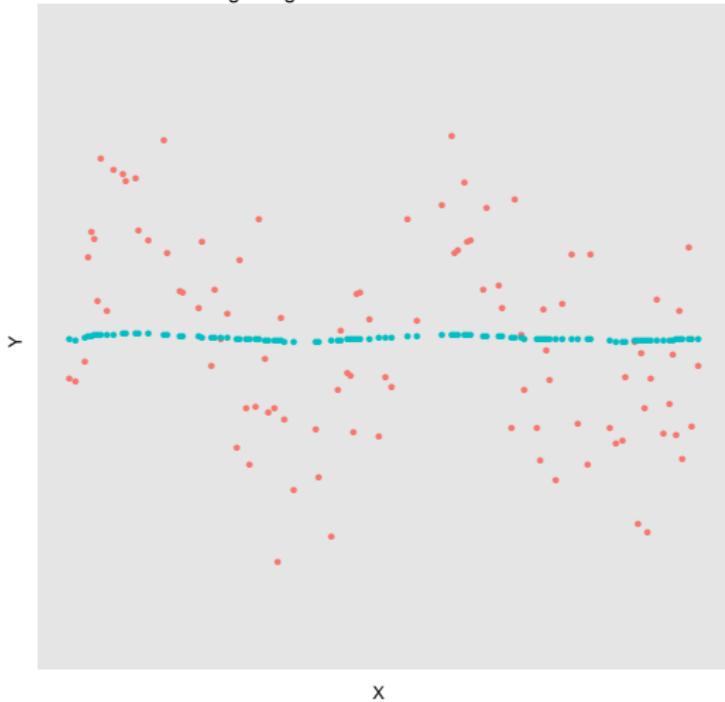
Ridge Regression with Lambda = 32.2



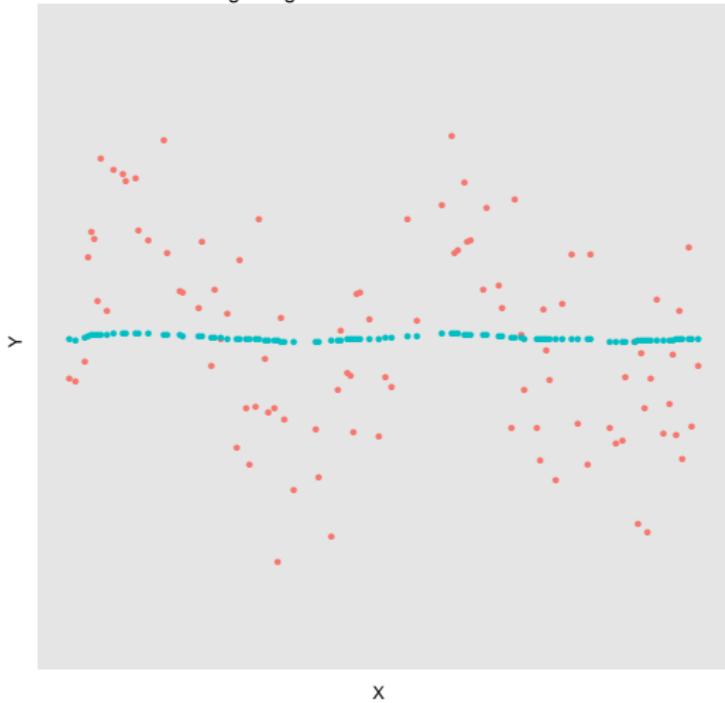
Ridge Regression with Lambda = 29.3



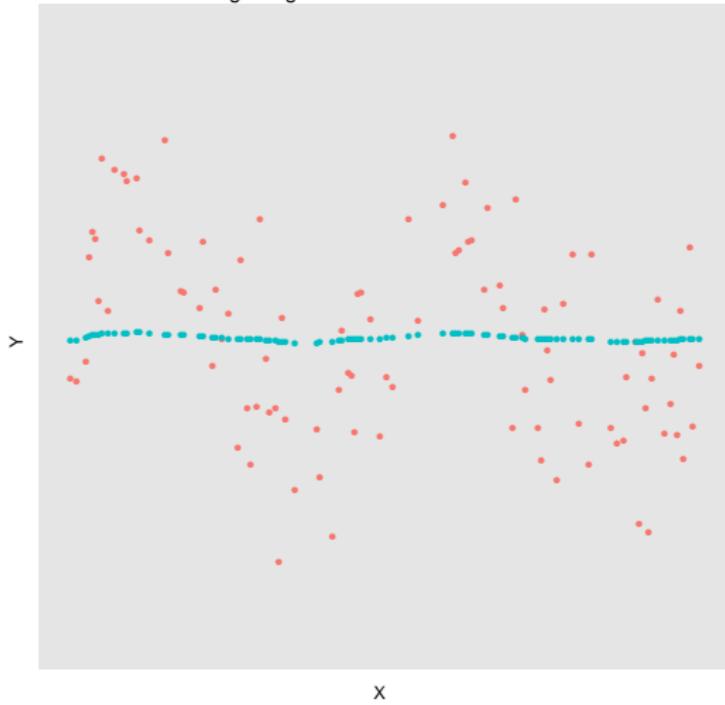
Ridge Regression with Lambda = 26.7



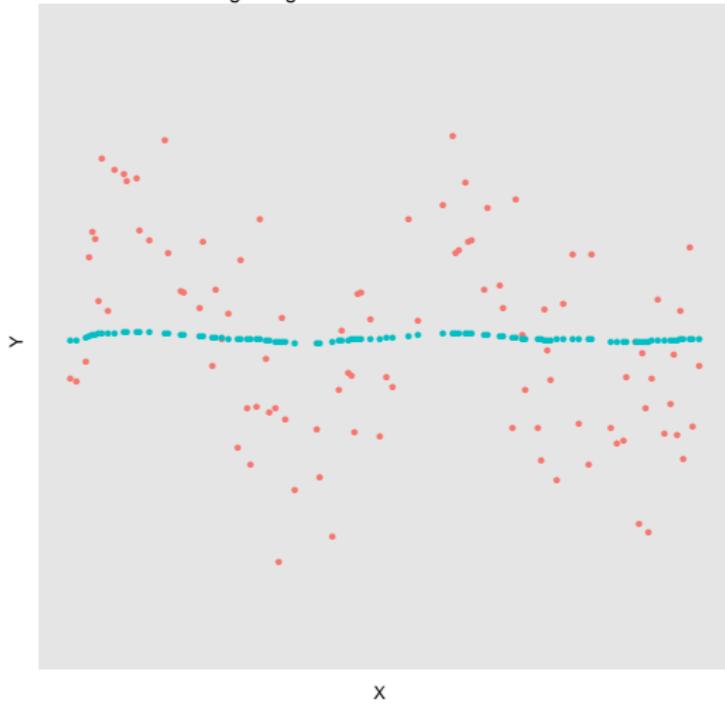
Ridge Regression with Lambda = 24.3



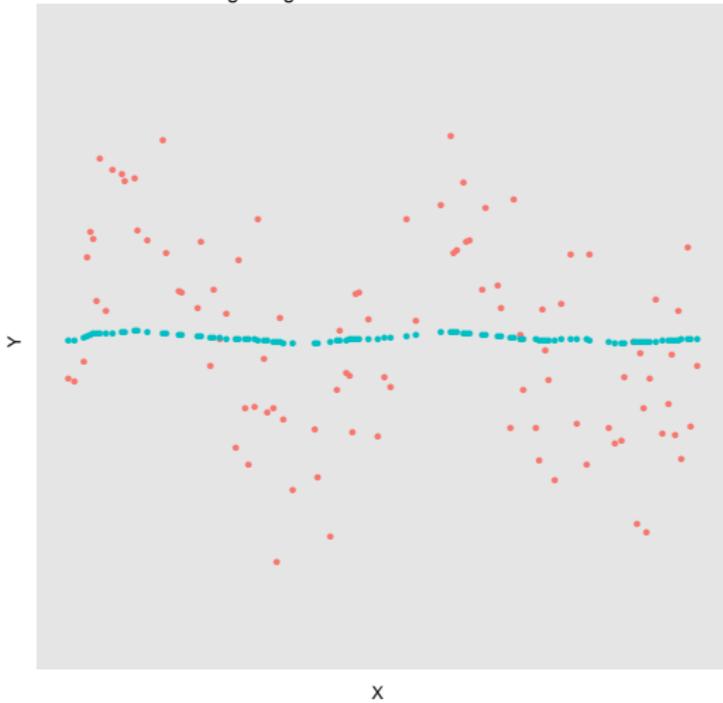
Ridge Regression with Lambda = 22.2



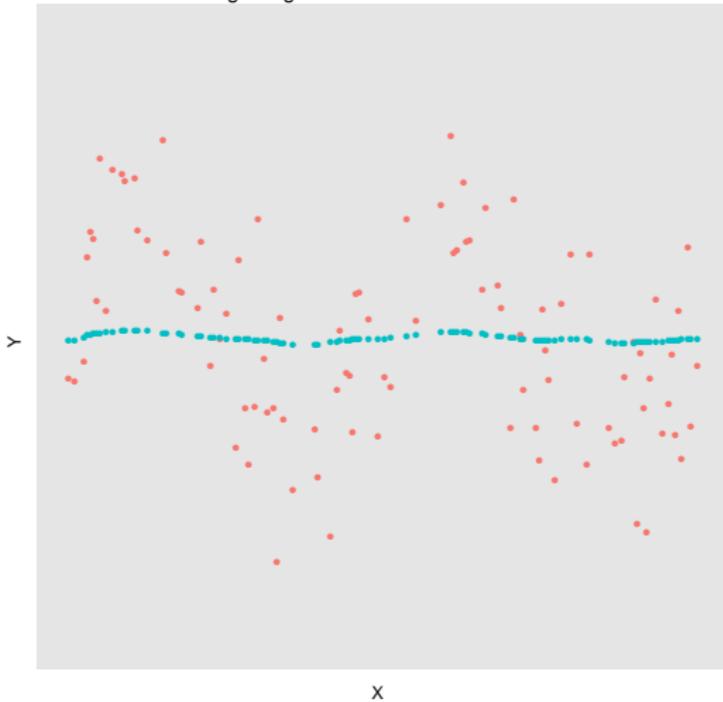
Ridge Regression with Lambda = 20.2



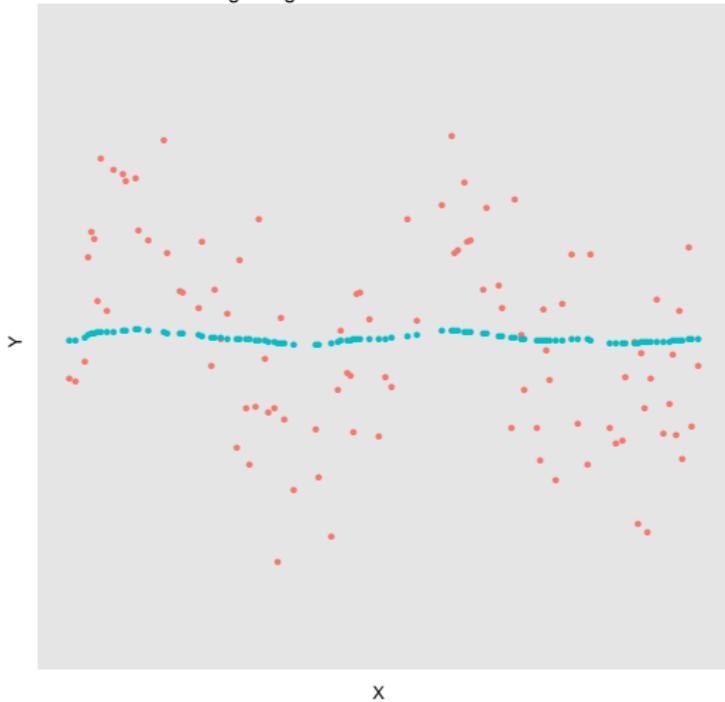
Ridge Regression with Lambda = 18.4



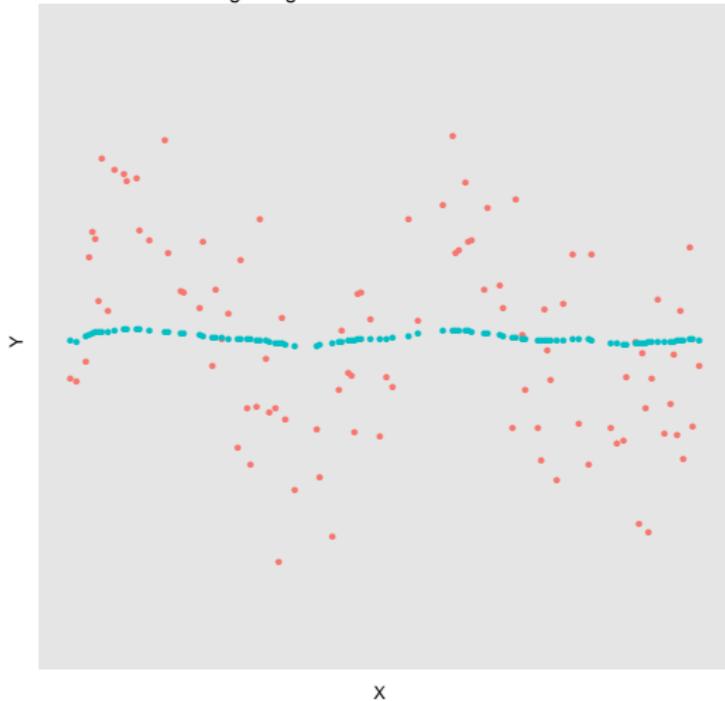
Ridge Regression with Lambda = 16.8



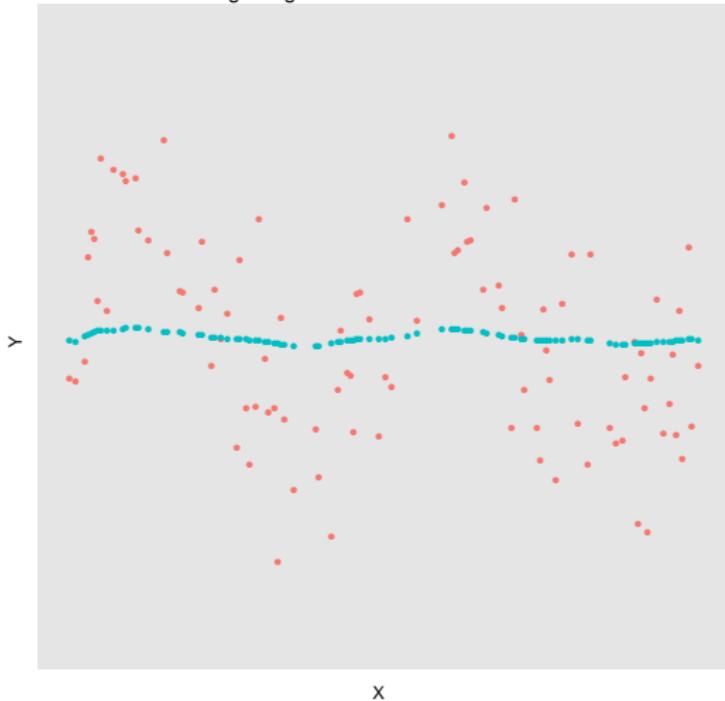
Ridge Regression with Lambda = 15.3



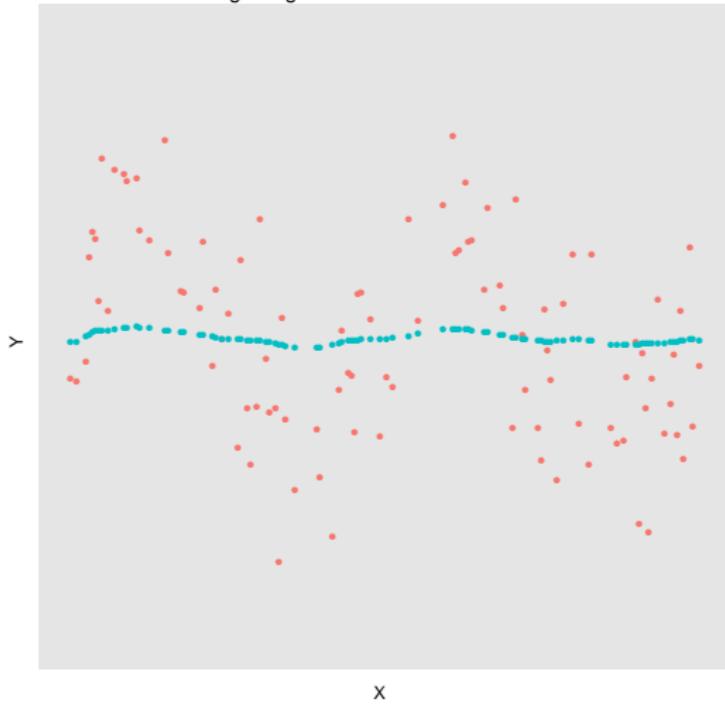
Ridge Regression with Lambda = 13.9



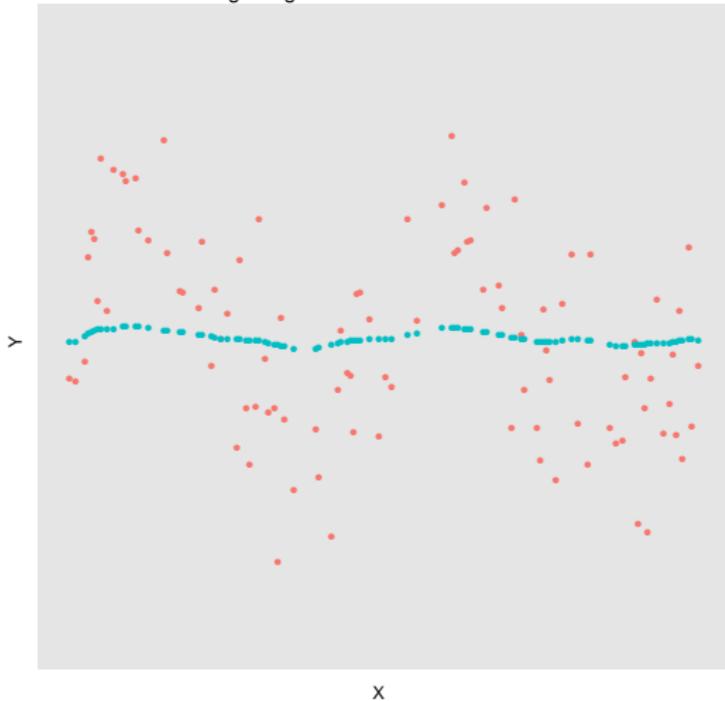
Ridge Regression with Lambda = 12.7



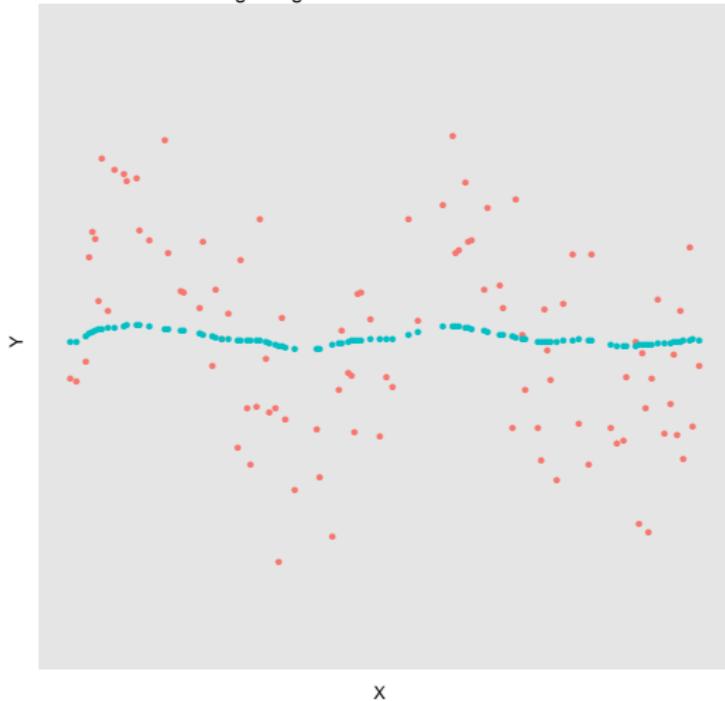
Ridge Regression with Lambda = 11.6



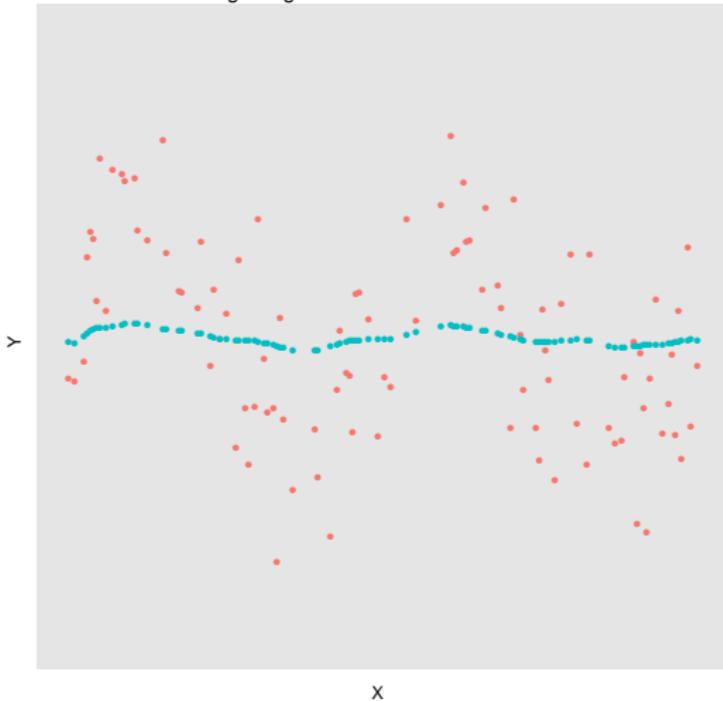
Ridge Regression with Lambda = 10.5



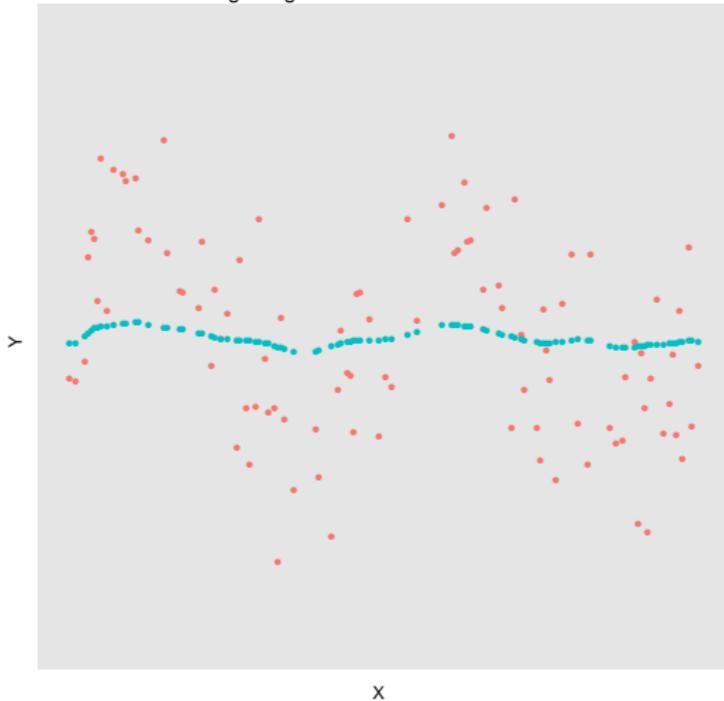
Ridge Regression with Lambda = 9.6



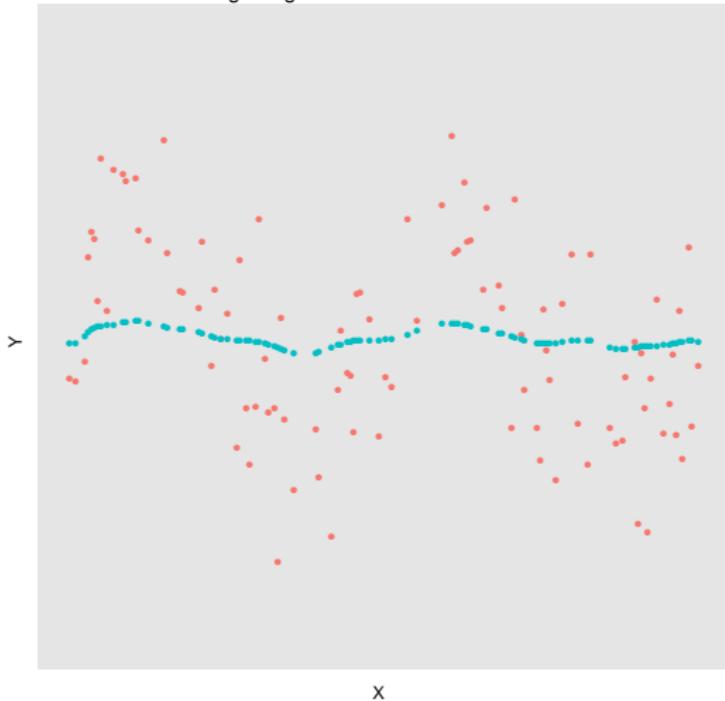
Ridge Regression with Lambda = 8.74



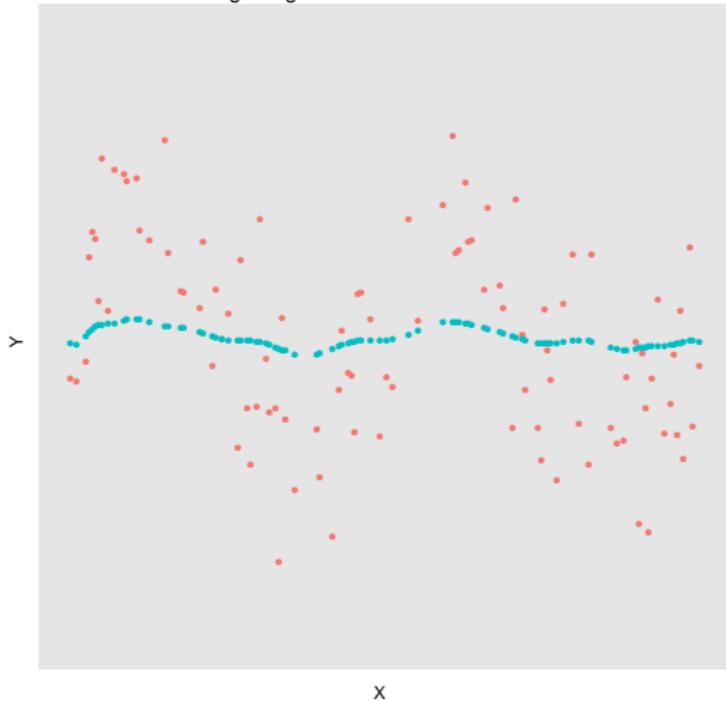
Ridge Regression with Lambda = 7.97



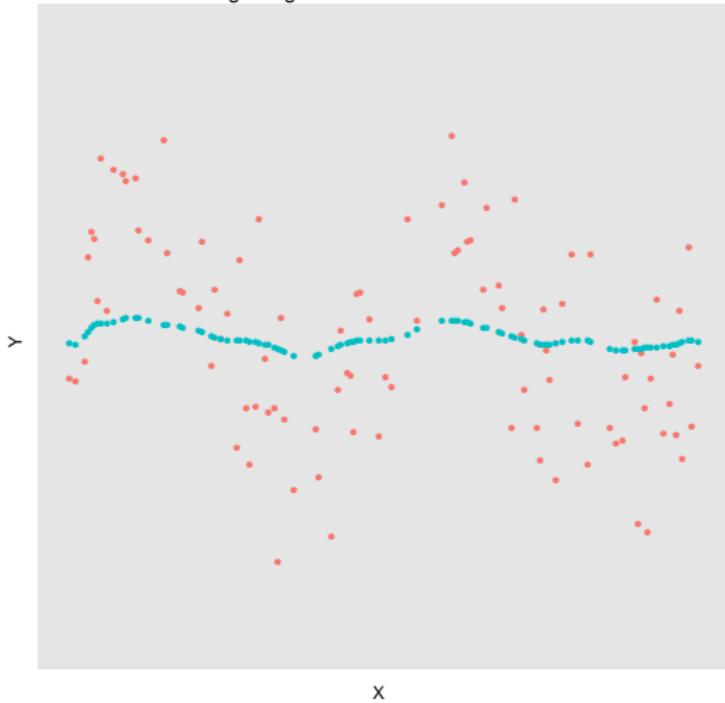
Ridge Regression with Lambda = 7.26



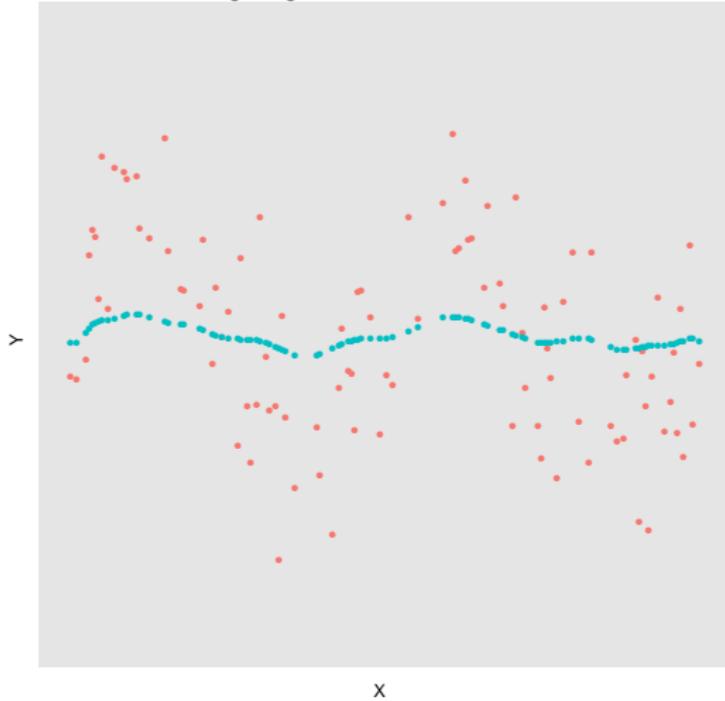
Ridge Regression with Lambda = 6.61



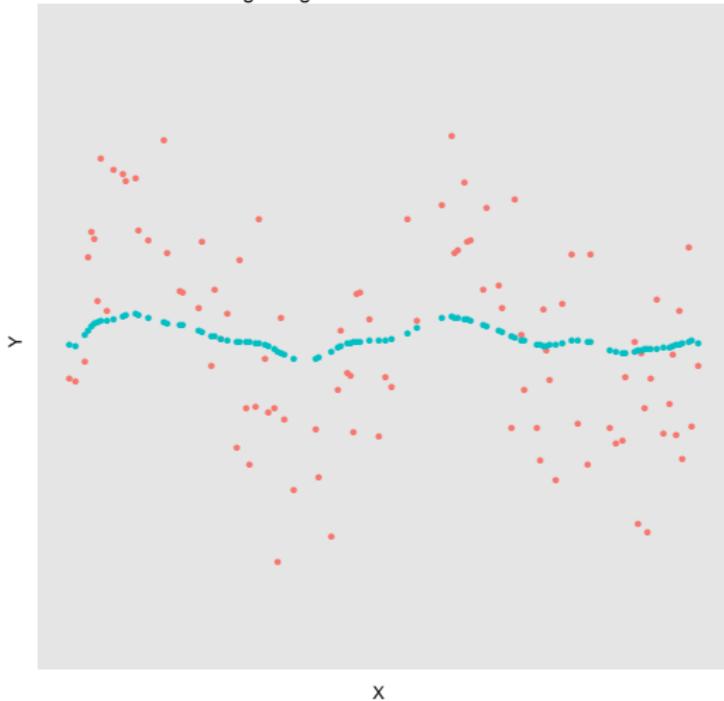
Ridge Regression with Lambda = 6.03



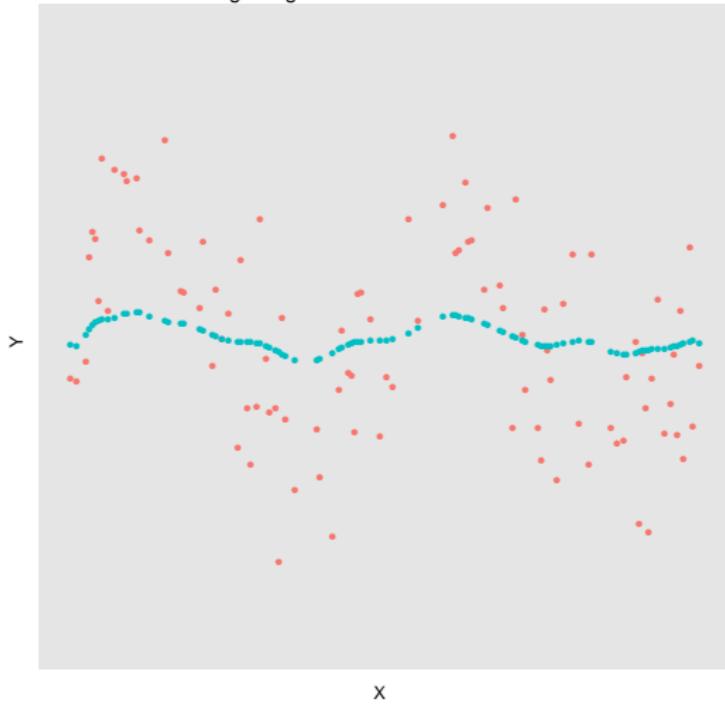
Ridge Regression with Lambda = 5.49



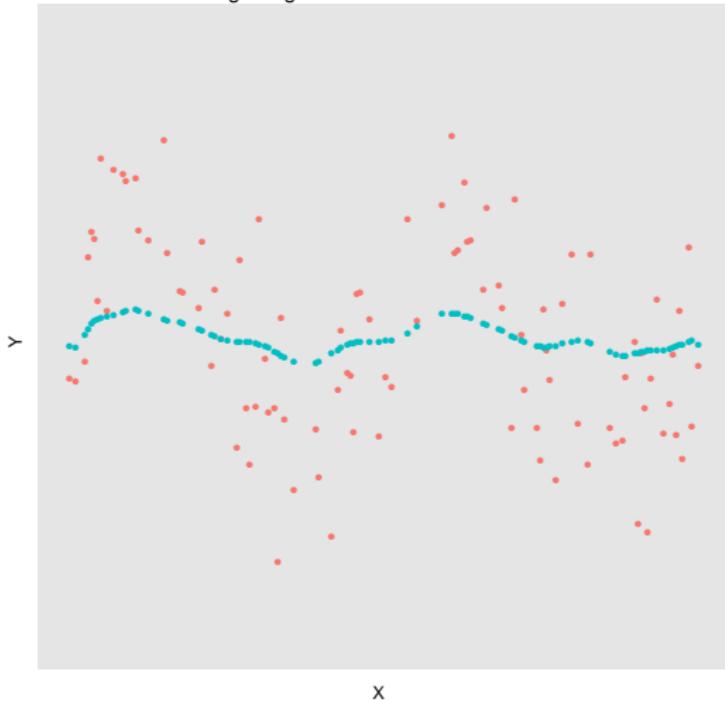
Ridge Regression with Lambda = 5



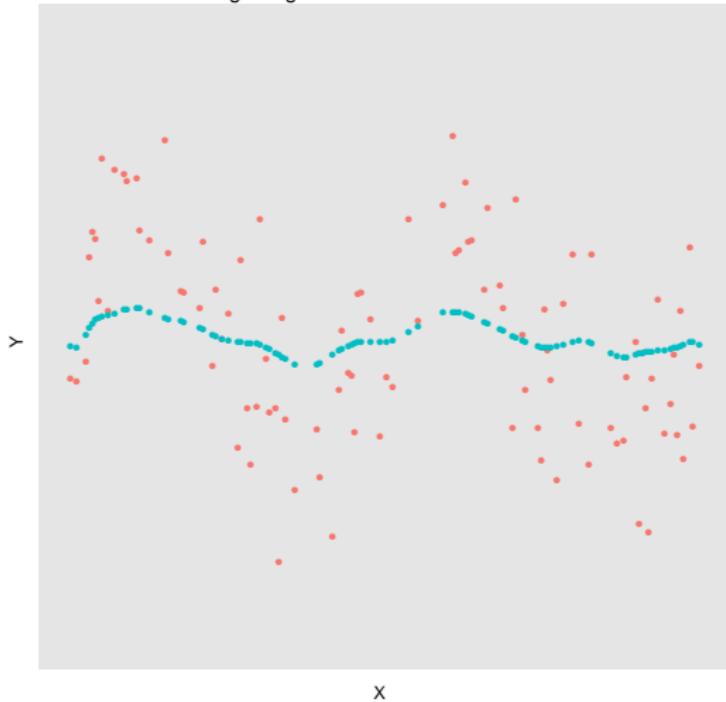
Ridge Regression with Lambda = 4.56



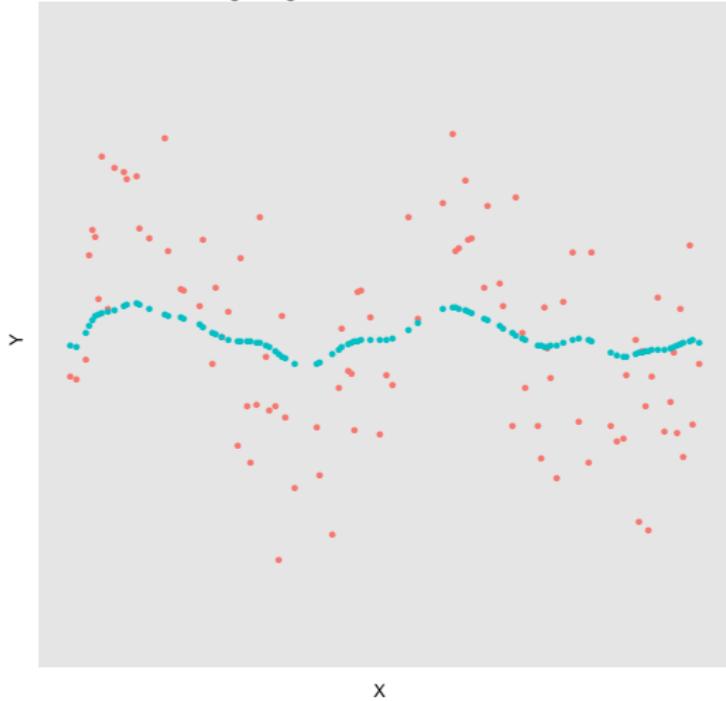
Ridge Regression with Lambda = 4.15



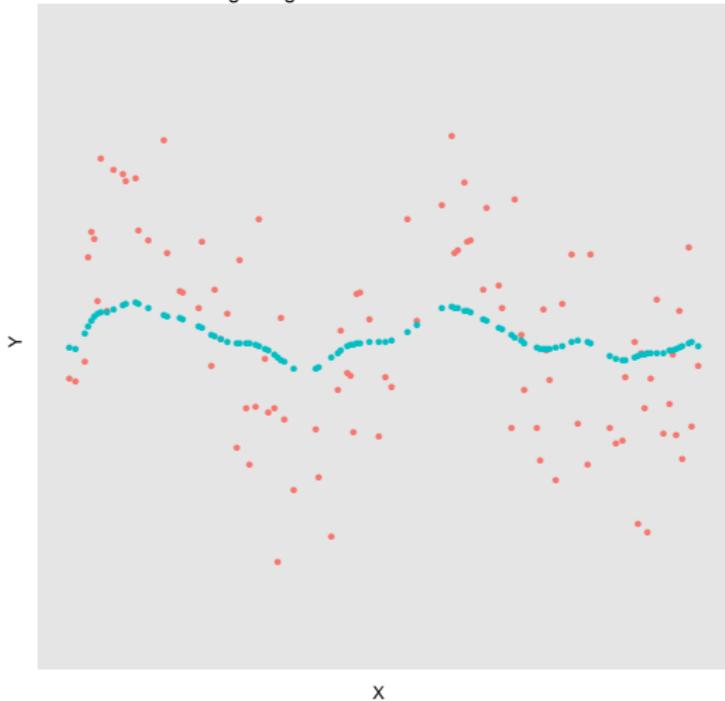
Ridge Regression with Lambda = 3.78



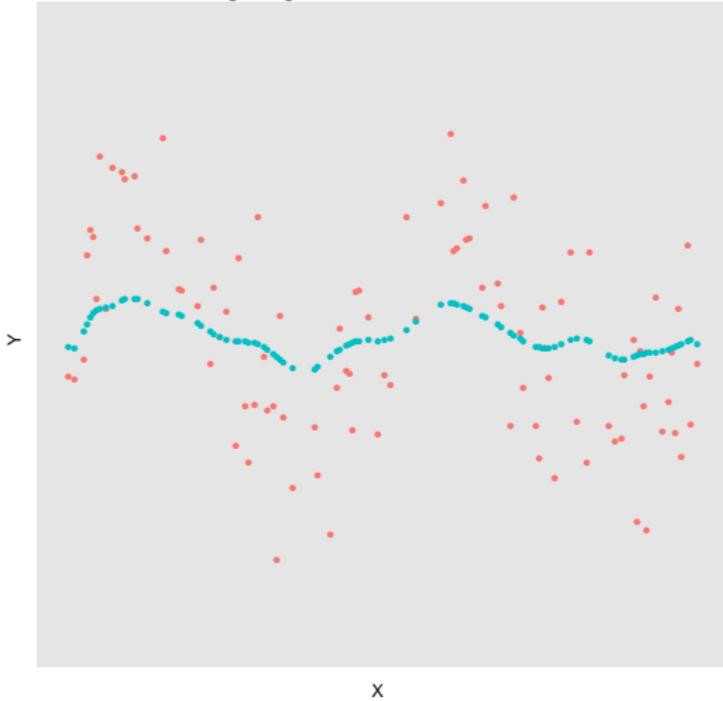
Ridge Regression with Lambda = 3.45



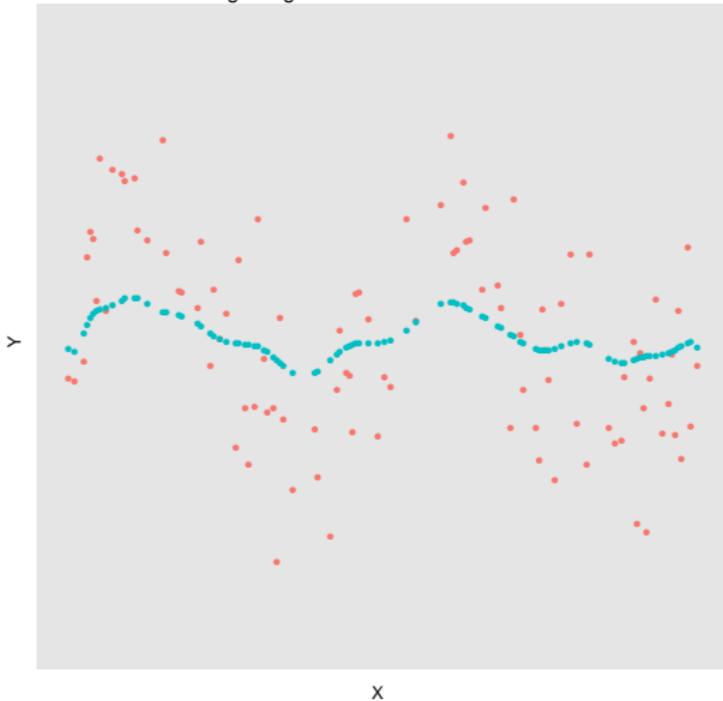
Ridge Regression with Lambda = 3.14



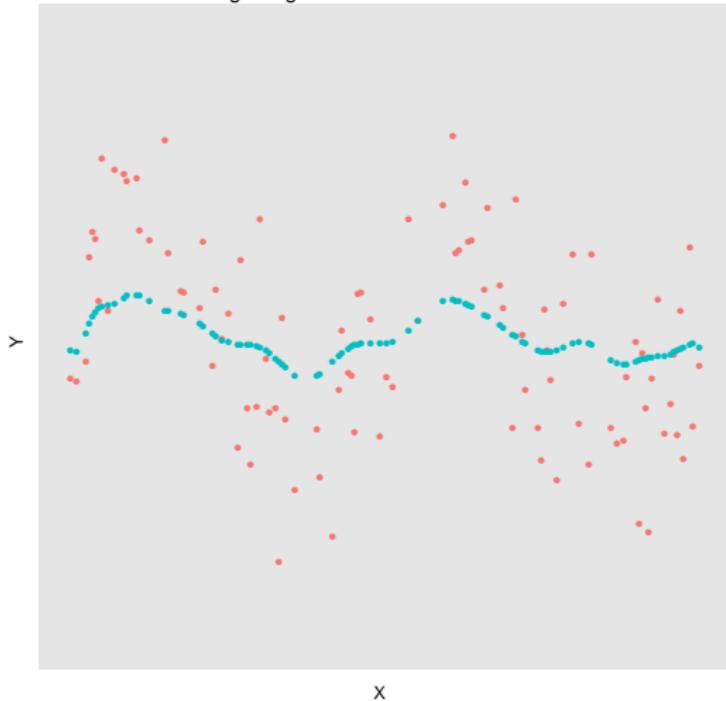
Ridge Regression with Lambda = 2.86



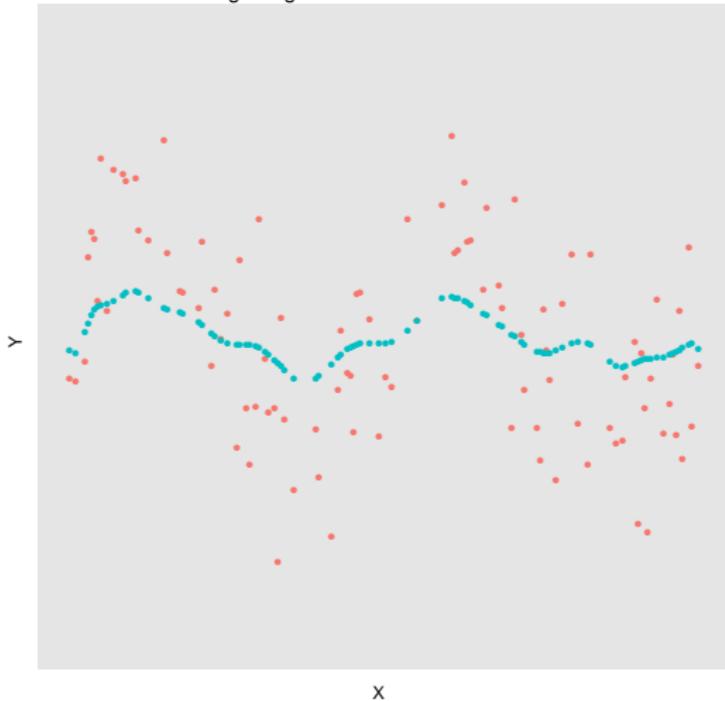
Ridge Regression with Lambda = 2.61



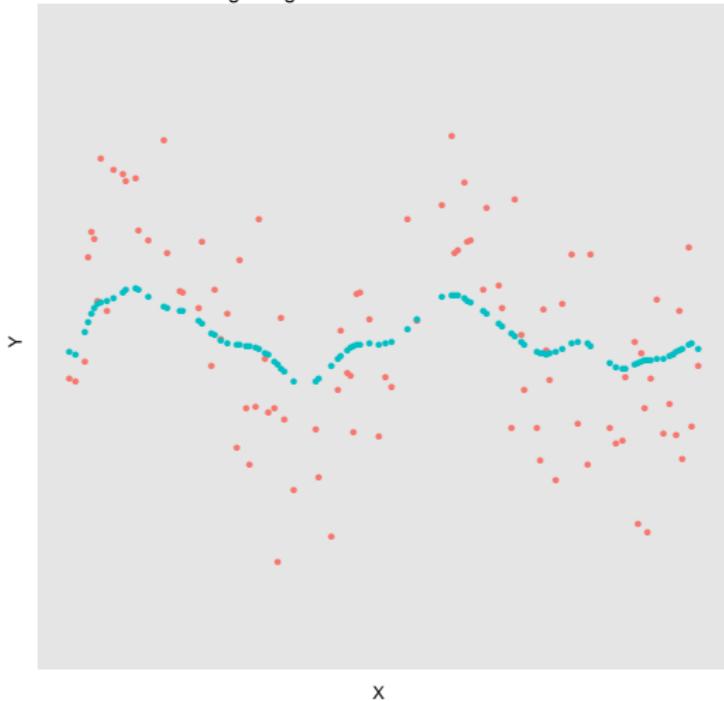
Ridge Regression with Lambda = 2.38



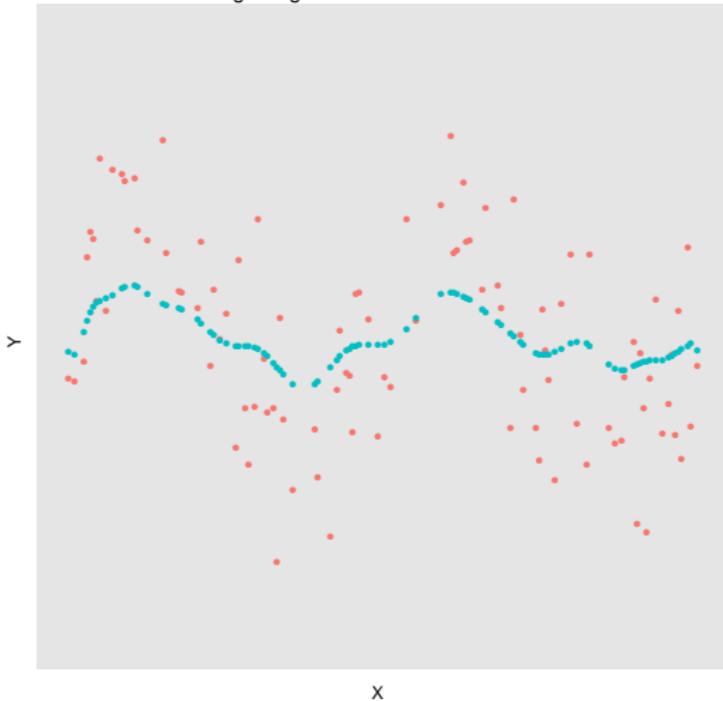
Ridge Regression with Lambda = 2.17



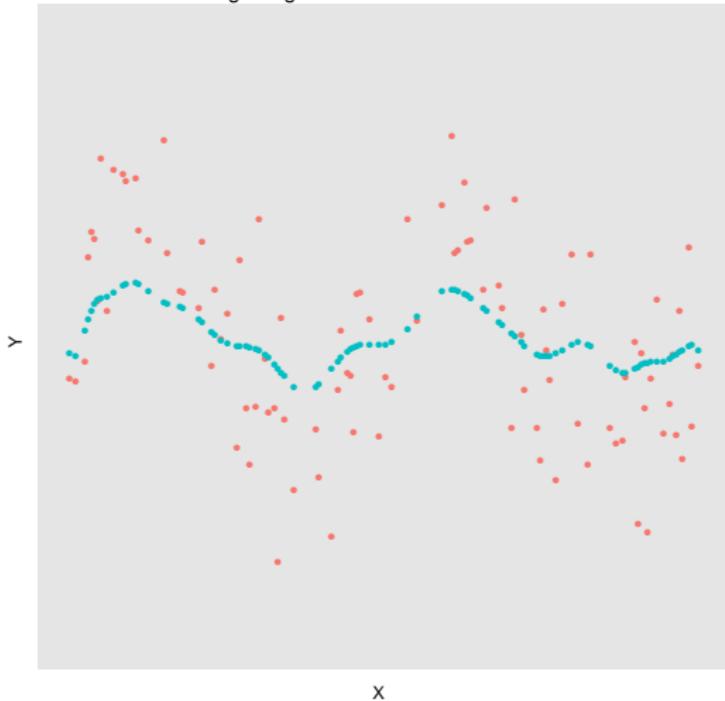
Ridge Regression with Lambda = 1.97



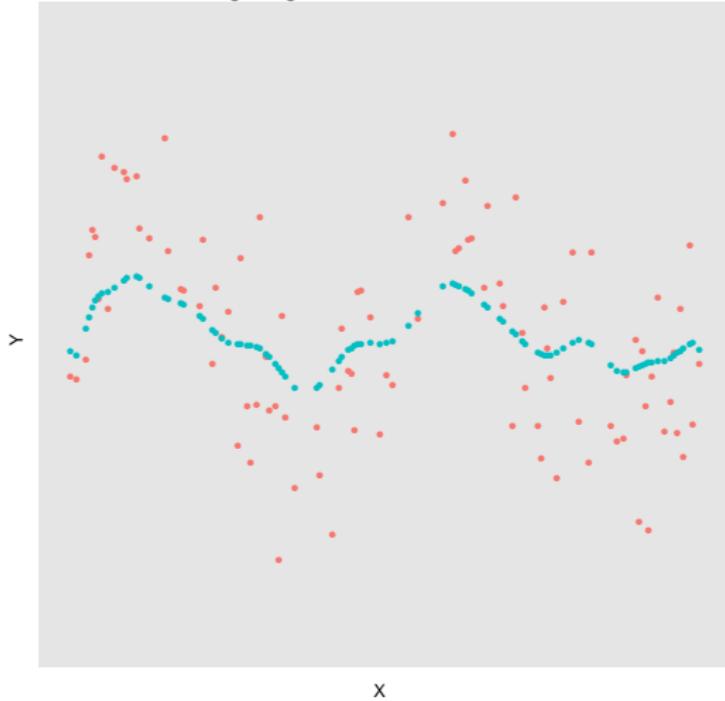
Ridge Regression with Lambda = 1.8



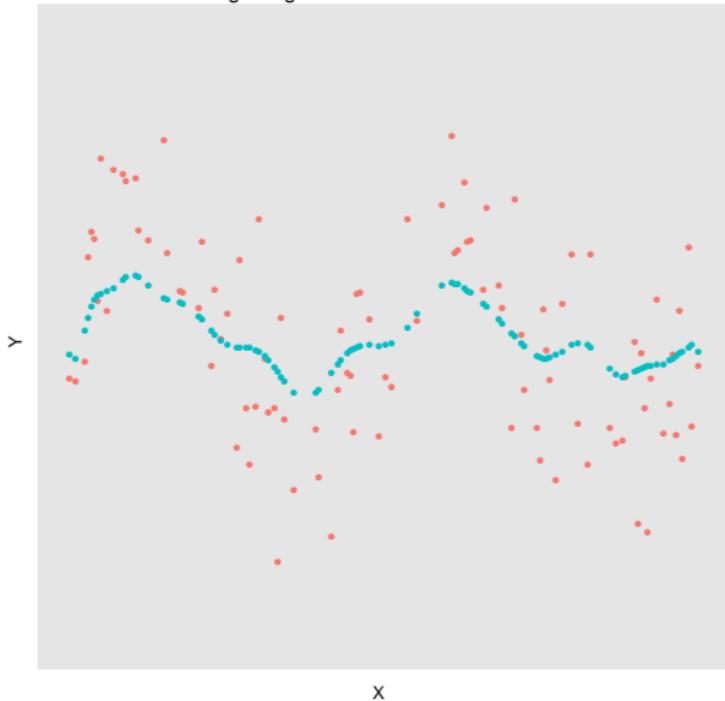
Ridge Regression with Lambda = 1.64



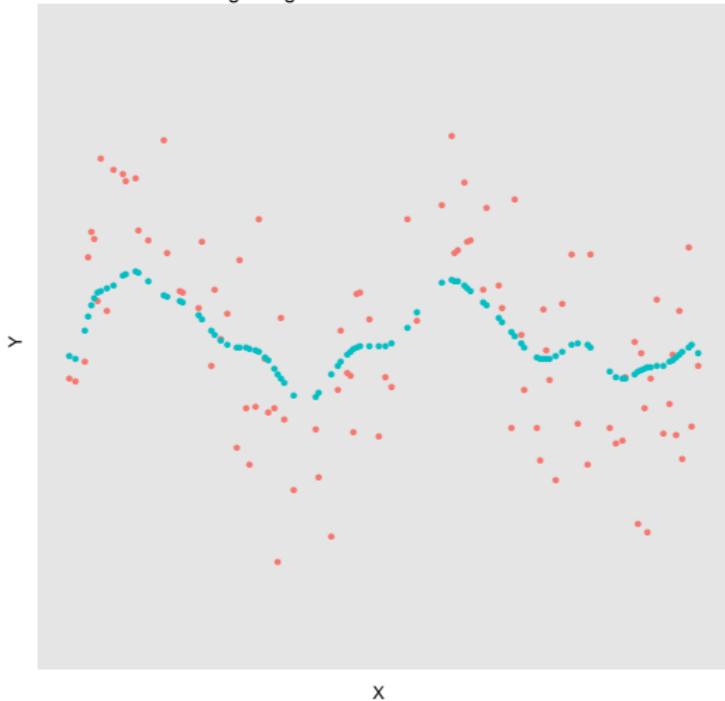
Ridge Regression with Lambda = 1.49



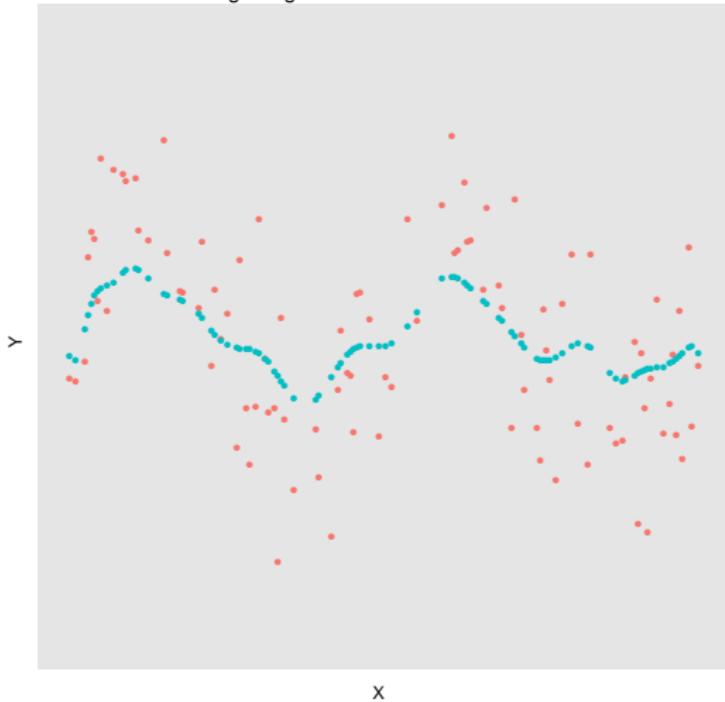
Ridge Regression with Lambda = 1.36



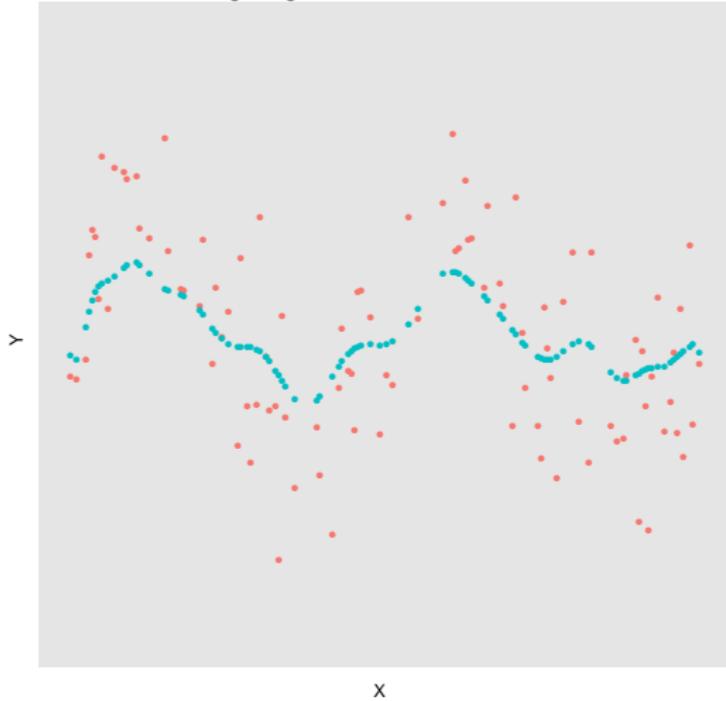
Ridge Regression with Lambda = 1.24



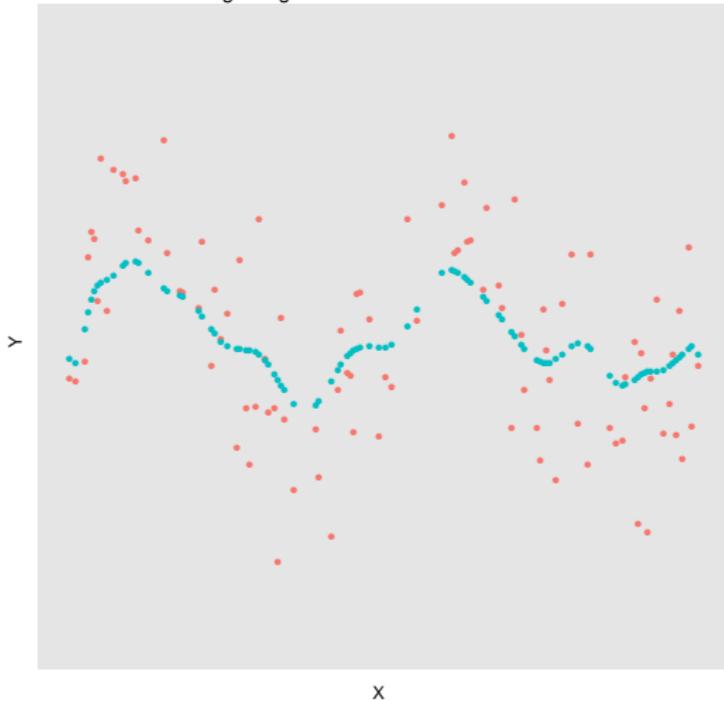
Ridge Regression with Lambda = 1.13



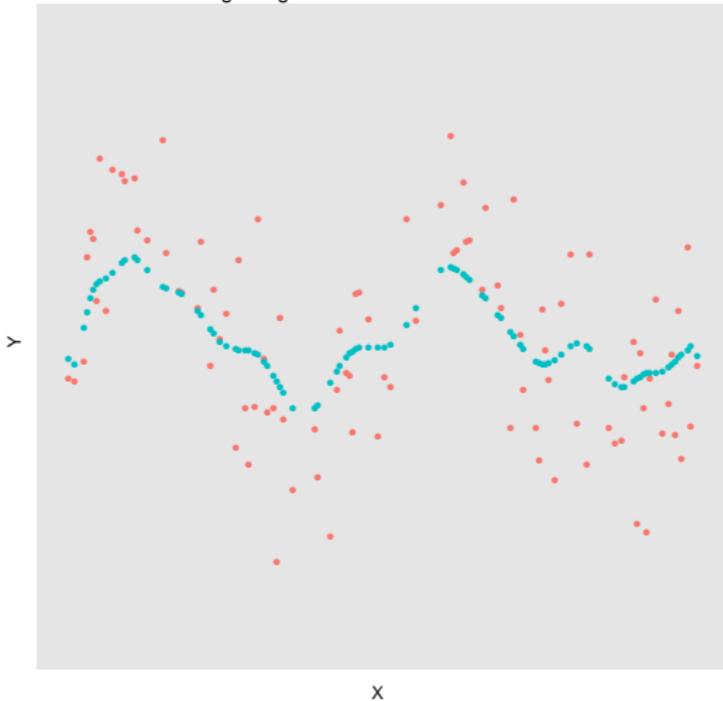
Ridge Regression with Lambda = 1.03



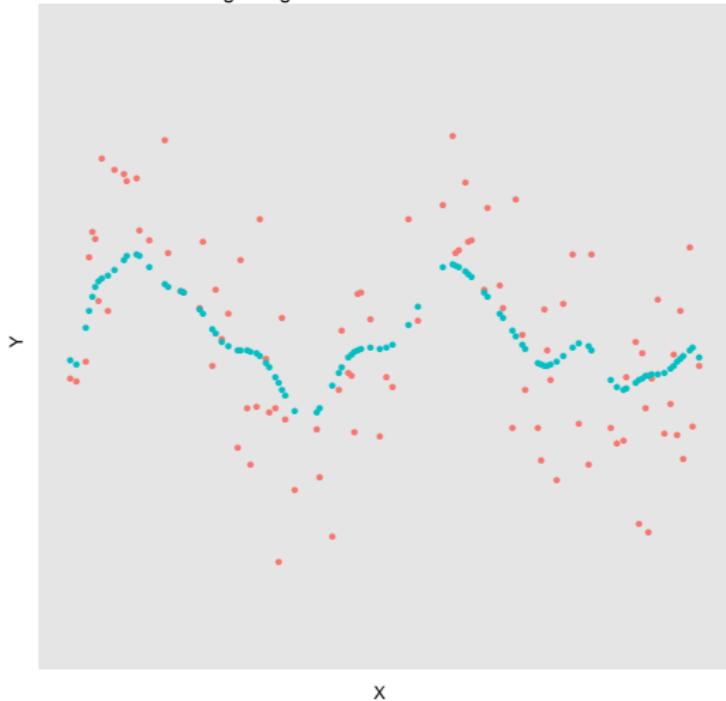
Ridge Regression with Lambda = 0.937



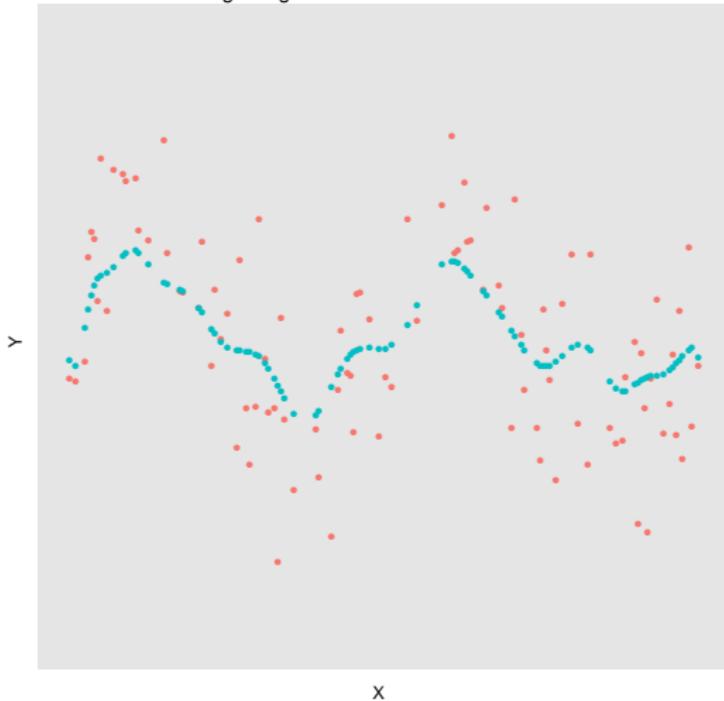
Ridge Regression with Lambda = 0.854



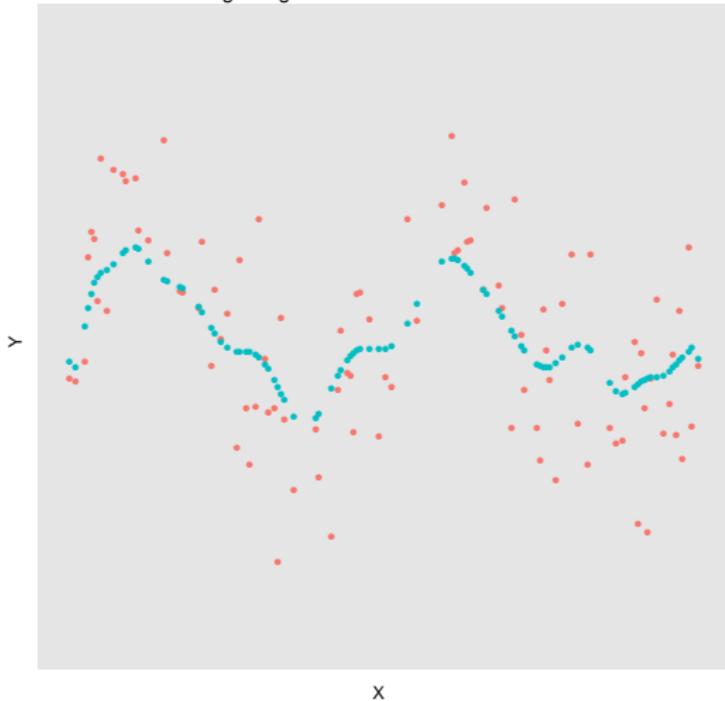
Ridge Regression with Lambda = 0.778



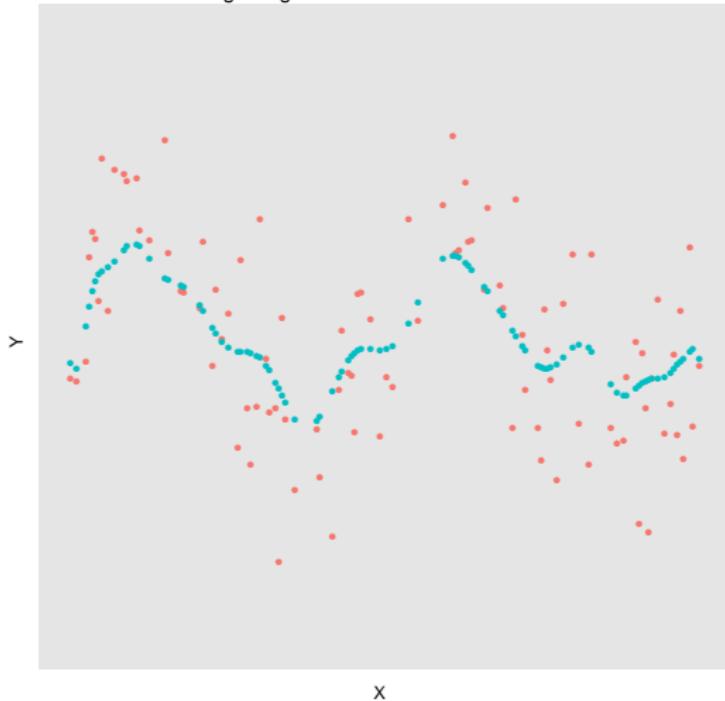
Ridge Regression with Lambda = 0.709



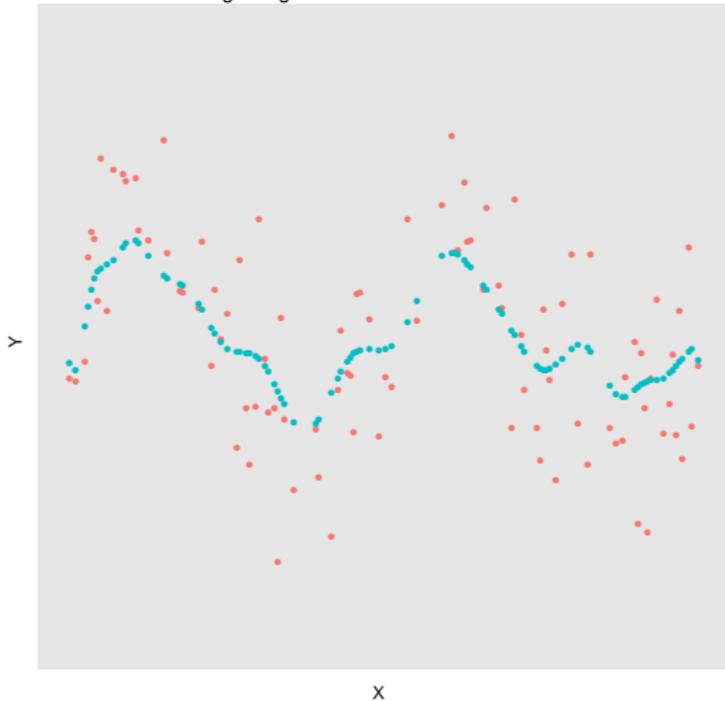
Ridge Regression with Lambda = 0.646



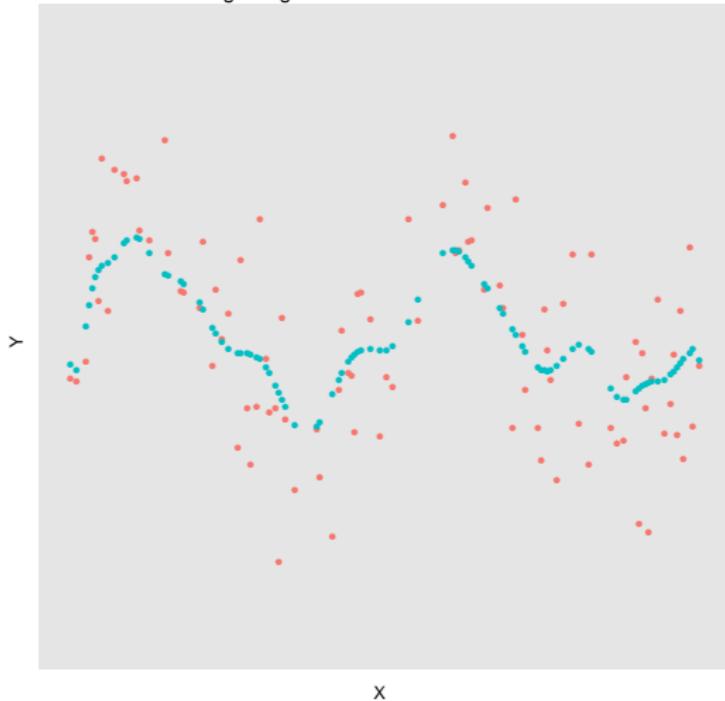
Ridge Regression with Lambda = 0.589



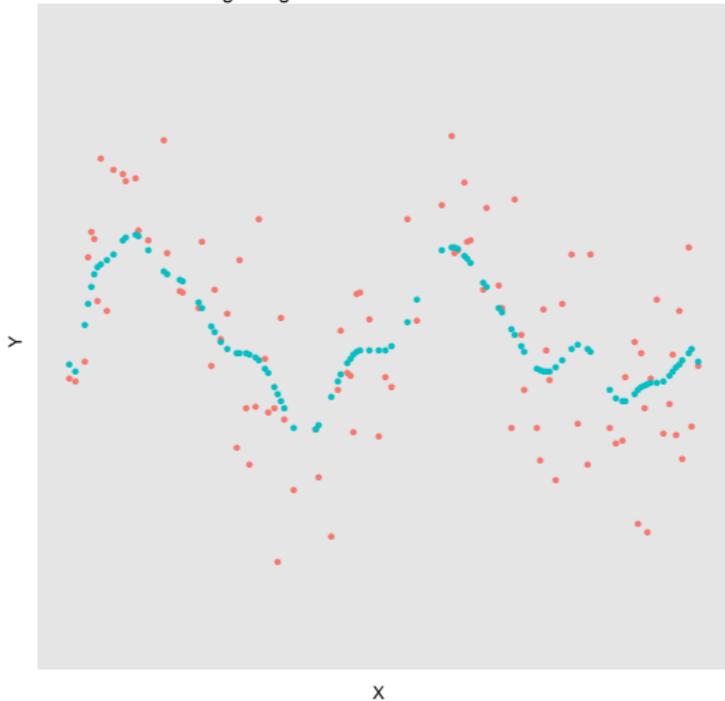
Ridge Regression with Lambda = 0.536



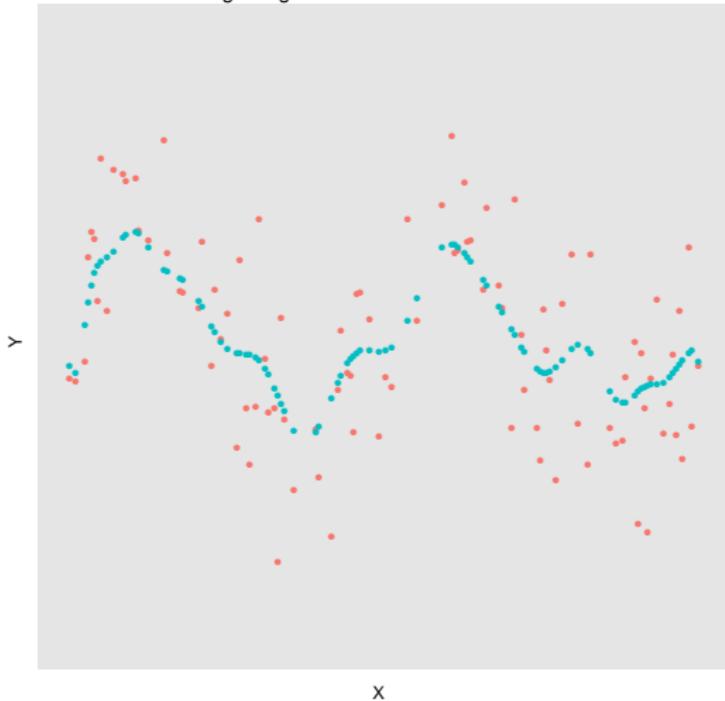
Ridge Regression with Lambda = 0.489



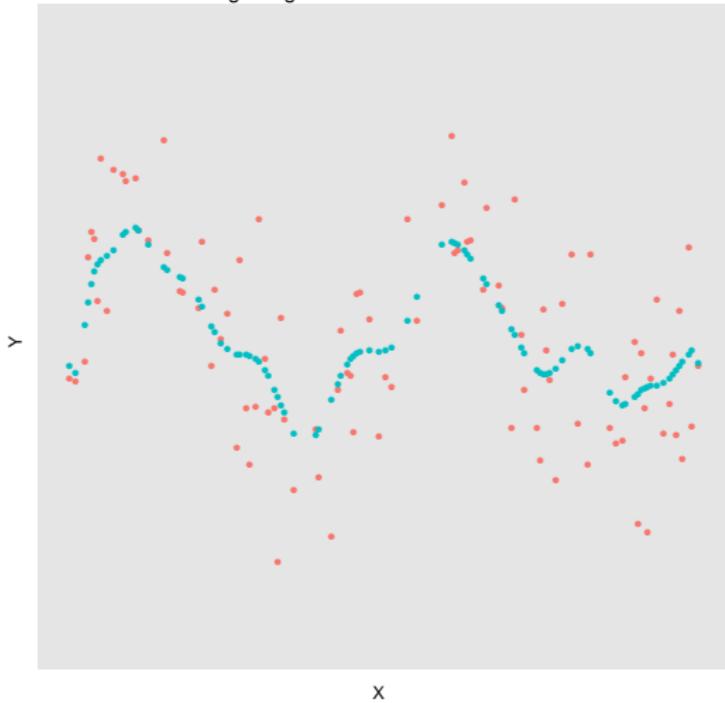
Ridge Regression with Lambda = 0.445



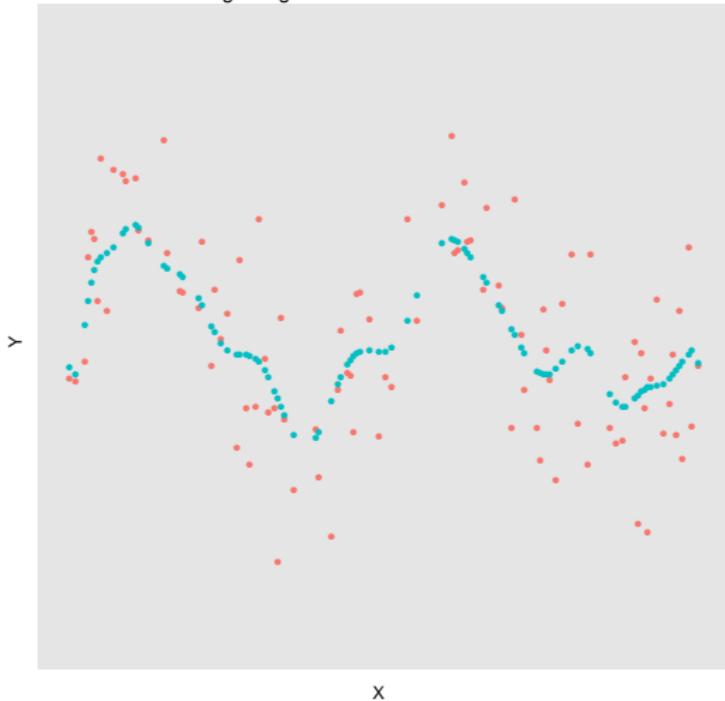
Ridge Regression with Lambda = 0.406



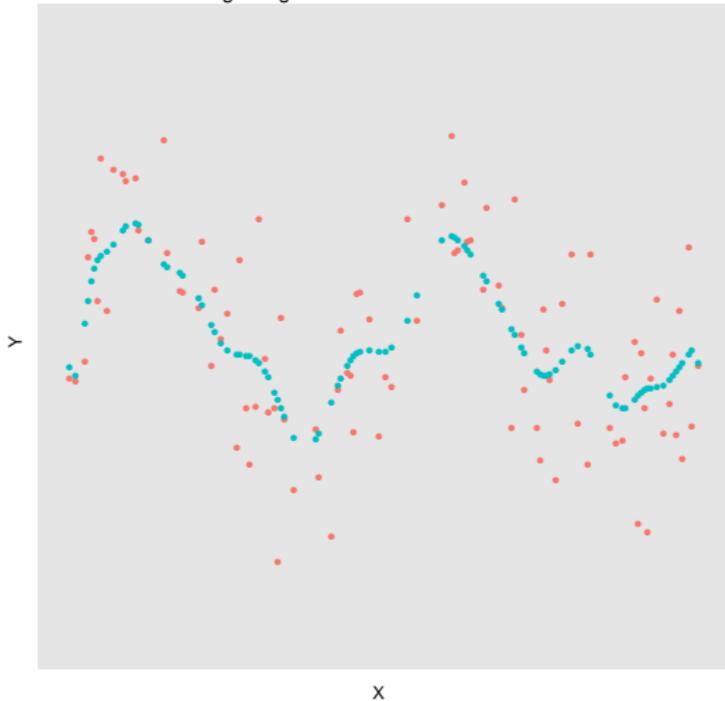
Ridge Regression with Lambda = 0.37



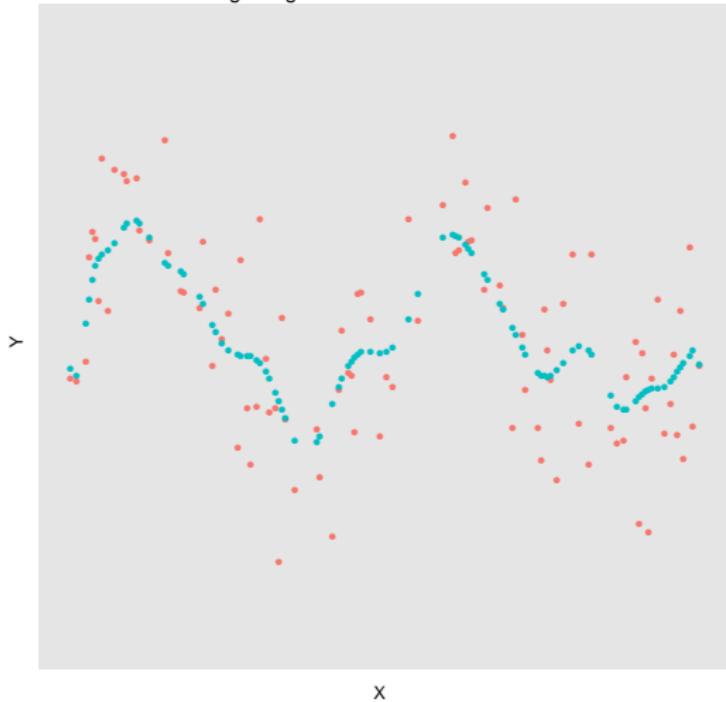
Ridge Regression with Lambda = 0.337



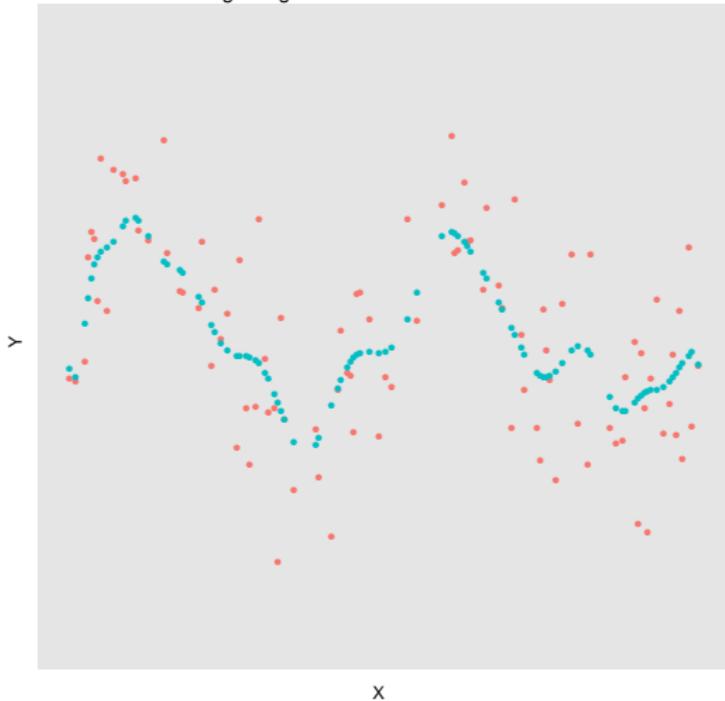
Ridge Regression with Lambda = 0.307



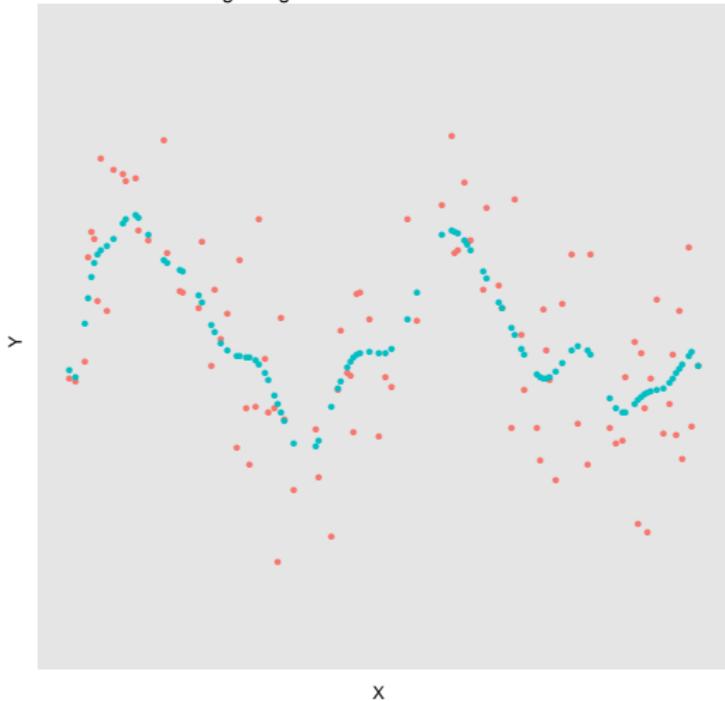
Ridge Regression with Lambda = 0.28



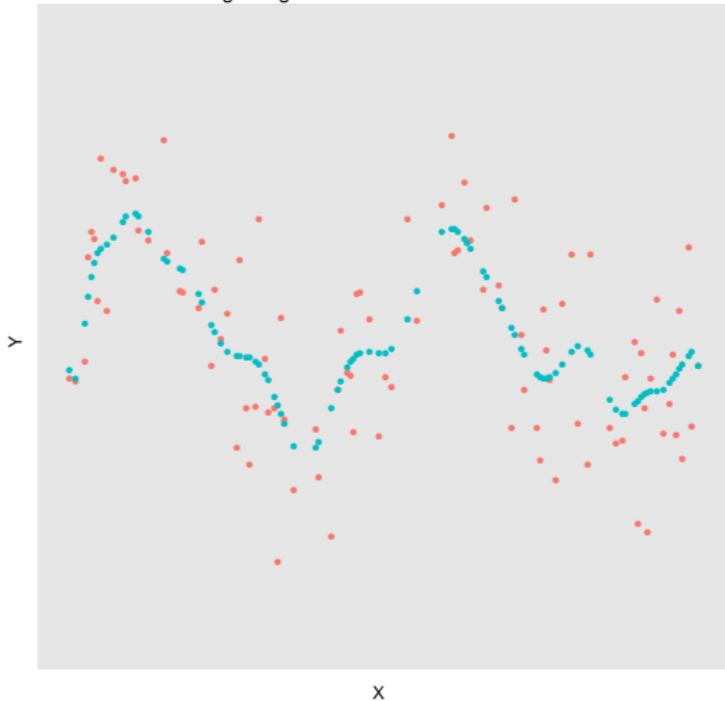
Ridge Regression with Lambda = 0.255



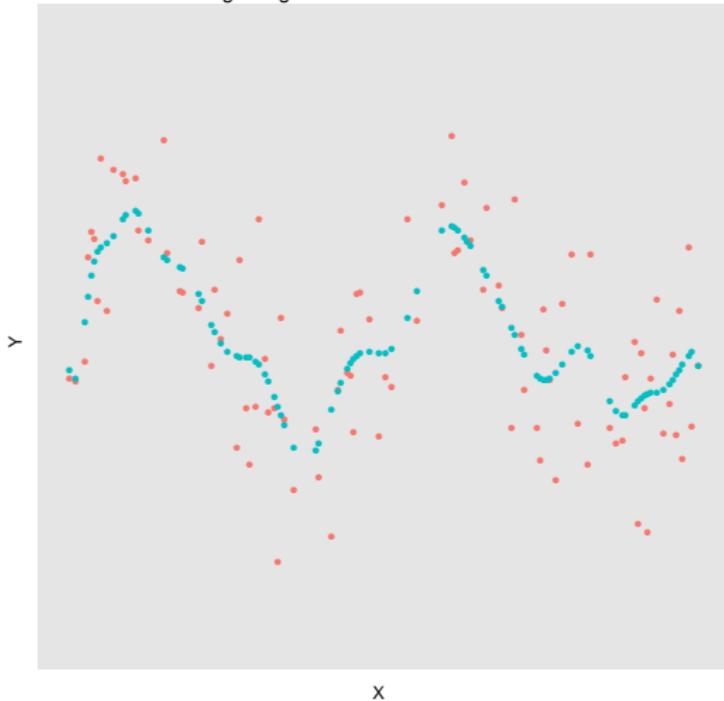
Ridge Regression with Lambda = 0.232



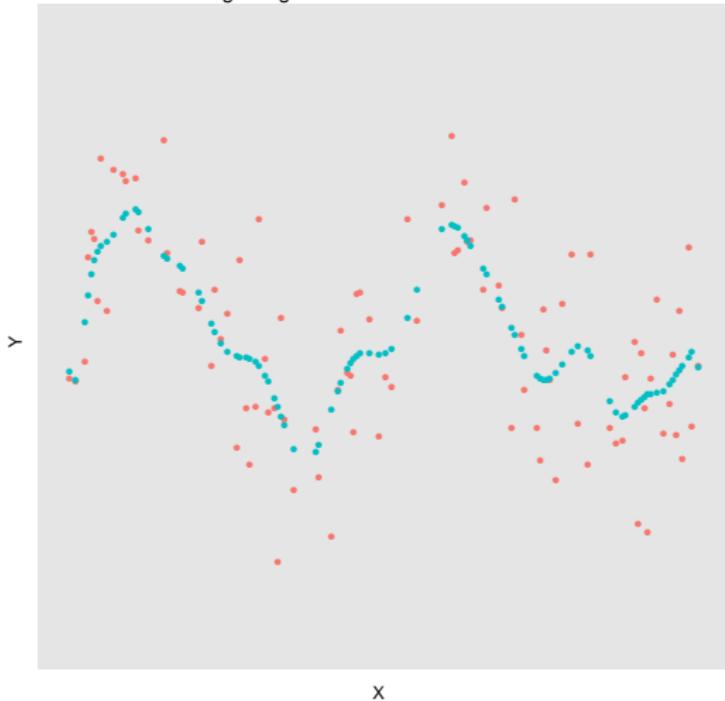
Ridge Regression with Lambda = 0.212



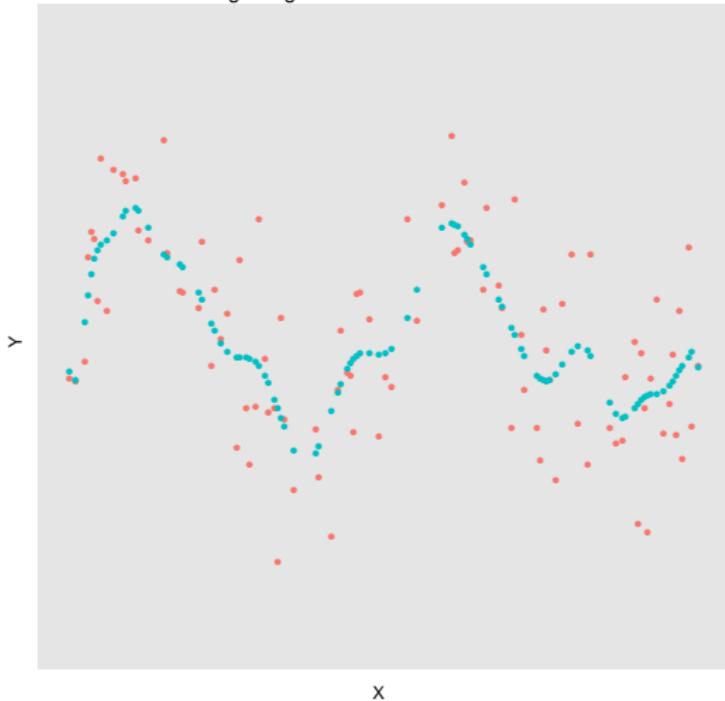
Ridge Regression with Lambda = 0.193



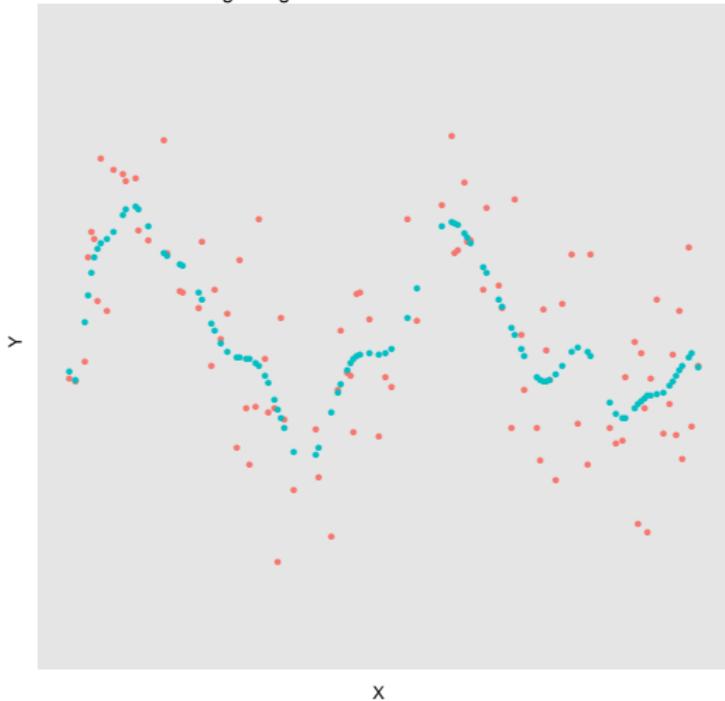
Ridge Regression with Lambda = 0.176



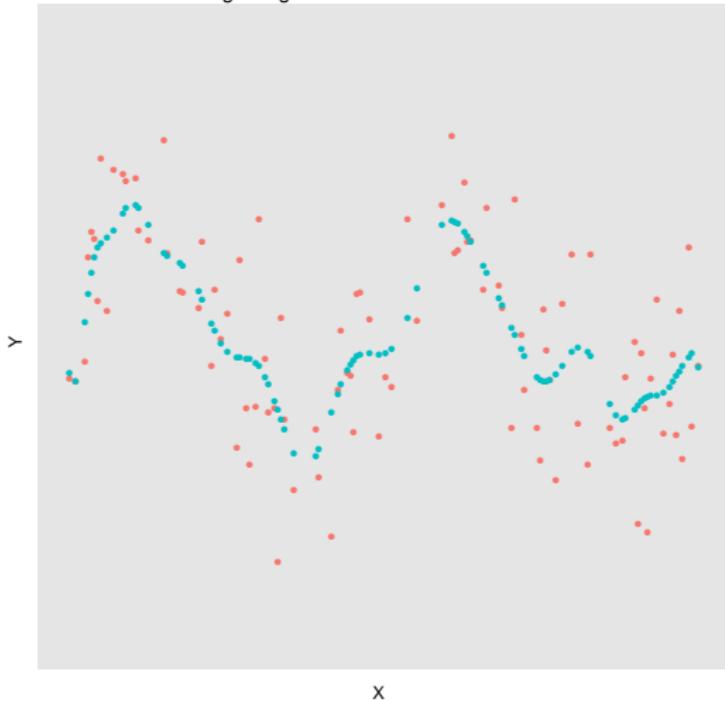
Ridge Regression with Lambda = 0.16



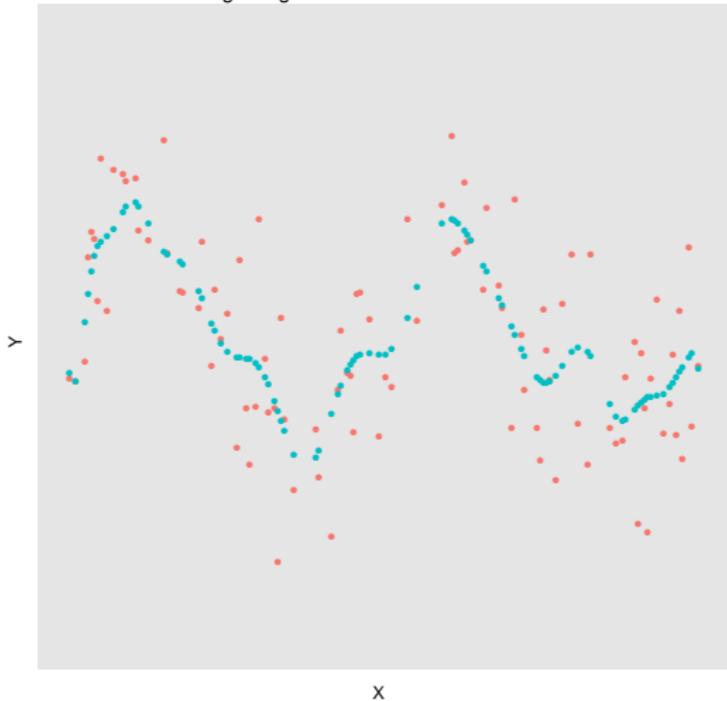
Ridge Regression with Lambda = 0.146



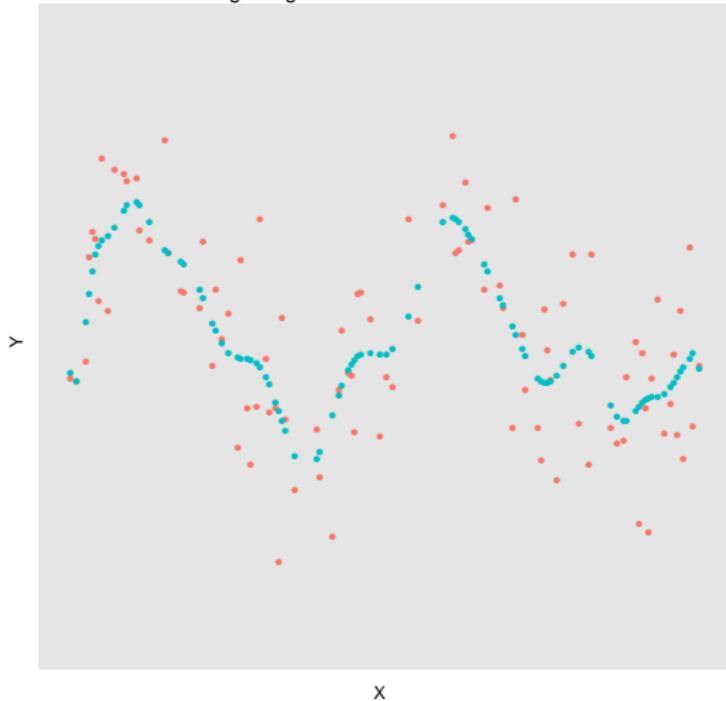
Ridge Regression with Lambda = 0.133



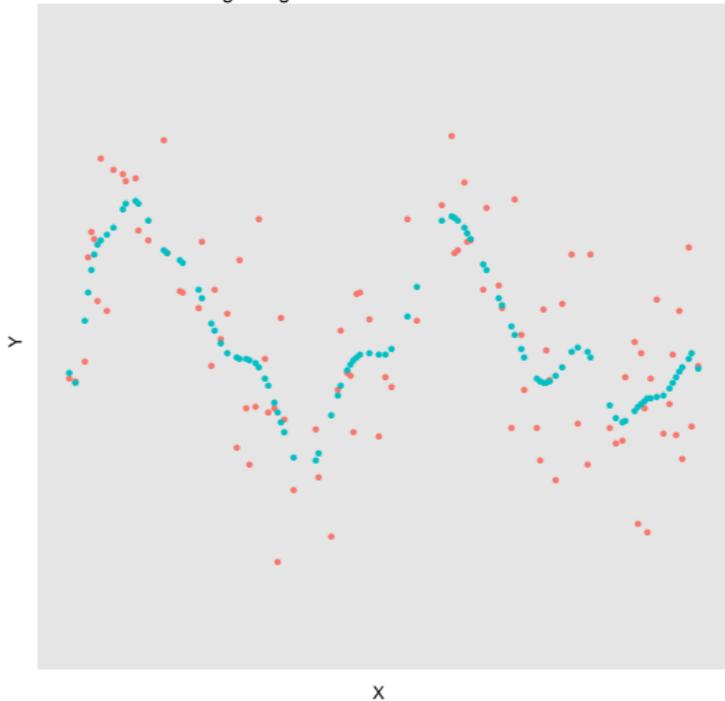
Ridge Regression with Lambda = 0.121



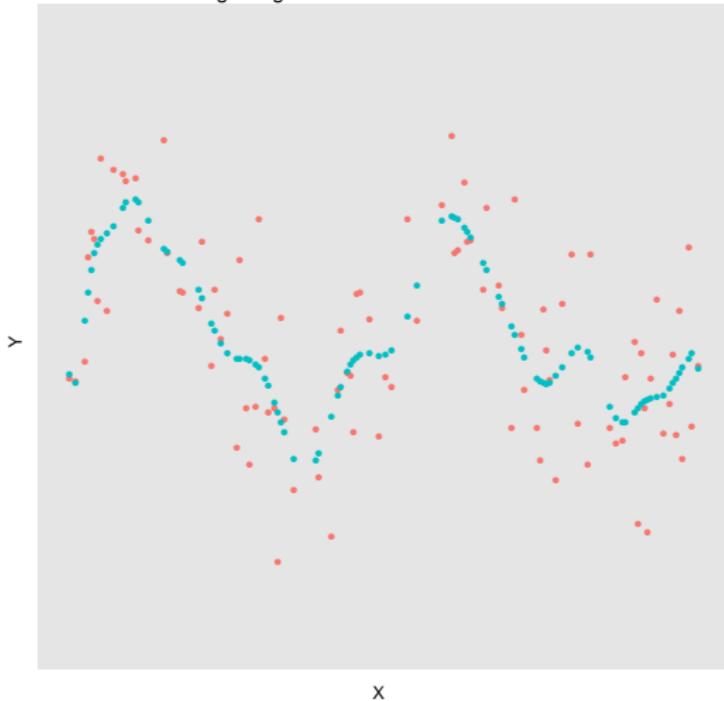
Ridge Regression with Lambda = 0.11



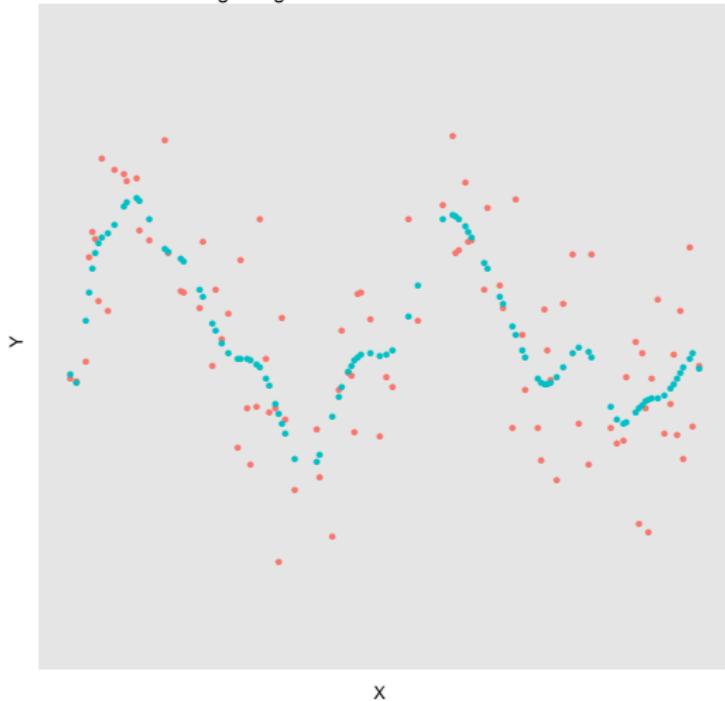
Ridge Regression with Lambda = 0.101



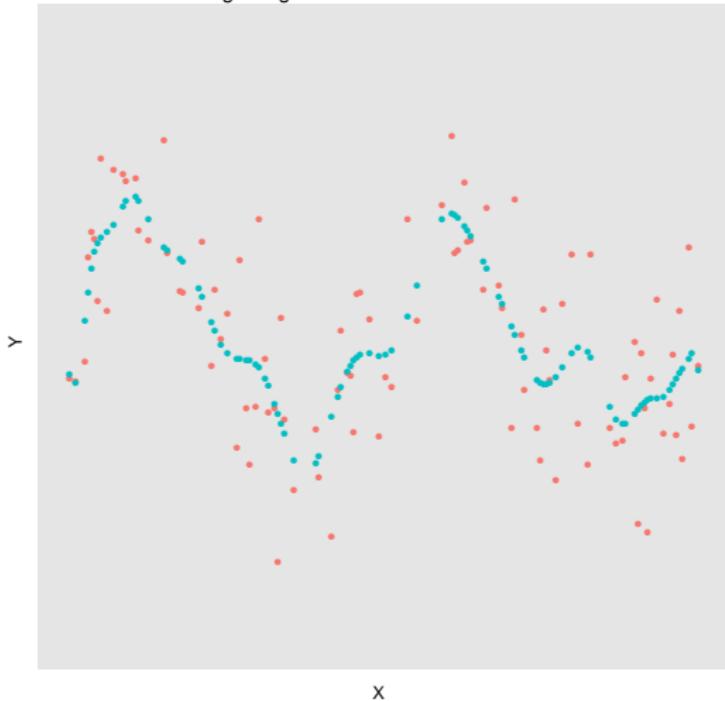
Ridge Regression with Lambda = 0.0916



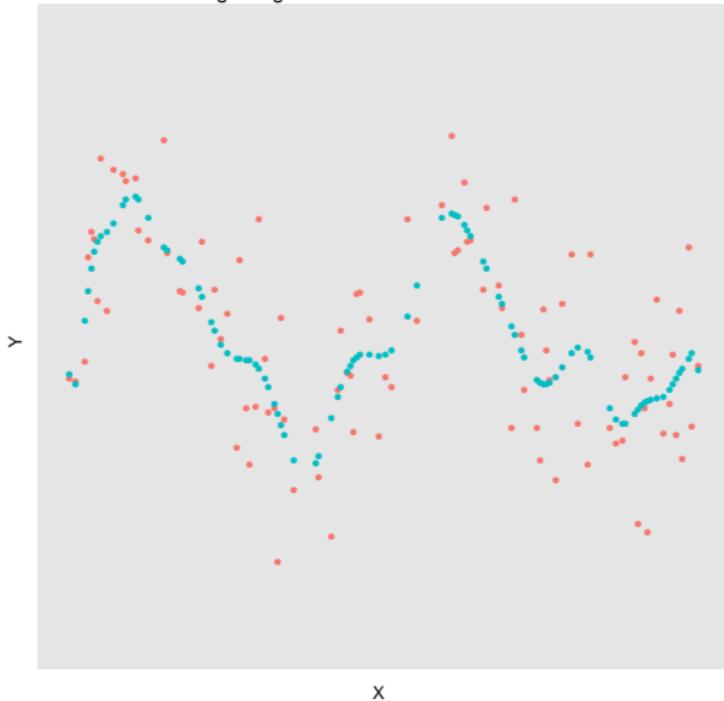
Ridge Regression with Lambda = 0.0835



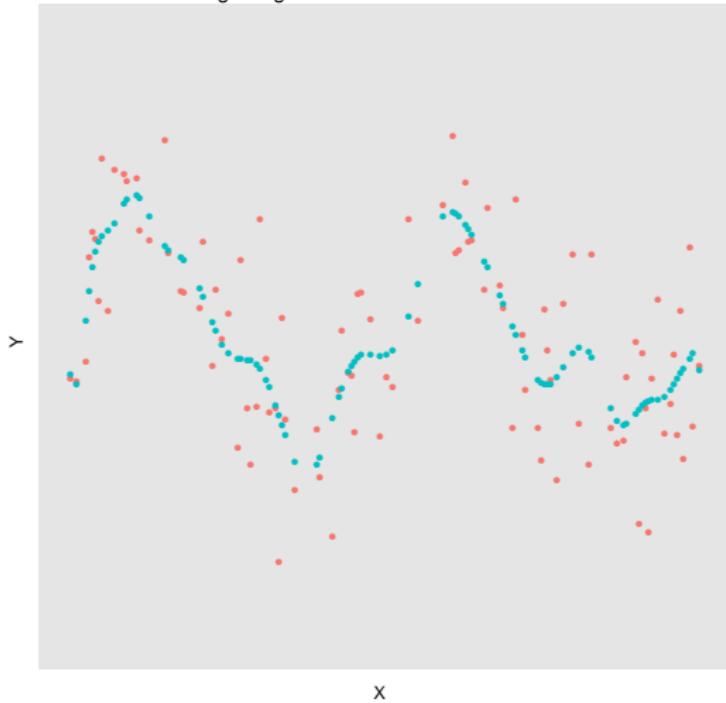
Ridge Regression with Lambda = 0.076



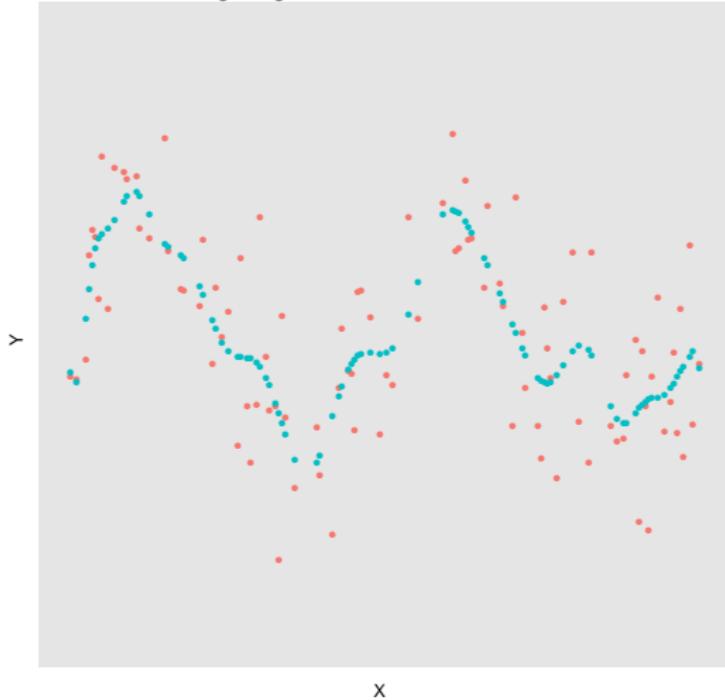
Ridge Regression with Lambda = 0.0693



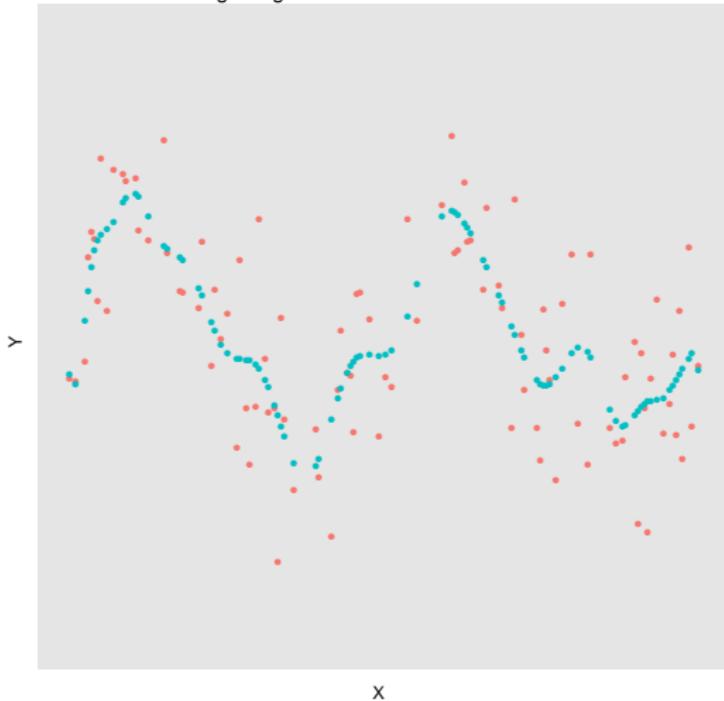
Ridge Regression with Lambda = 0.0631



Ridge Regression with Lambda = 0.0575



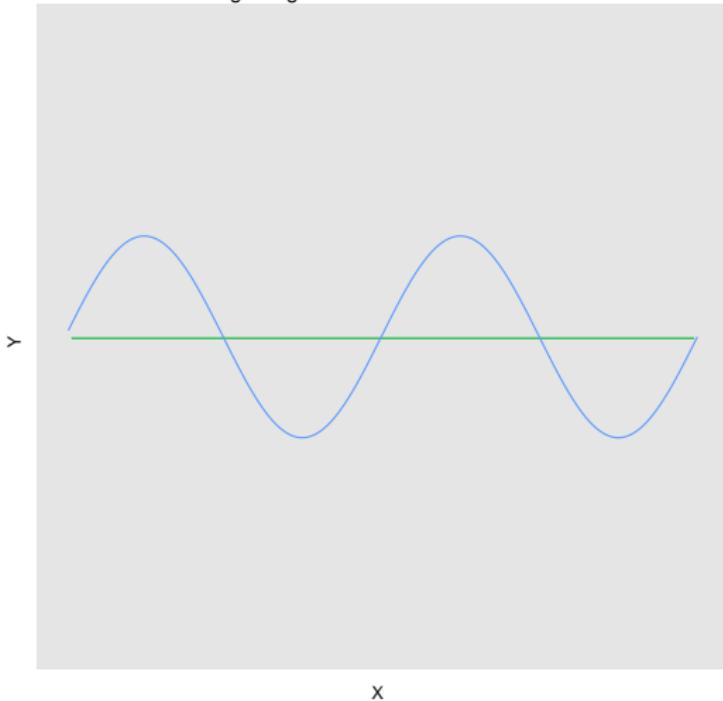
Ridge Regression with Lambda = 0.0524



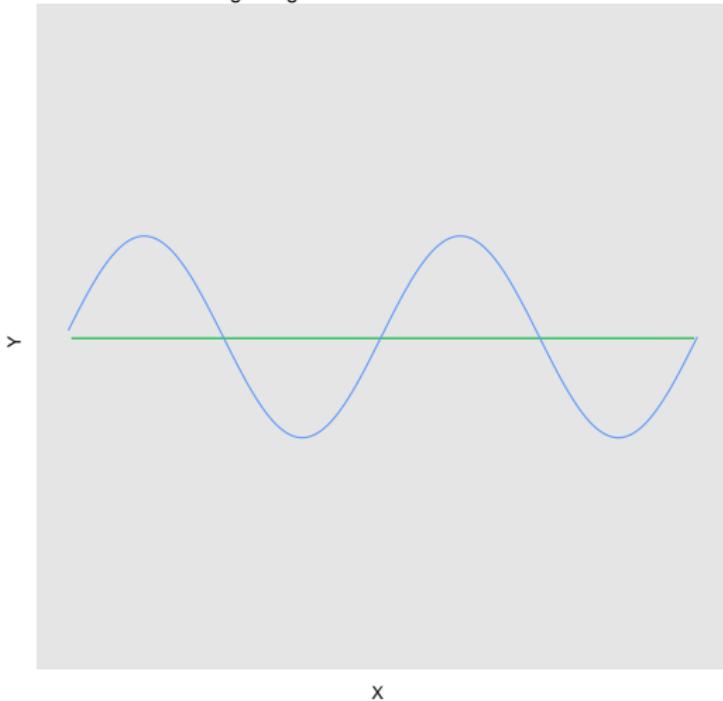
Decreasing λ makes the model more flexible

How does that flexibility relate to finding the underlying pattern?

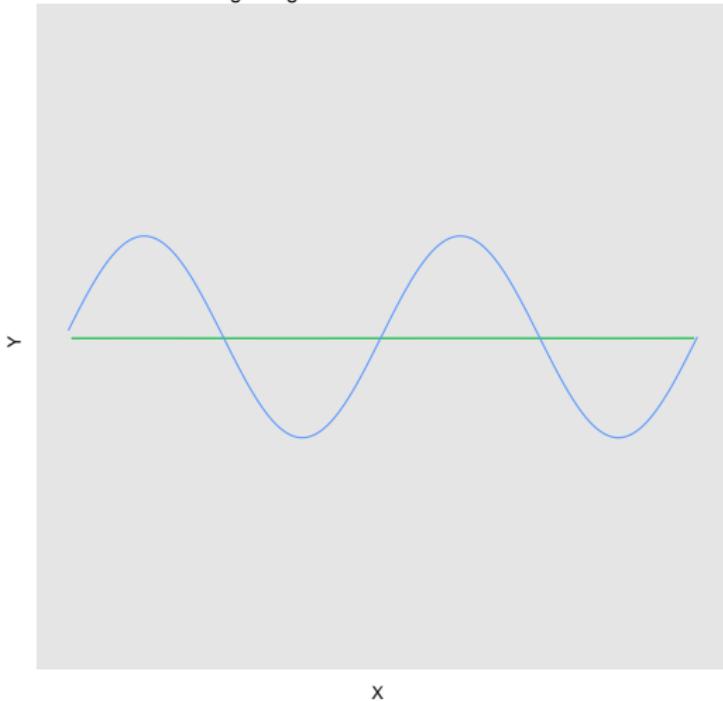
Ridge Regression with Lambda = 524



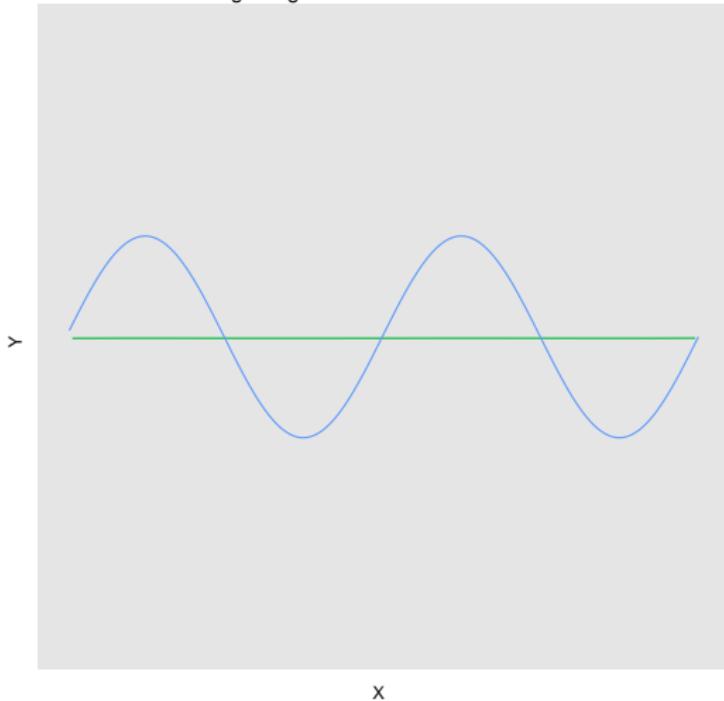
Ridge Regression with Lambda = 478



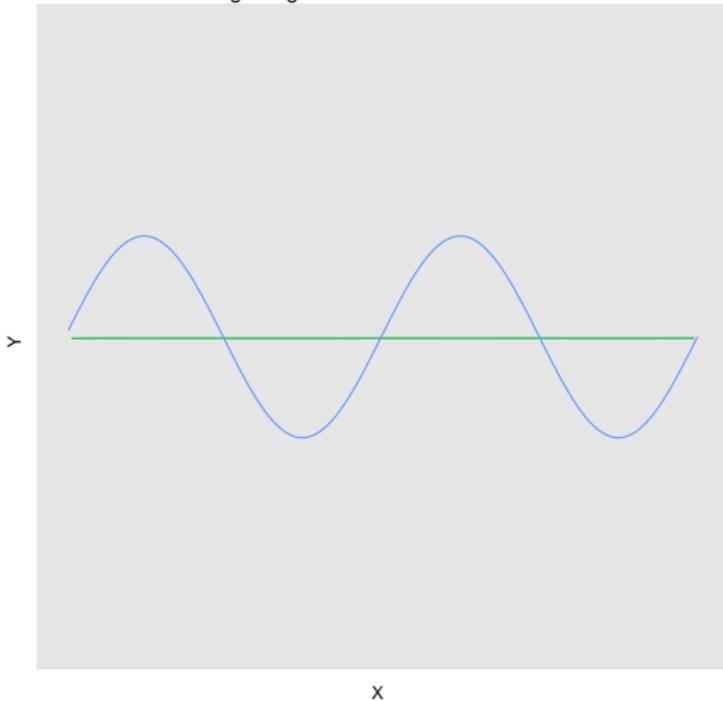
Ridge Regression with Lambda = 435



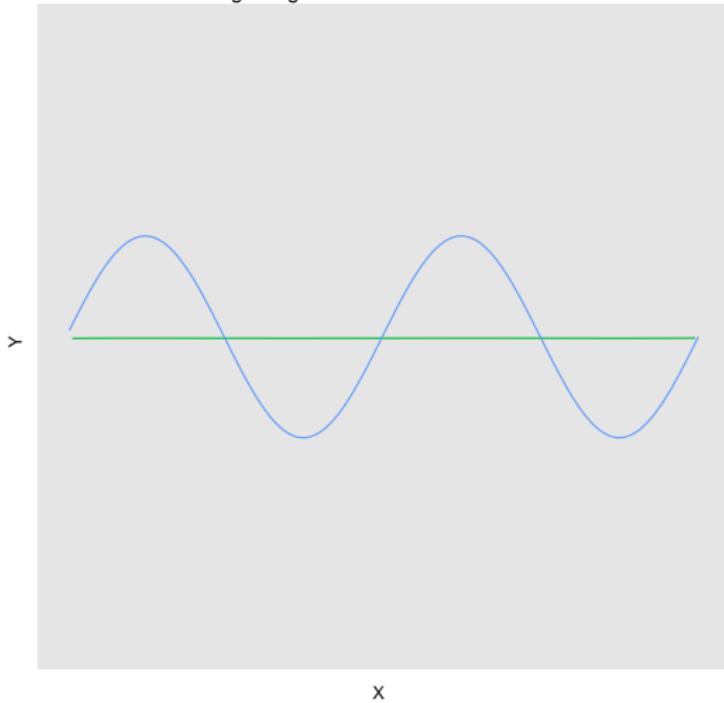
Ridge Regression with Lambda = 396



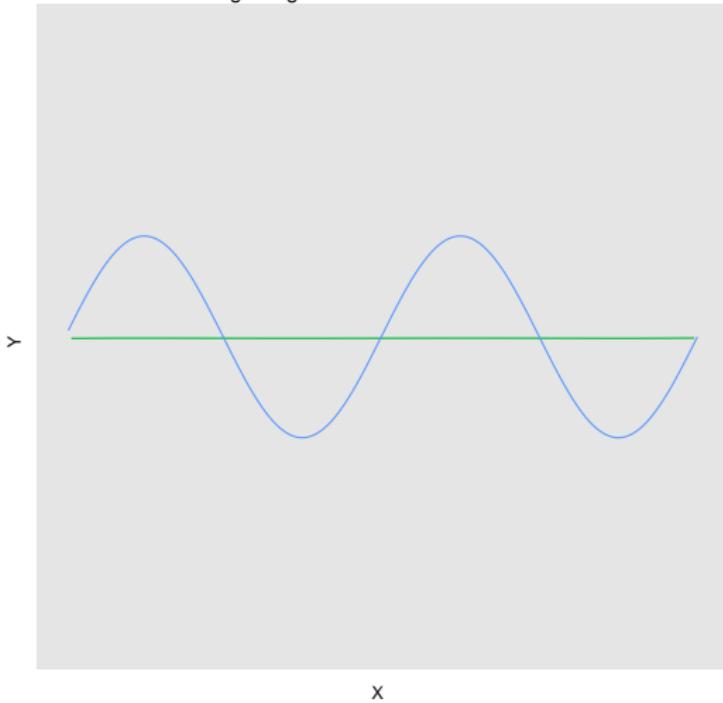
Ridge Regression with Lambda = 361



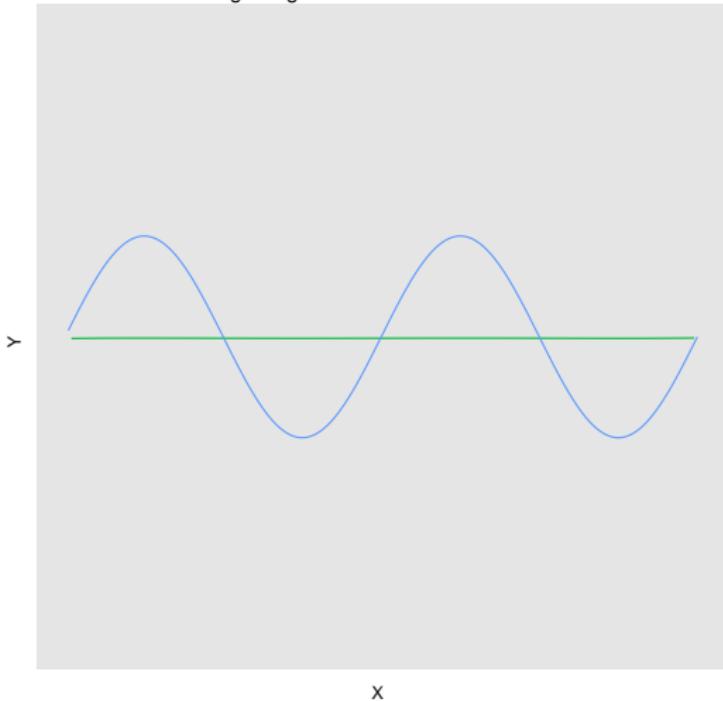
Ridge Regression with Lambda = 329



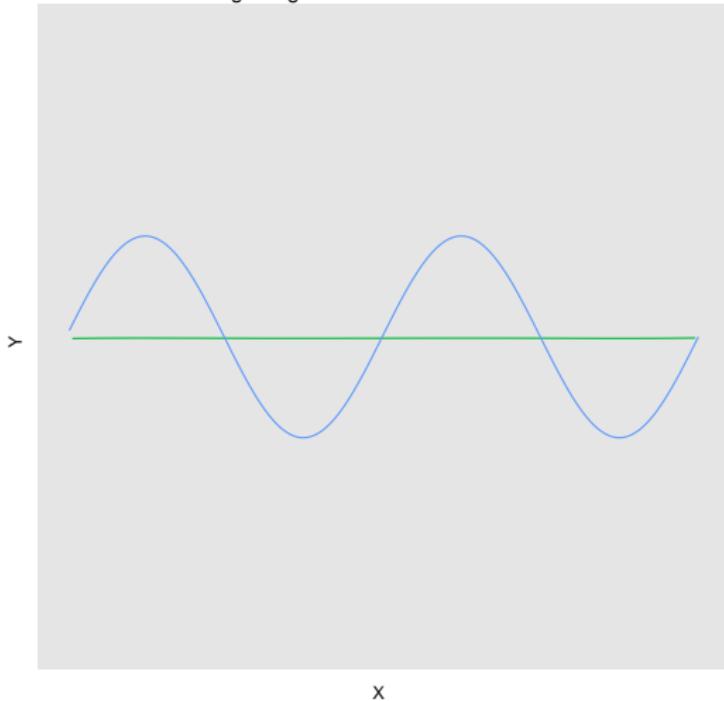
Ridge Regression with Lambda = 300



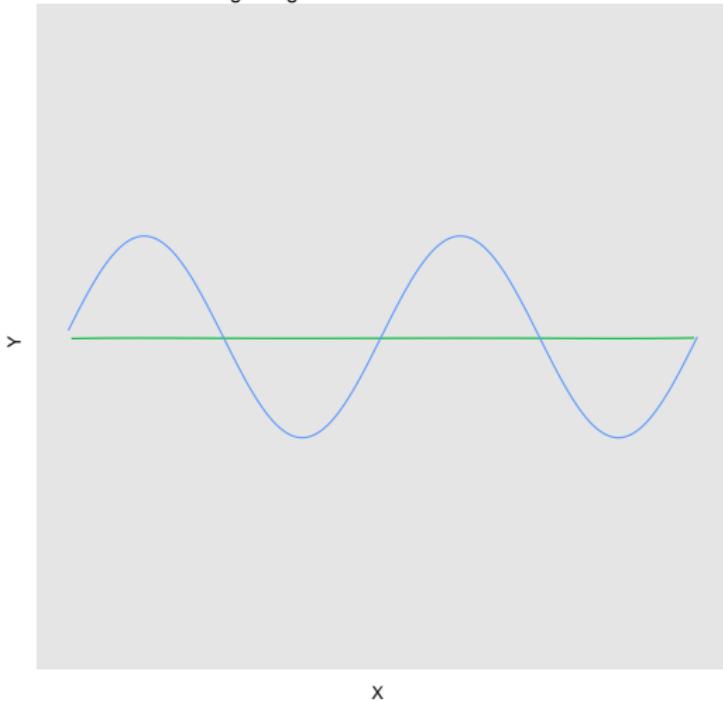
Ridge Regression with Lambda = 273



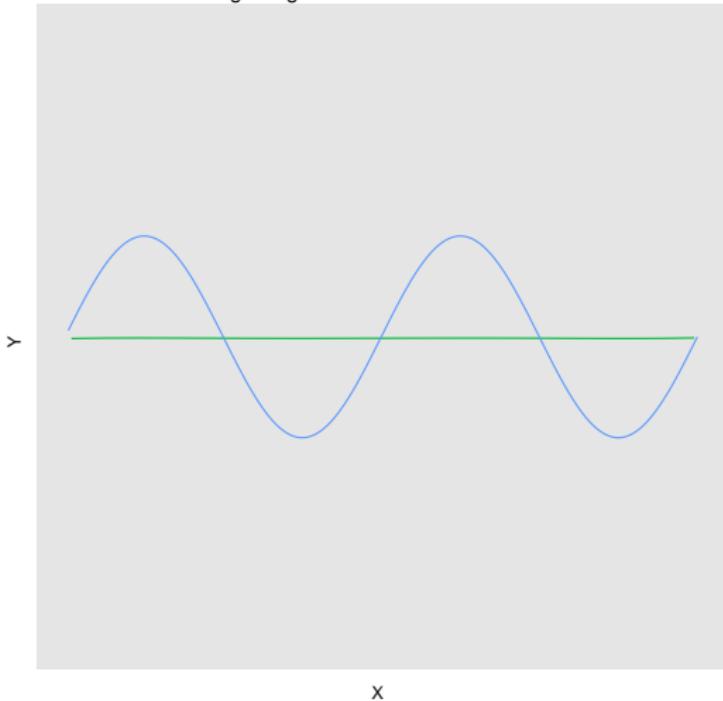
Ridge Regression with Lambda = 249



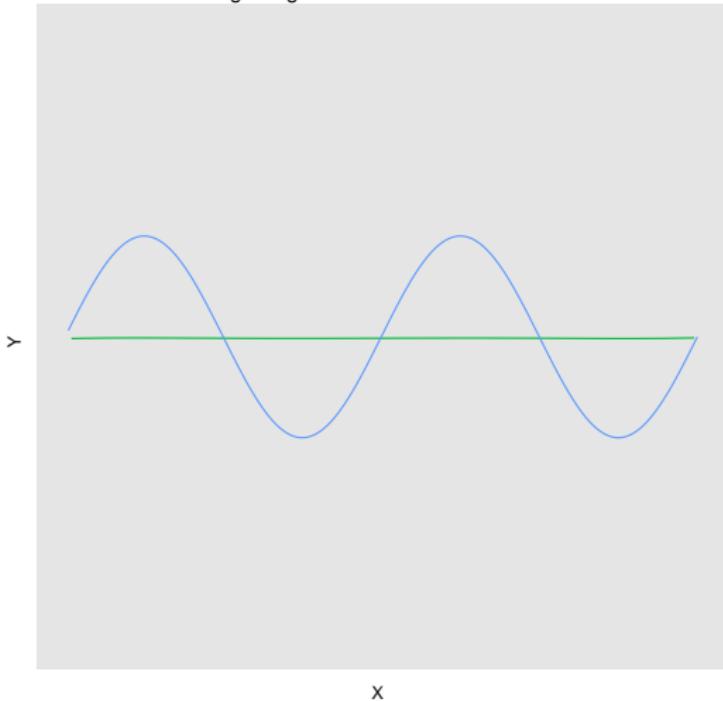
Ridge Regression with Lambda = 227



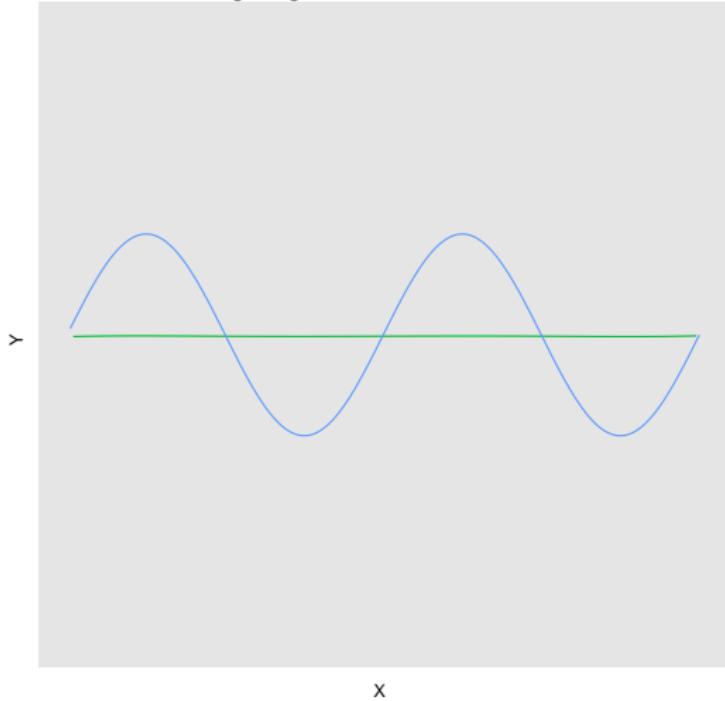
Ridge Regression with Lambda = 207



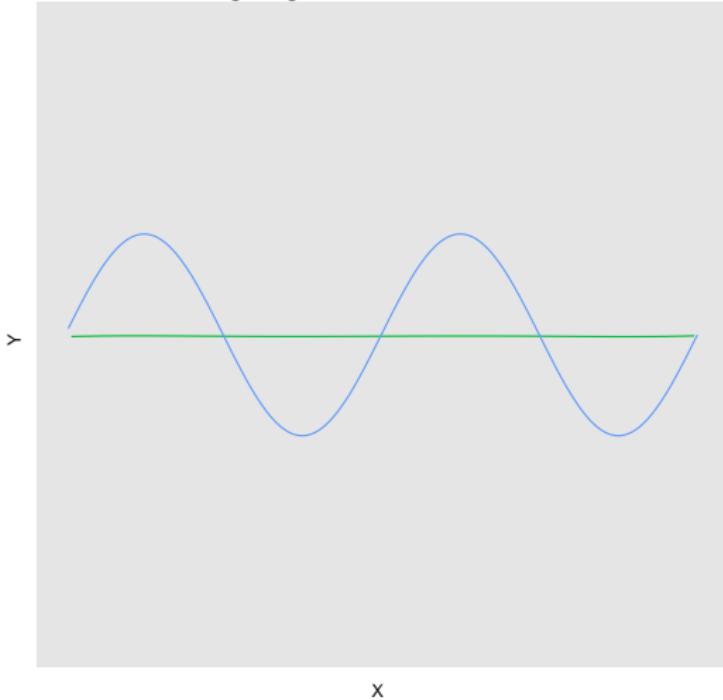
Ridge Regression with Lambda = 188



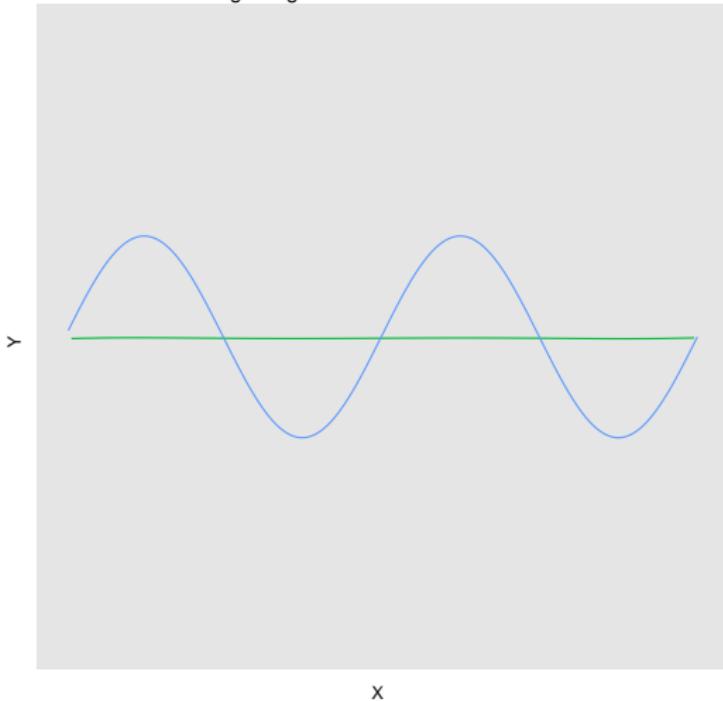
Ridge Regression with Lambda = 172



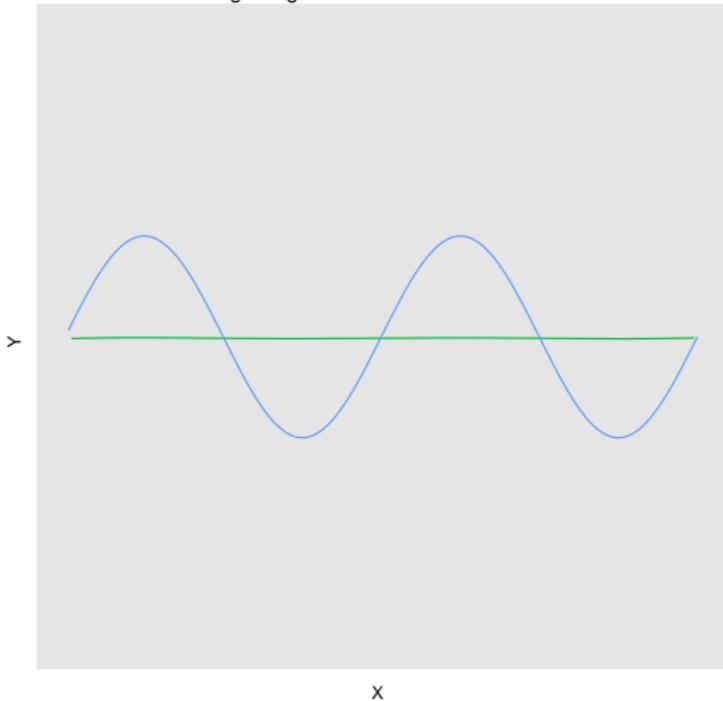
Ridge Regression with Lambda = 156



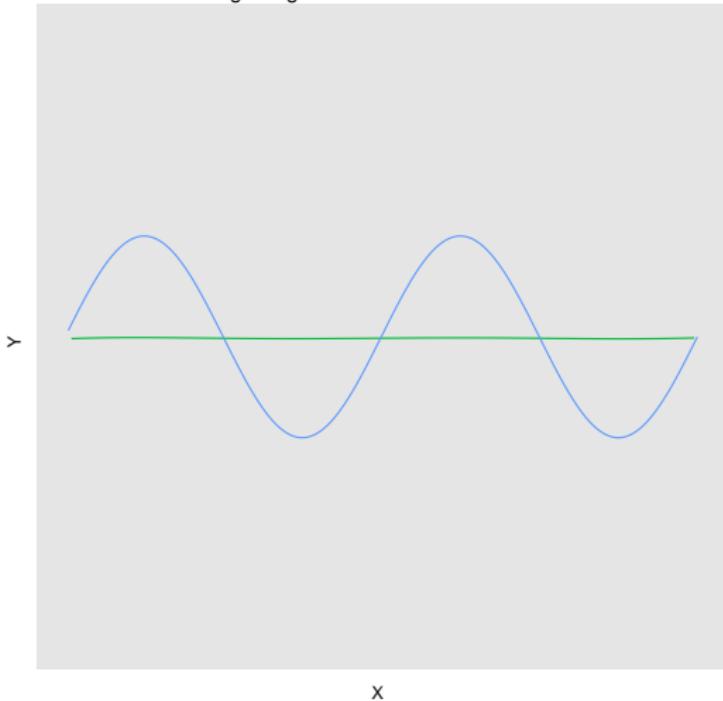
Ridge Regression with Lambda = 142



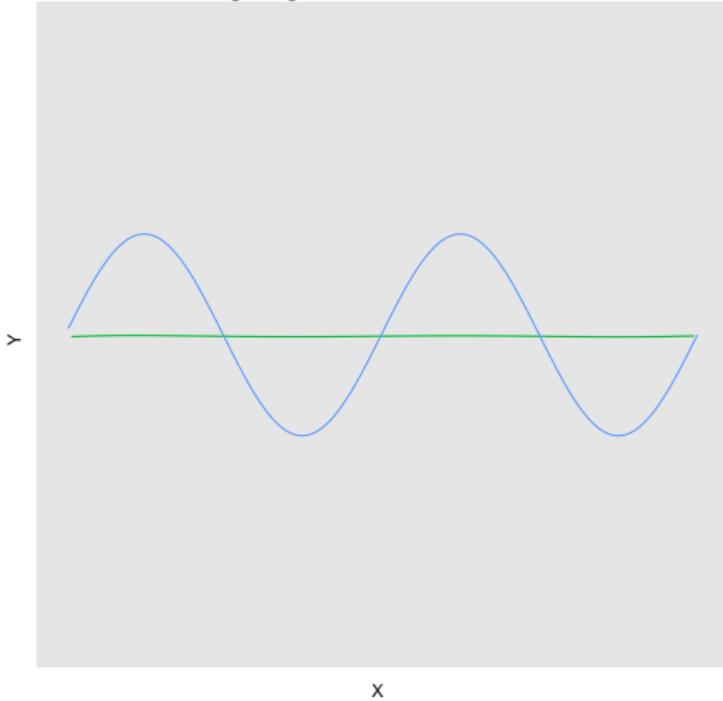
Ridge Regression with Lambda = 130



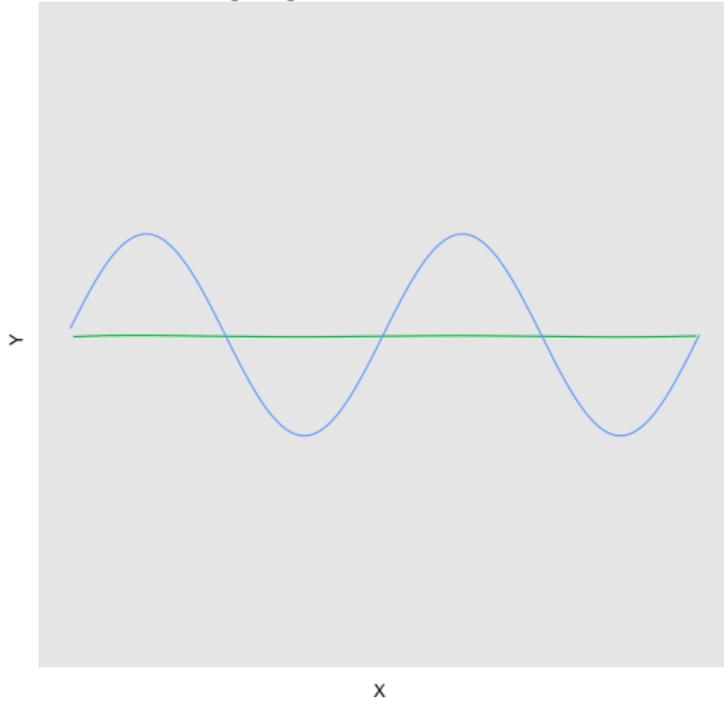
Ridge Regression with Lambda = 118



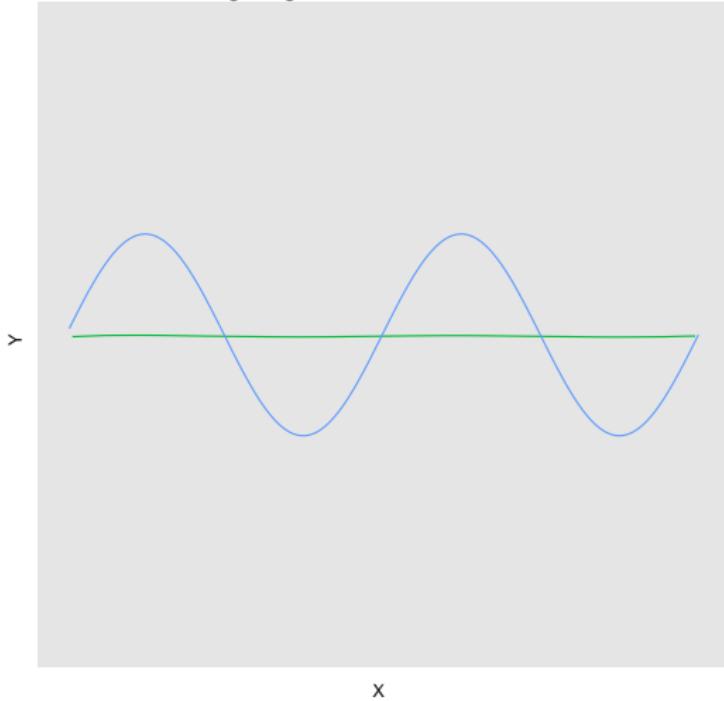
Ridge Regression with Lambda = 108



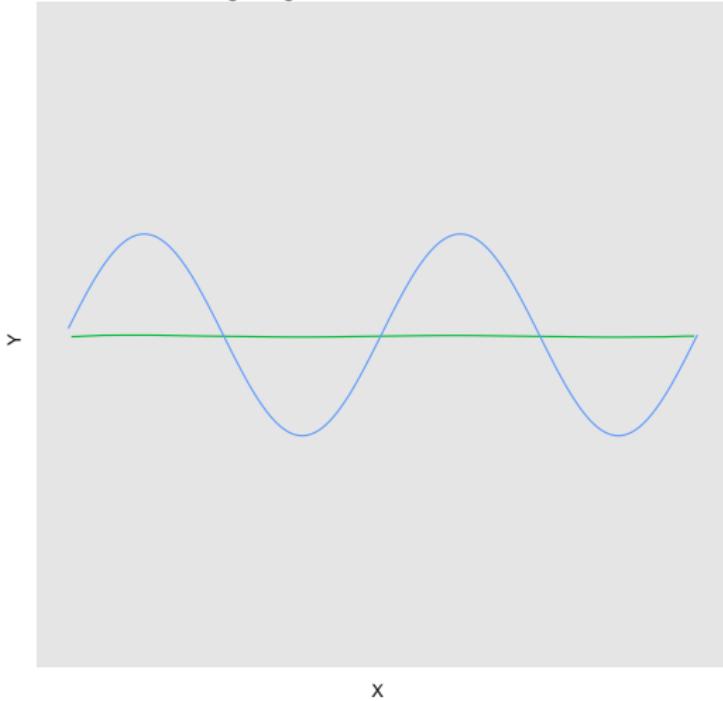
Ridge Regression with Lambda = 98.2



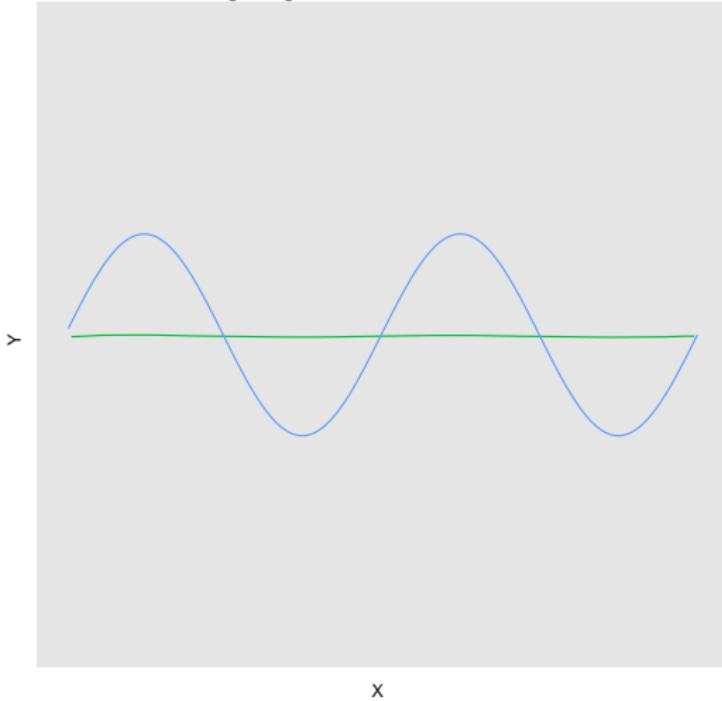
Ridge Regression with Lambda = 89.5



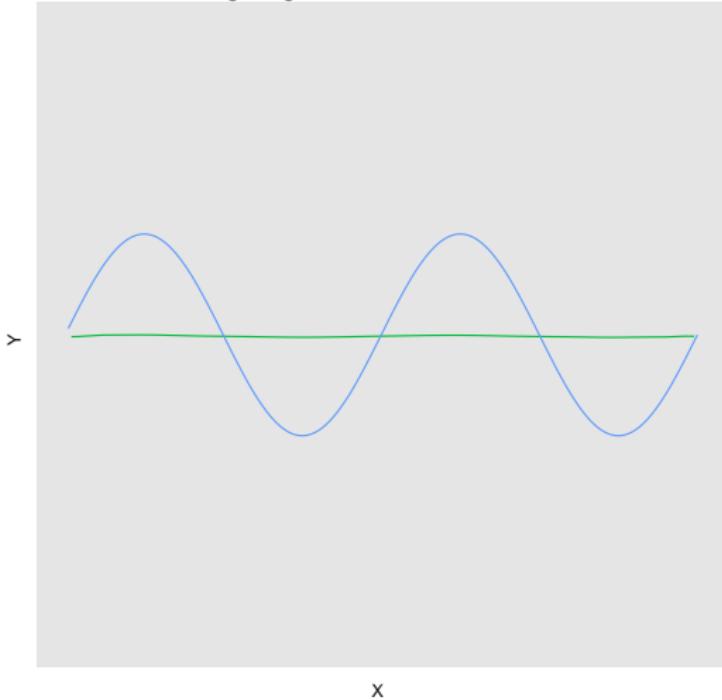
Ridge Regression with Lambda = 81.5



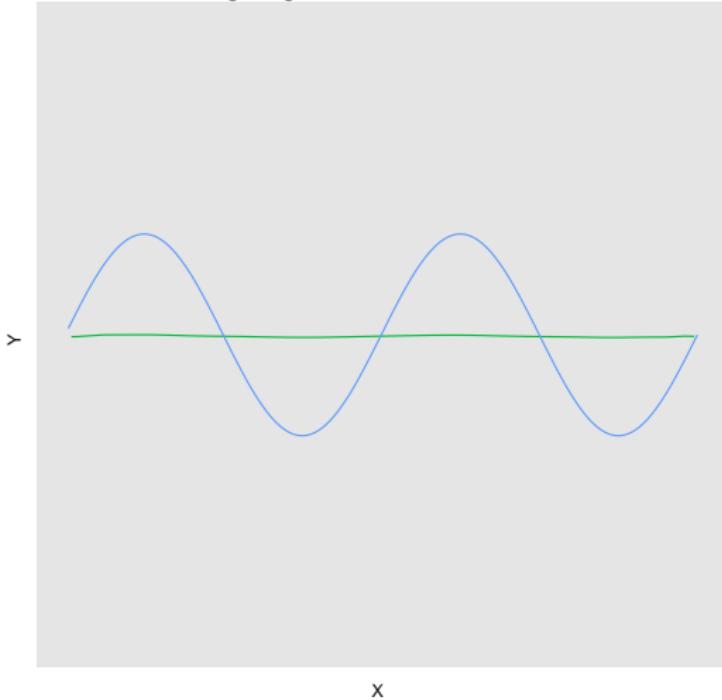
Ridge Regression with Lambda = 74.3



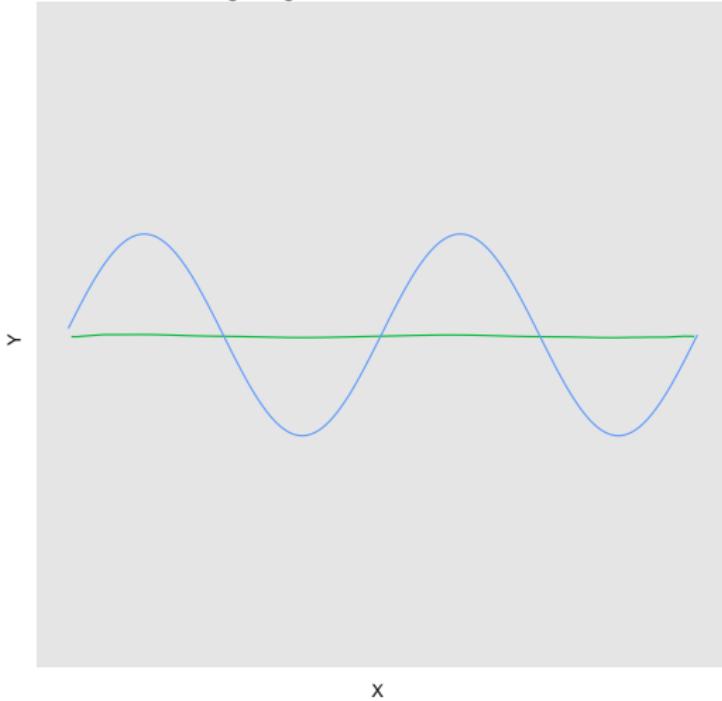
Ridge Regression with Lambda = 67.7



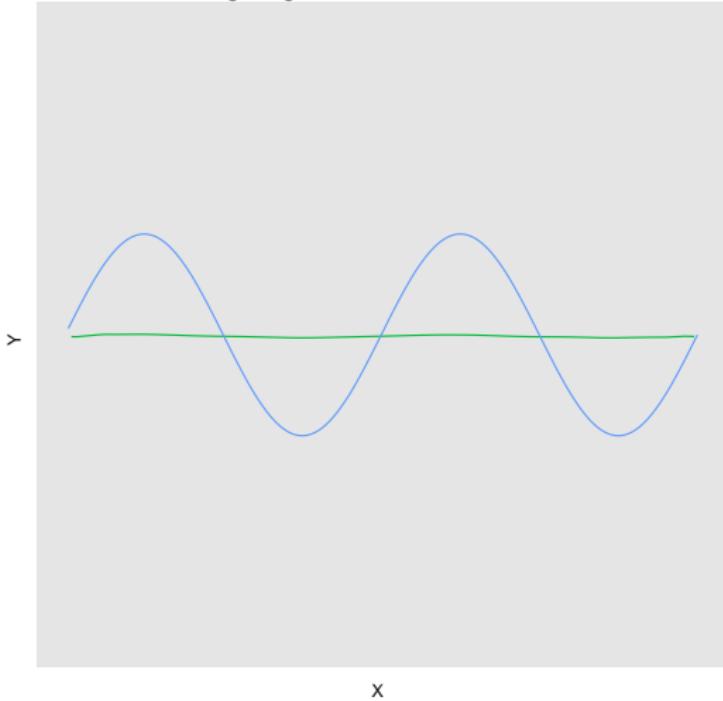
Ridge Regression with Lambda = 61.7



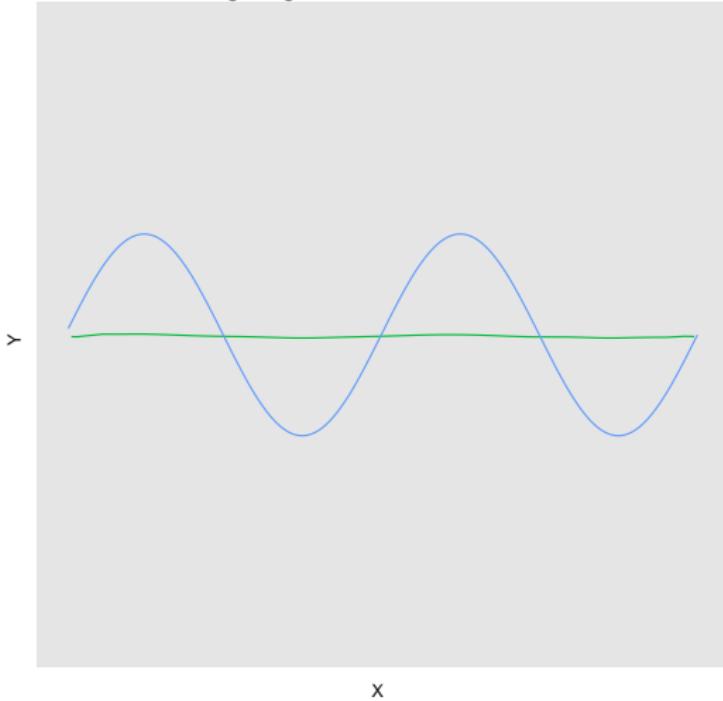
Ridge Regression with Lambda = 56.2



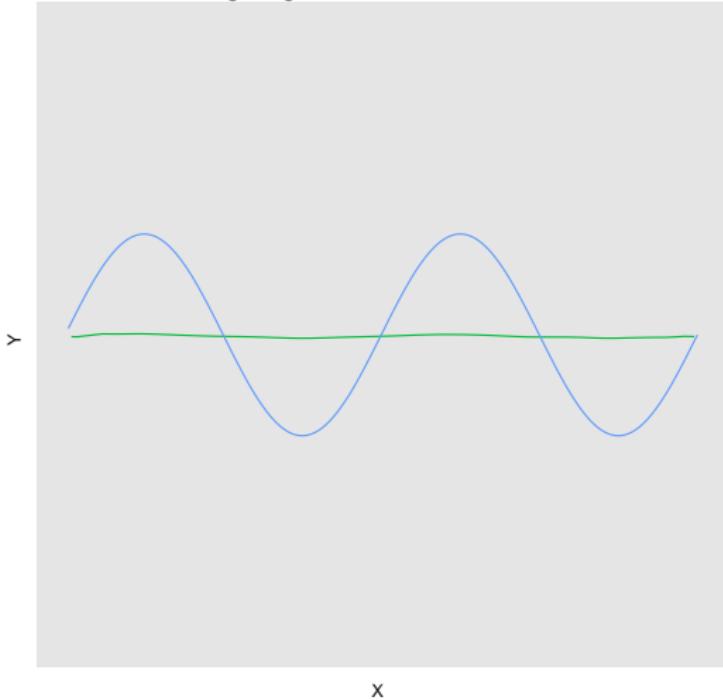
Ridge Regression with Lambda = 51.2



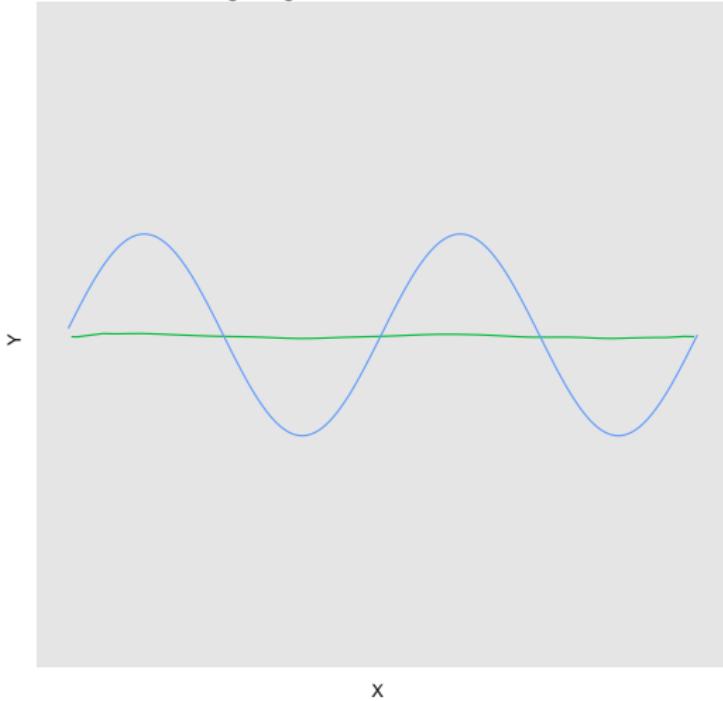
Ridge Regression with Lambda = 46.7



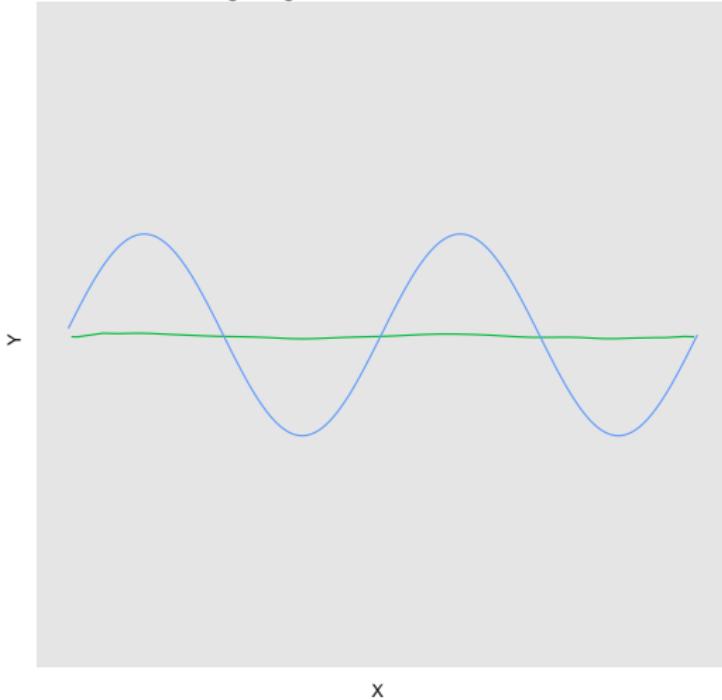
Ridge Regression with Lambda = 42.5



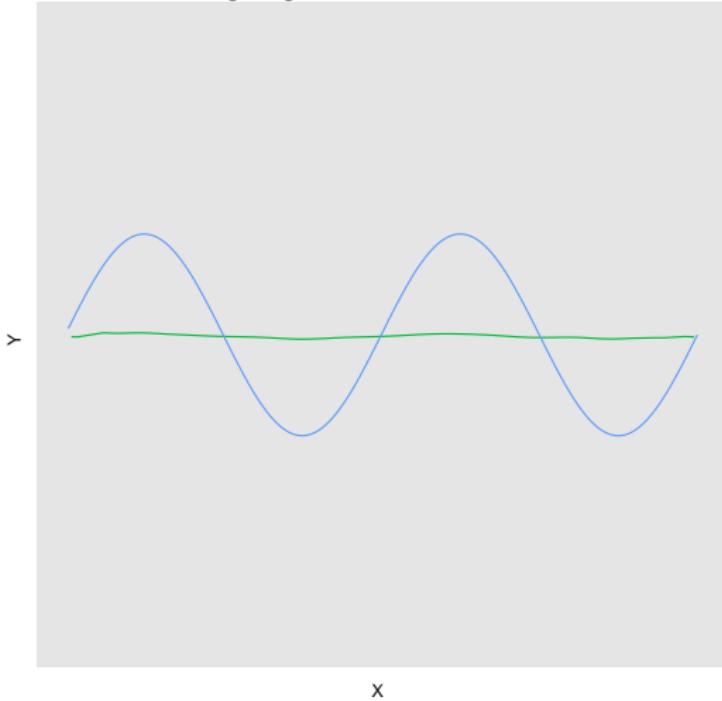
Ridge Regression with Lambda = 38.7



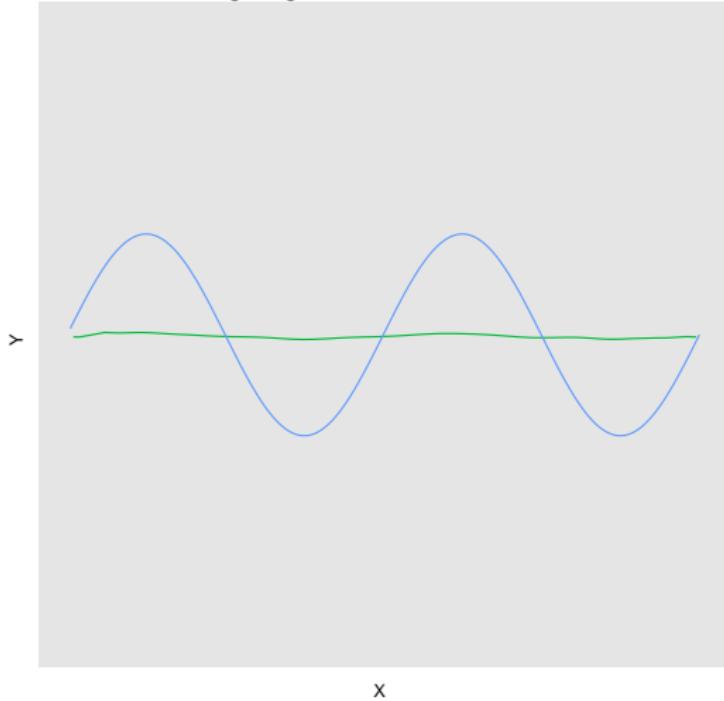
Ridge Regression with Lambda = 35.3



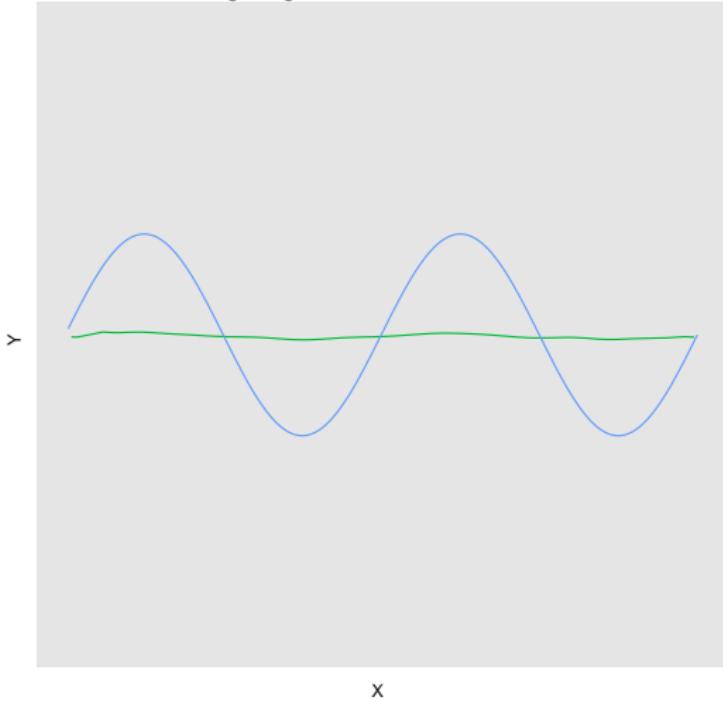
Ridge Regression with Lambda = 32.2



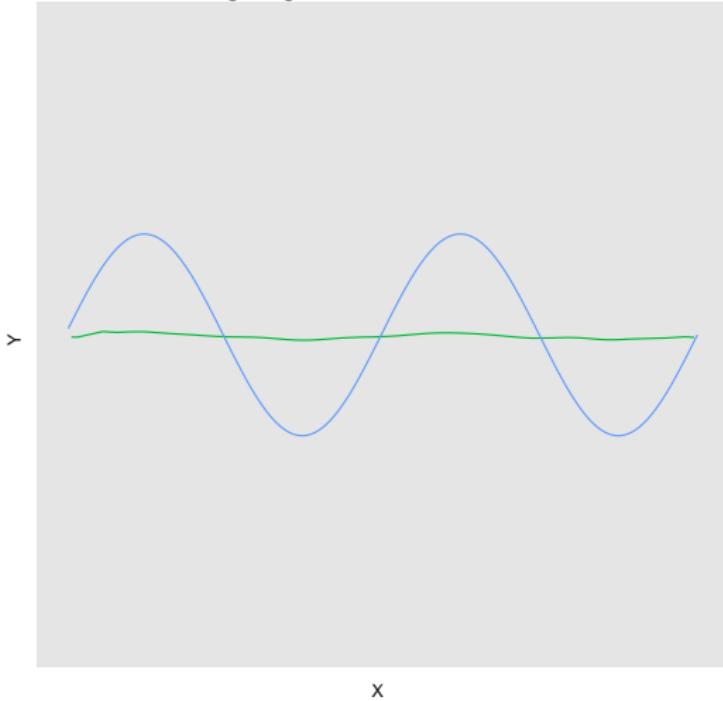
Ridge Regression with Lambda = 29.3



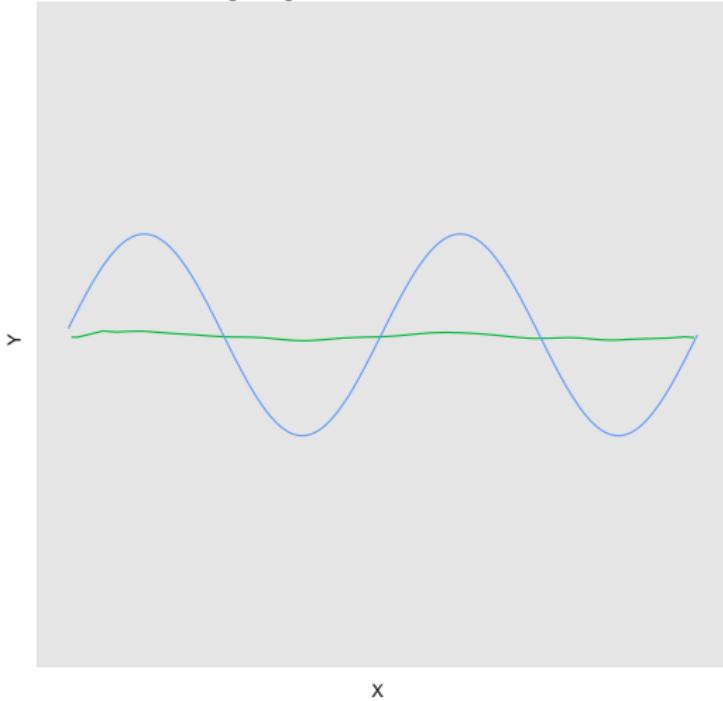
Ridge Regression with Lambda = 26.7



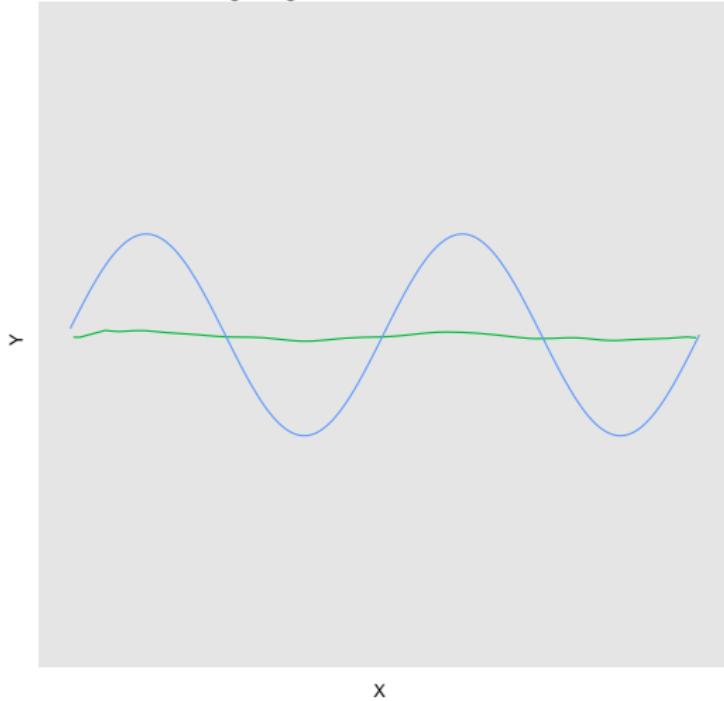
Ridge Regression with Lambda = 24.3



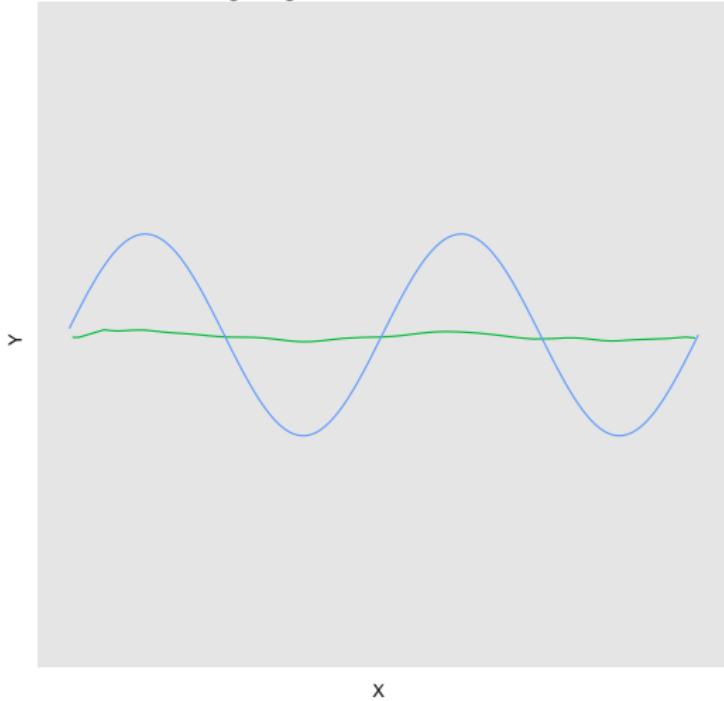
Ridge Regression with Lambda = 22.2



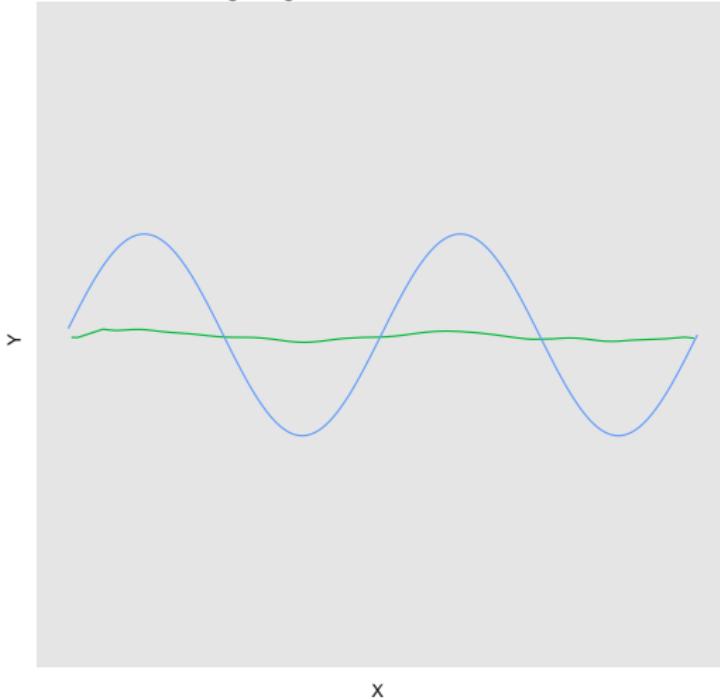
Ridge Regression with Lambda = 20.2



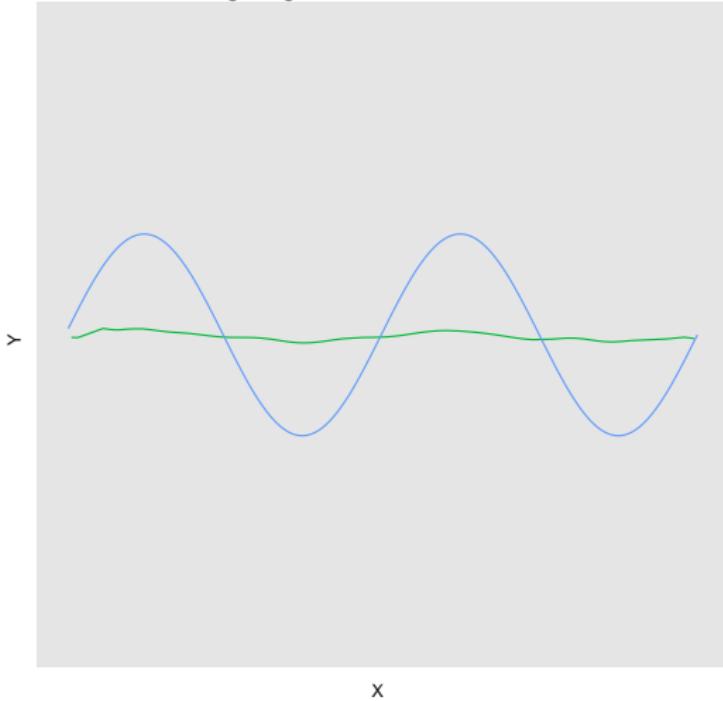
Ridge Regression with Lambda = 18.4



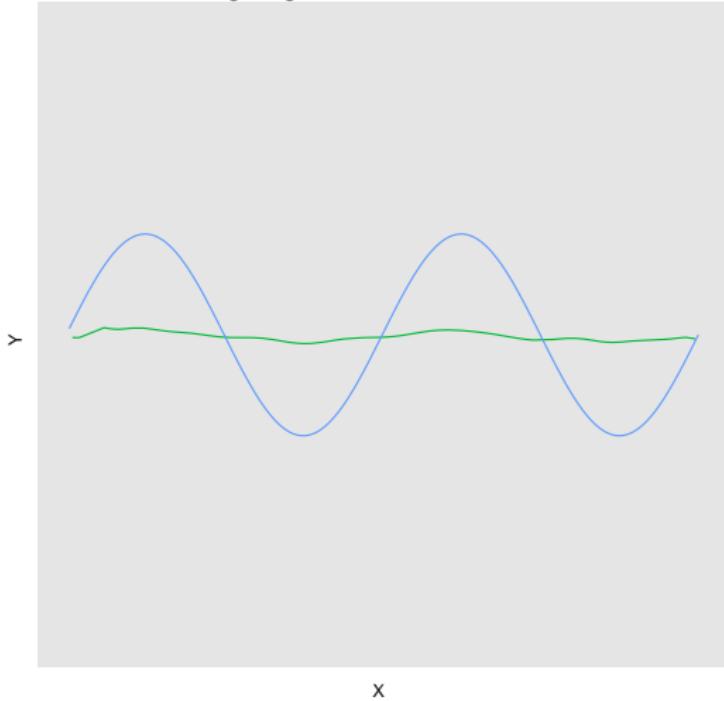
Ridge Regression with Lambda = 16.8



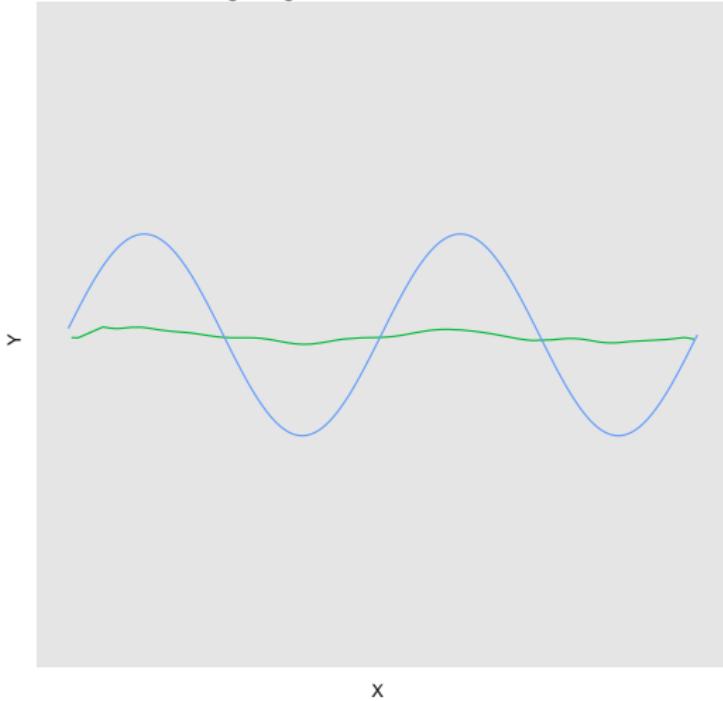
Ridge Regression with Lambda = 15.3



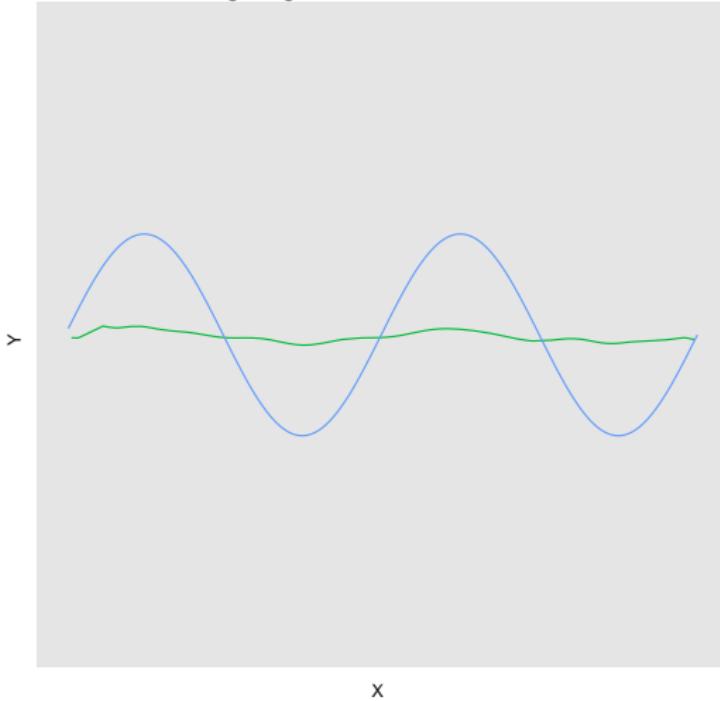
Ridge Regression with Lambda = 13.9



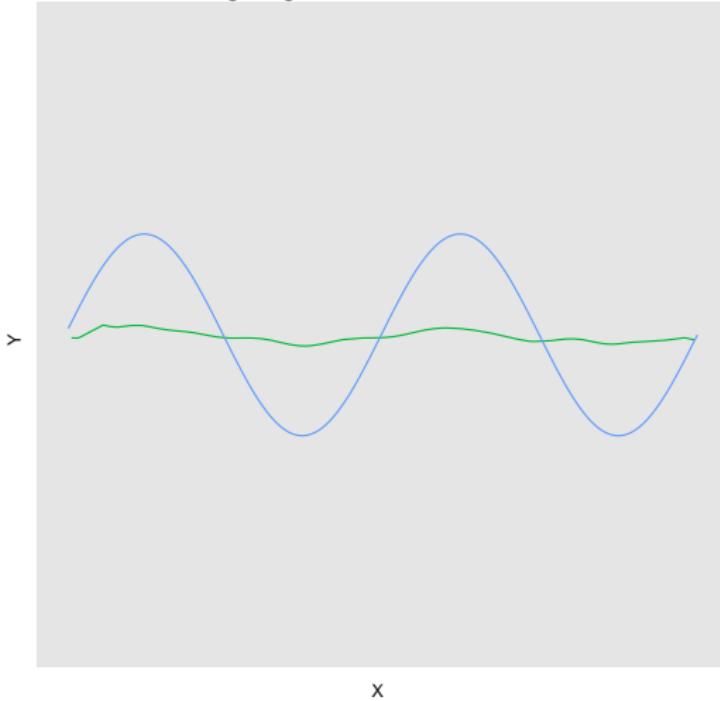
Ridge Regression with Lambda = 12.7



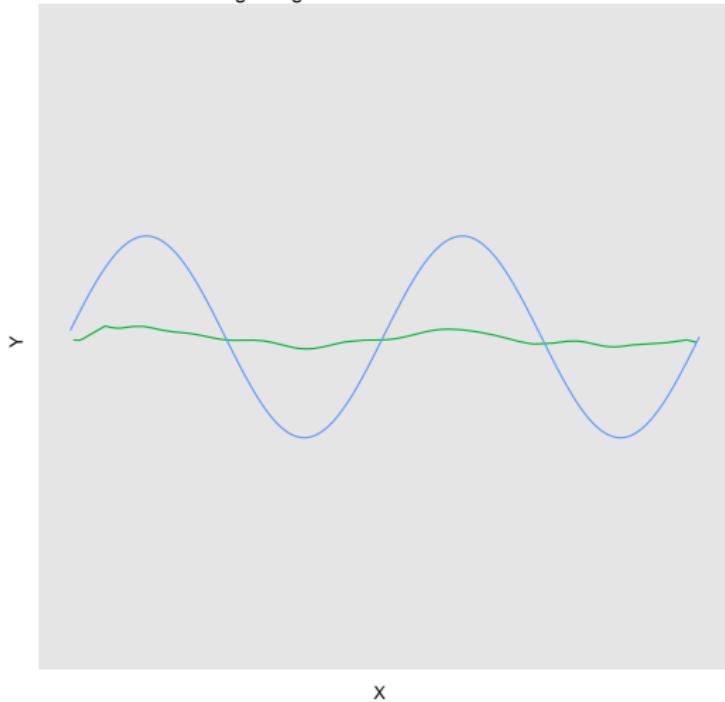
Ridge Regression with Lambda = 11.6



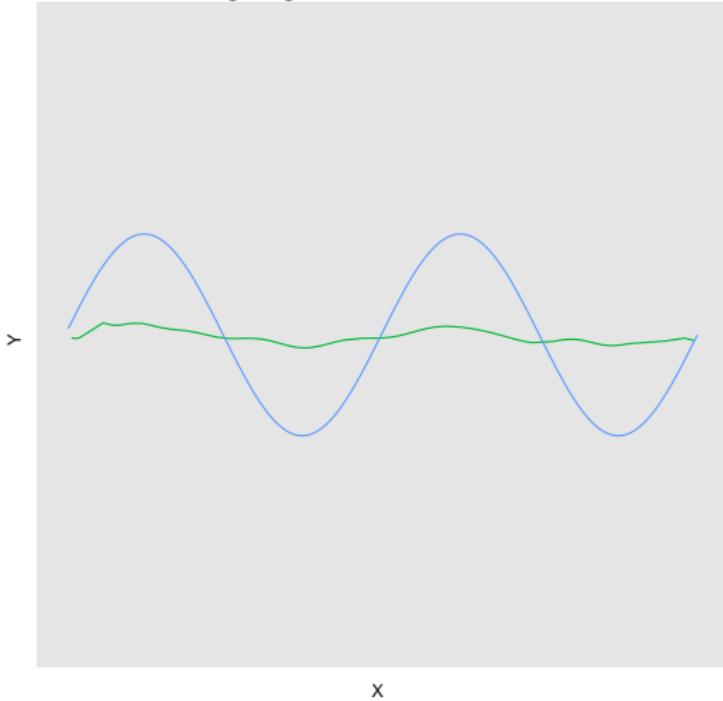
Ridge Regression with Lambda = 10.5



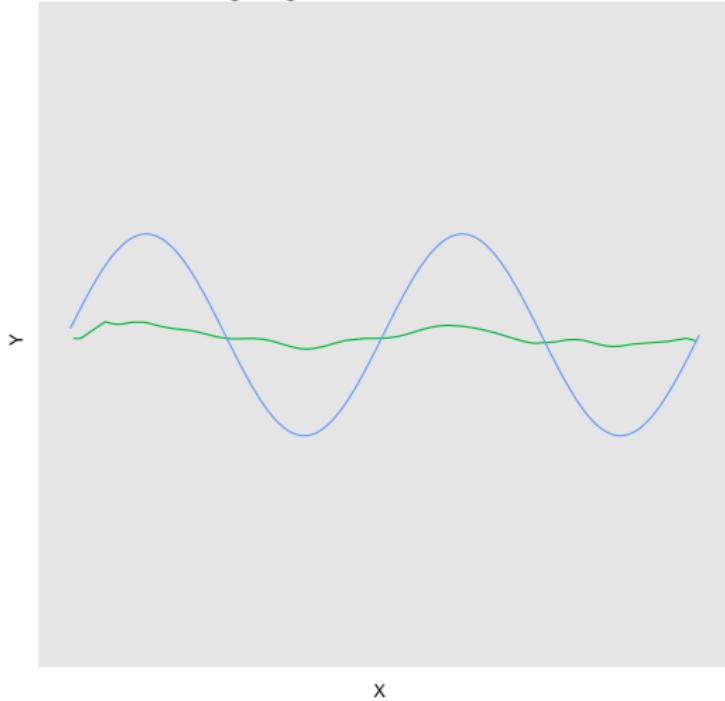
Ridge Regression with Lambda = 9.6



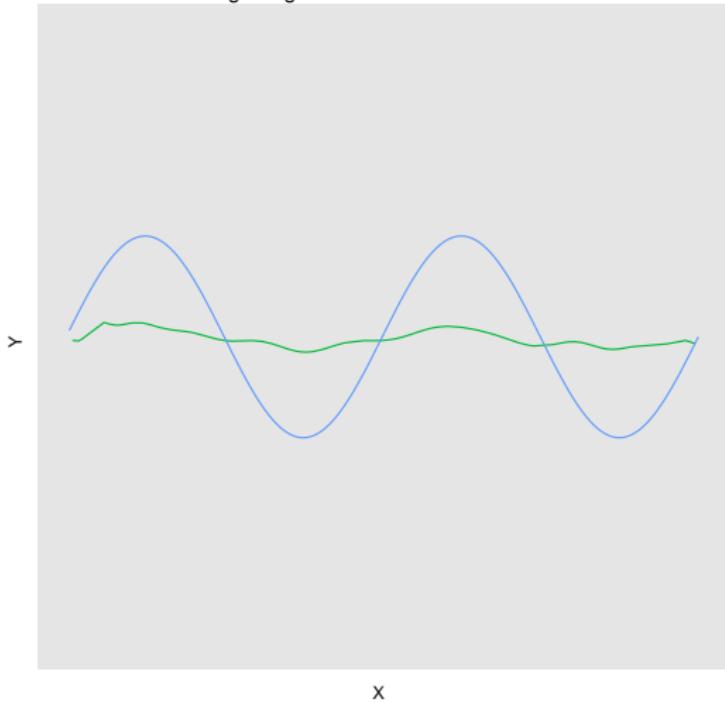
Ridge Regression with Lambda = 8.74



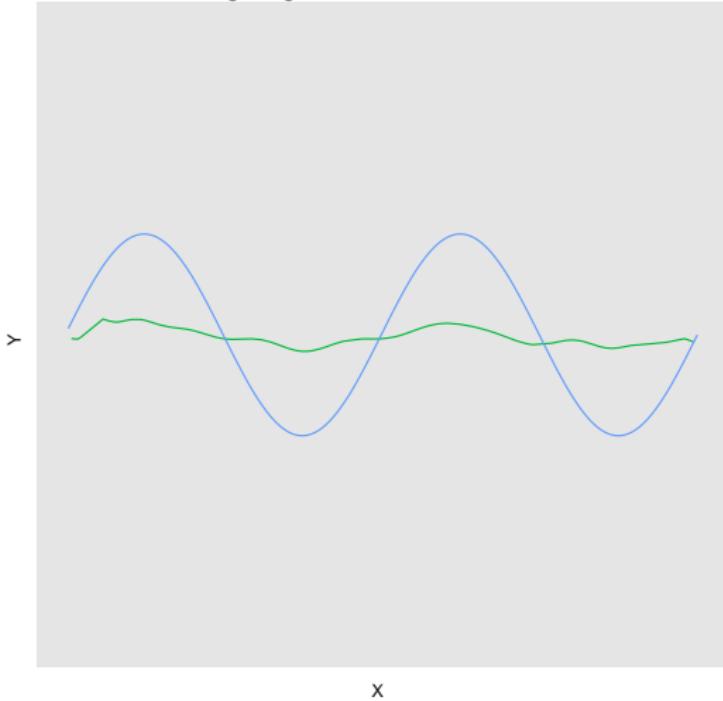
Ridge Regression with Lambda = 7.97



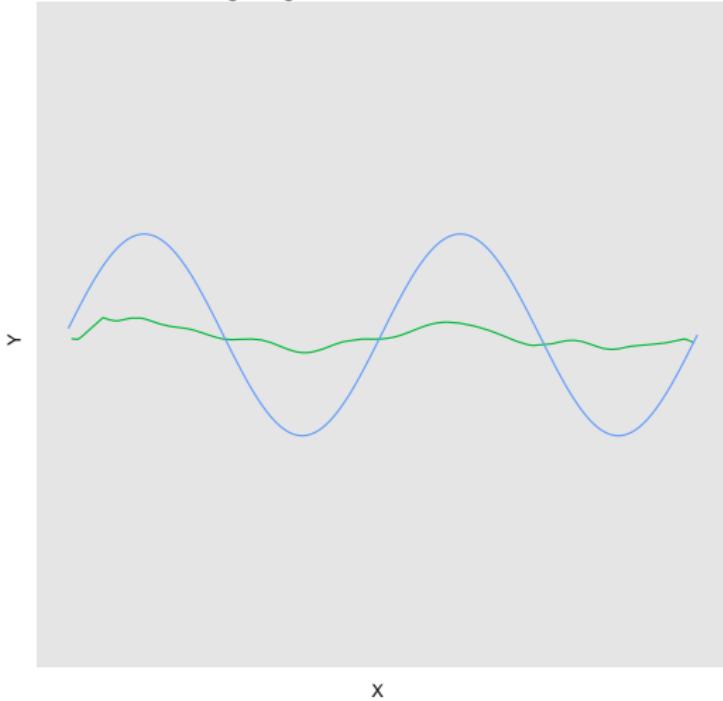
Ridge Regression with Lambda = 7.26



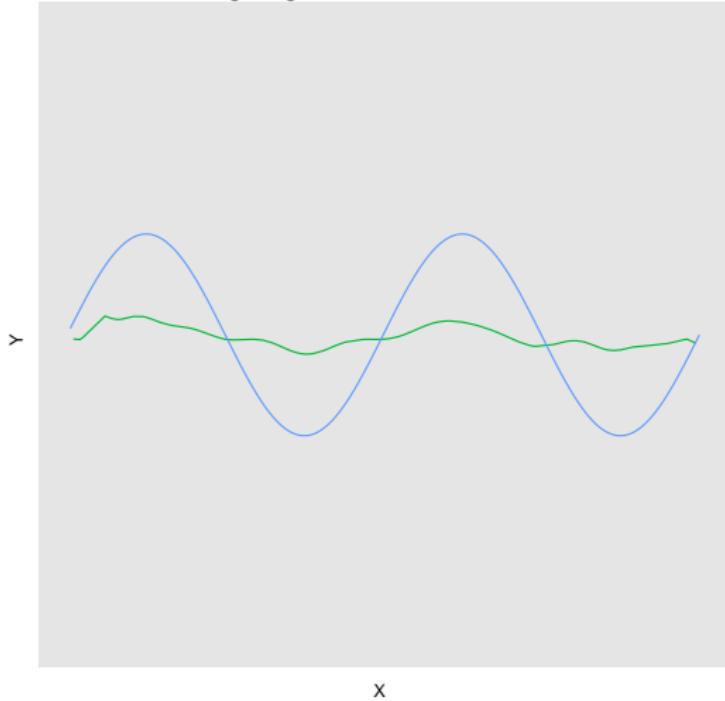
Ridge Regression with Lambda = 6.61



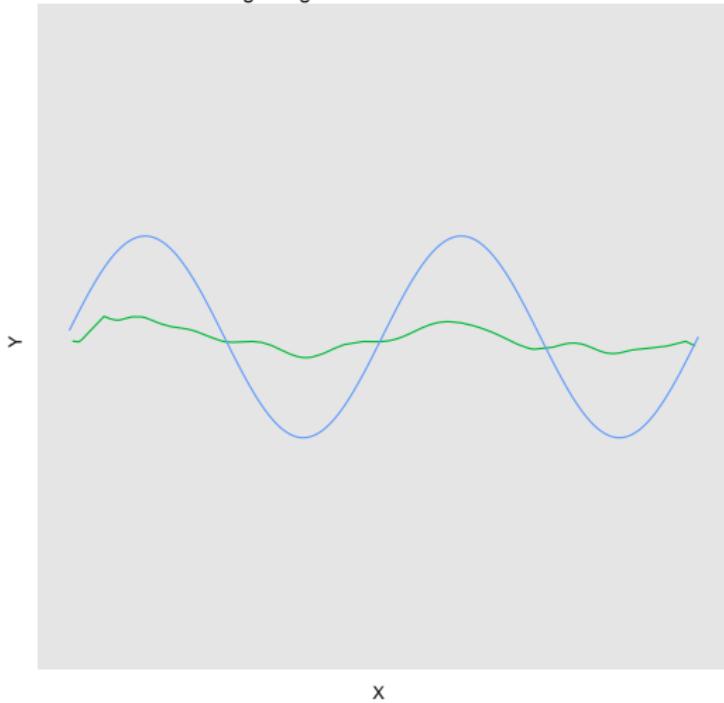
Ridge Regression with Lambda = 6.03



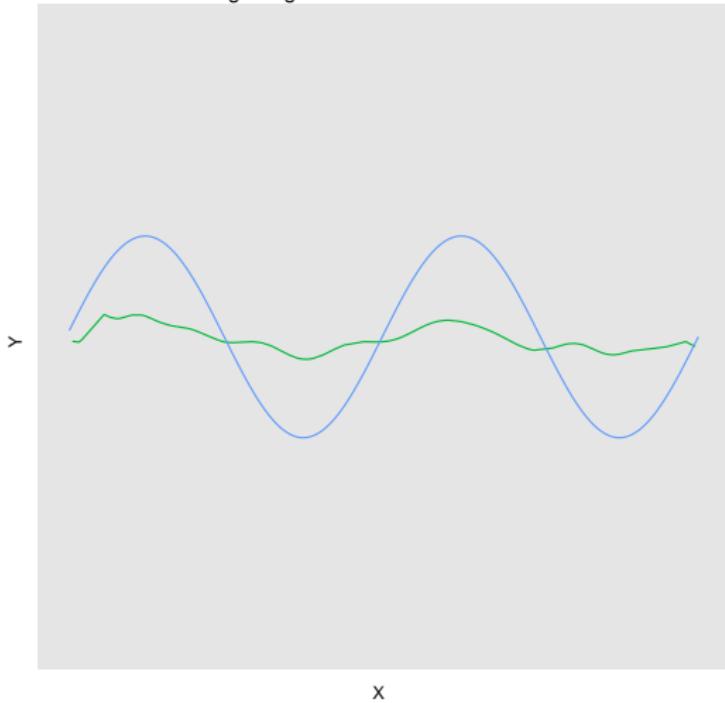
Ridge Regression with Lambda = 5.49



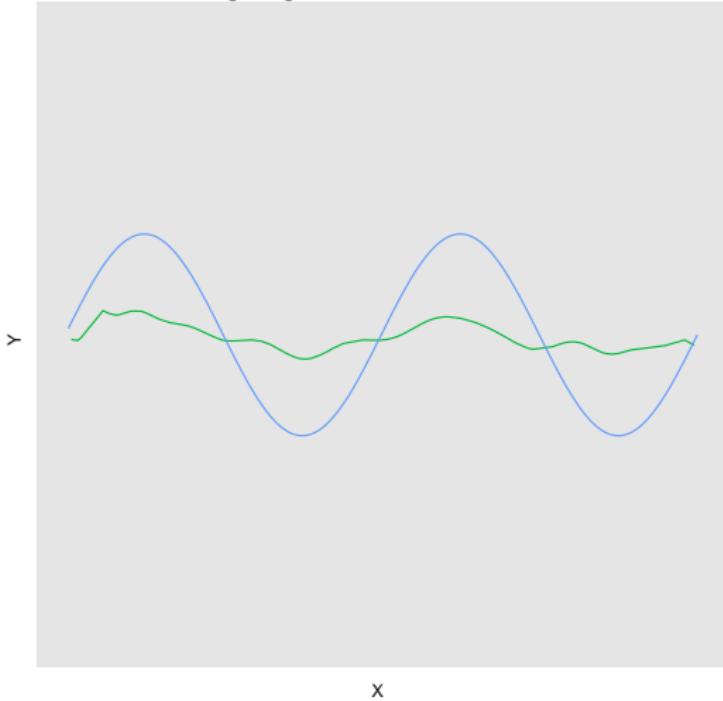
Ridge Regression with Lambda = 5



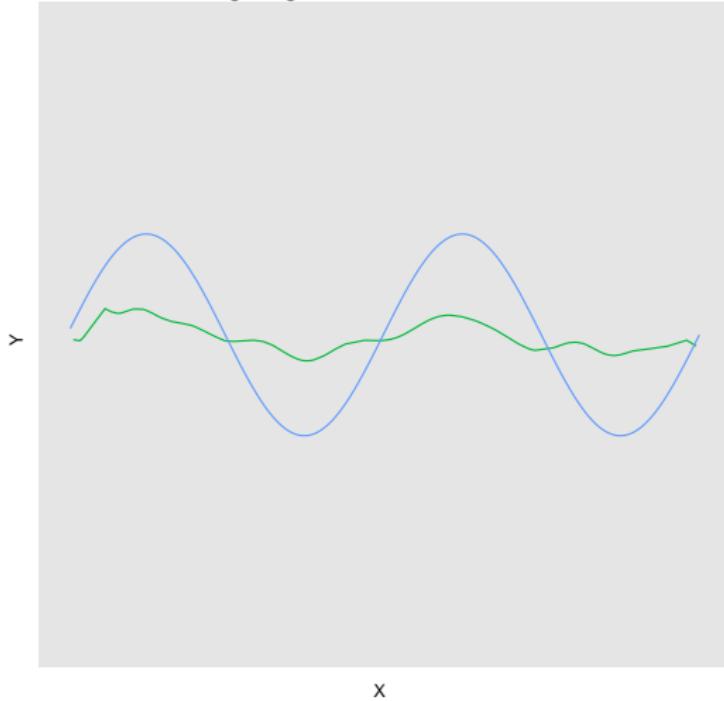
Ridge Regression with Lambda = 4.56



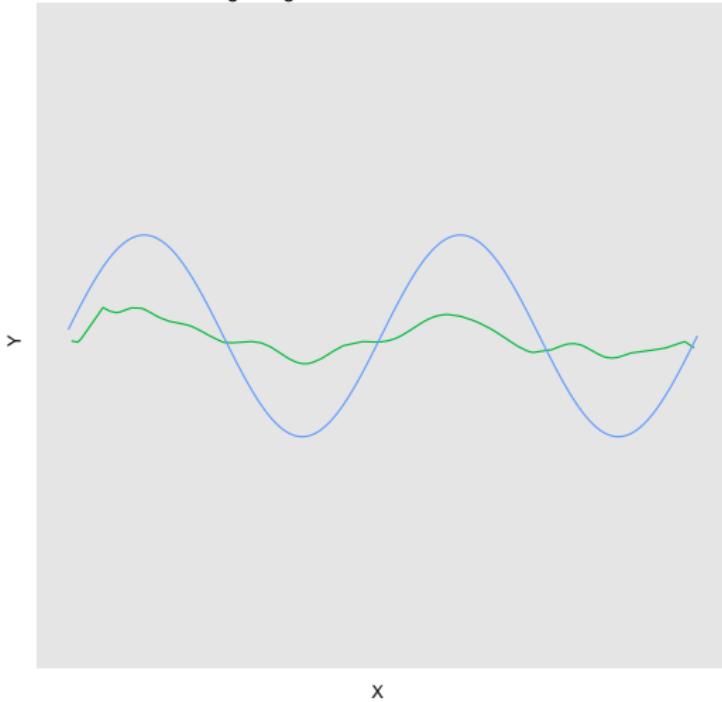
Ridge Regression with Lambda = 4.15



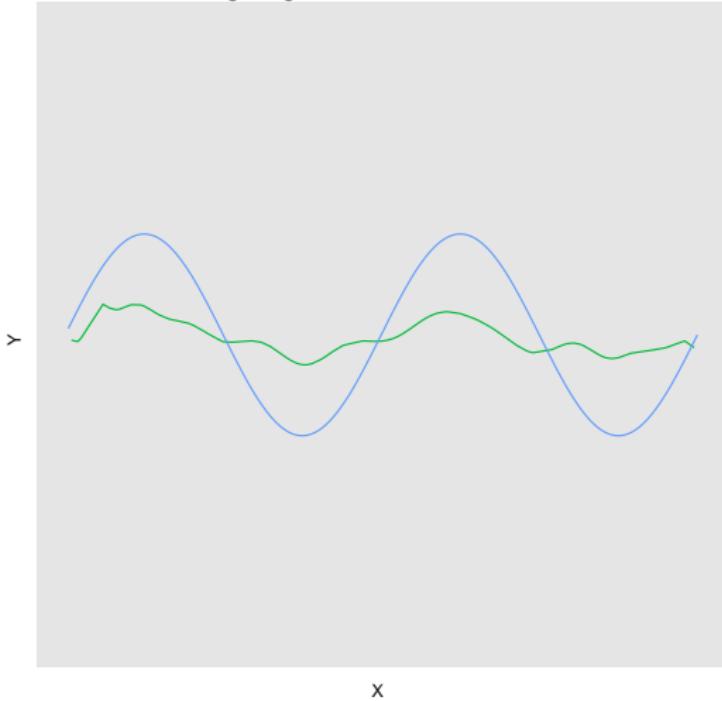
Ridge Regression with Lambda = 3.78



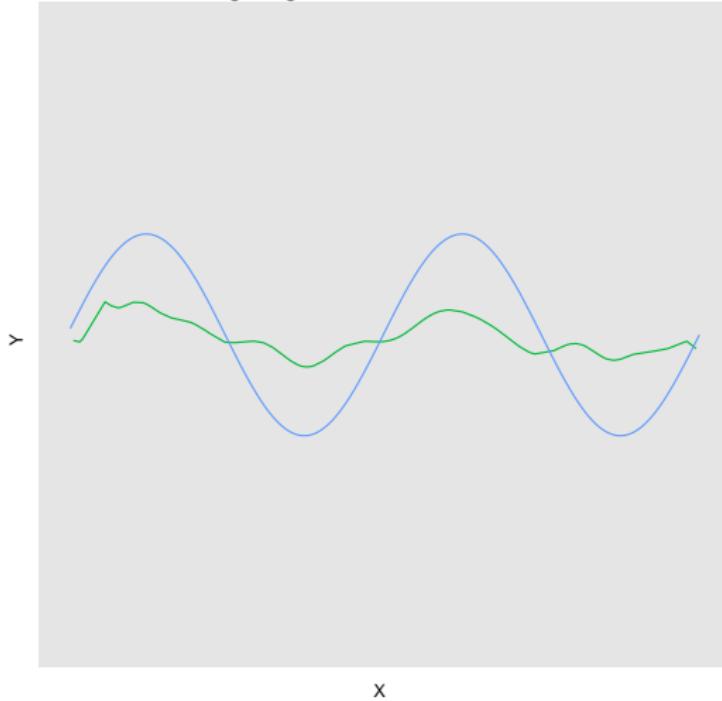
Ridge Regression with Lambda = 3.45



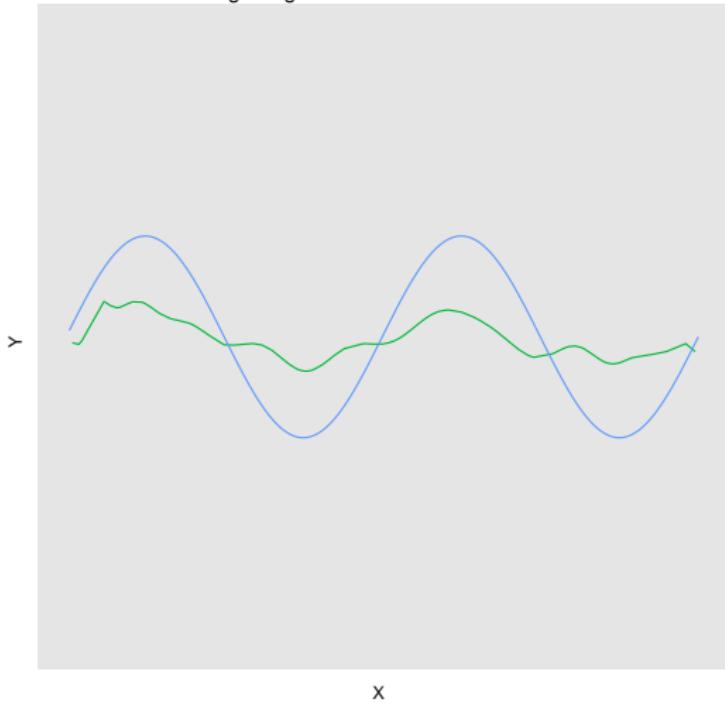
Ridge Regression with Lambda = 3.14



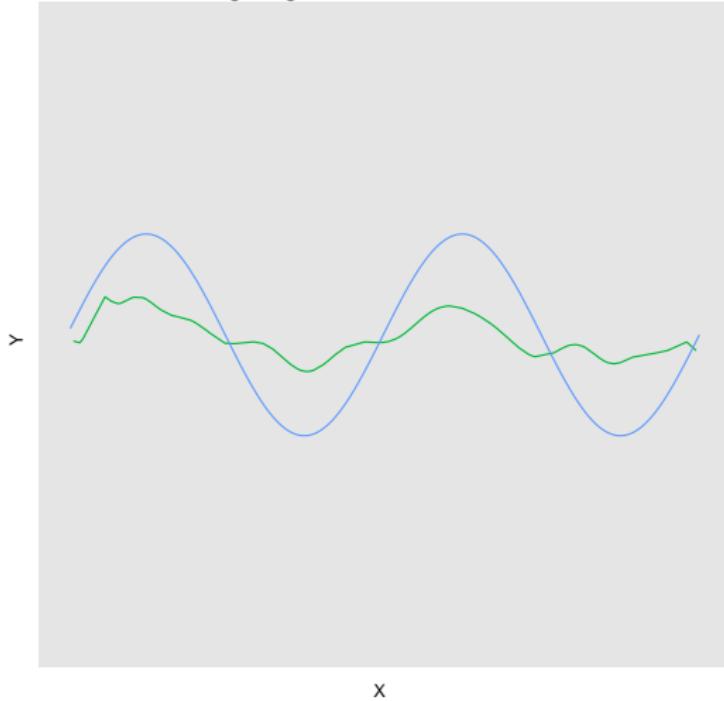
Ridge Regression with Lambda = 2.86



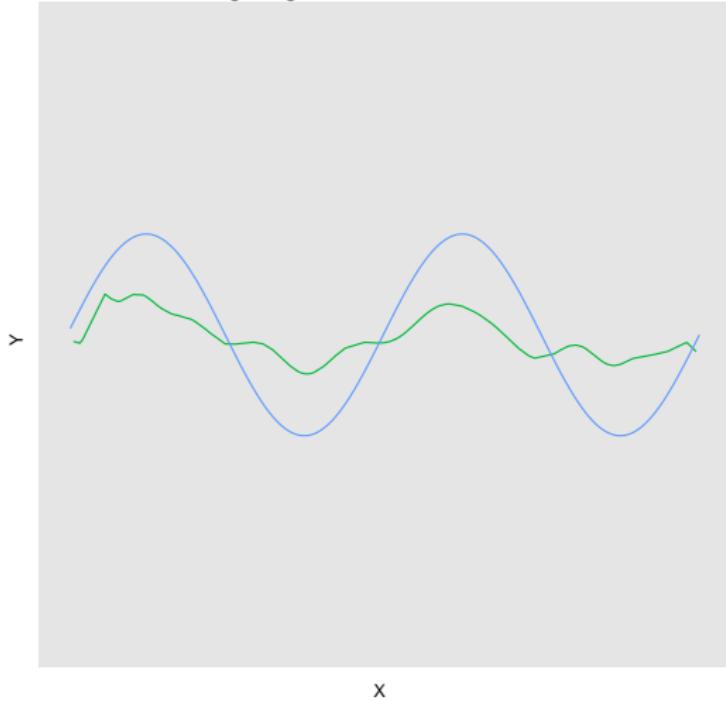
Ridge Regression with Lambda = 2.61



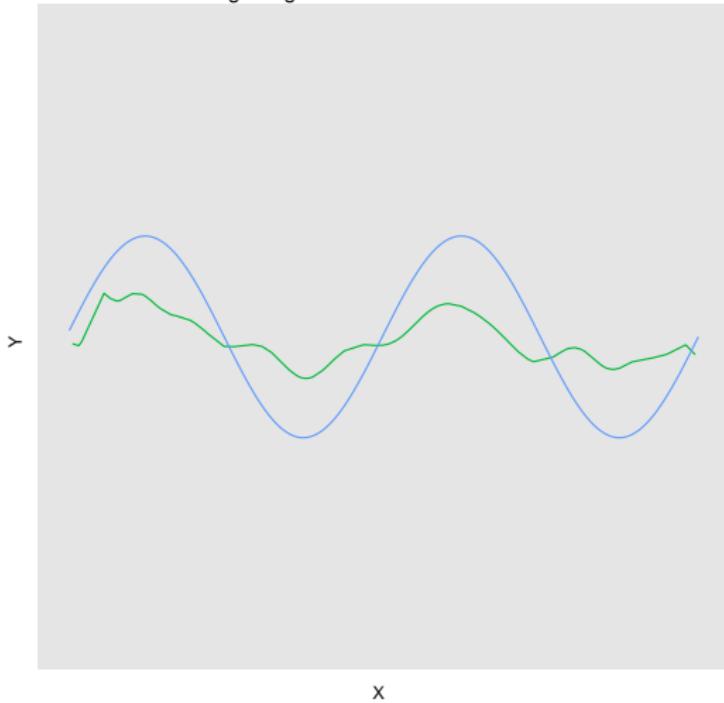
Ridge Regression with Lambda = 2.38



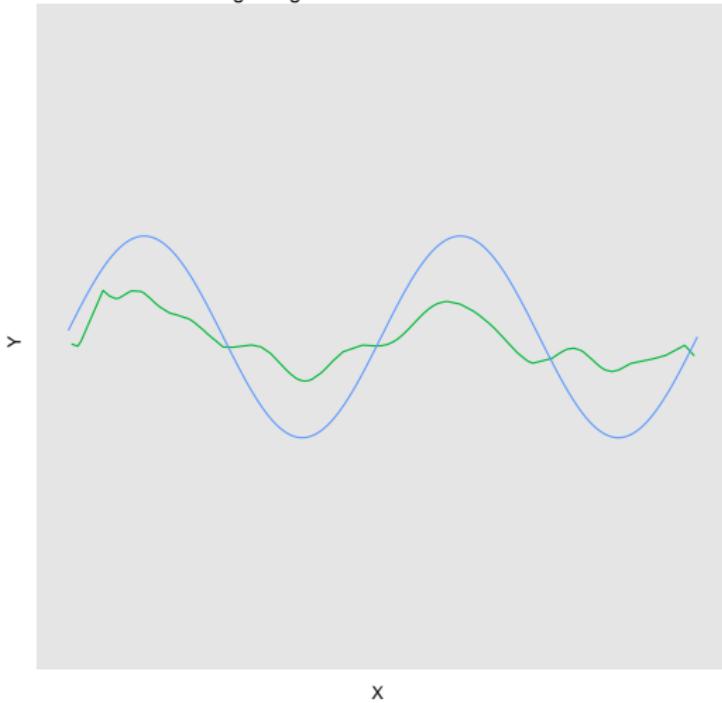
Ridge Regression with Lambda = 2.17



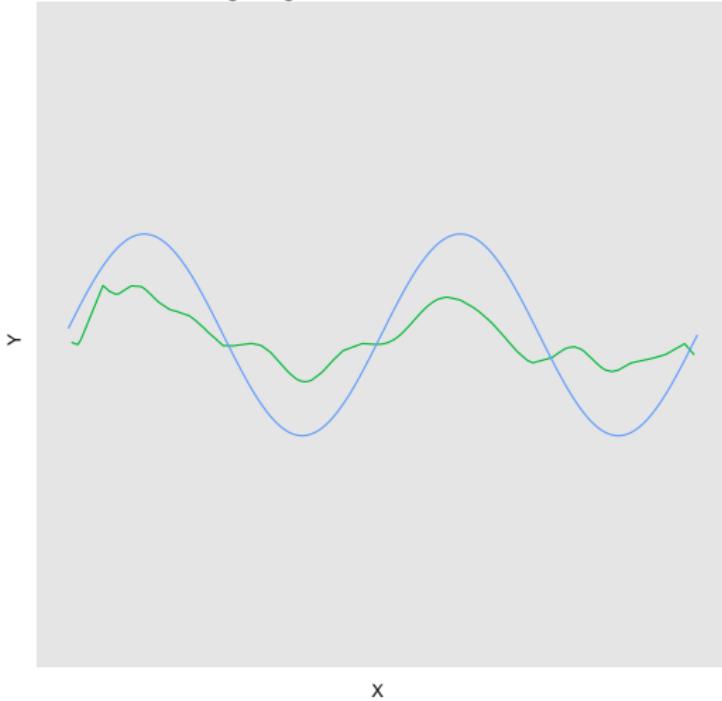
Ridge Regression with Lambda = 1.97



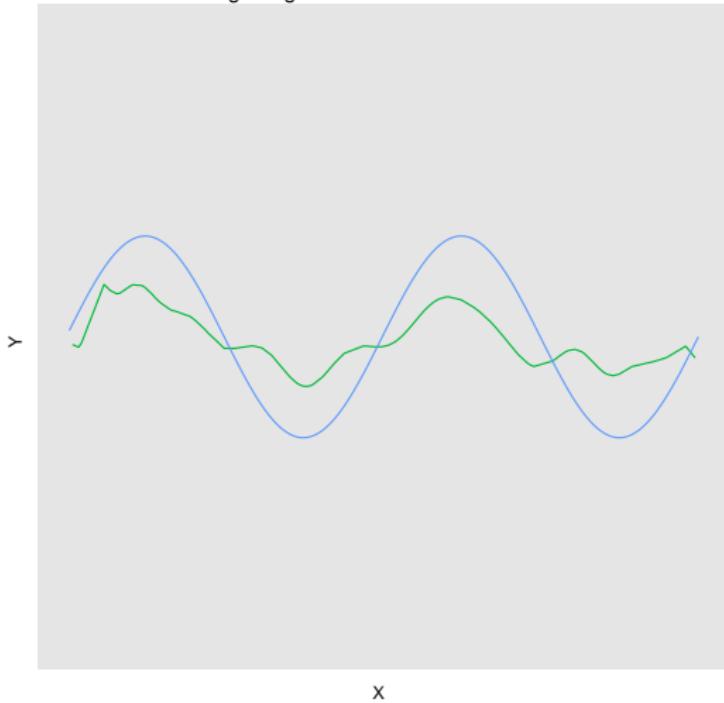
Ridge Regression with Lambda = 1.8



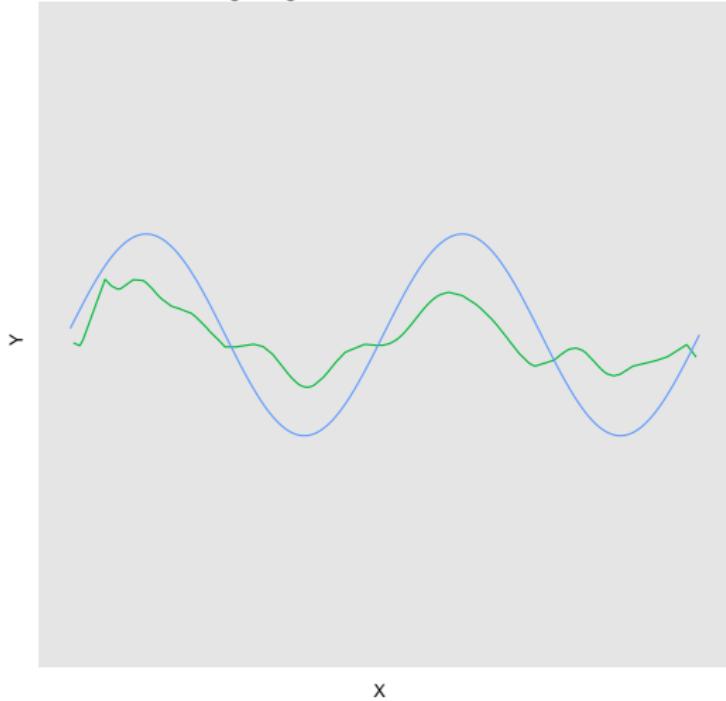
Ridge Regression with Lambda = 1.64



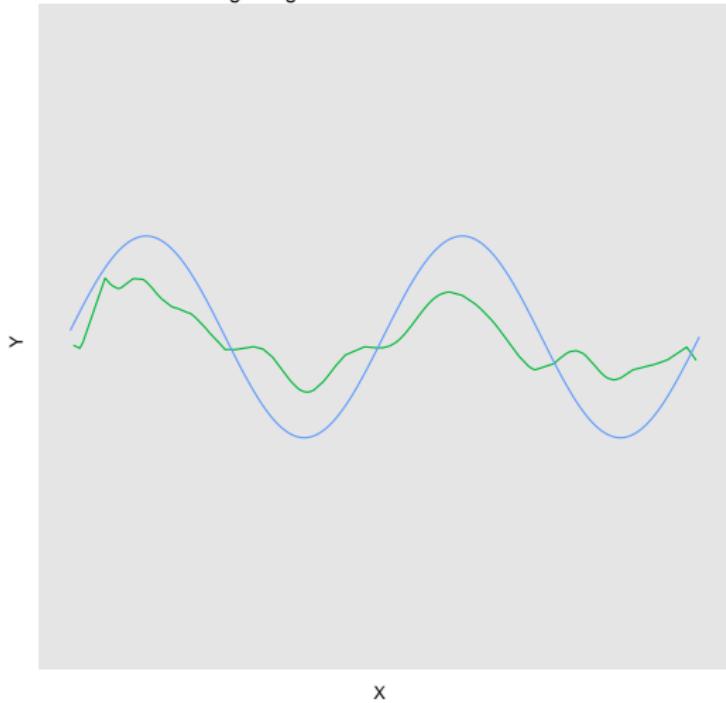
Ridge Regression with Lambda = 1.49



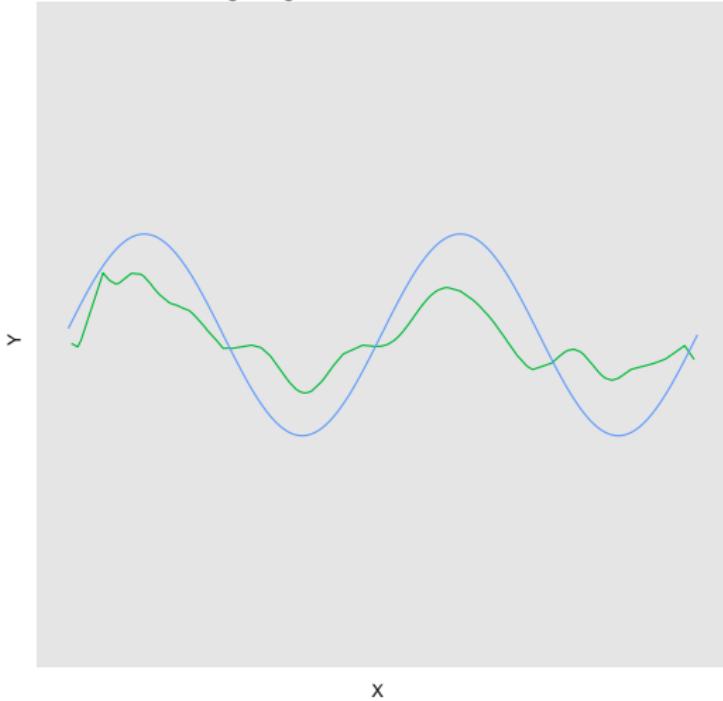
Ridge Regression with Lambda = 1.36



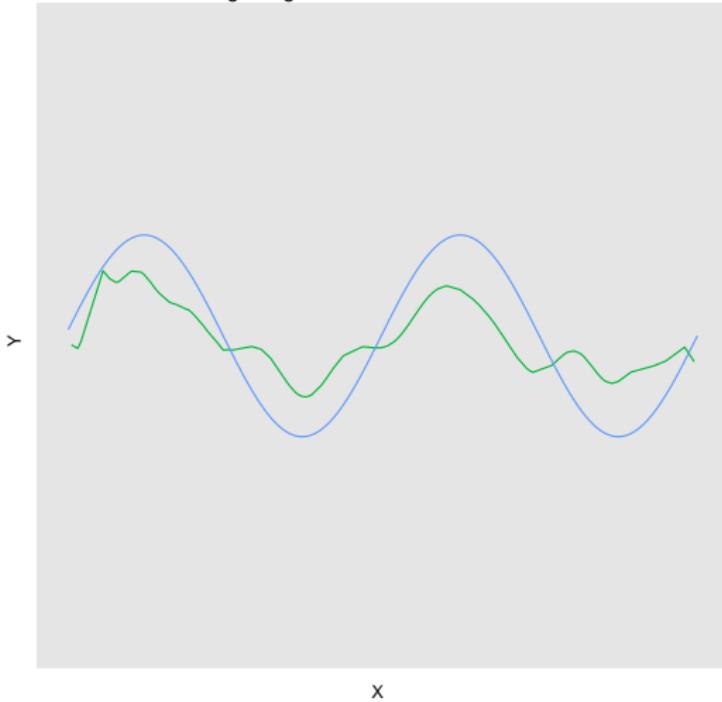
Ridge Regression with Lambda = 1.24



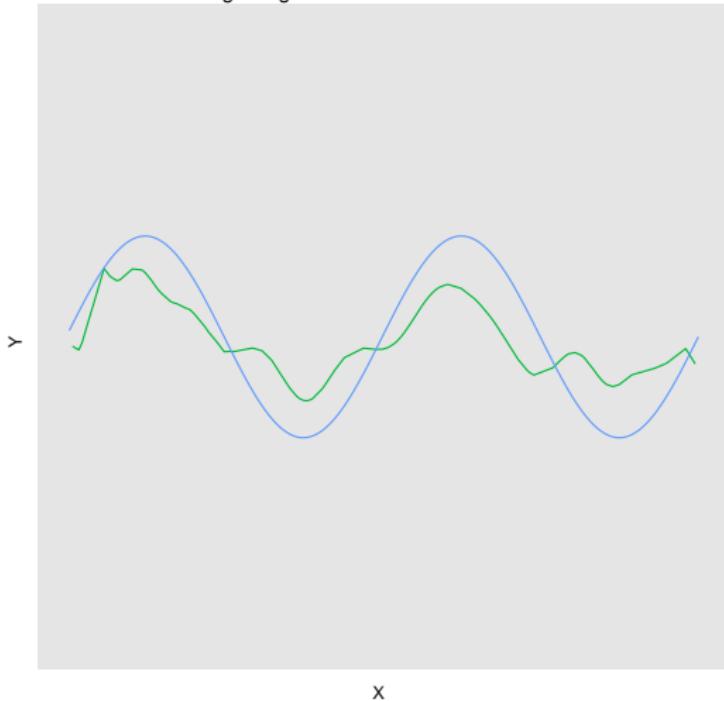
Ridge Regression with Lambda = 1.13



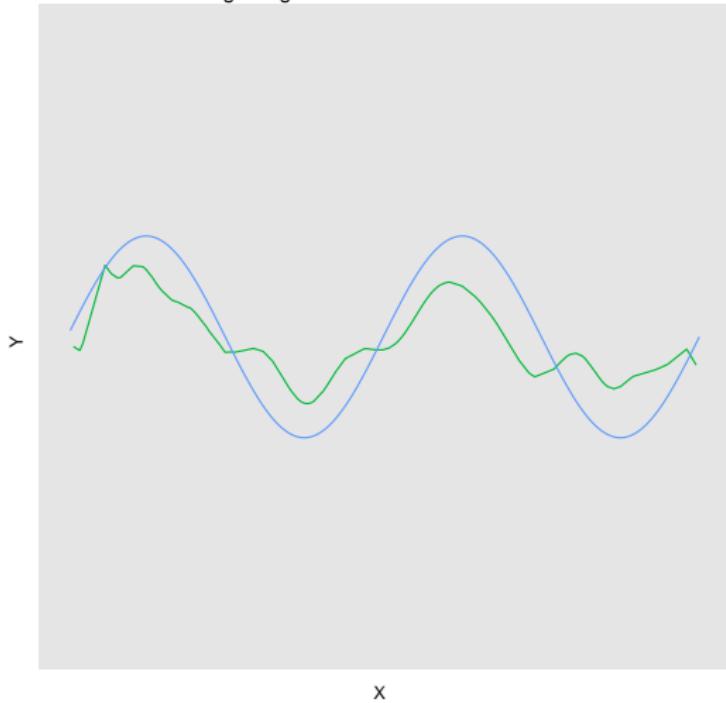
Ridge Regression with Lambda = 1.03



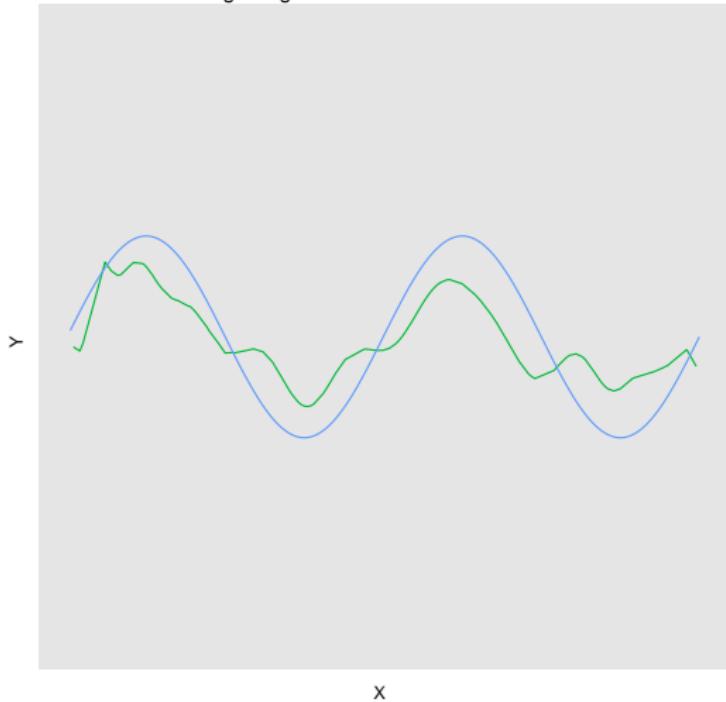
Ridge Regression with Lambda = 0.937



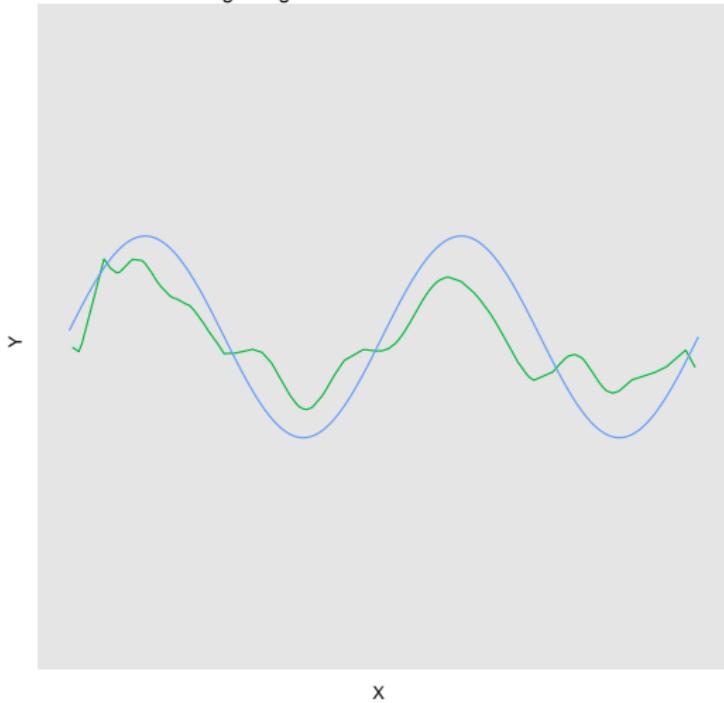
Ridge Regression with Lambda = 0.854



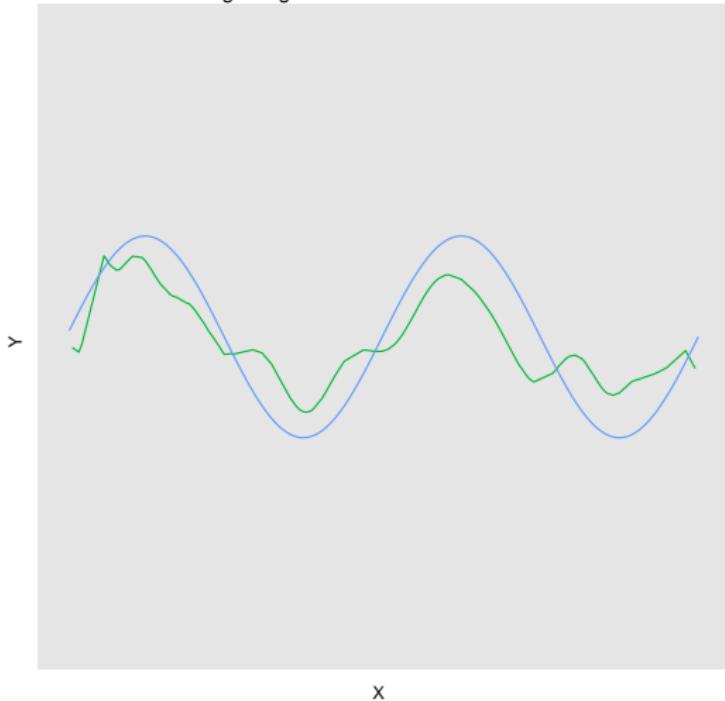
Ridge Regression with Lambda = 0.778



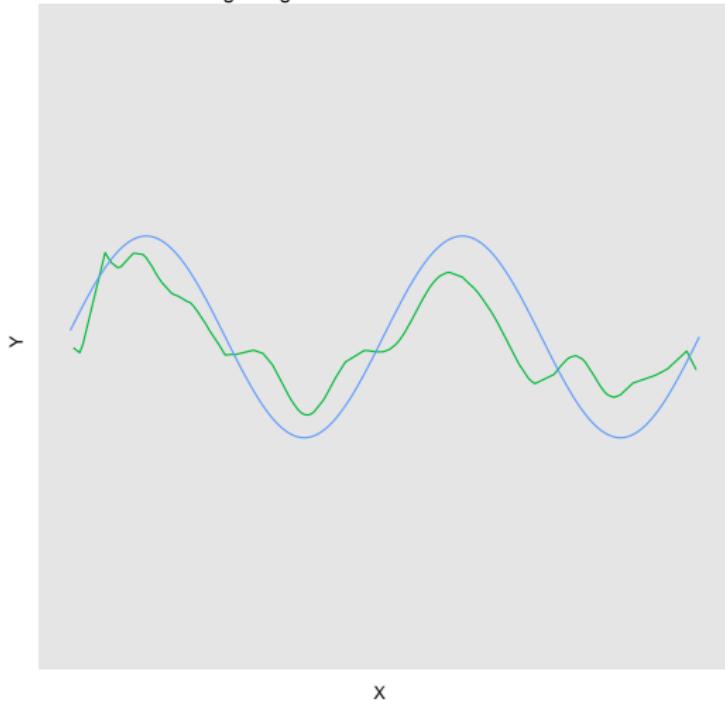
Ridge Regression with Lambda = 0.709



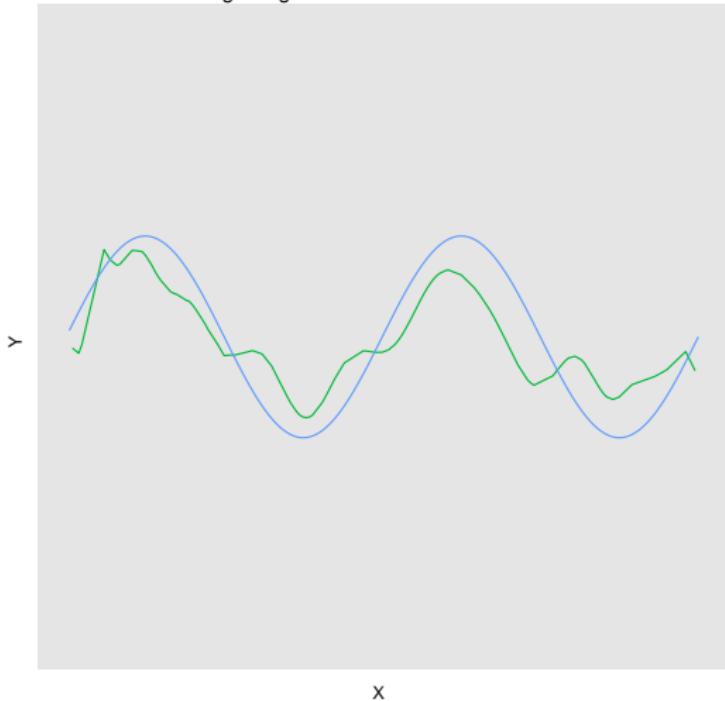
Ridge Regression with Lambda = 0.646



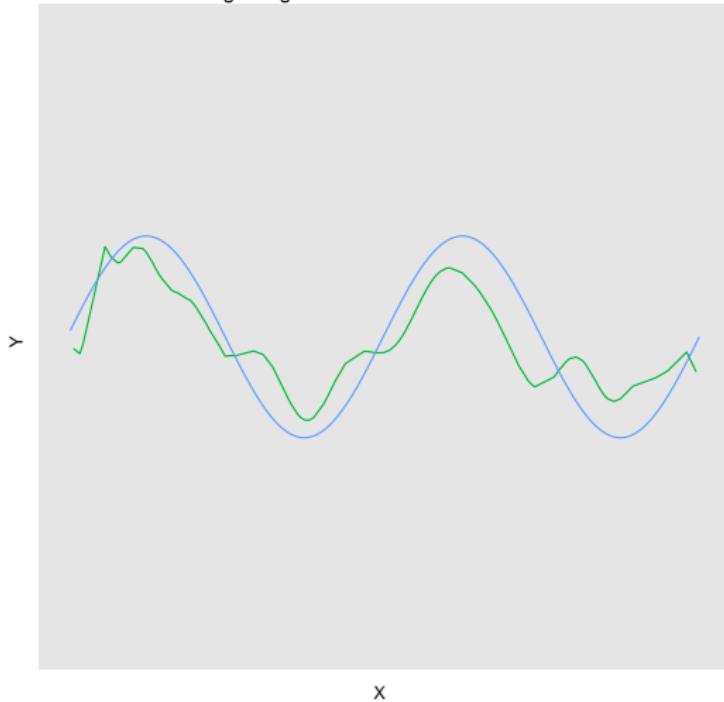
Ridge Regression with Lambda = 0.589



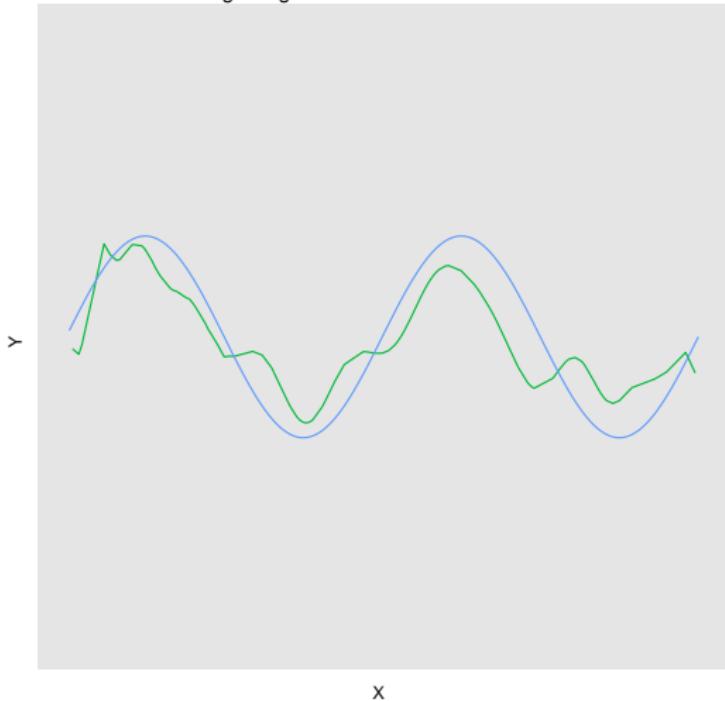
Ridge Regression with Lambda = 0.536



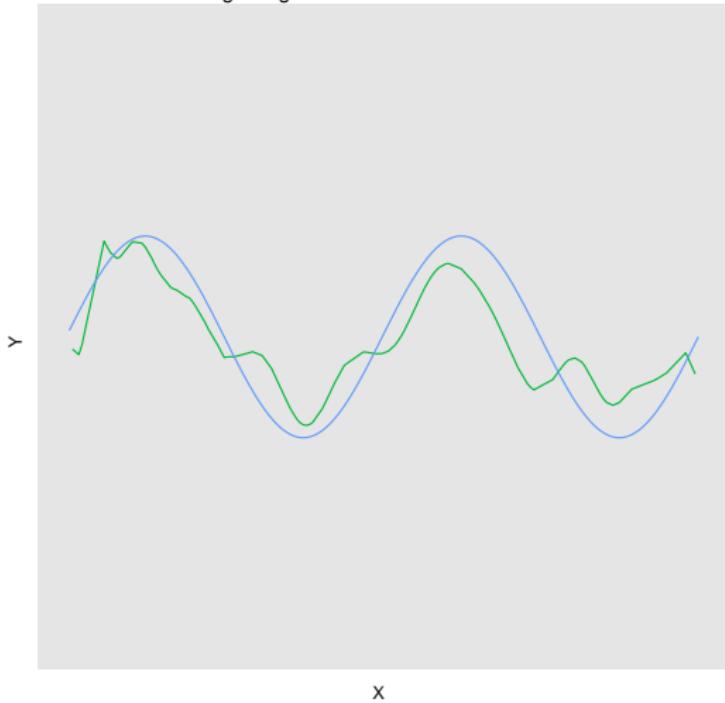
Ridge Regression with Lambda = 0.489



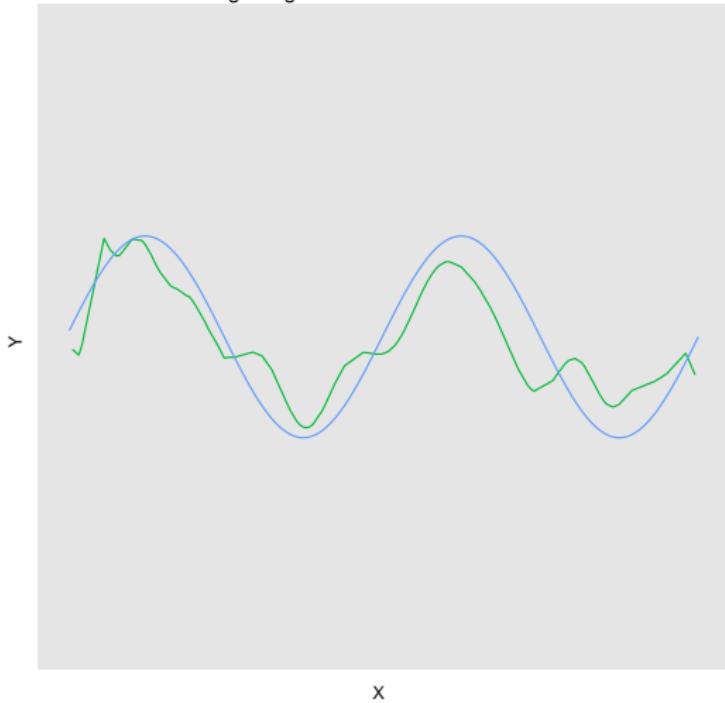
Ridge Regression with Lambda = 0.445



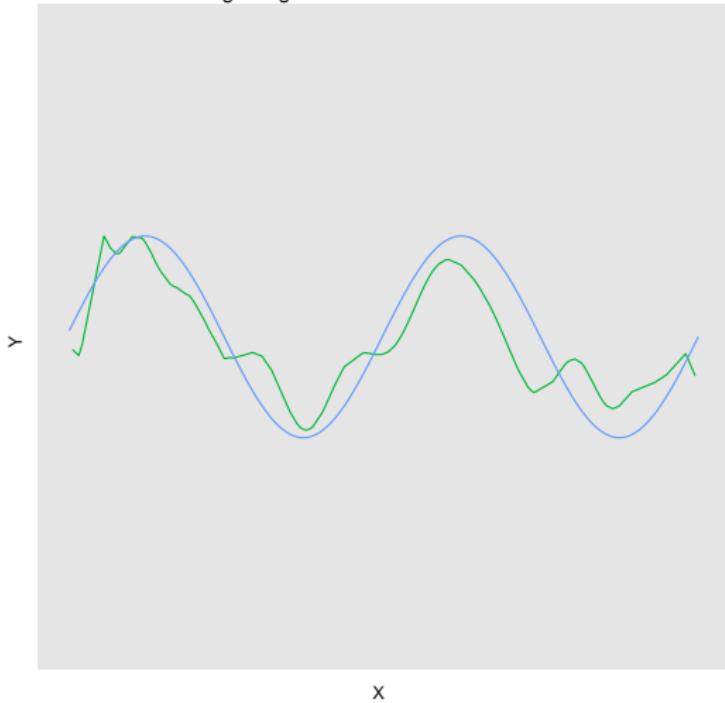
Ridge Regression with Lambda = 0.406



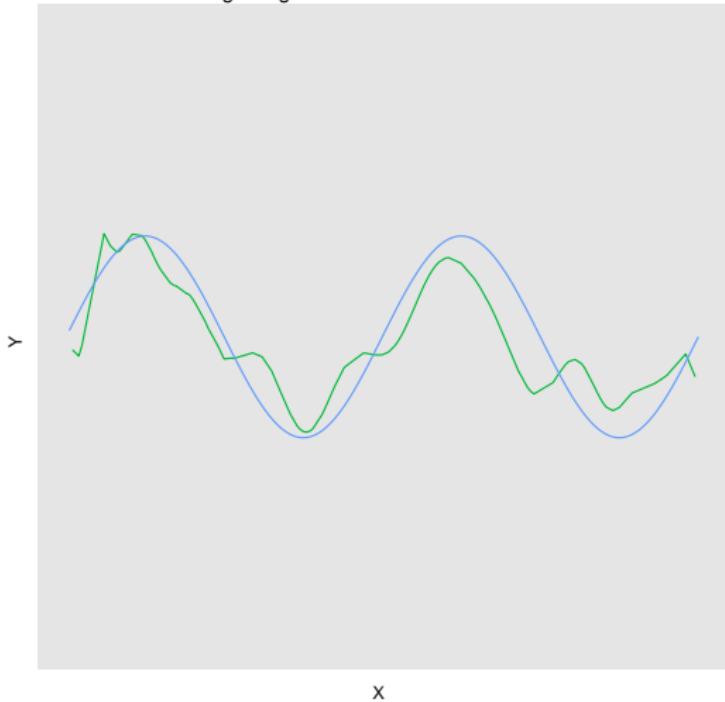
Ridge Regression with Lambda = 0.37



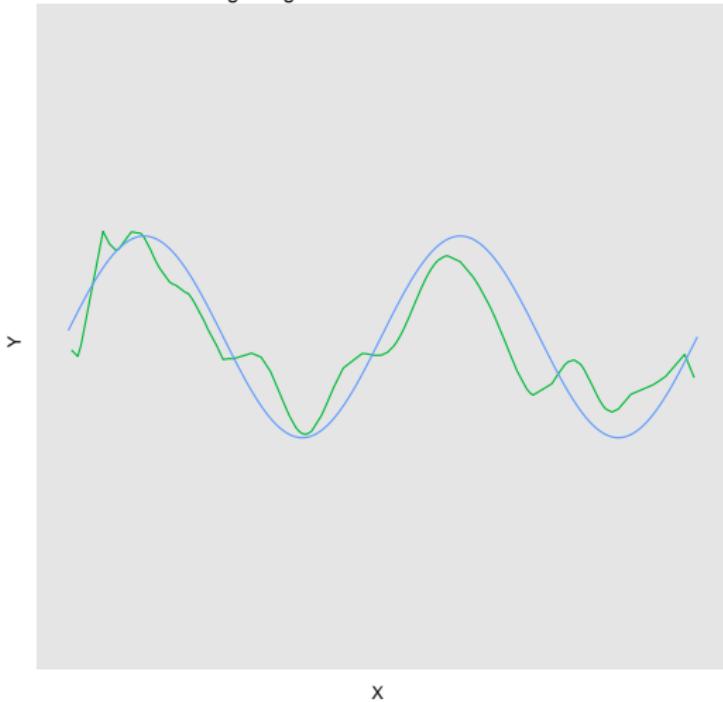
Ridge Regression with Lambda = 0.337



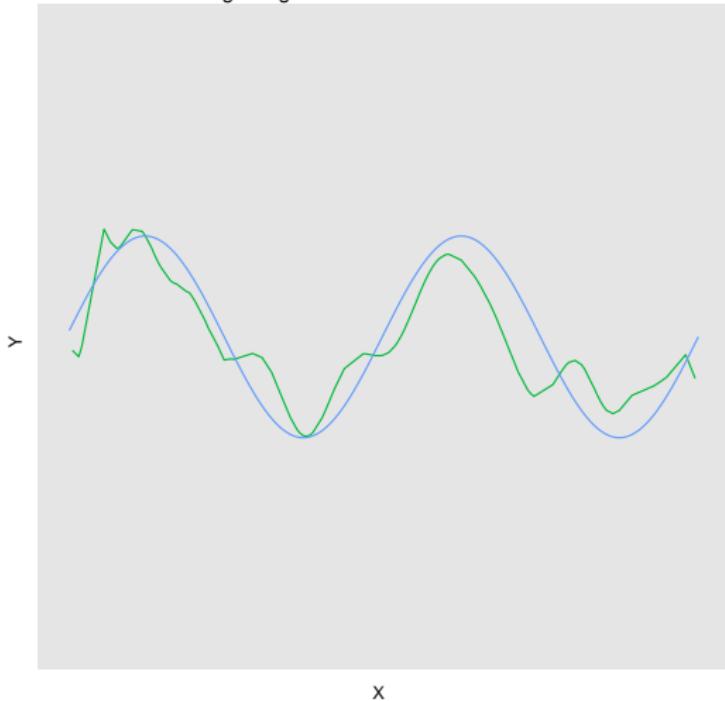
Ridge Regression with Lambda = 0.307



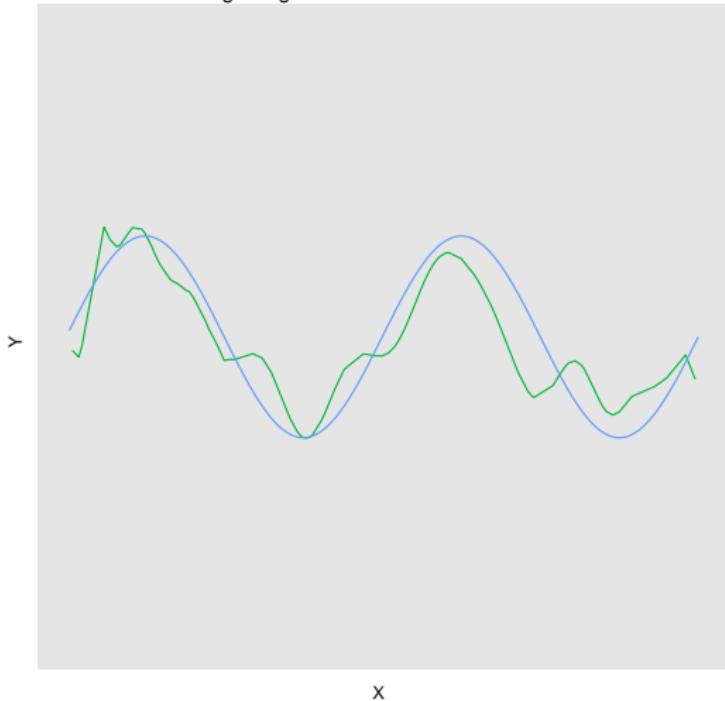
Ridge Regression with Lambda = 0.28



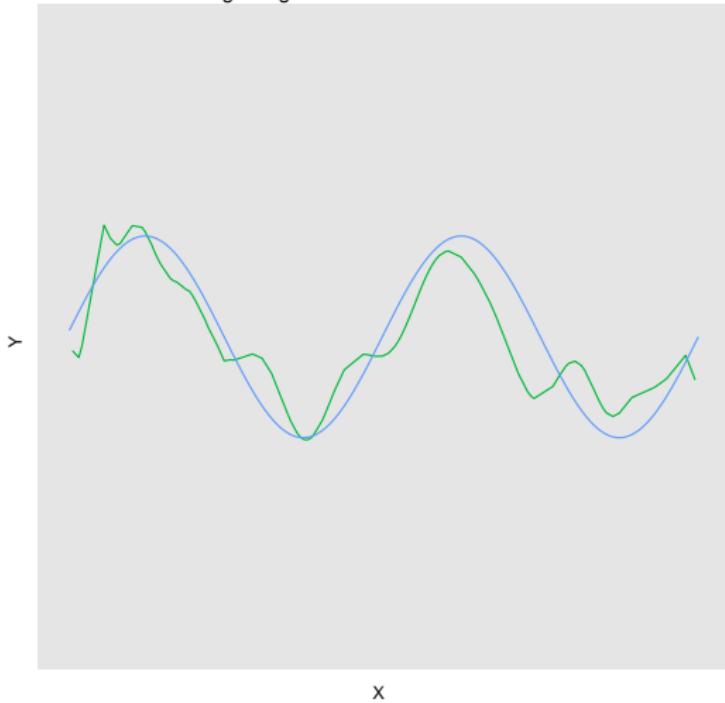
Ridge Regression with Lambda = 0.255



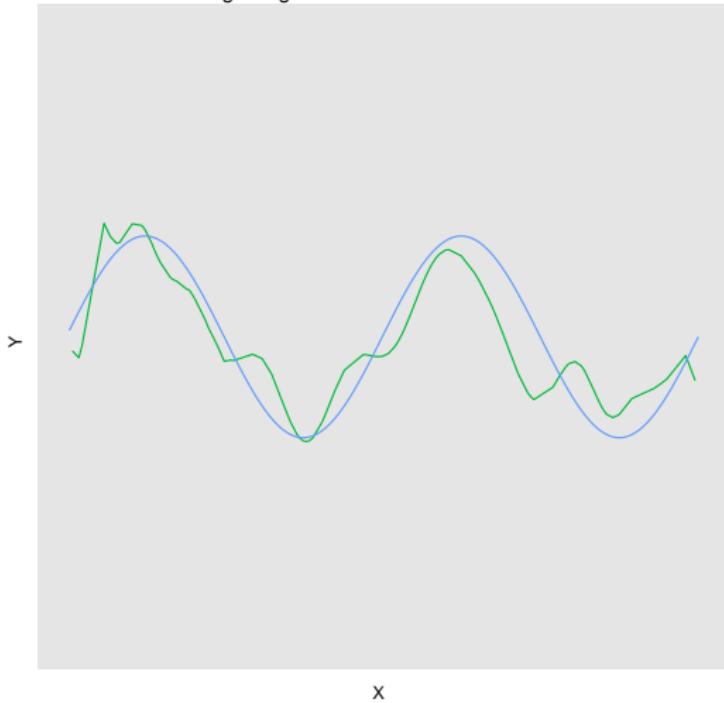
Ridge Regression with Lambda = 0.232



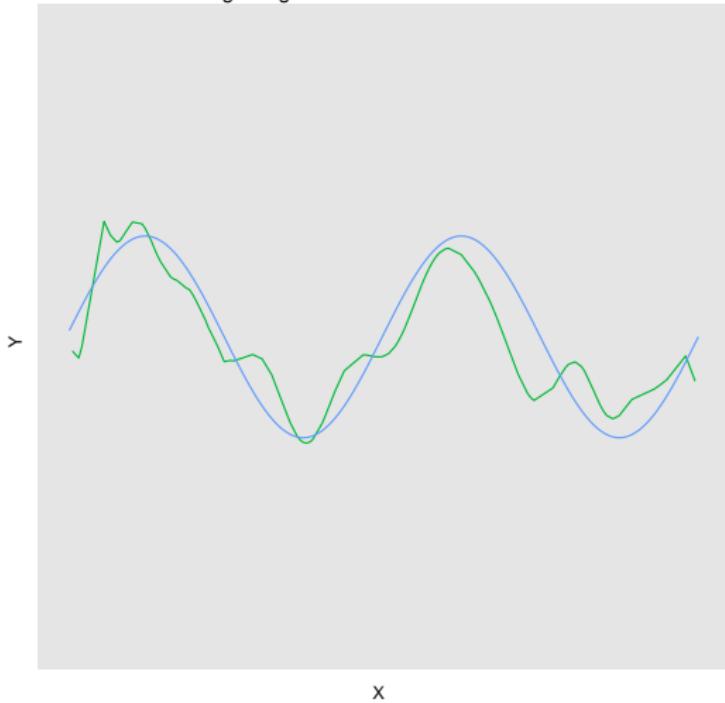
Ridge Regression with Lambda = 0.212



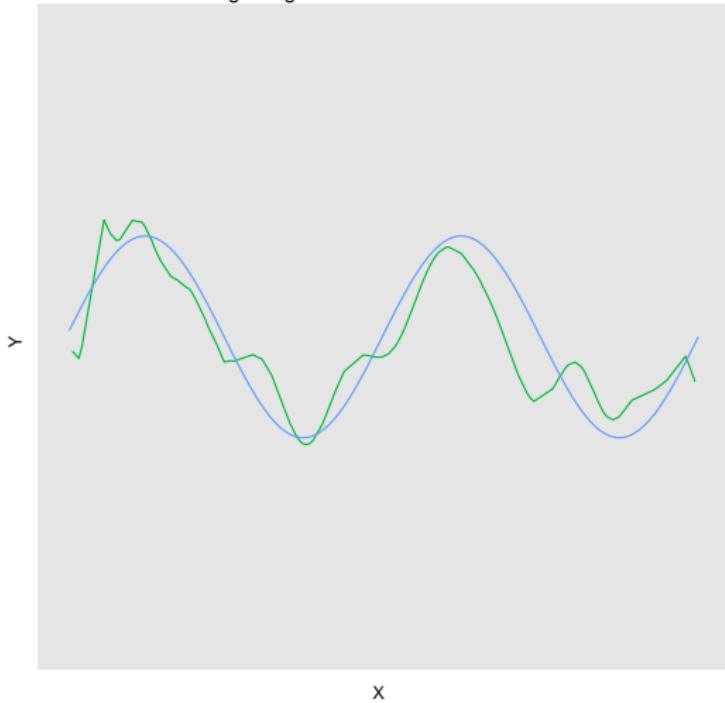
Ridge Regression with Lambda = 0.193



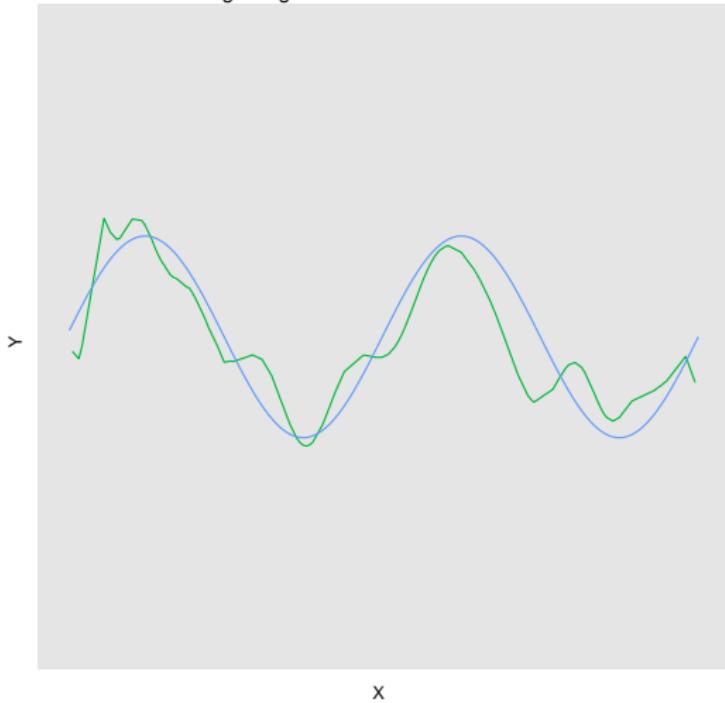
Ridge Regression with Lambda = 0.176



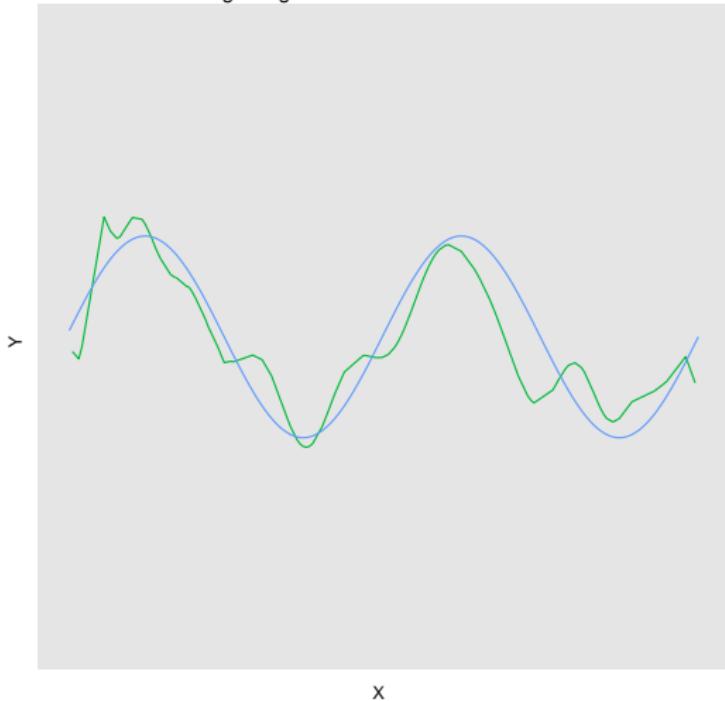
Ridge Regression with Lambda = 0.16



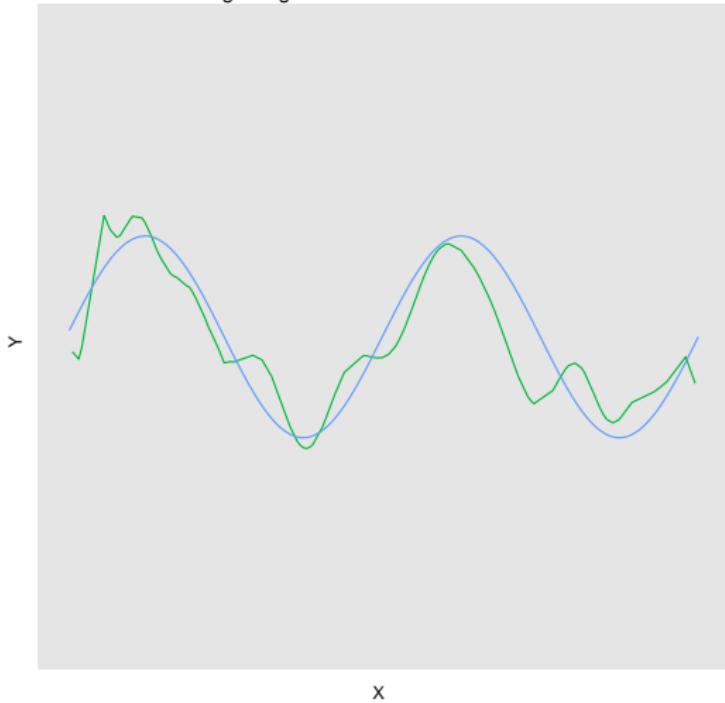
Ridge Regression with Lambda = 0.146



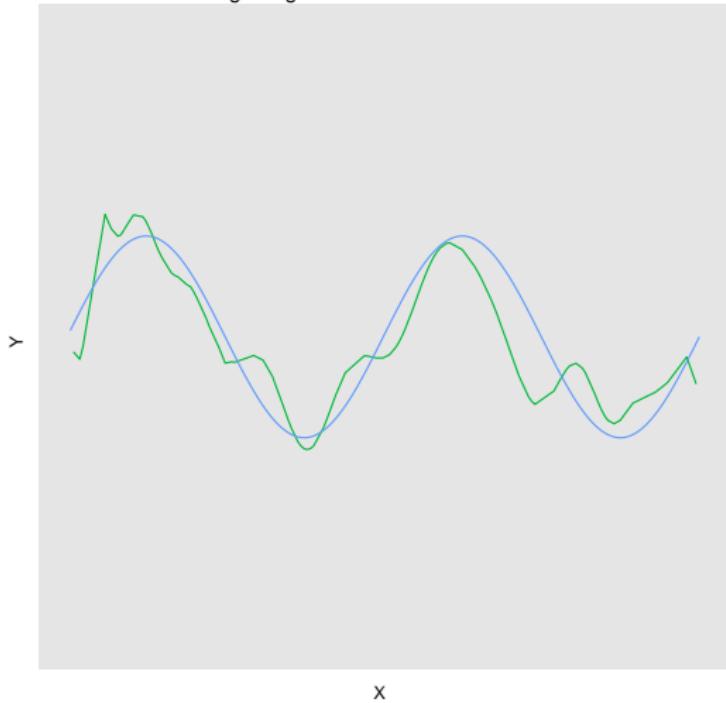
Ridge Regression with Lambda = 0.133



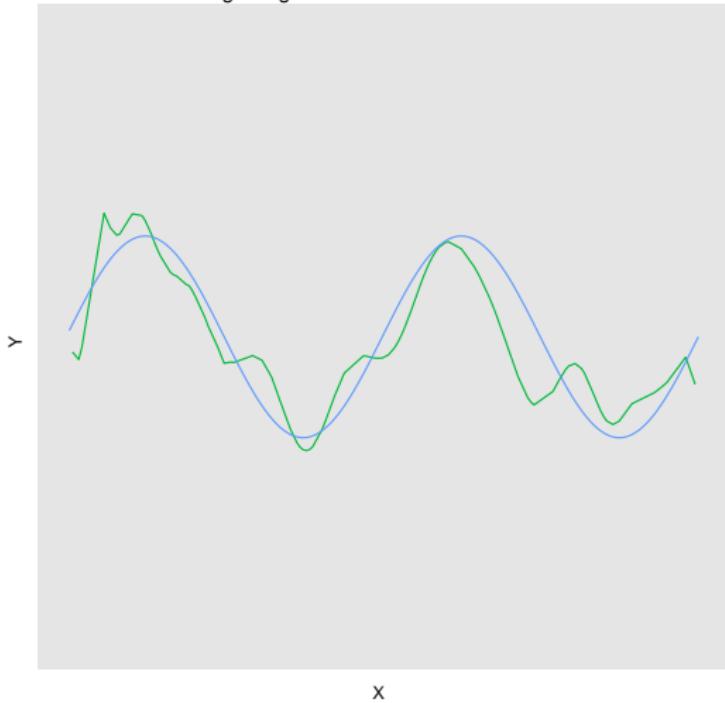
Ridge Regression with Lambda = 0.121



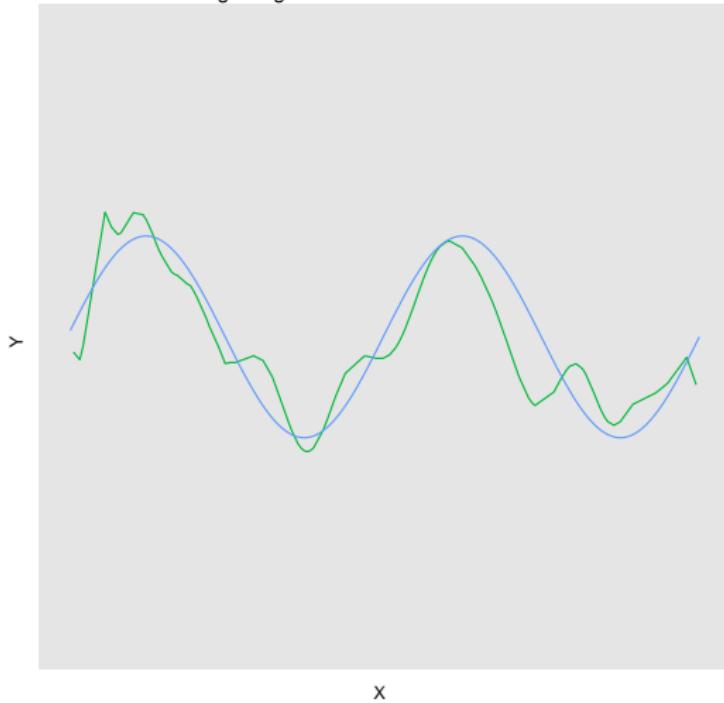
Ridge Regression with Lambda = 0.11



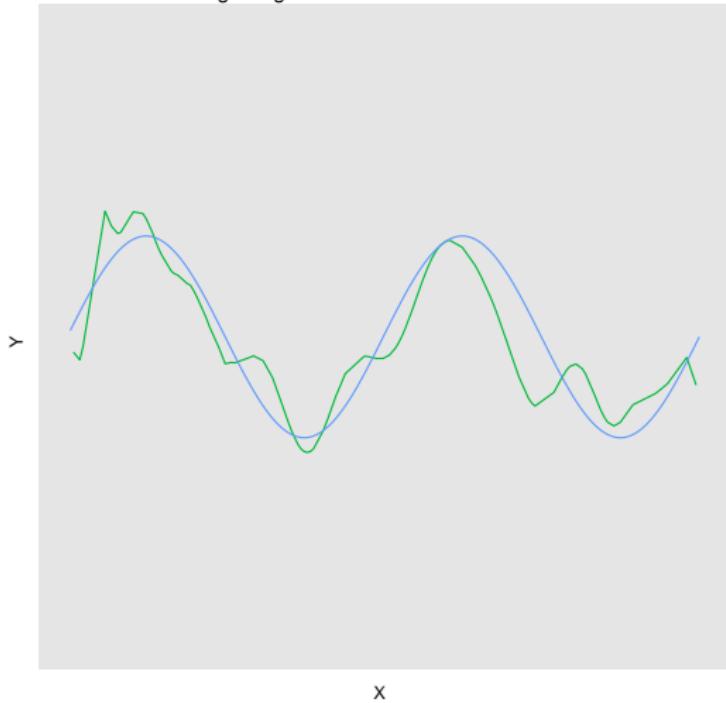
Ridge Regression with Lambda = 0.101



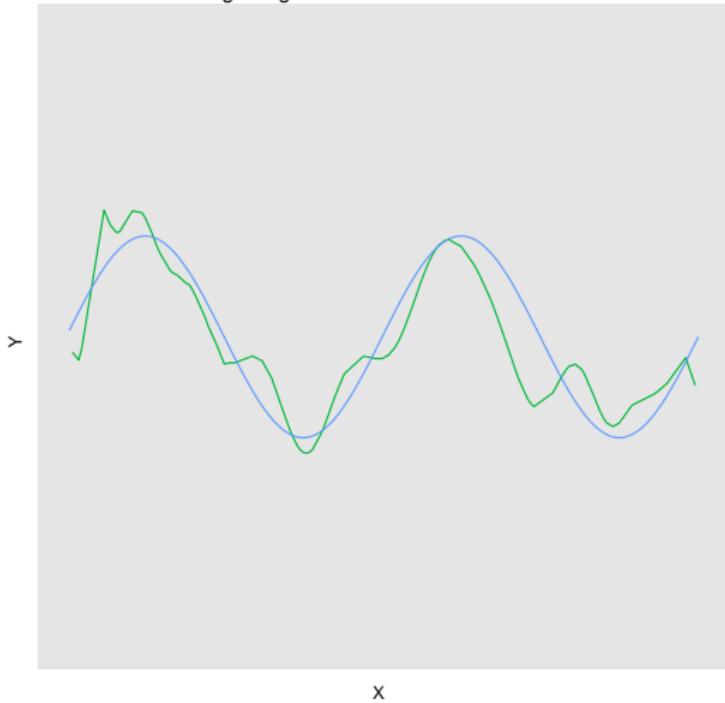
Ridge Regression with Lambda = 0.0916



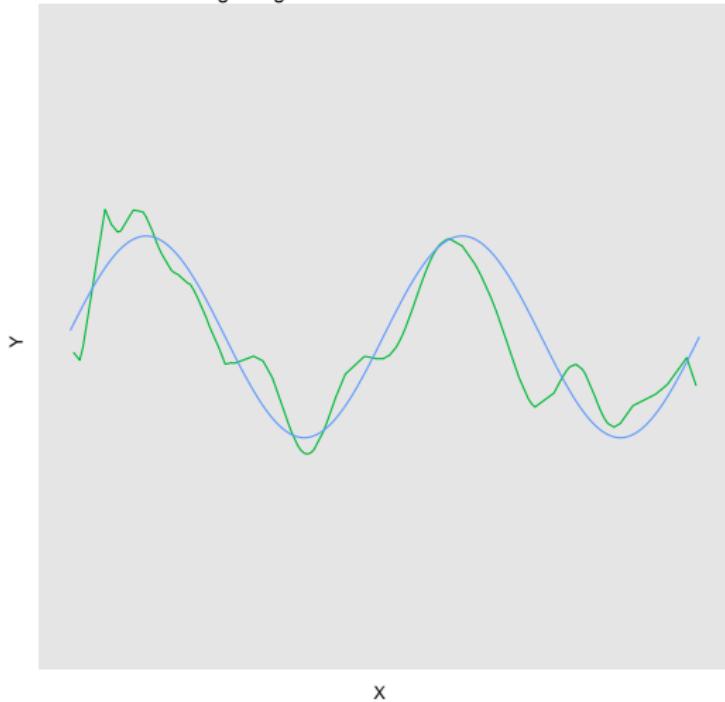
Ridge Regression with Lambda = 0.0835



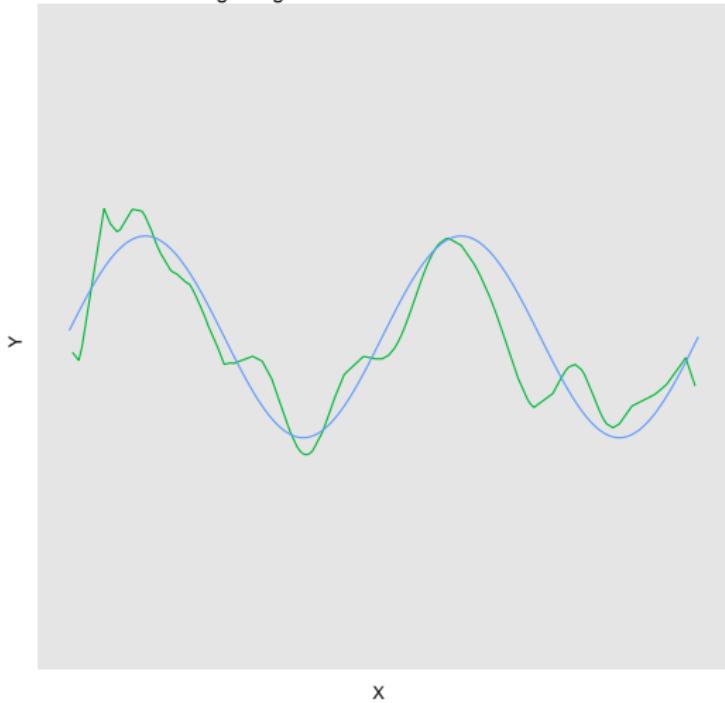
Ridge Regression with Lambda = 0.076



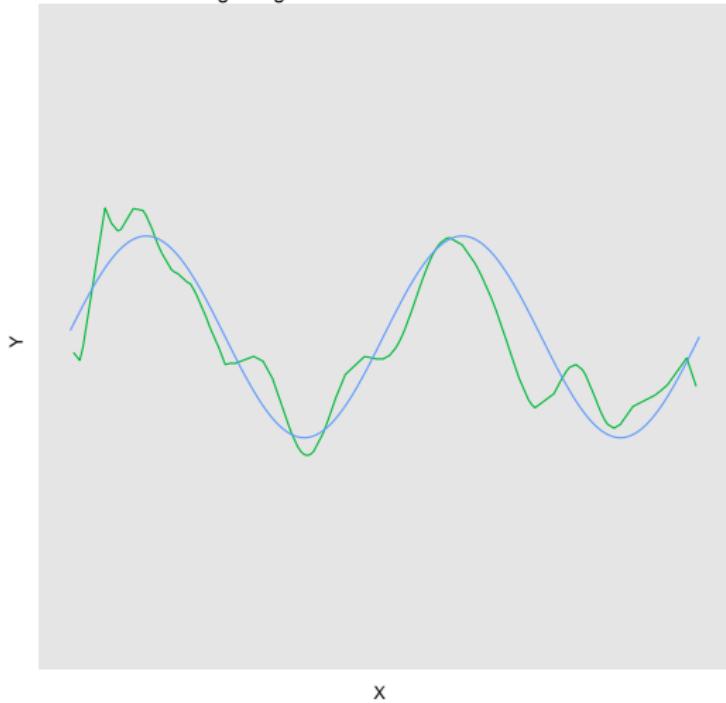
Ridge Regression with Lambda = 0.0693



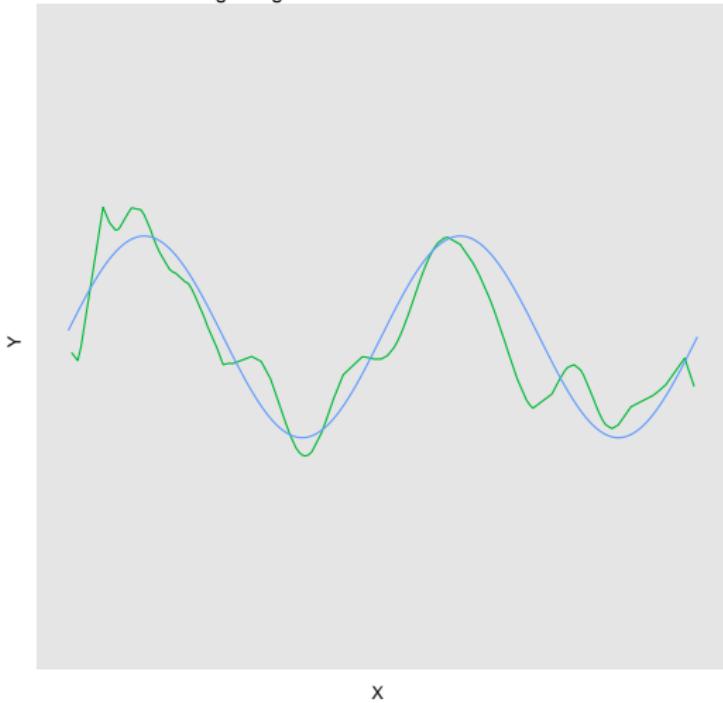
Ridge Regression with Lambda = 0.0631



Ridge Regression with Lambda = 0.0575



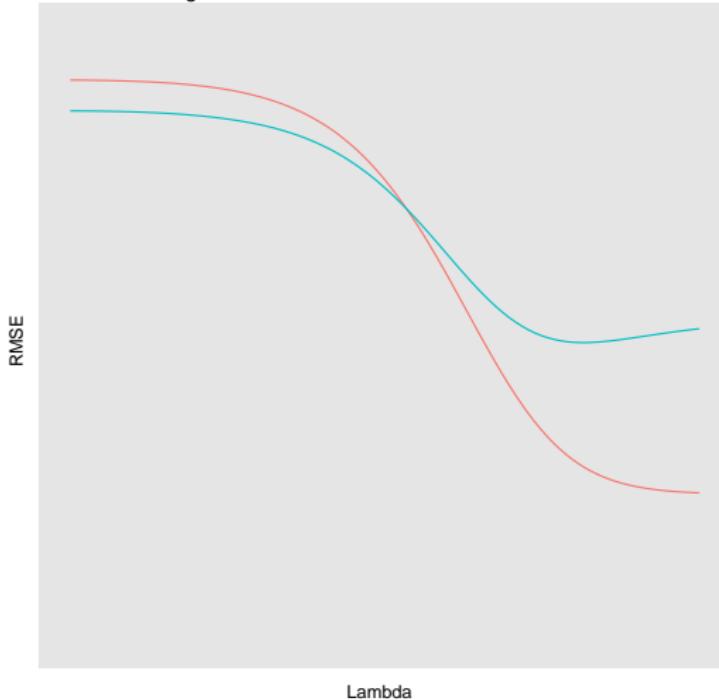
Ridge Regression with Lambda = 0.0524



Regularization forces us to use simpler models

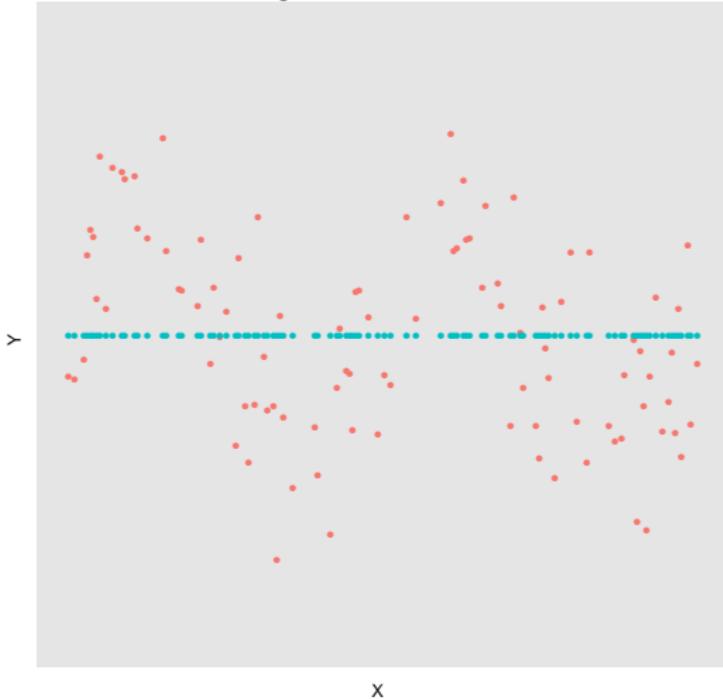
Weaker regularizers help at first, but we will overfit eventually

Training Set Performance versus Test Set Performance

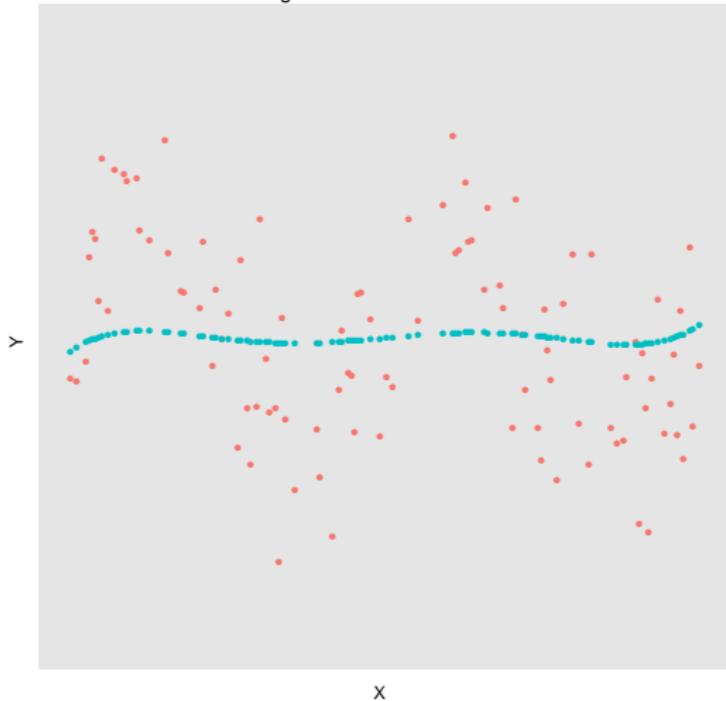


What happens if we use the Lasso instead of ridge regression?

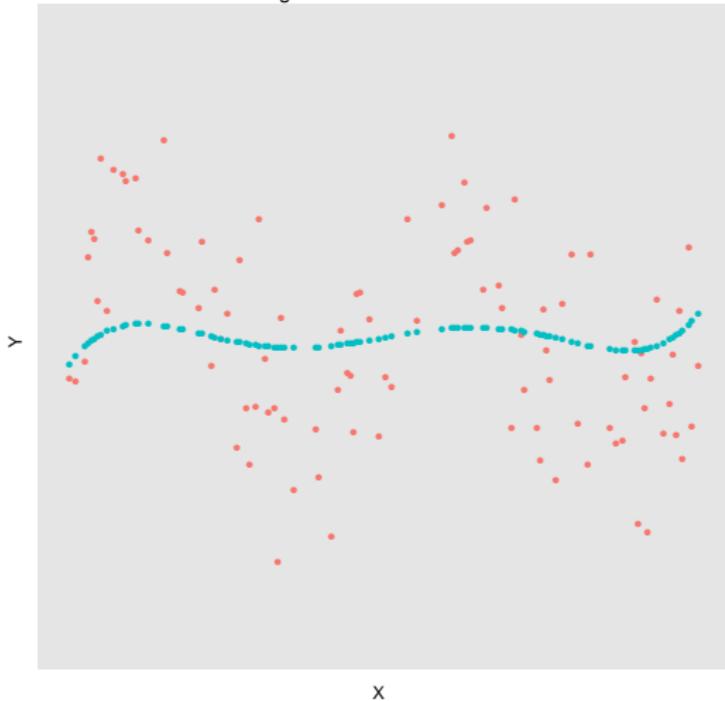
Lasso Regression with Lambda = 0.524



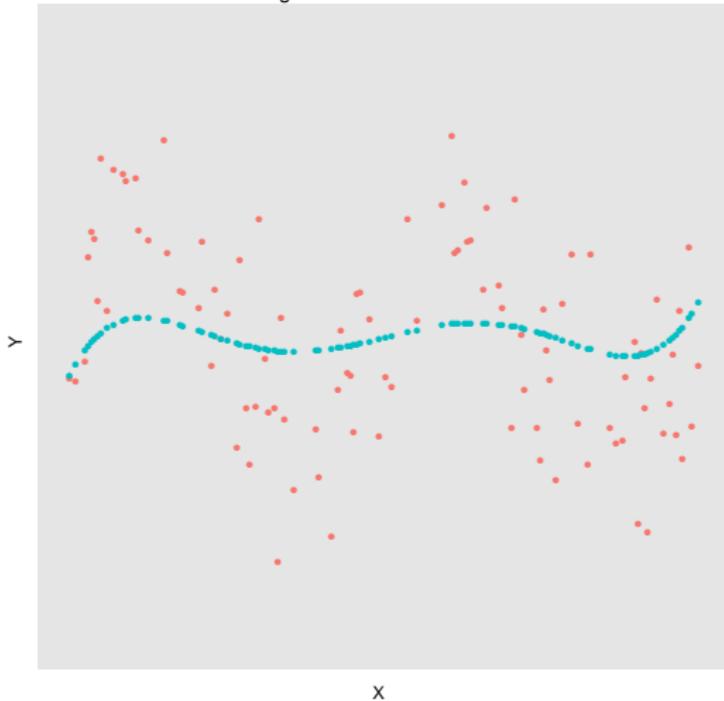
Lasso Regression with Lambda = 0.478



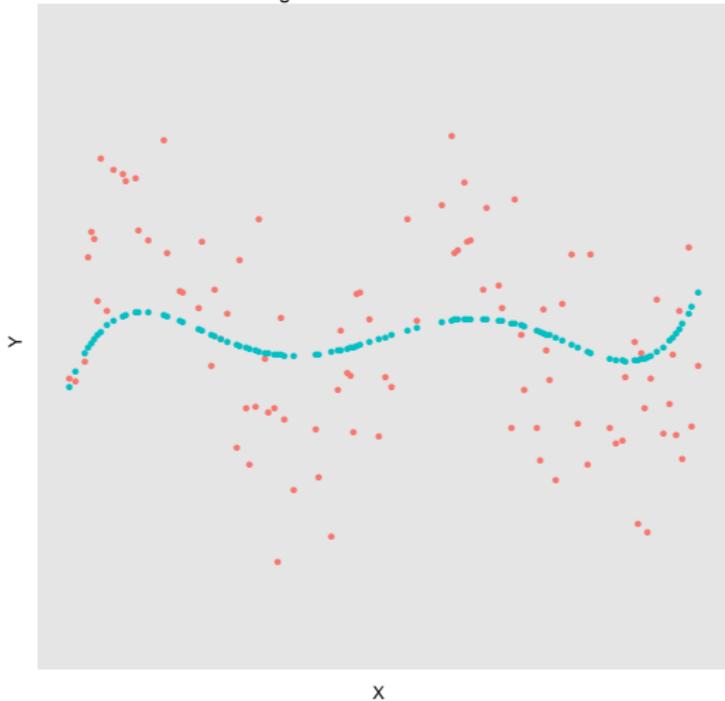
Lasso Regression with Lambda = 0.435



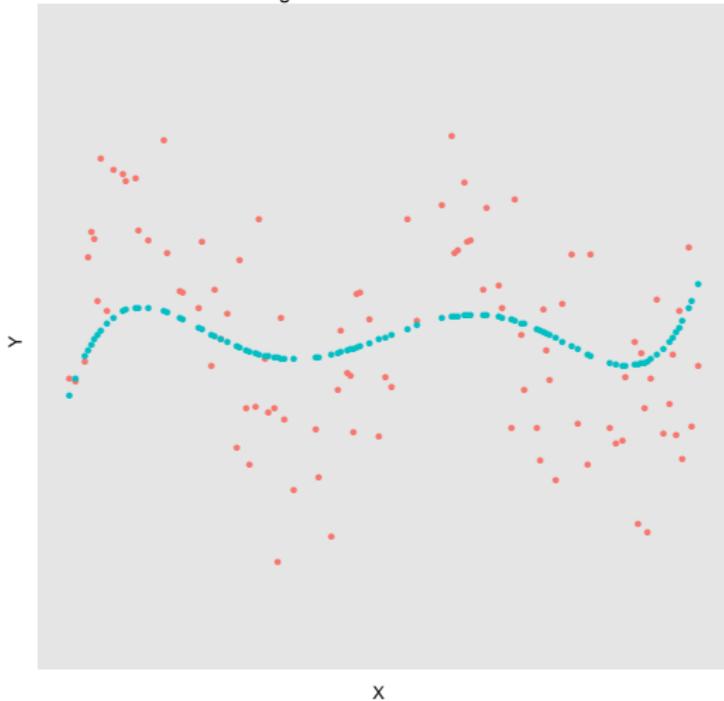
Lasso Regression with Lambda = 0.396



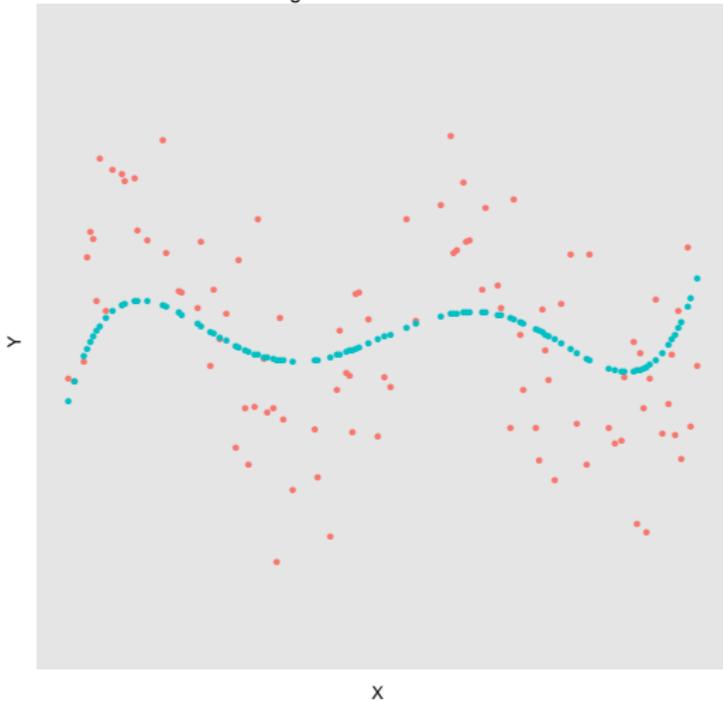
Lasso Regression with Lambda = 0.361



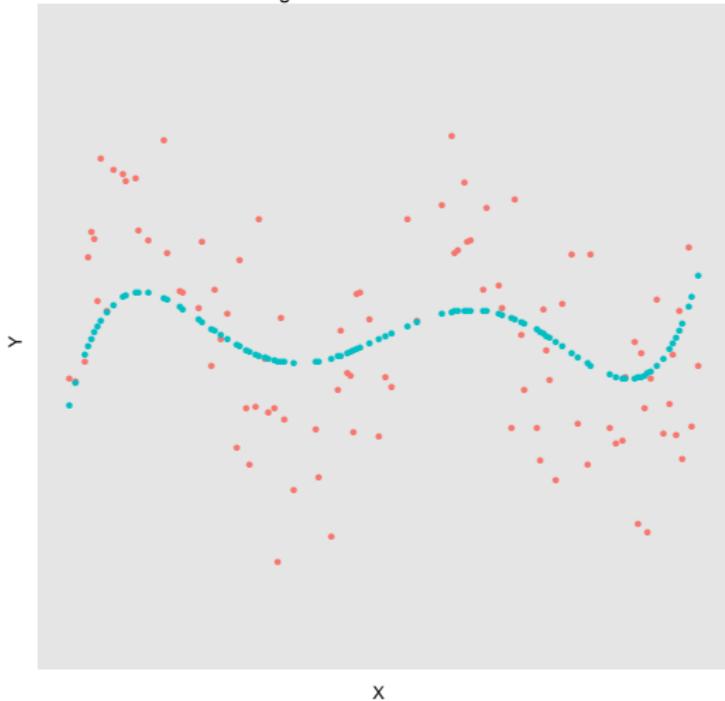
Lasso Regression with Lambda = 0.329



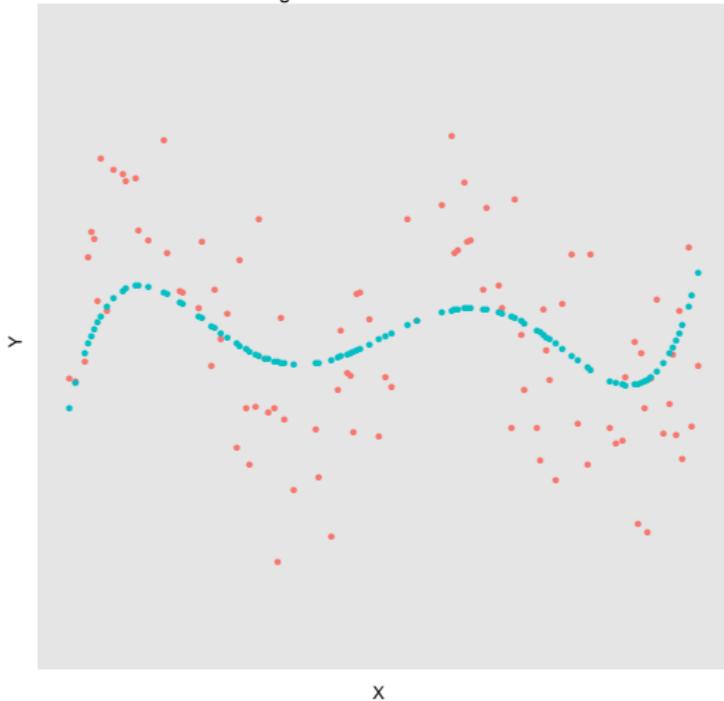
Lasso Regression with Lambda = 0.3



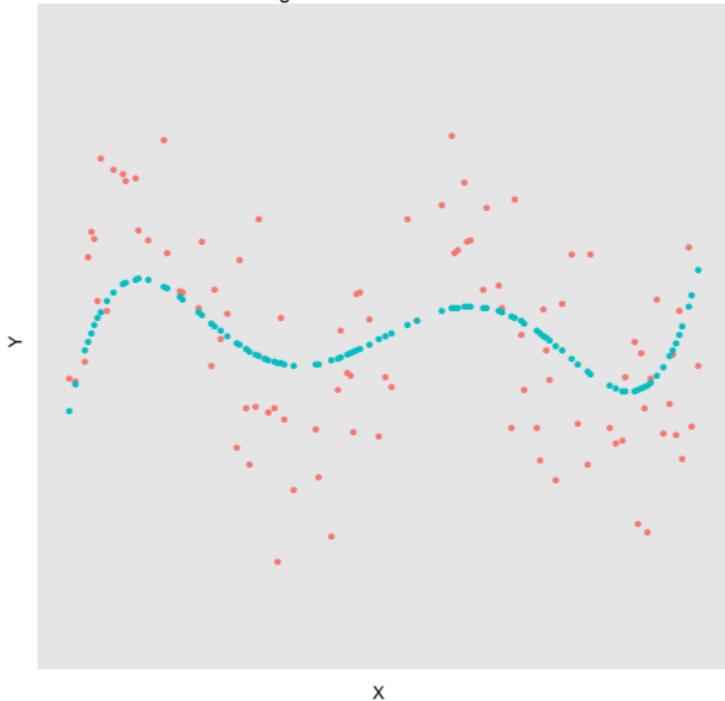
Lasso Regression with Lambda = 0.273



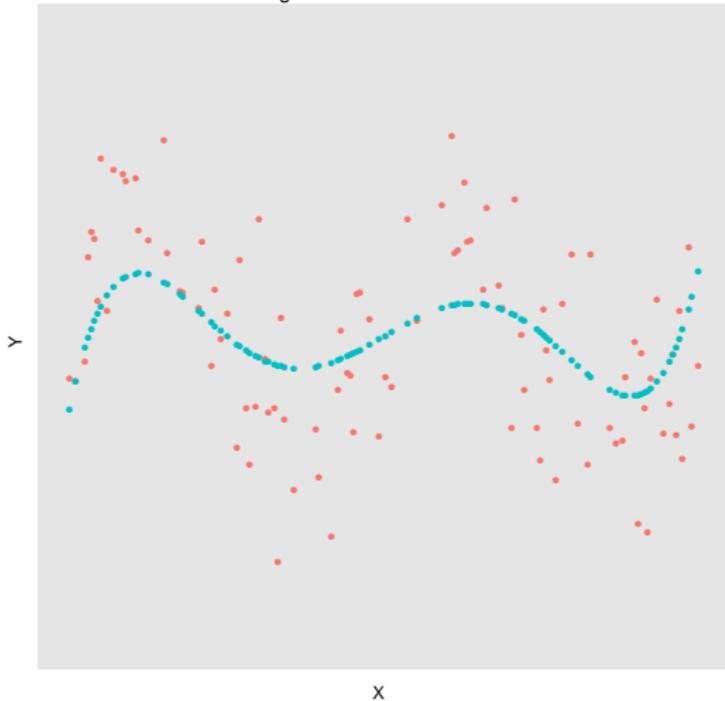
Lasso Regression with Lambda = 0.249



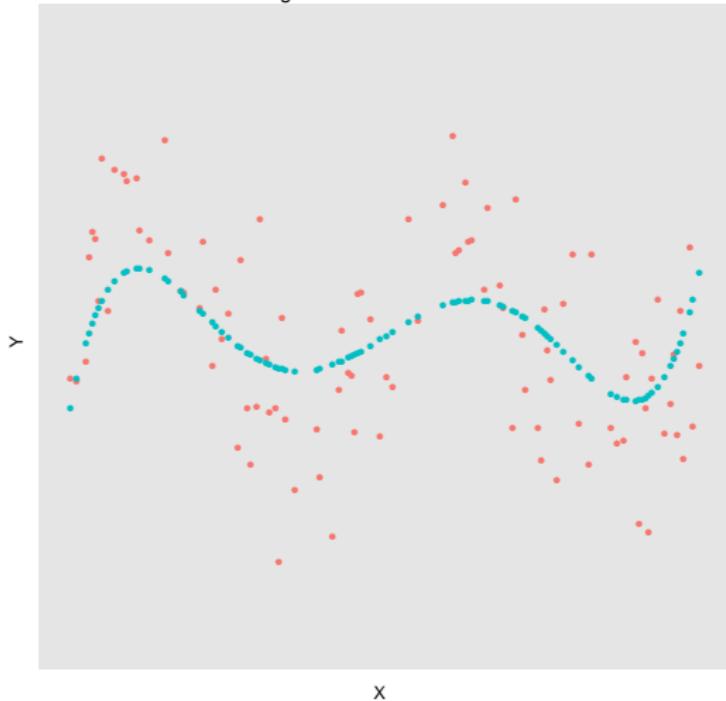
Lasso Regression with Lambda = 0.227



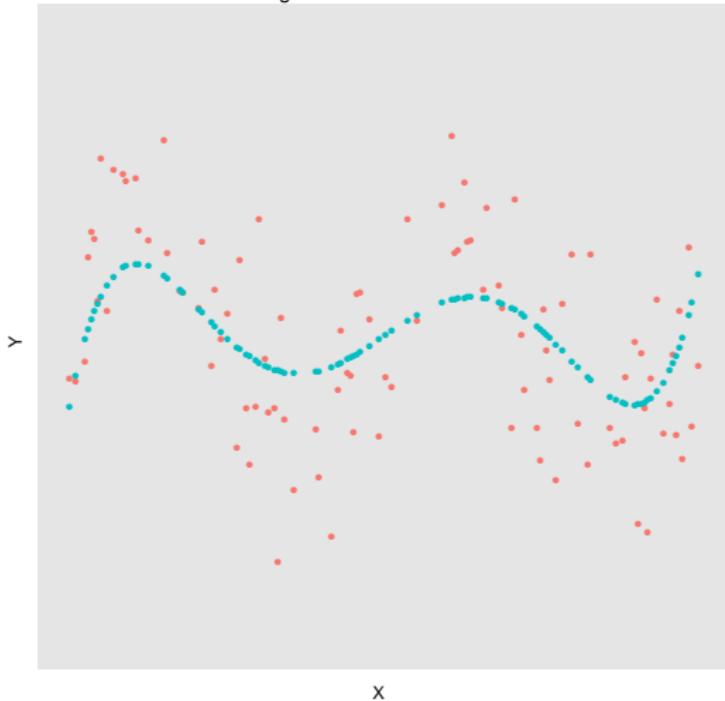
Lasso Regression with Lambda = 0.207



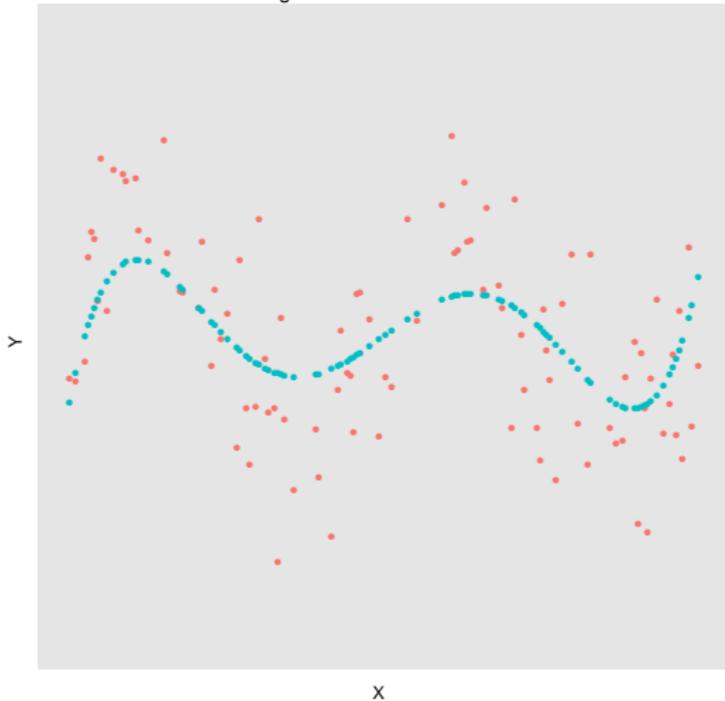
Lasso Regression with Lambda = 0.188



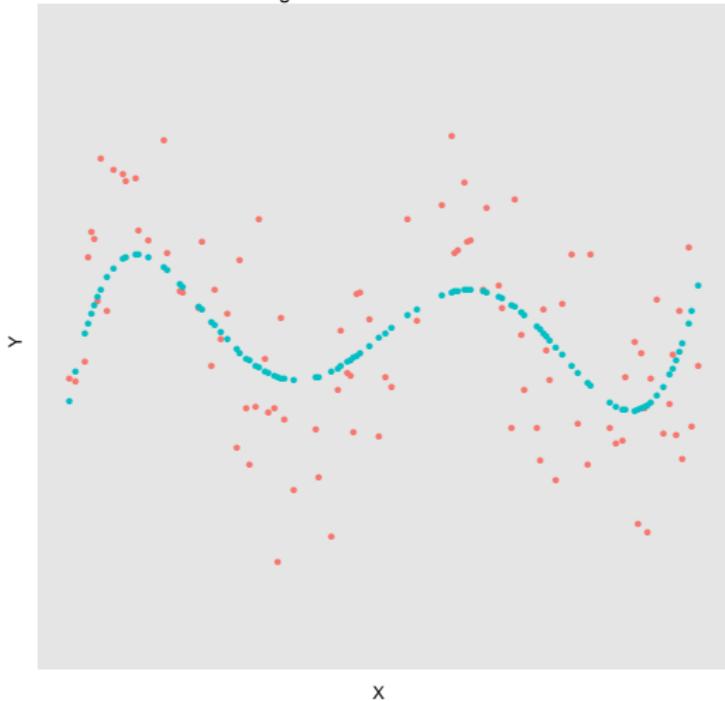
Lasso Regression with Lambda = 0.172



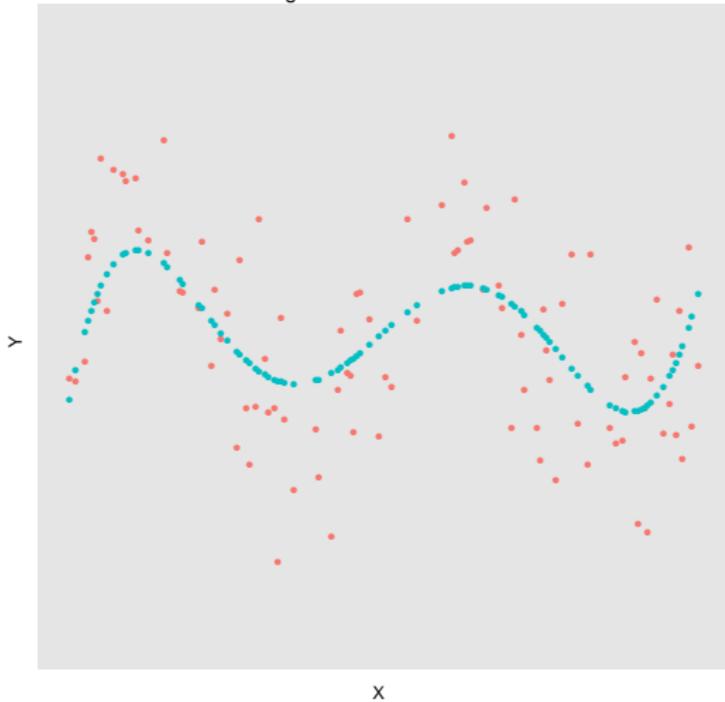
Lasso Regression with Lambda = 0.156



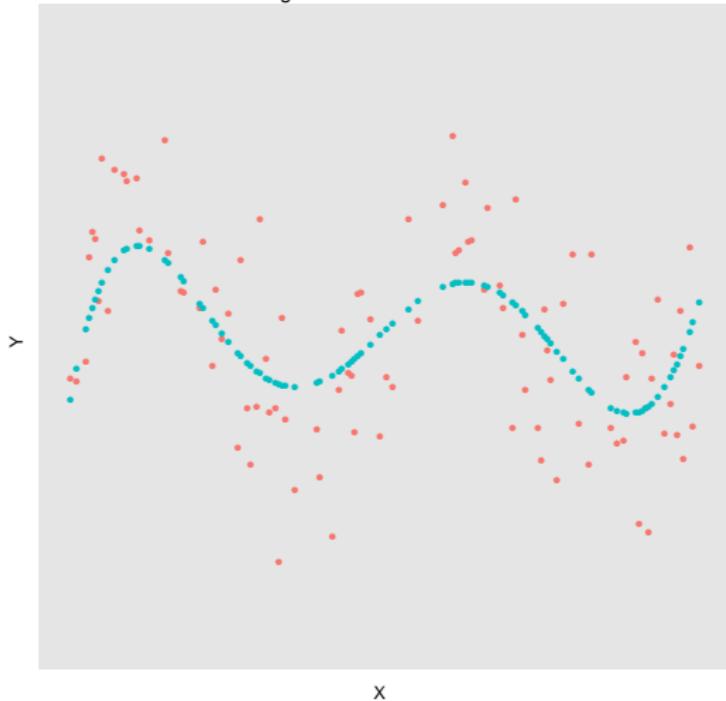
Lasso Regression with Lambda = 0.142



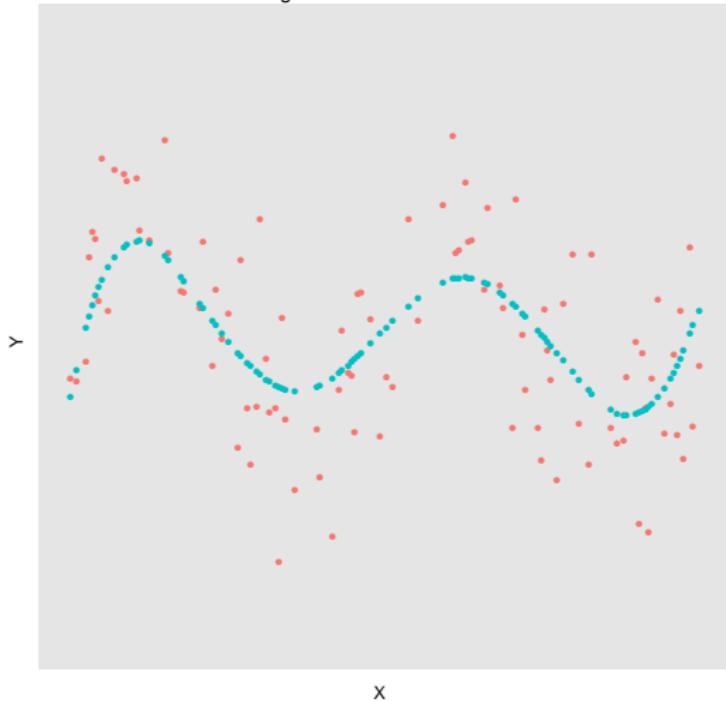
Lasso Regression with Lambda = 0.13



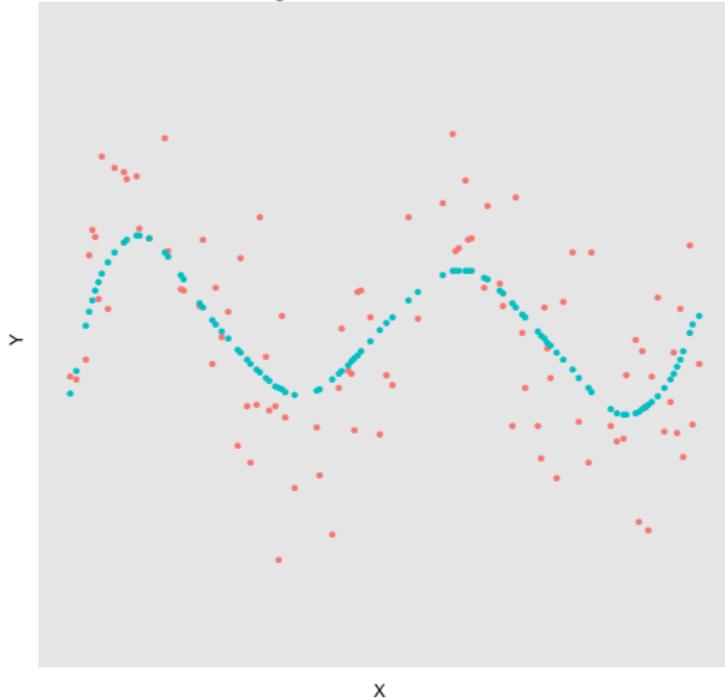
Lasso Regression with Lambda = 0.118



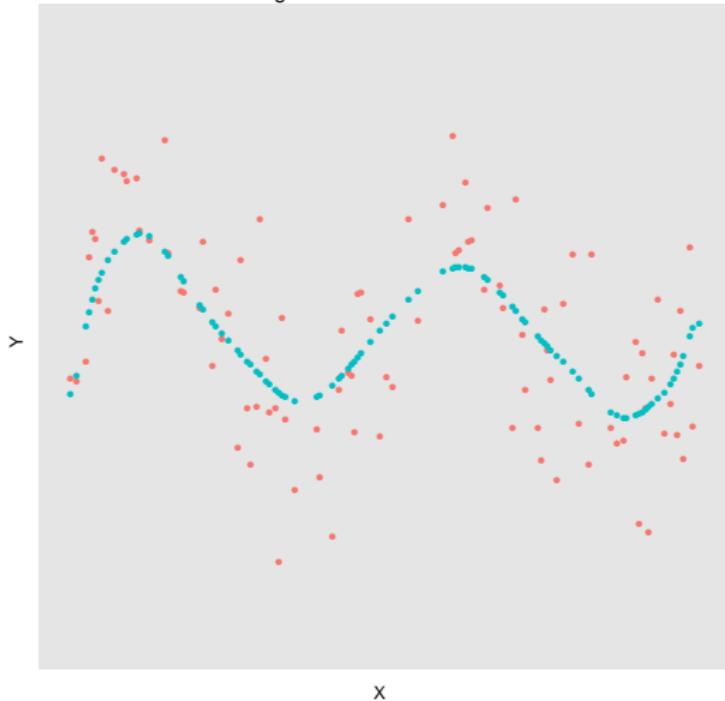
Lasso Regression with Lambda = 0.108



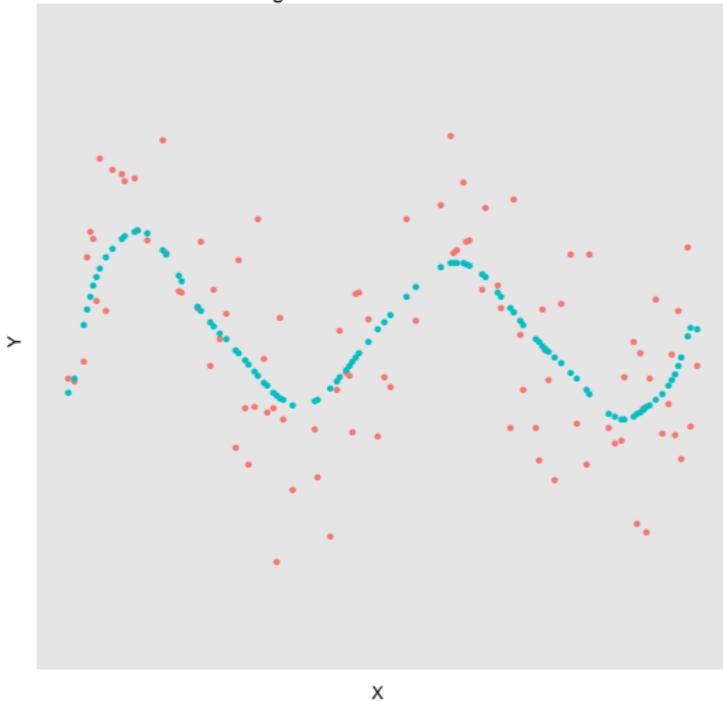
Lasso Regression with Lambda = 0.0982



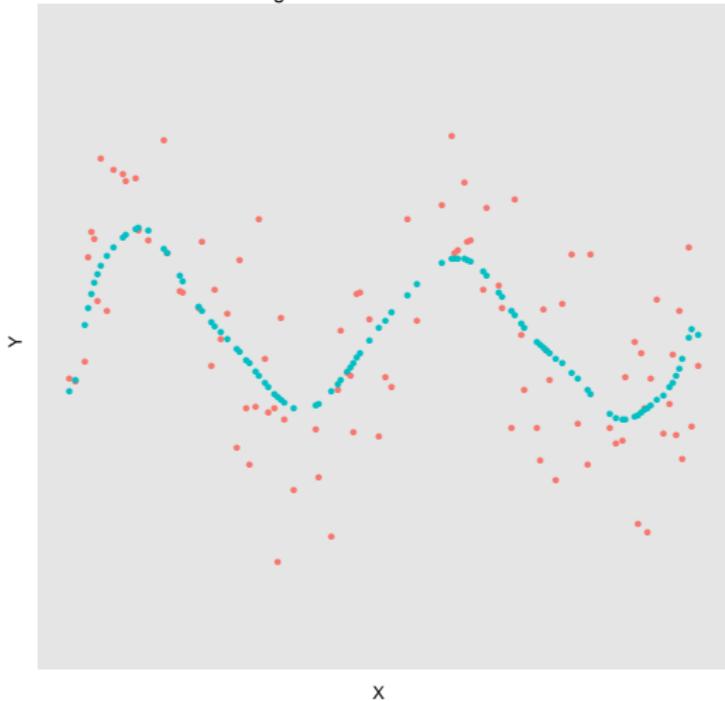
Lasso Regression with Lambda = 0.0895



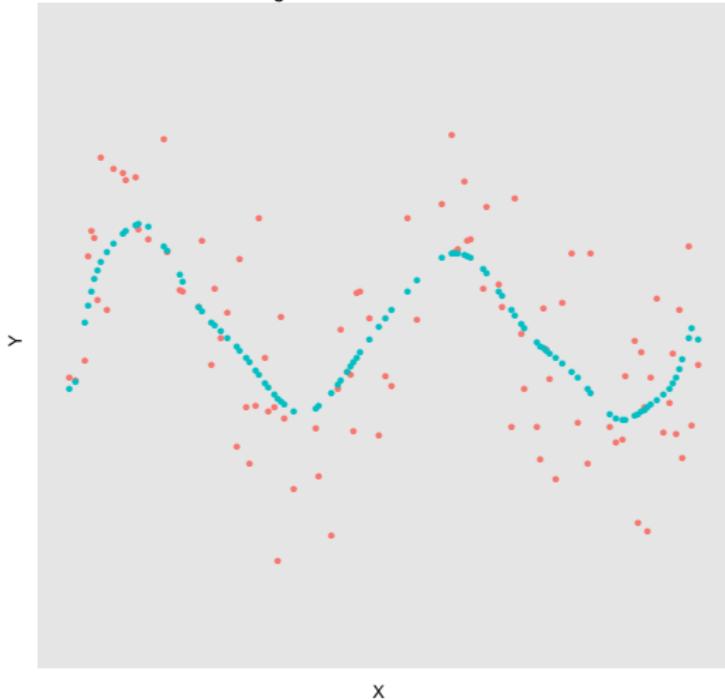
Lasso Regression with Lambda = 0.0815



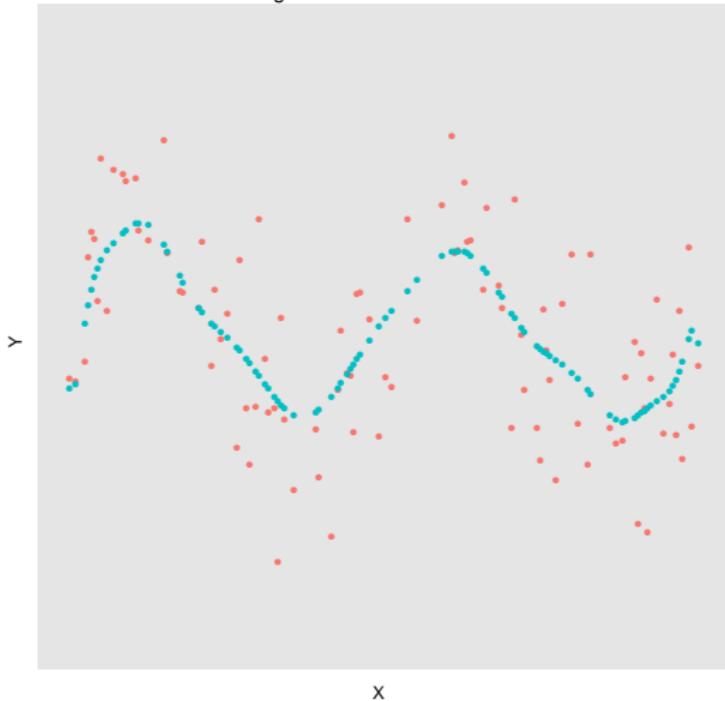
Lasso Regression with Lambda = 0.0743



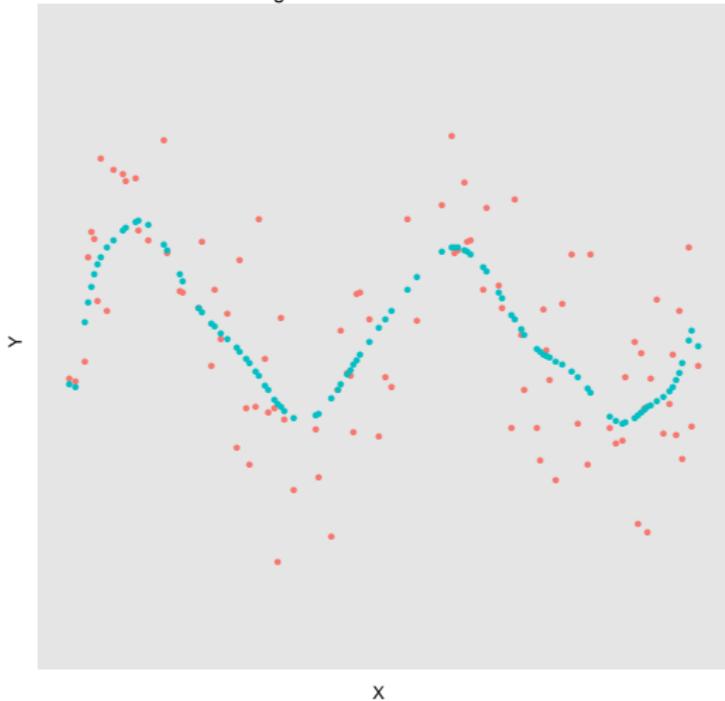
Lasso Regression with Lambda = 0.0677



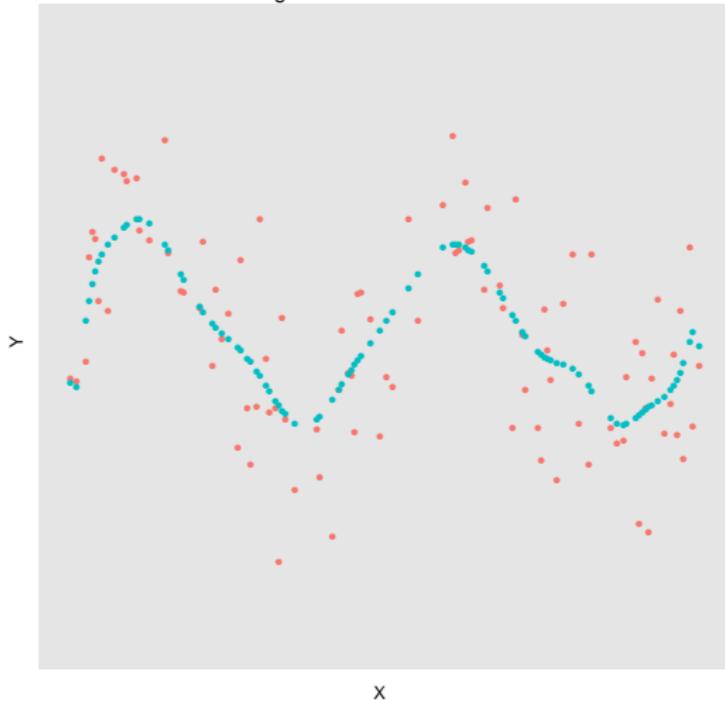
Lasso Regression with Lambda = 0.0617



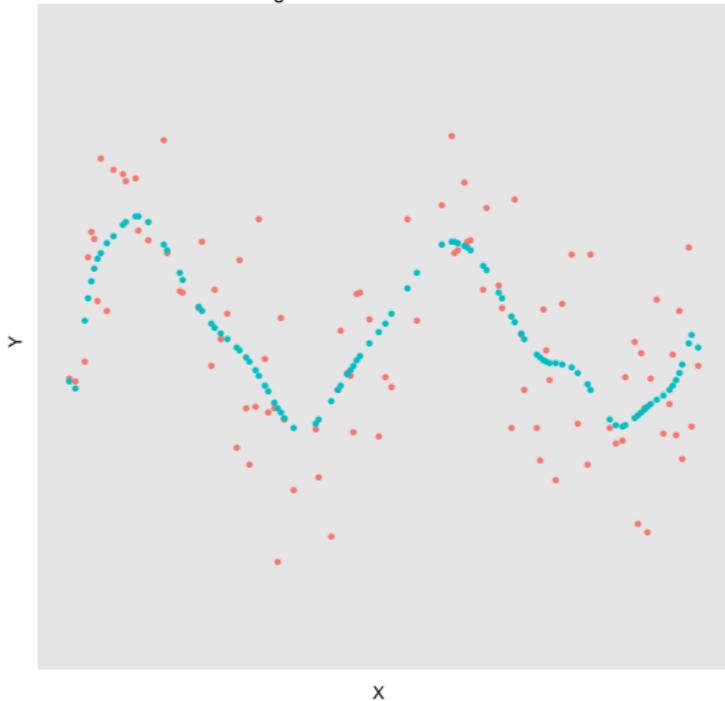
Lasso Regression with Lambda = 0.0562



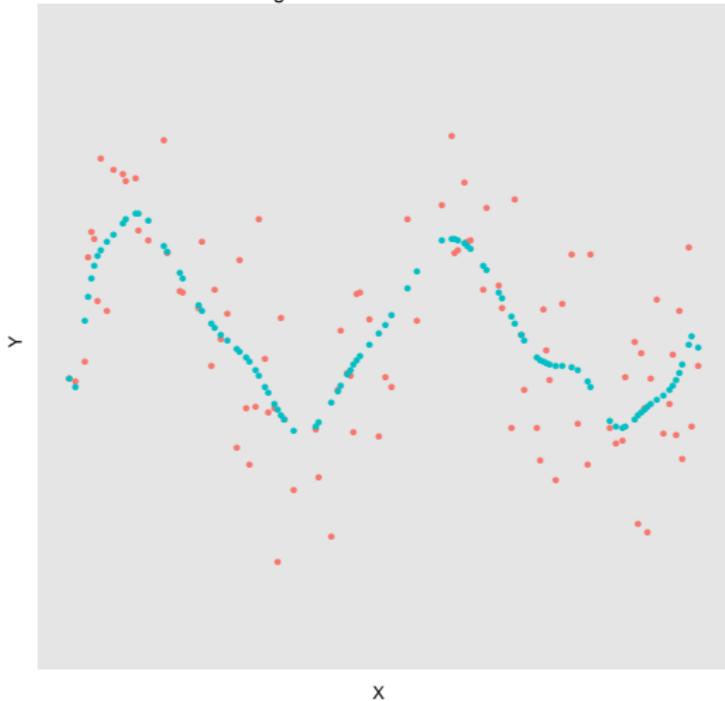
Lasso Regression with Lambda = 0.0512



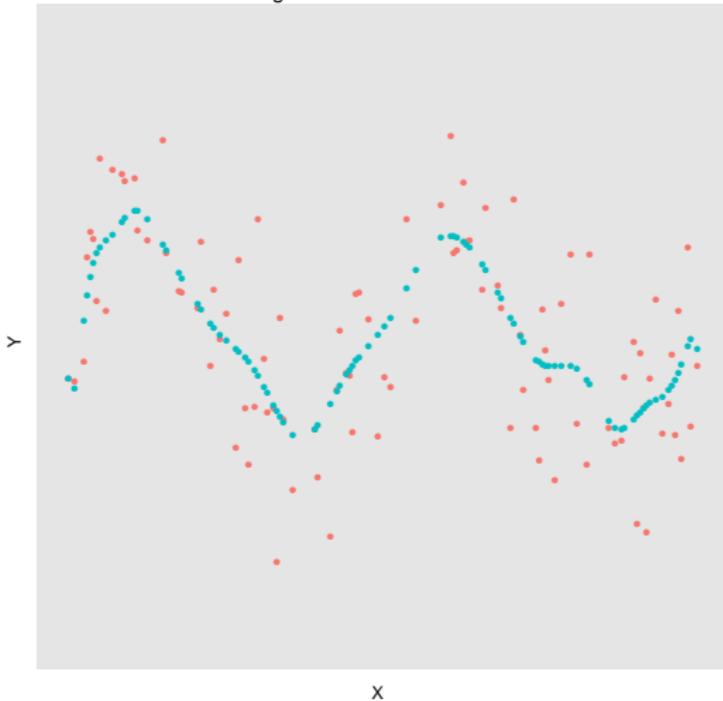
Lasso Regression with Lambda = 0.0467



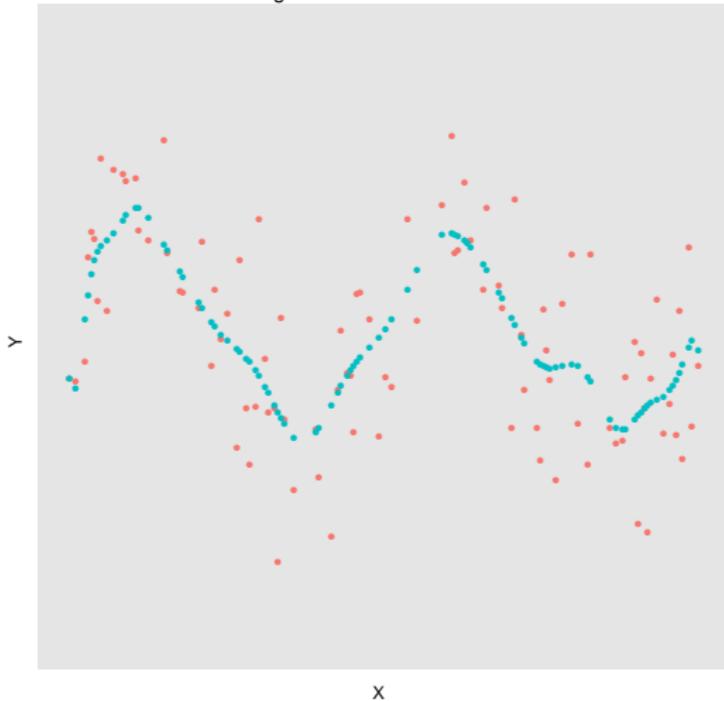
Lasso Regression with Lambda = 0.0425



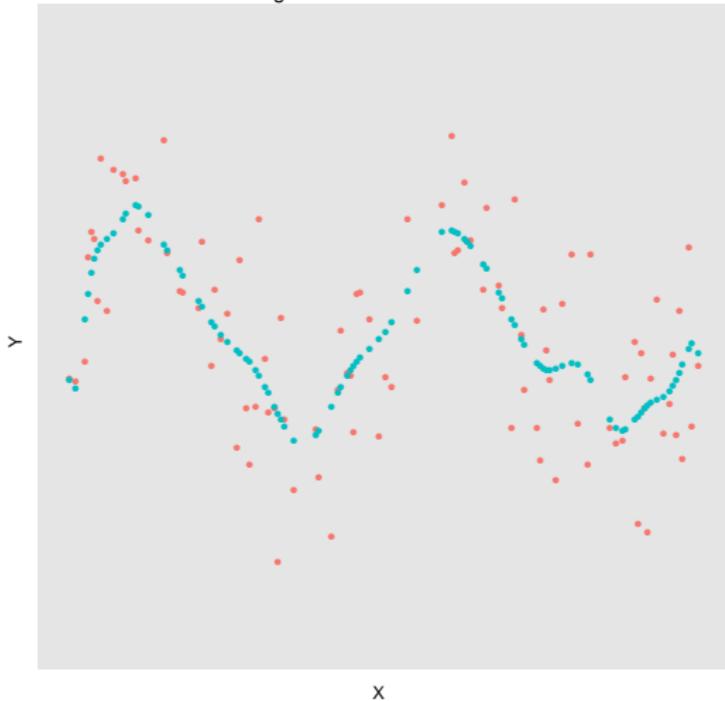
Lasso Regression with Lambda = 0.0387



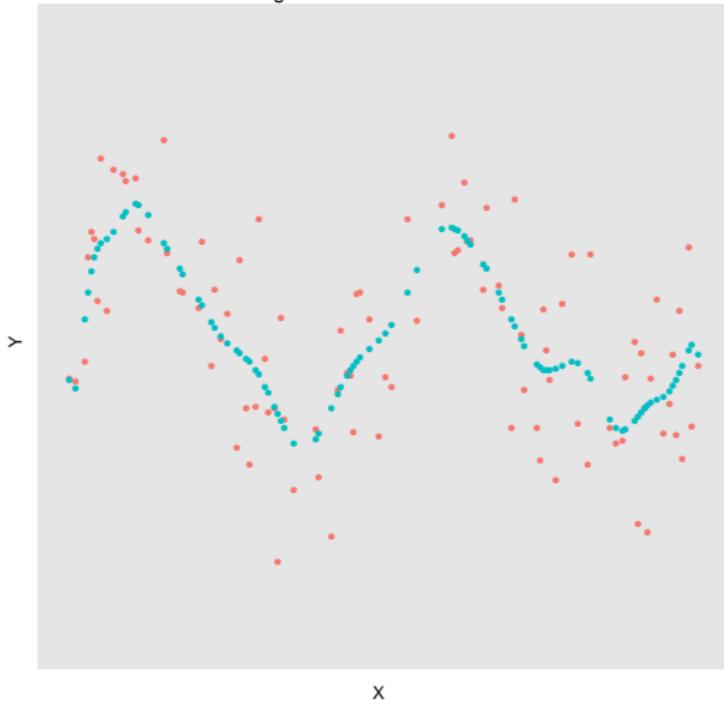
Lasso Regression with Lambda = 0.0353



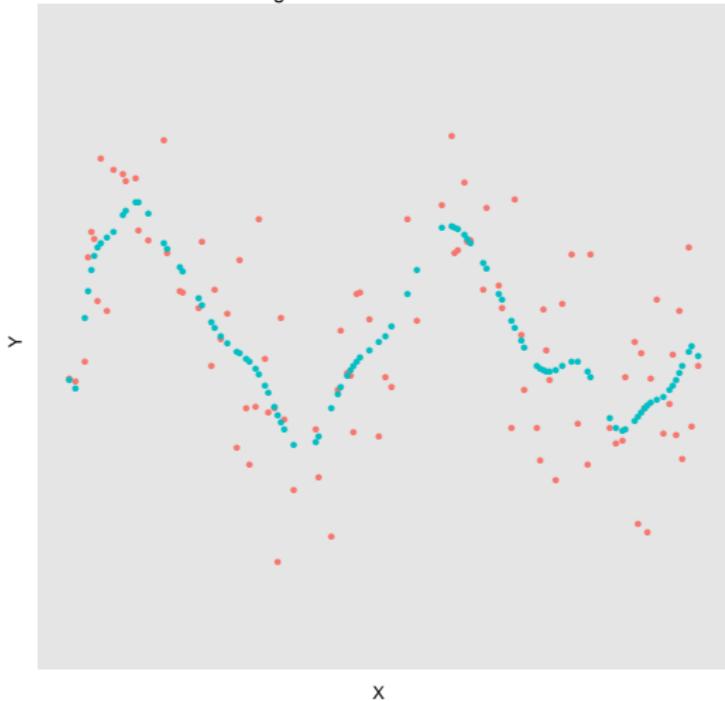
Lasso Regression with Lambda = 0.0322



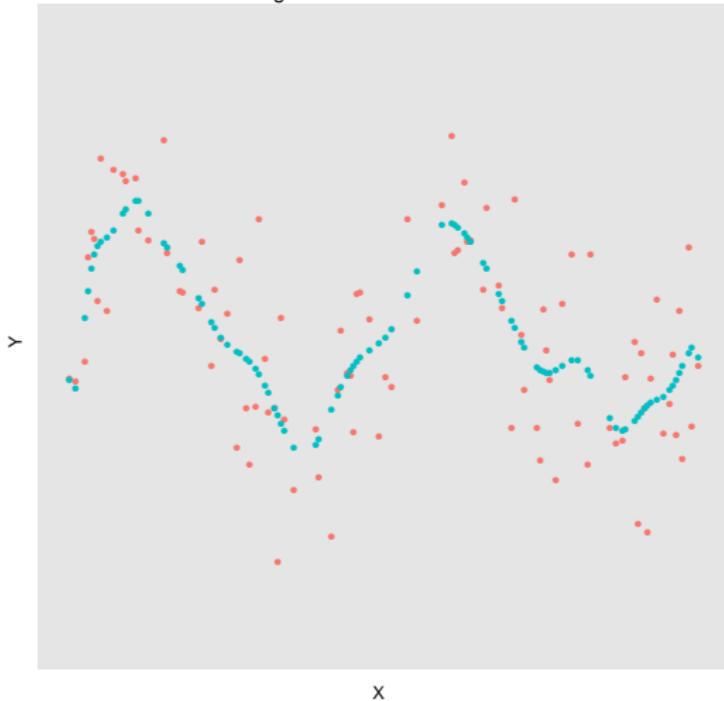
Lasso Regression with Lambda = 0.0293



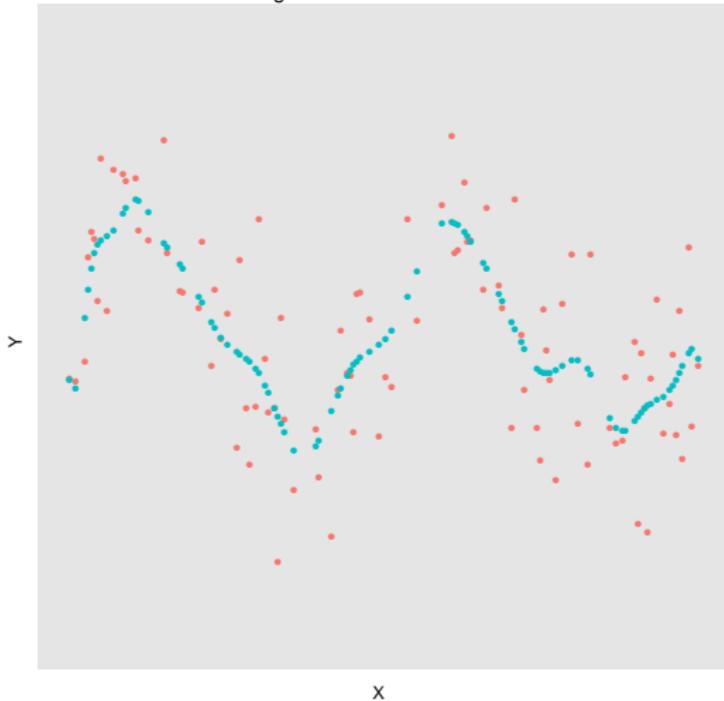
Lasso Regression with Lambda = 0.0267



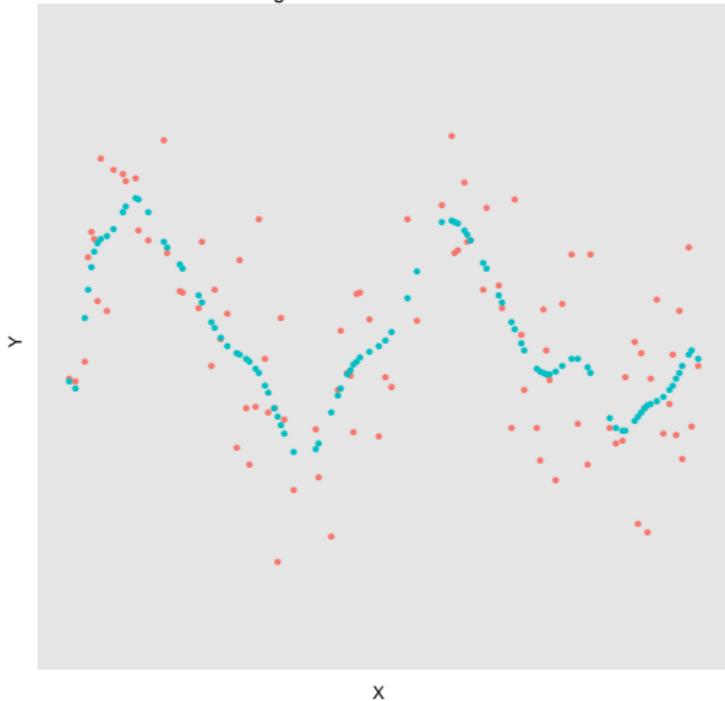
Lasso Regression with Lambda = 0.0243



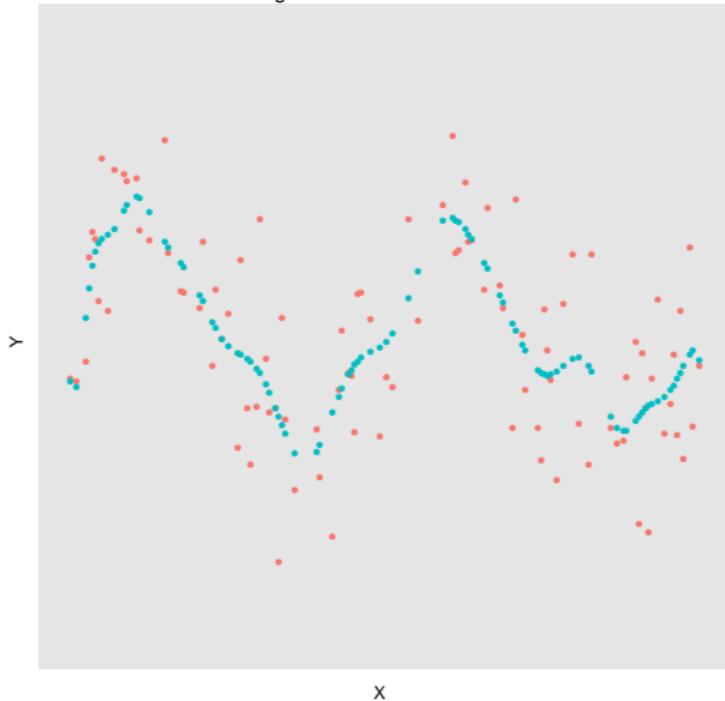
Lasso Regression with Lambda = 0.0222



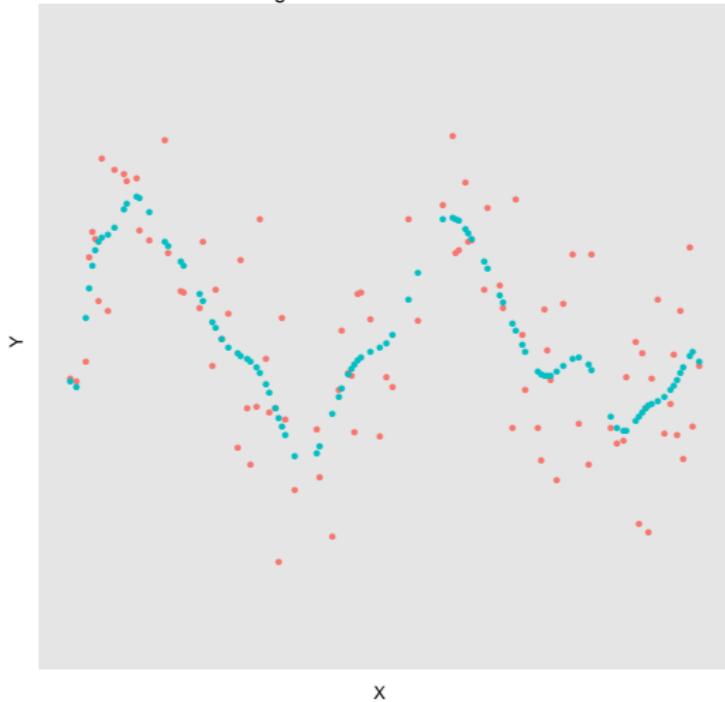
Lasso Regression with Lambda = 0.0202



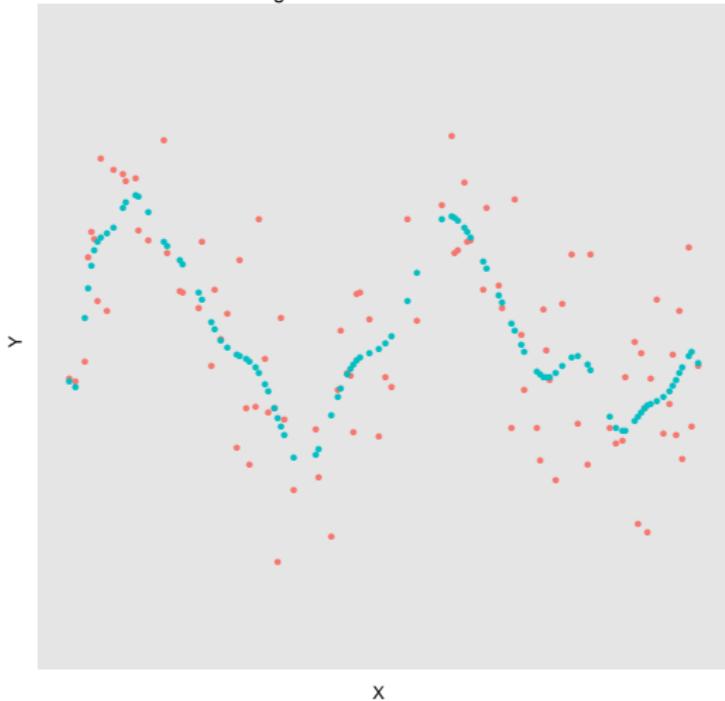
Lasso Regression with Lambda = 0.0184



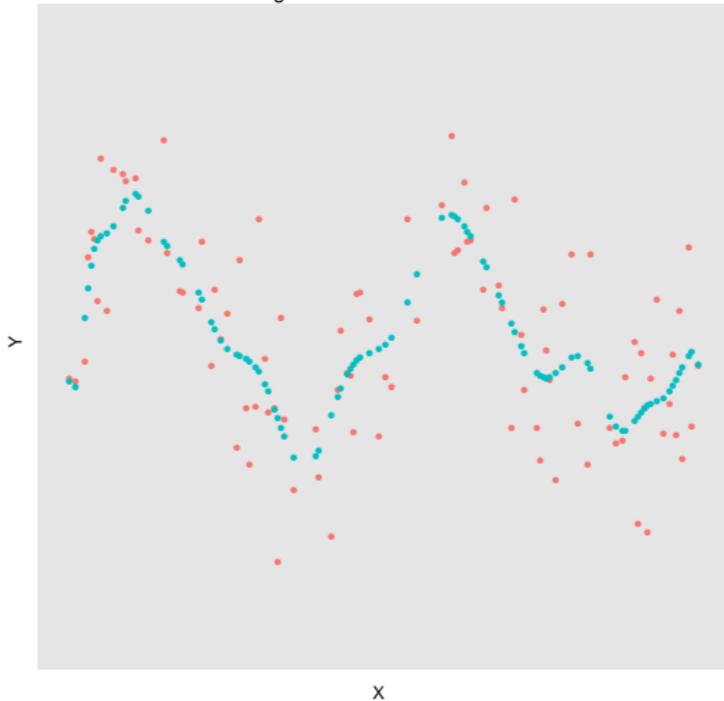
Lasso Regression with Lambda = 0.0168



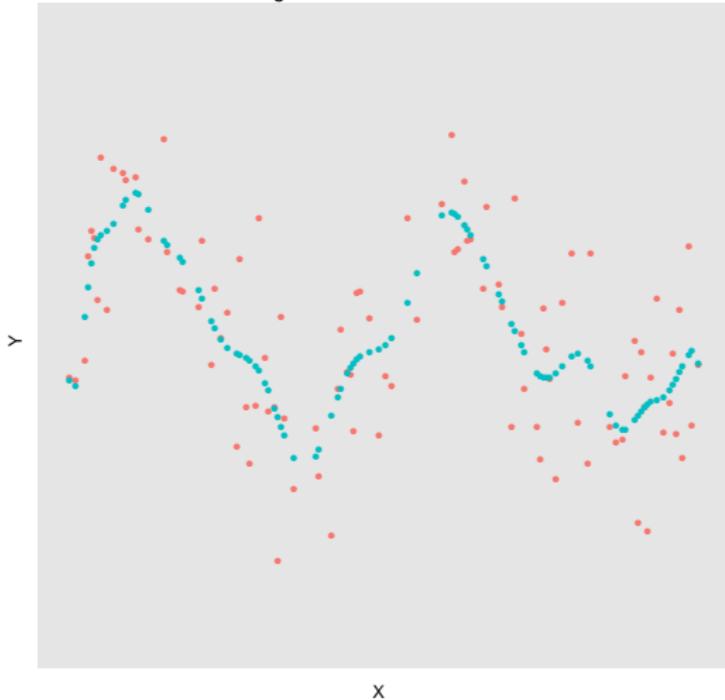
Lasso Regression with Lambda = 0.0153



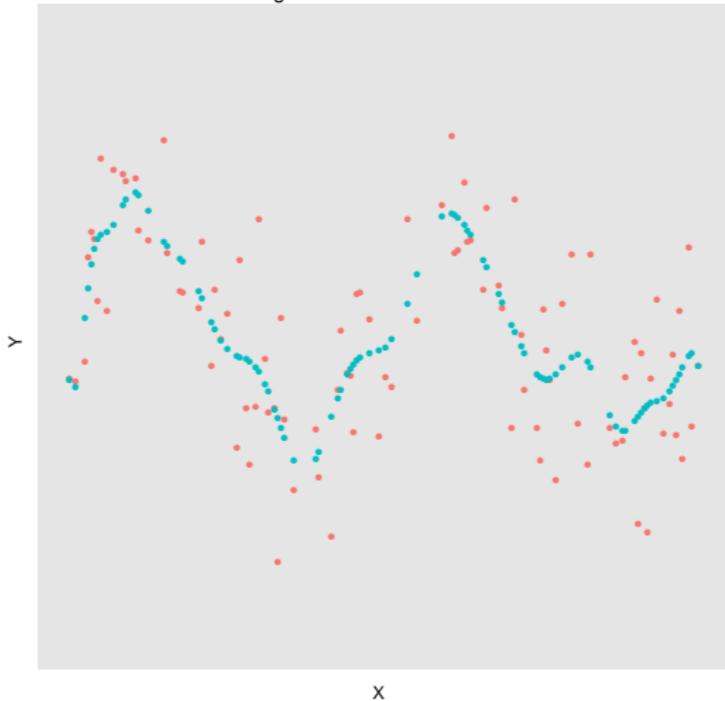
Lasso Regression with Lambda = 0.0139



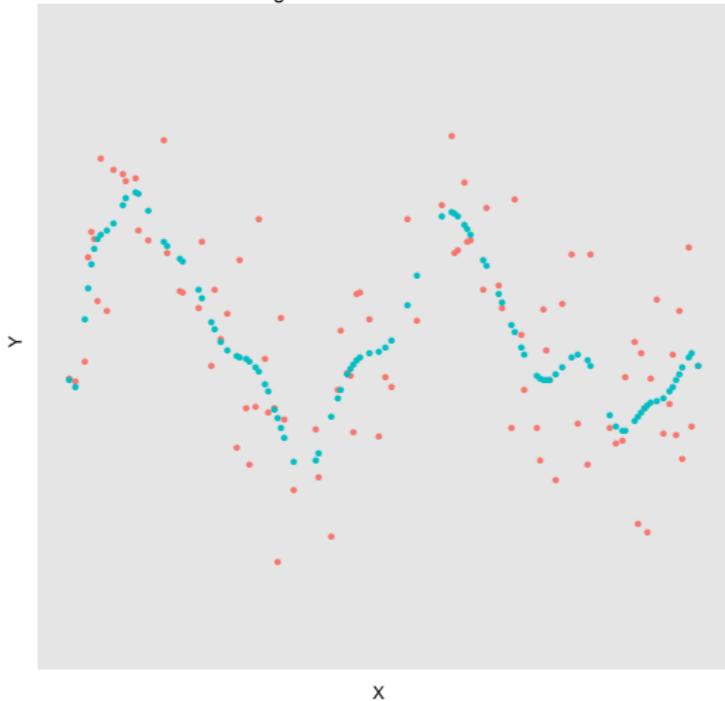
Lasso Regression with Lambda = 0.0127



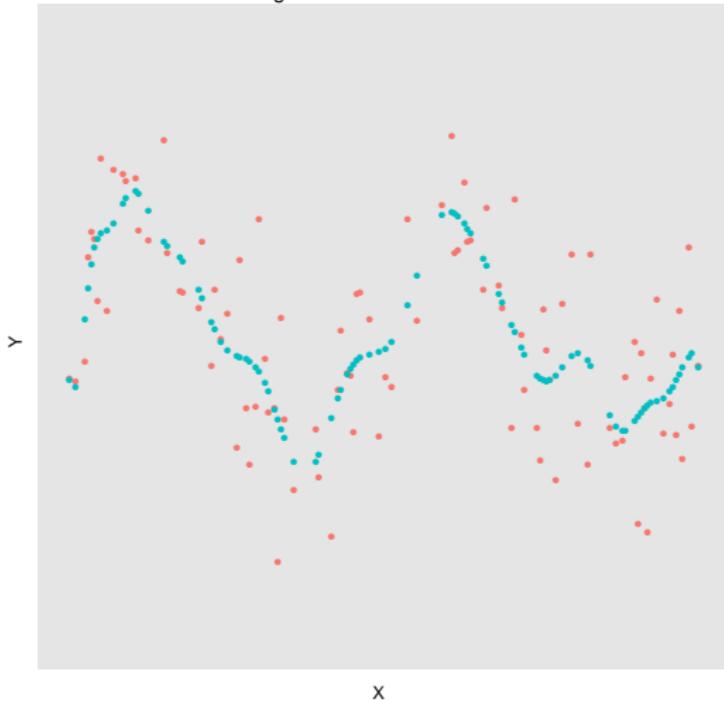
Lasso Regression with Lambda = 0.0116



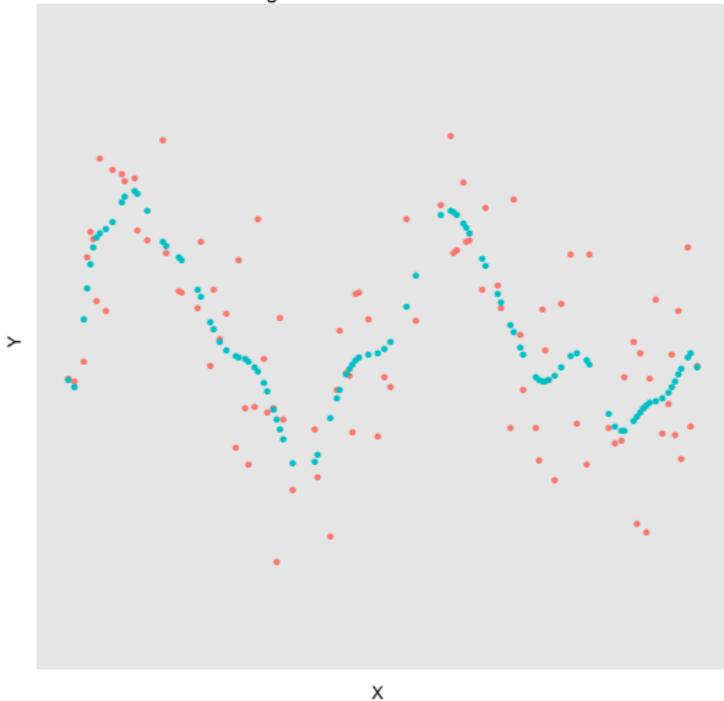
Lasso Regression with Lambda = 0.0105



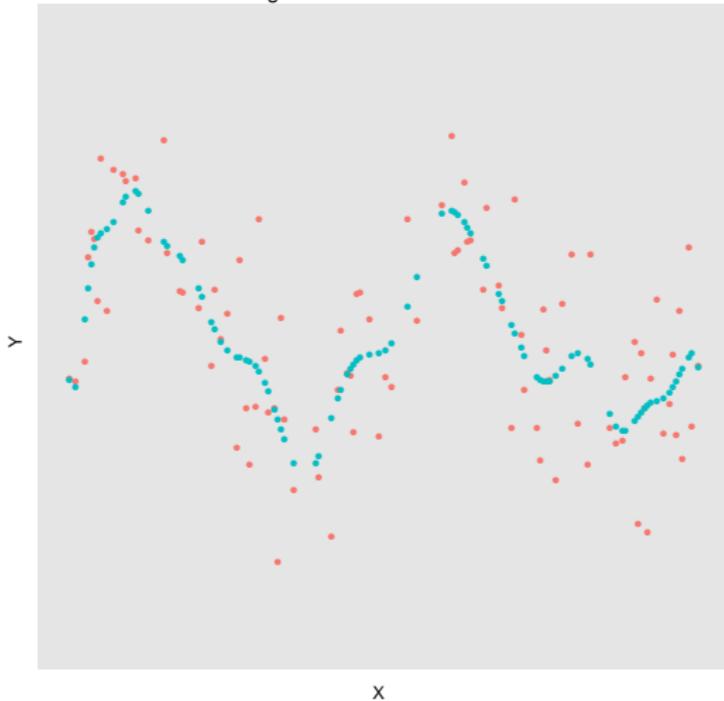
Lasso Regression with Lambda = 0.0096



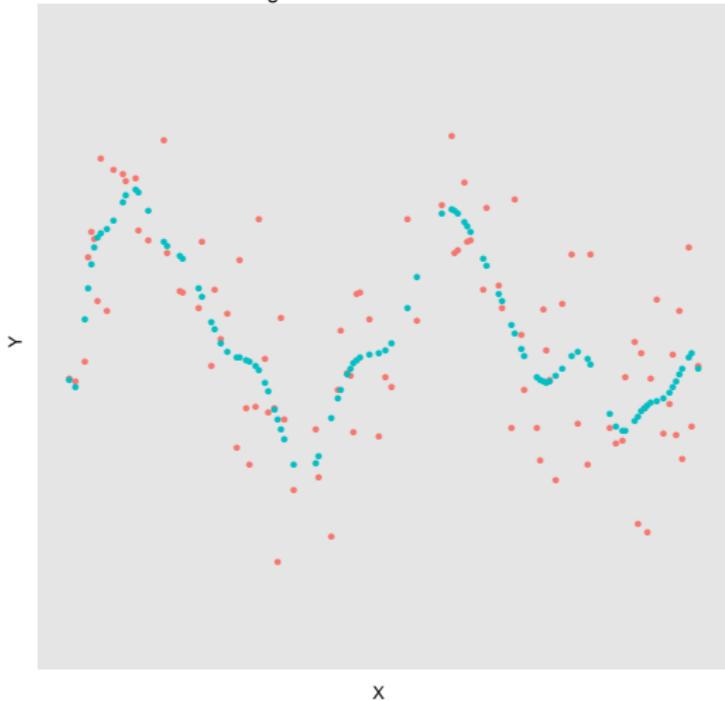
Lasso Regression with Lambda = 0.00874



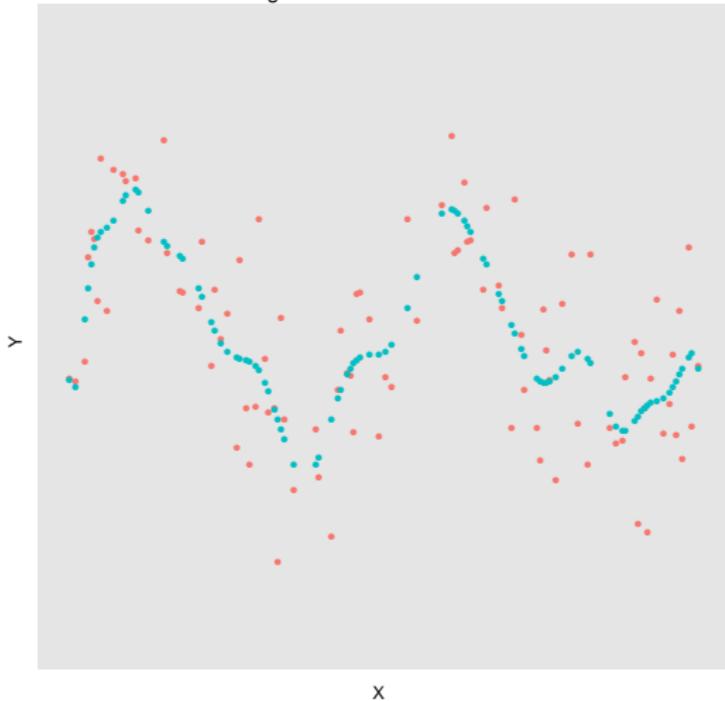
Lasso Regression with Lambda = 0.00797



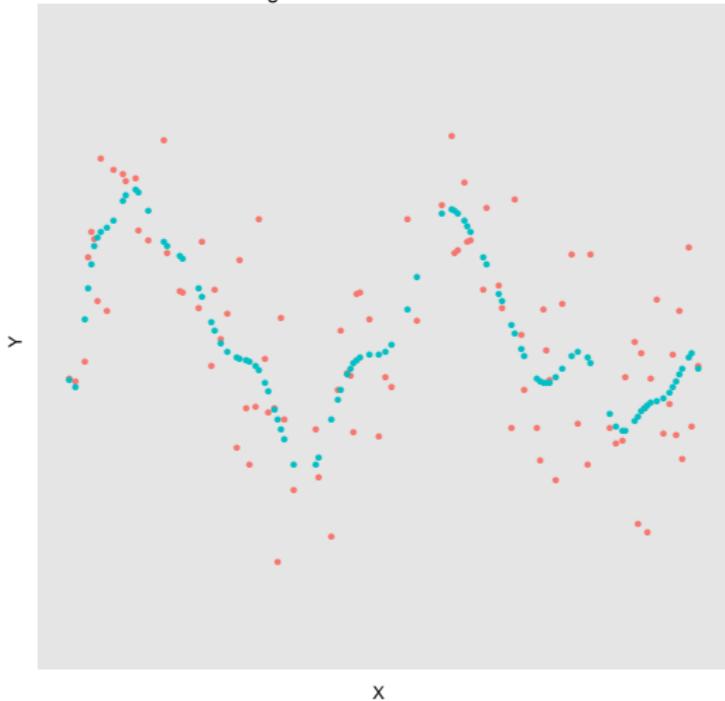
Lasso Regression with Lambda = 0.00726



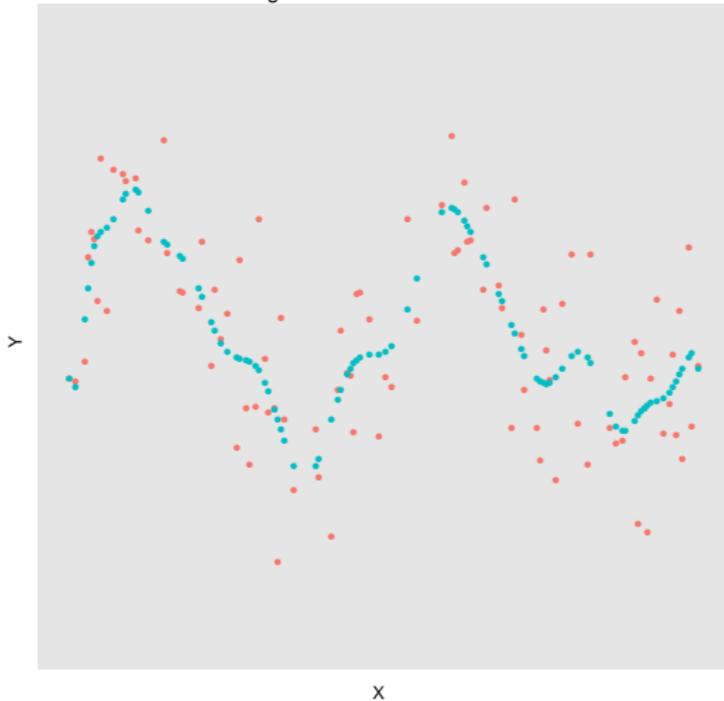
Lasso Regression with Lambda = 0.00661



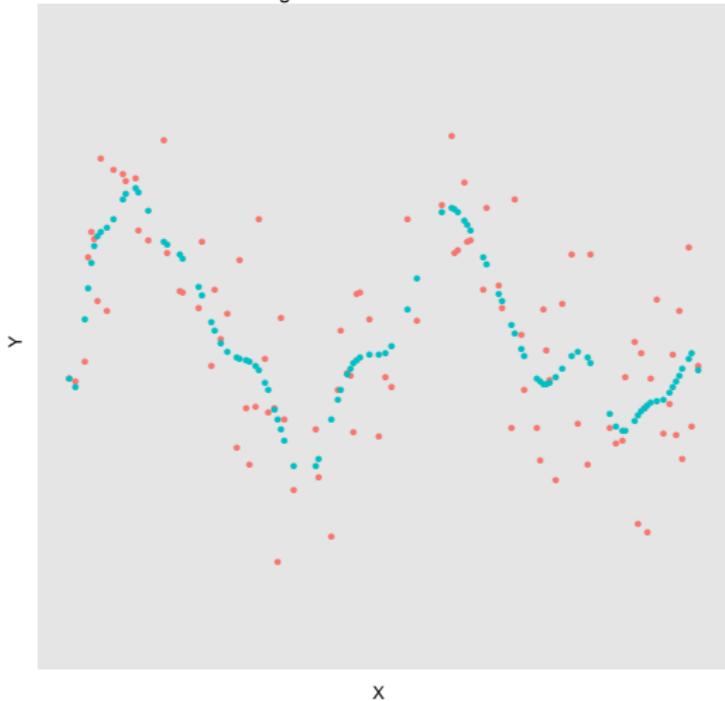
Lasso Regression with Lambda = 0.00603



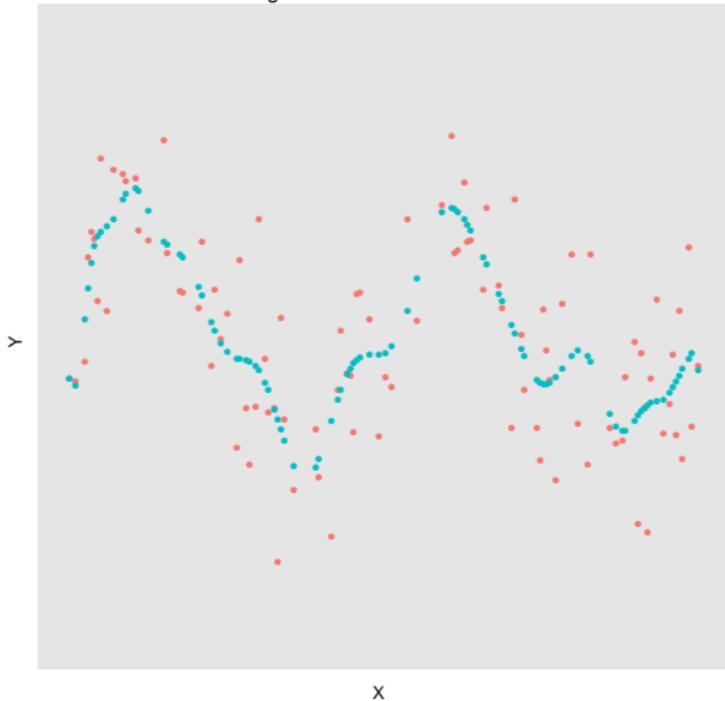
Lasso Regression with Lambda = 0.00549



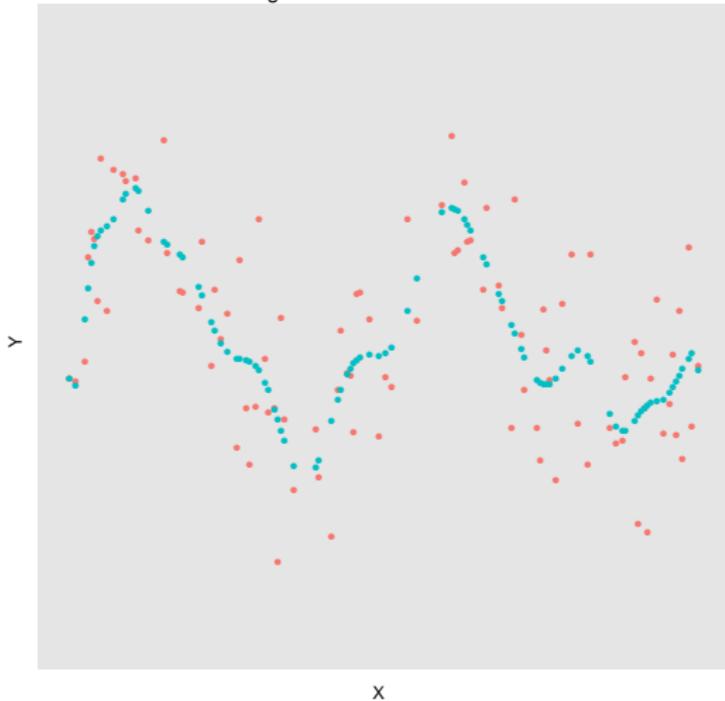
Lasso Regression with Lambda = 0.005



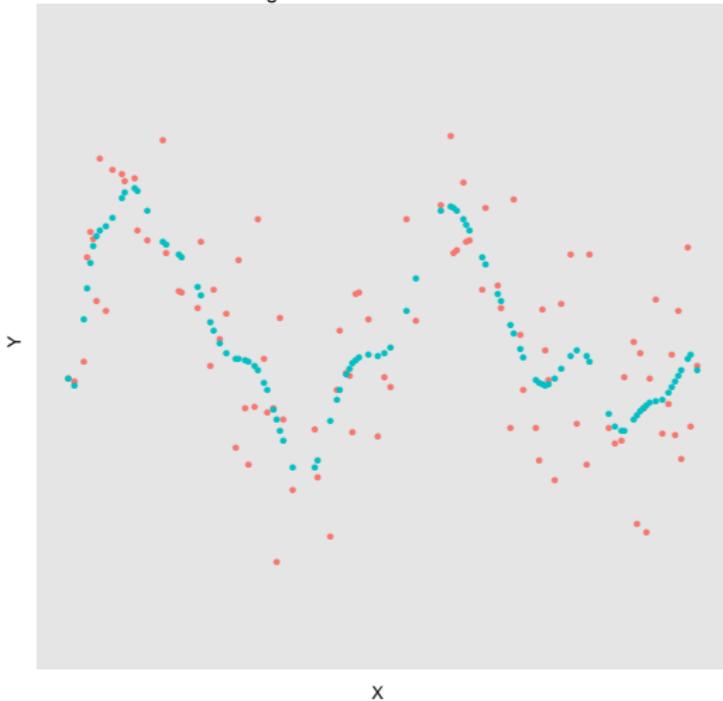
Lasso Regression with Lambda = 0.00456



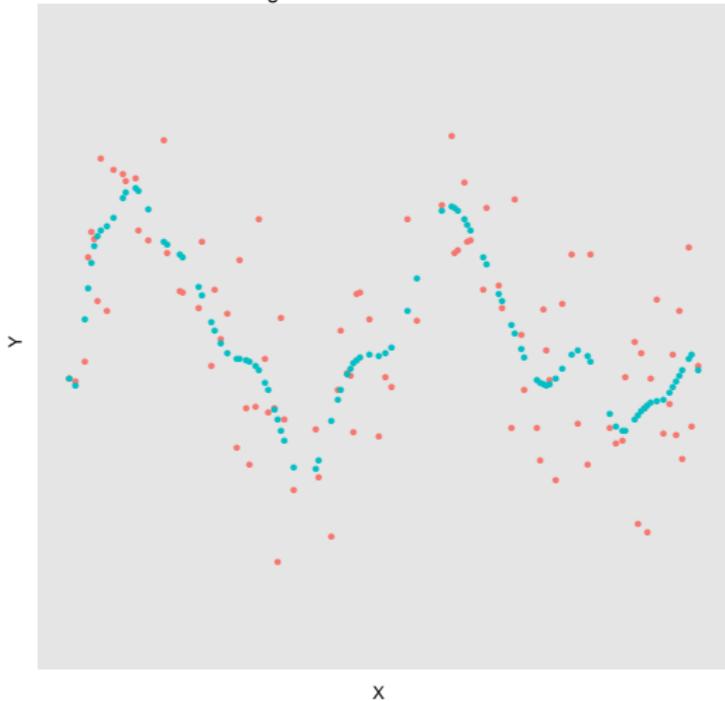
Lasso Regression with Lambda = 0.00415



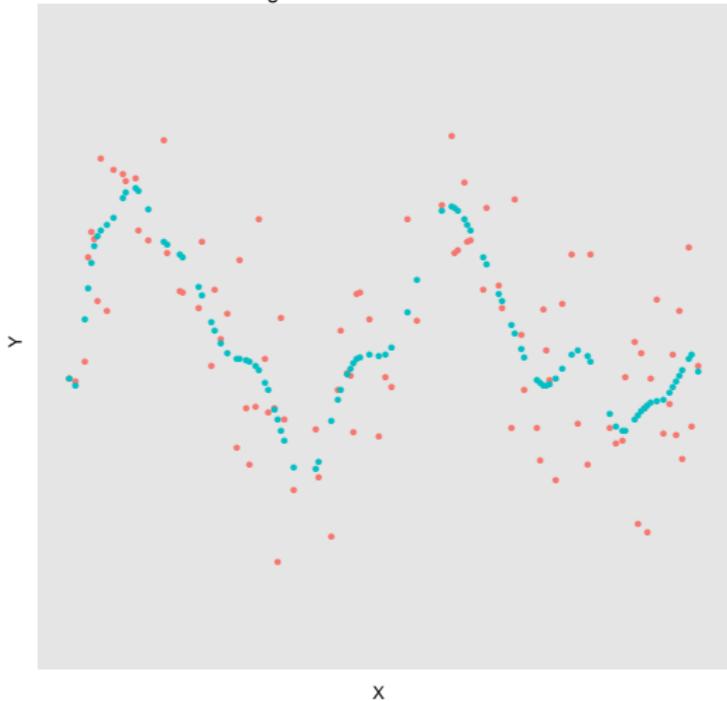
Lasso Regression with Lambda = 0.00378



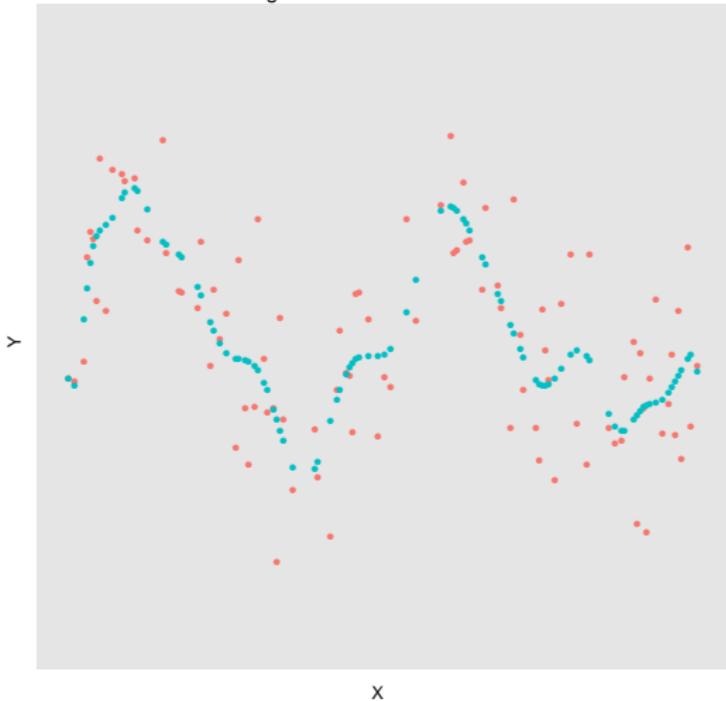
Lasso Regression with Lambda = 0.00345



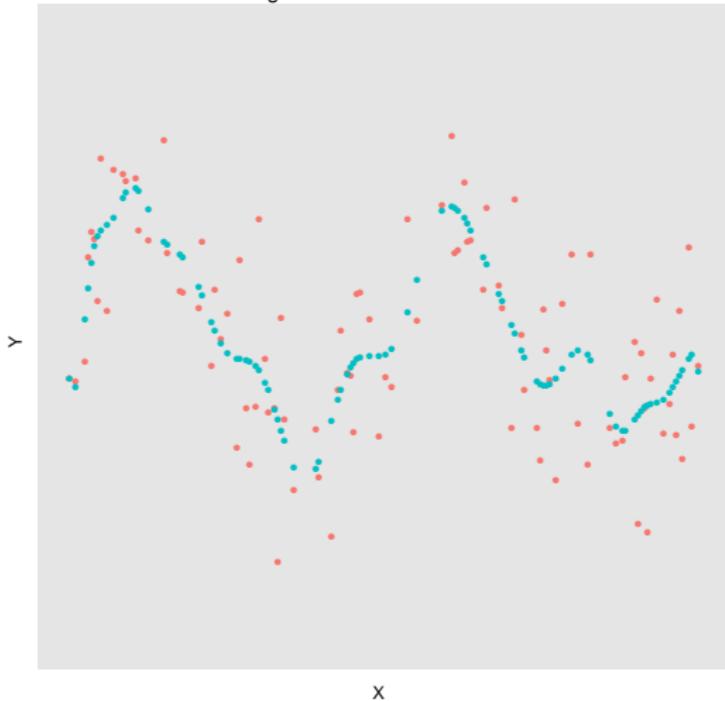
Lasso Regression with Lambda = 0.00314



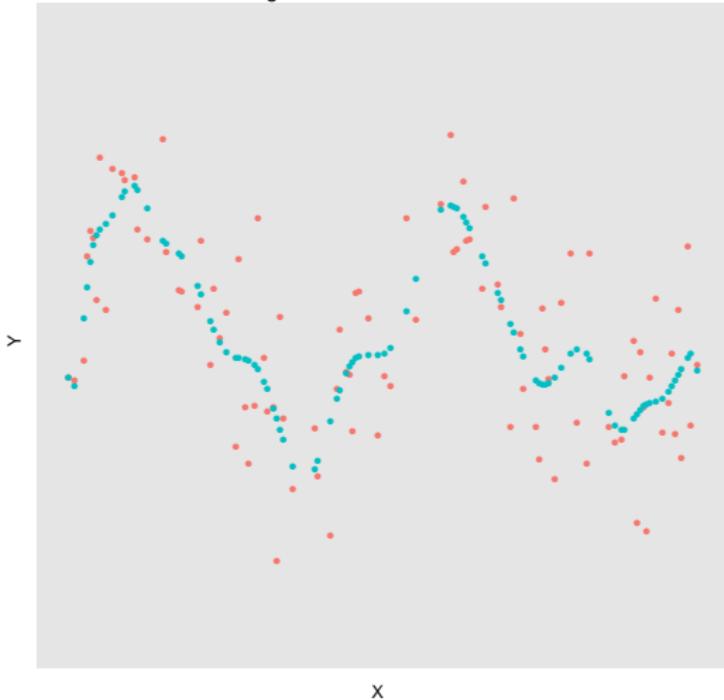
Lasso Regression with Lambda = 0.00286



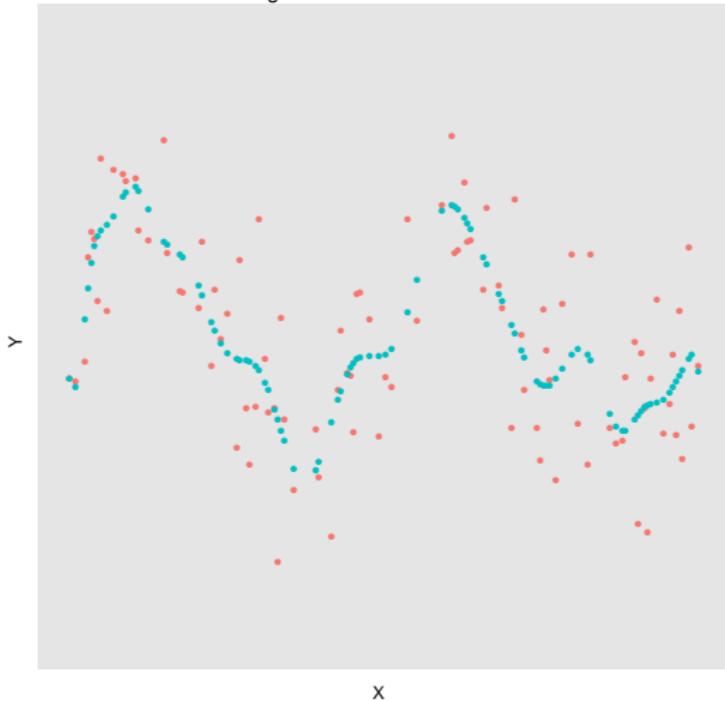
Lasso Regression with Lambda = 0.00261



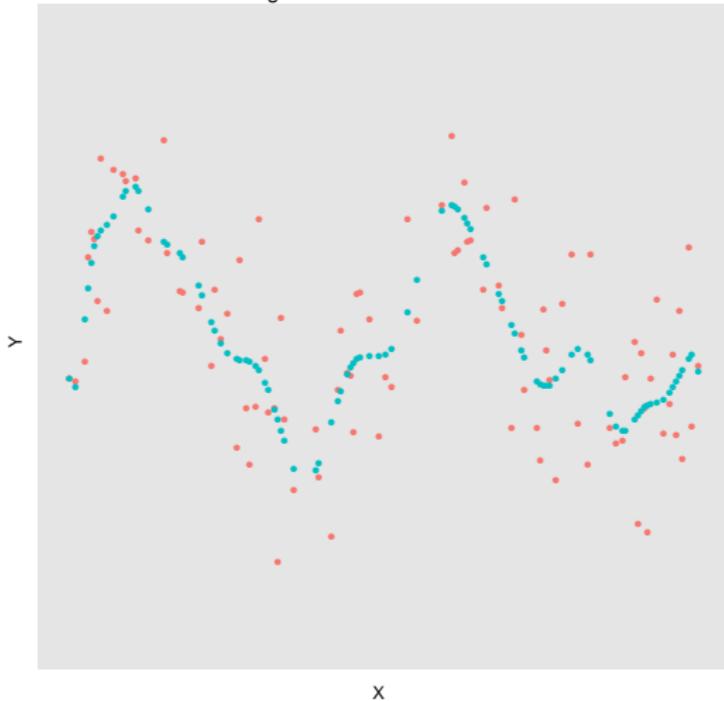
Lasso Regression with Lambda = 0.00238



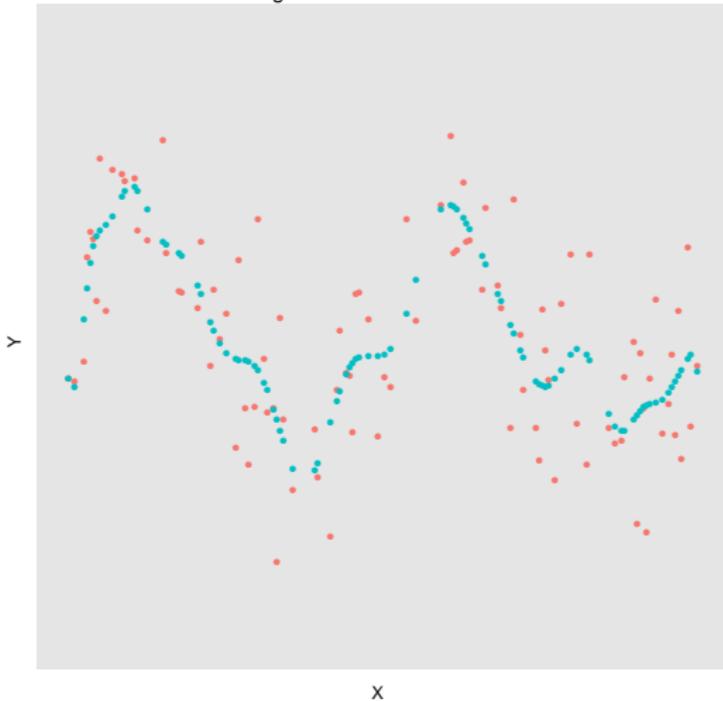
Lasso Regression with Lambda = 0.00217



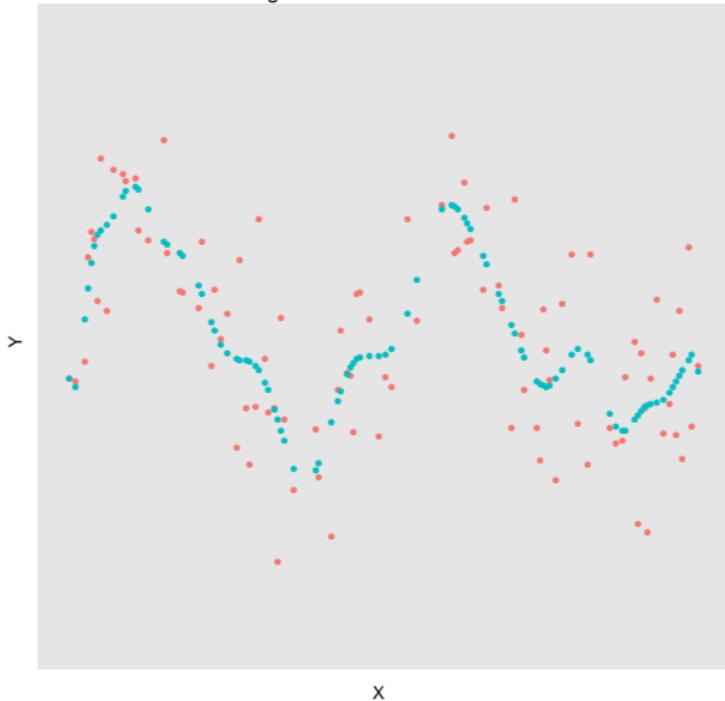
Lasso Regression with Lambda = 0.00197



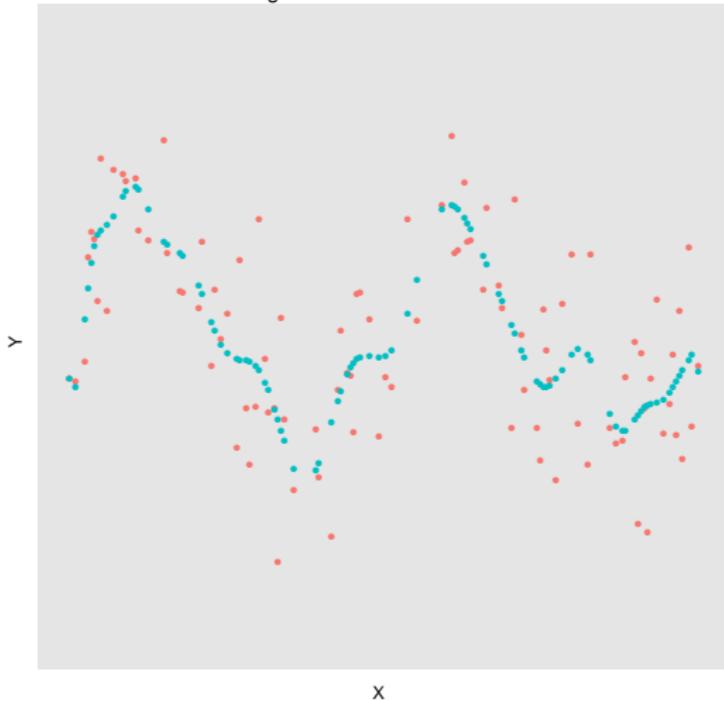
Lasso Regression with Lambda = 0.0018



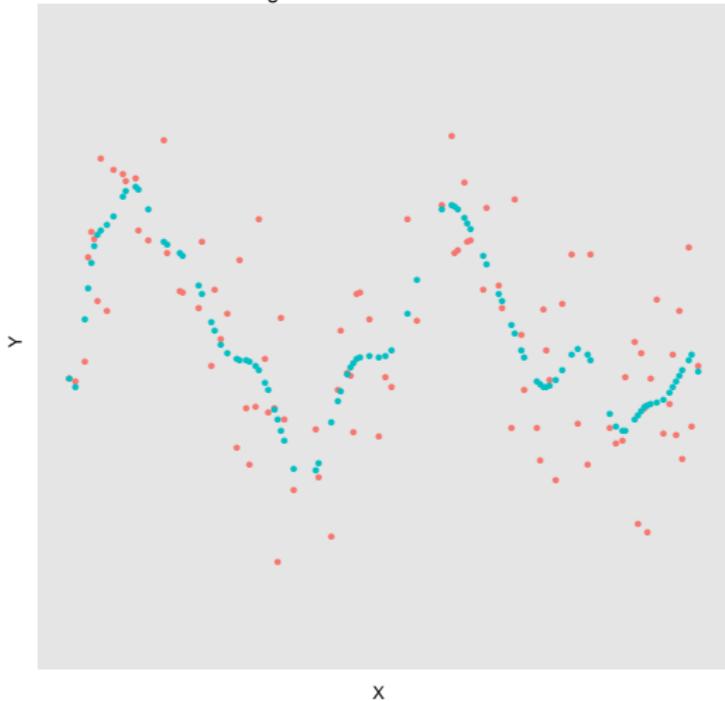
Lasso Regression with Lambda = 0.00164



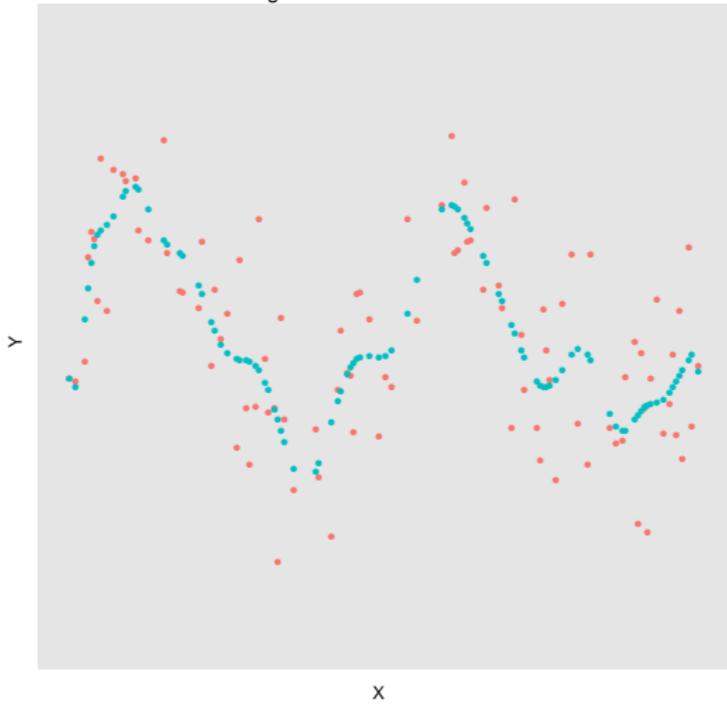
Lasso Regression with Lambda = 0.00149



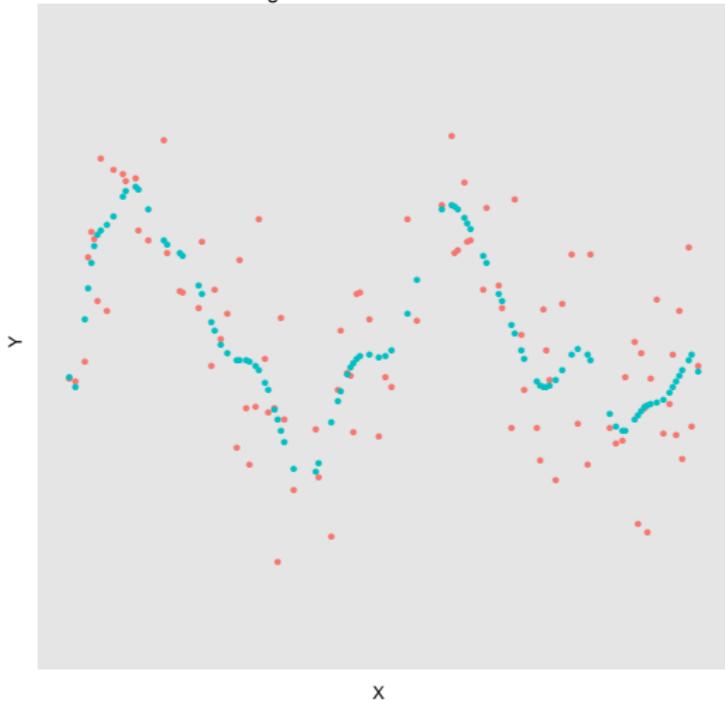
Lasso Regression with Lambda = 0.00136



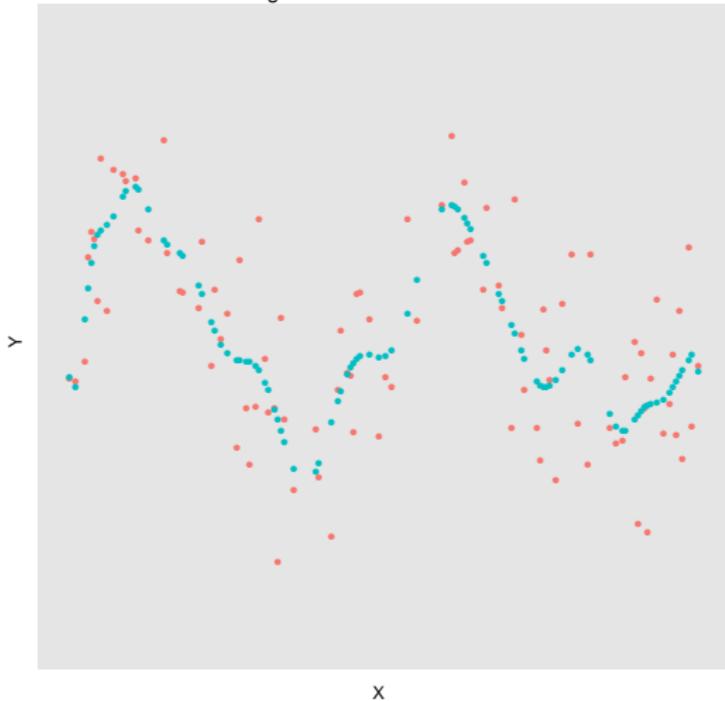
Lasso Regression with Lambda = 0.00124



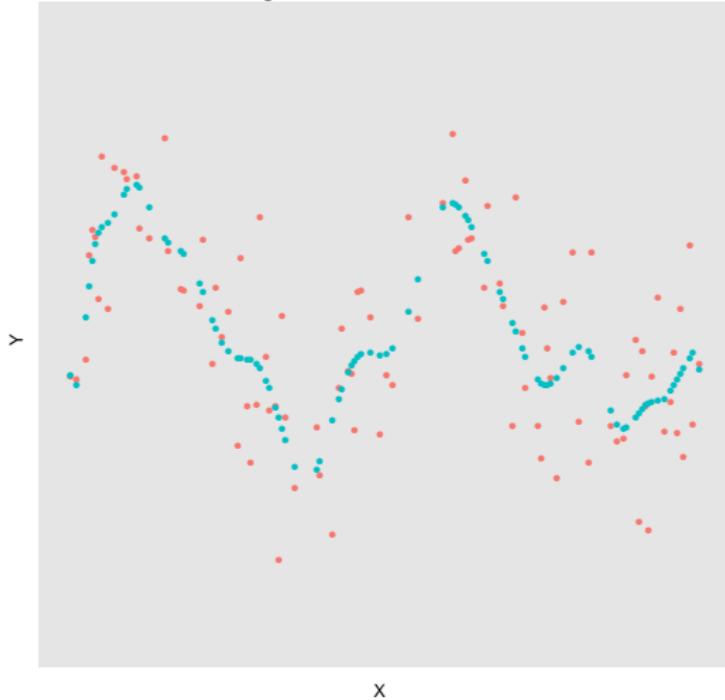
Lasso Regression with Lambda = 0.00113



Lasso Regression with Lambda = 0.00103



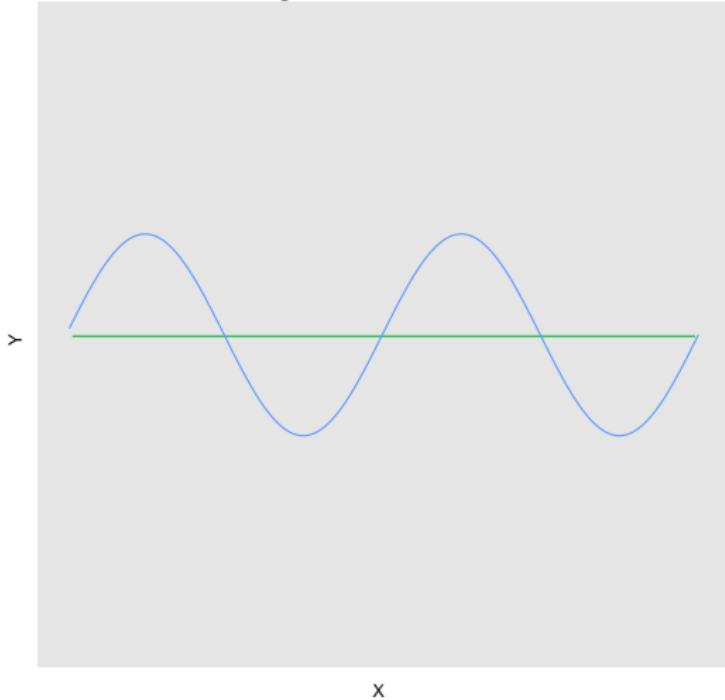
Lasso Regression with Lambda = 0.000937



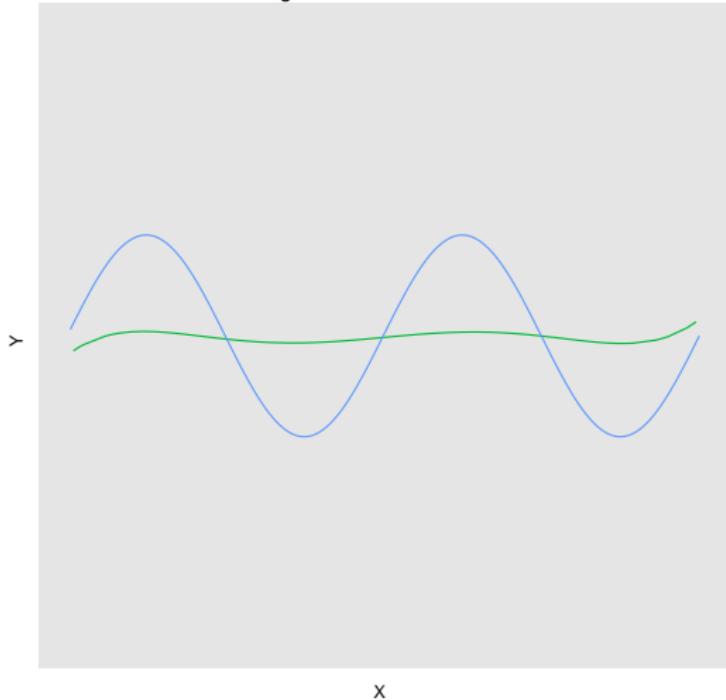
We build a smoother model

Do we get closer to the underlying pattern?

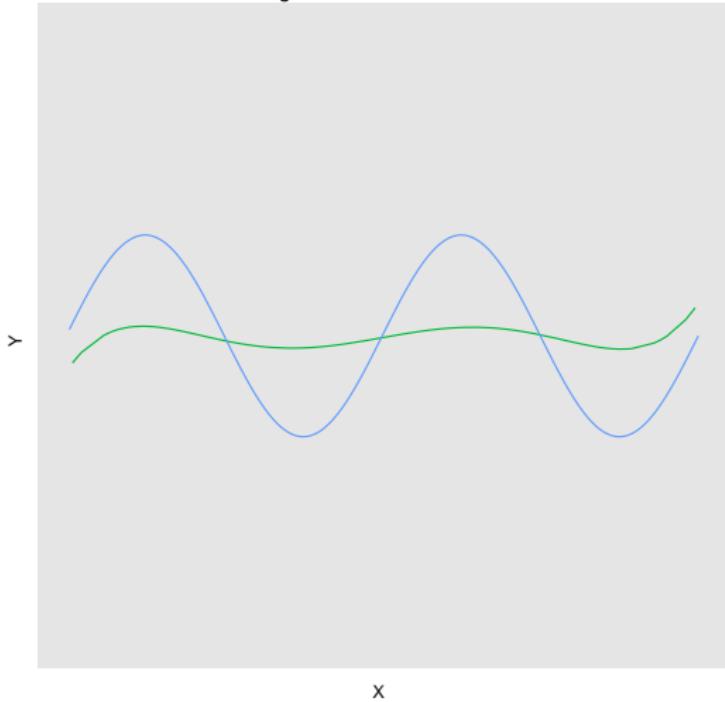
Lasso Regression with Lambda = 0.524



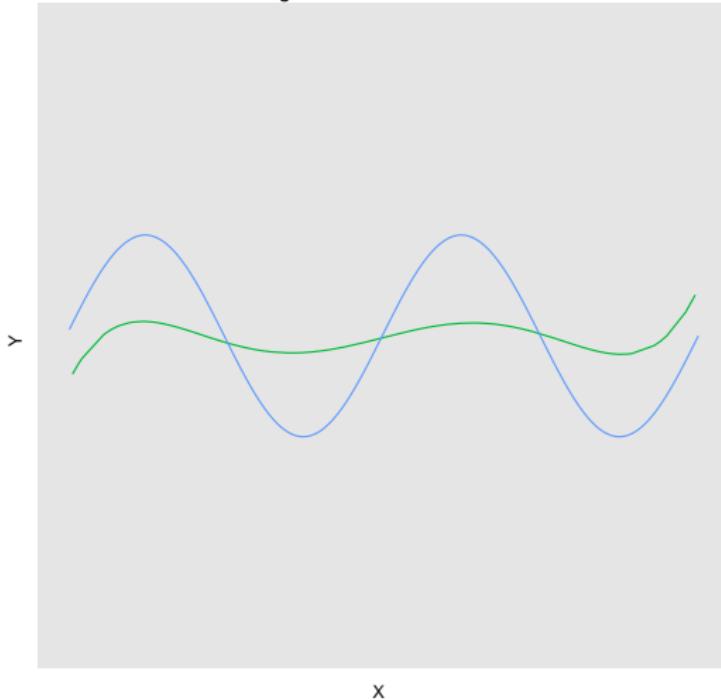
Lasso Regression with Lambda = 0.478



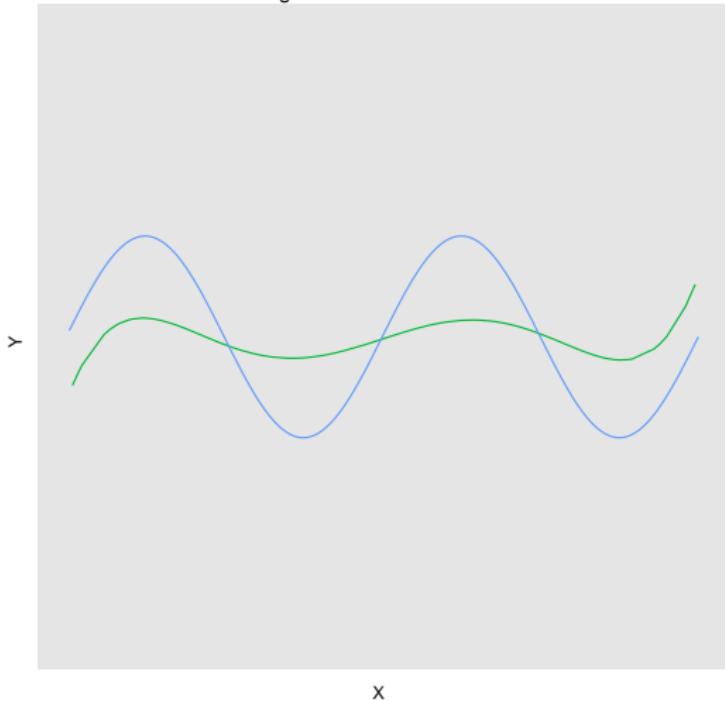
Lasso Regression with Lambda = 0.435



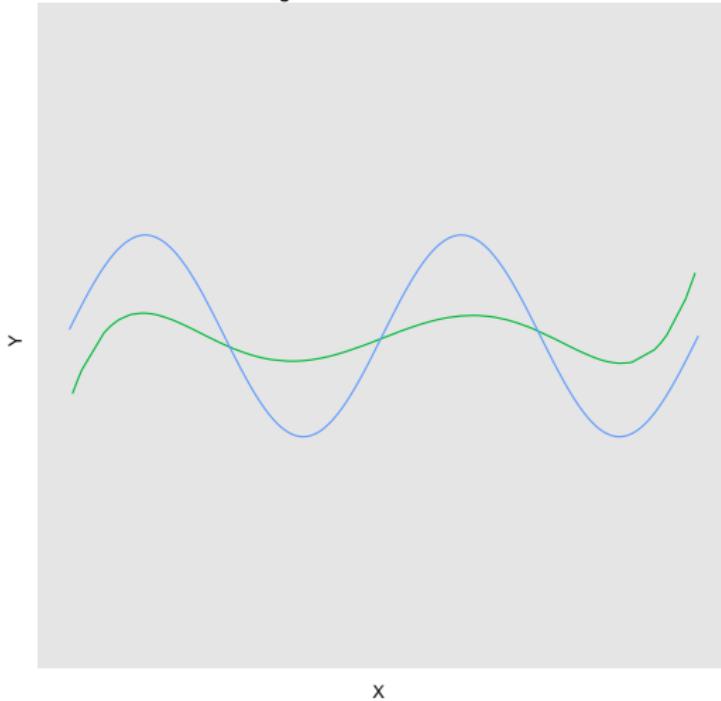
Lasso Regression with Lambda = 0.396



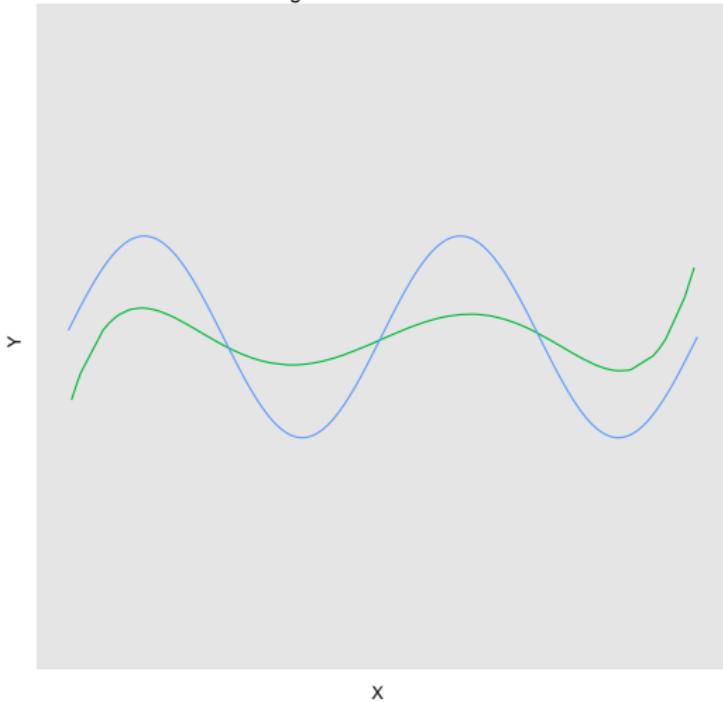
Lasso Regression with Lambda = 0.361



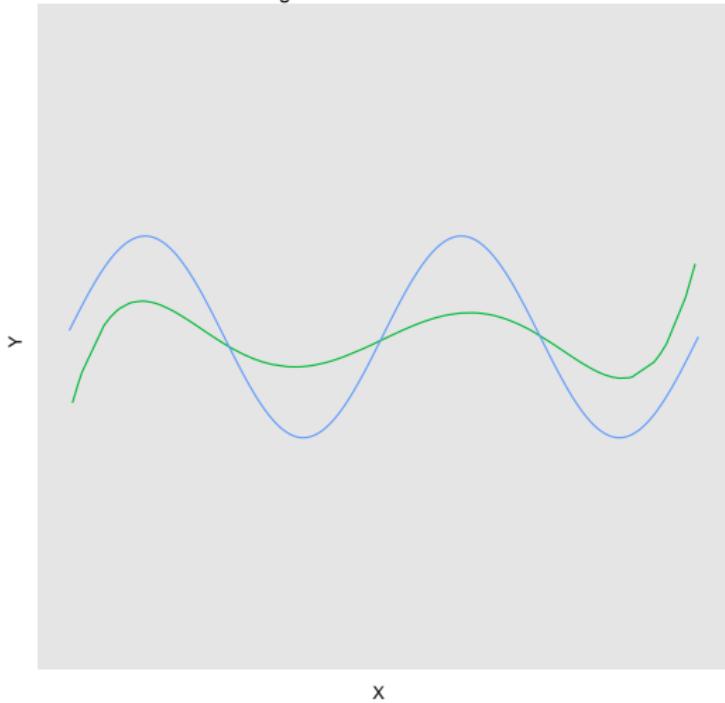
Lasso Regression with Lambda = 0.329



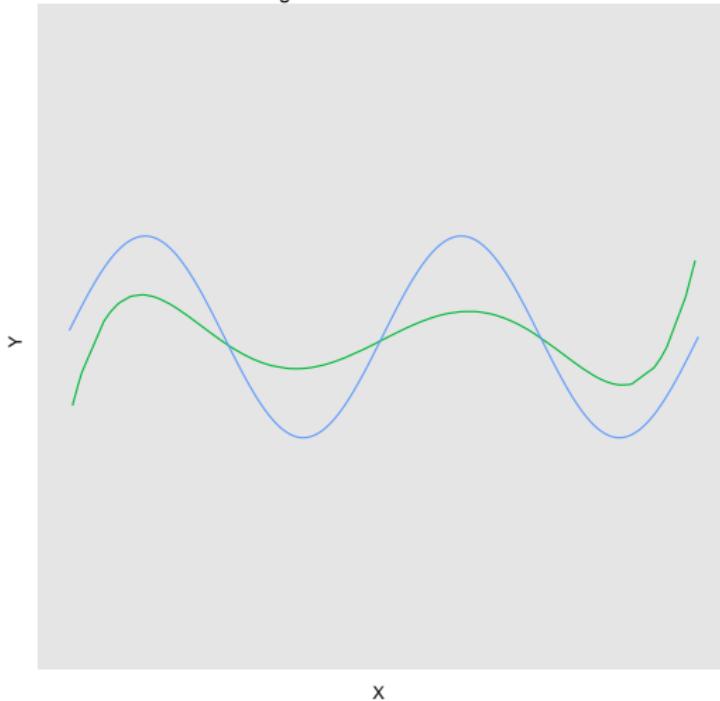
Lasso Regression with Lambda = 0.3



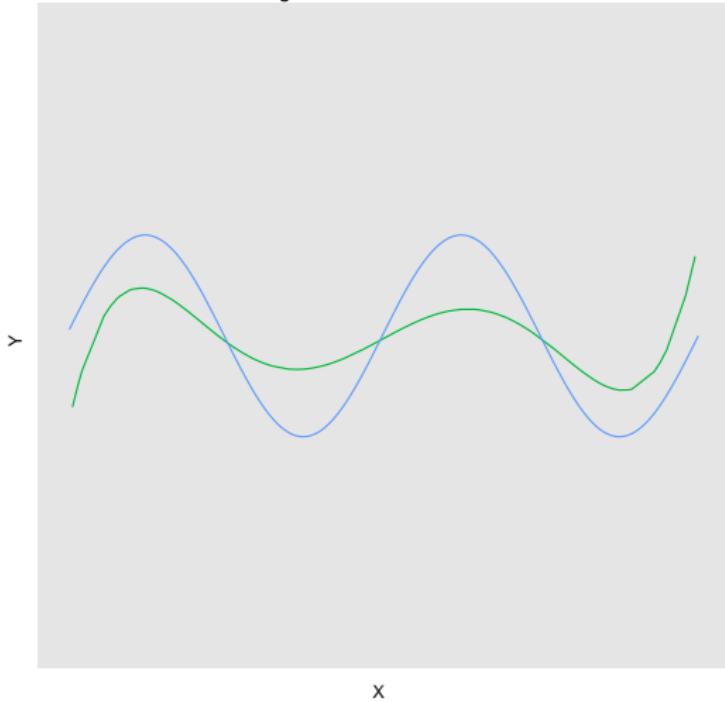
Lasso Regression with Lambda = 0.273



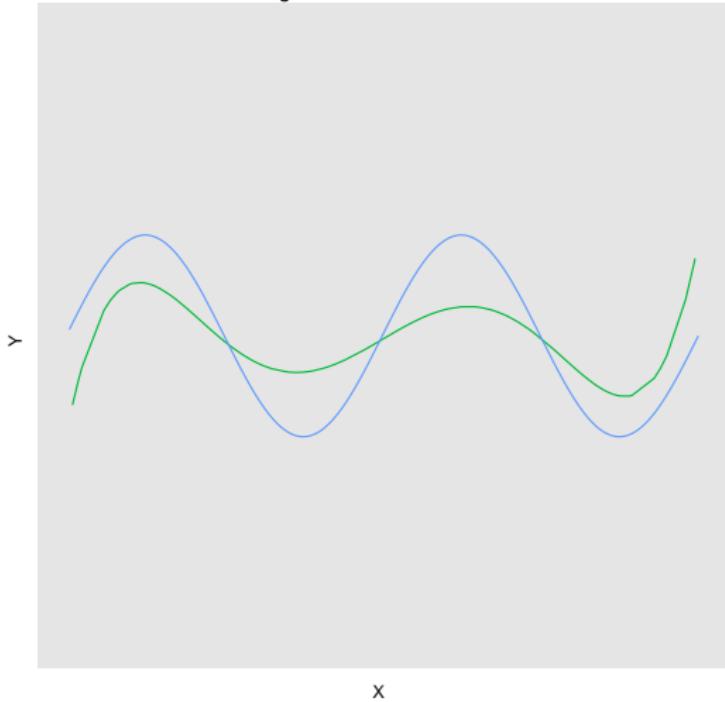
Lasso Regression with Lambda = 0.249



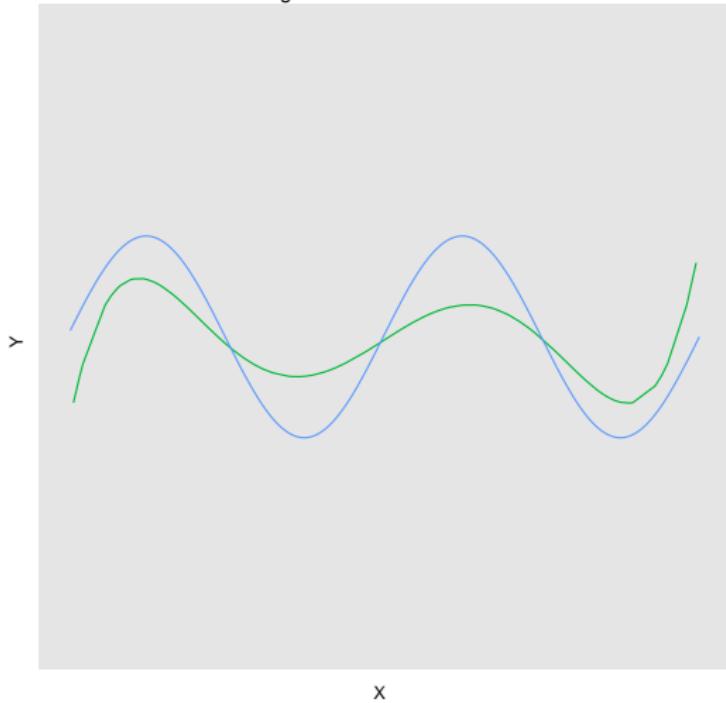
Lasso Regression with Lambda = 0.227



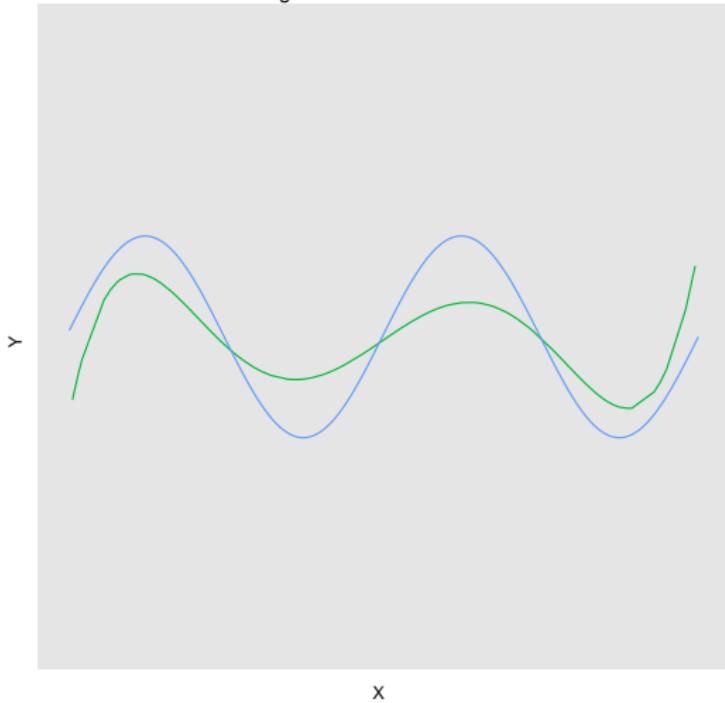
Lasso Regression with Lambda = 0.207



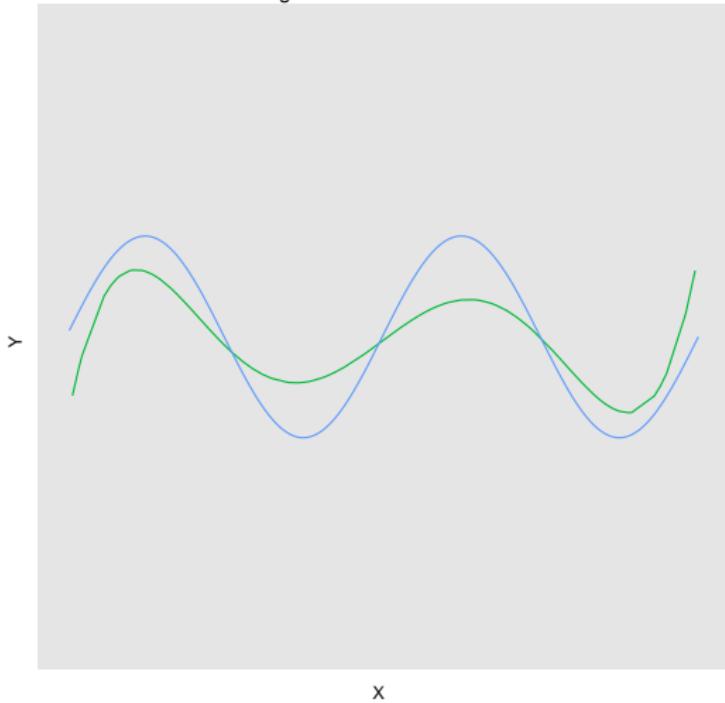
Lasso Regression with Lambda = 0.188



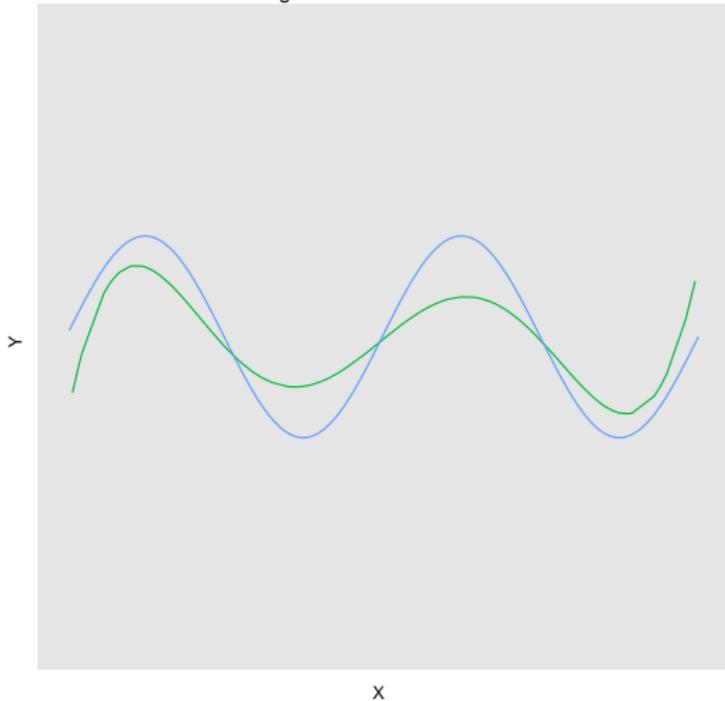
Lasso Regression with Lambda = 0.172



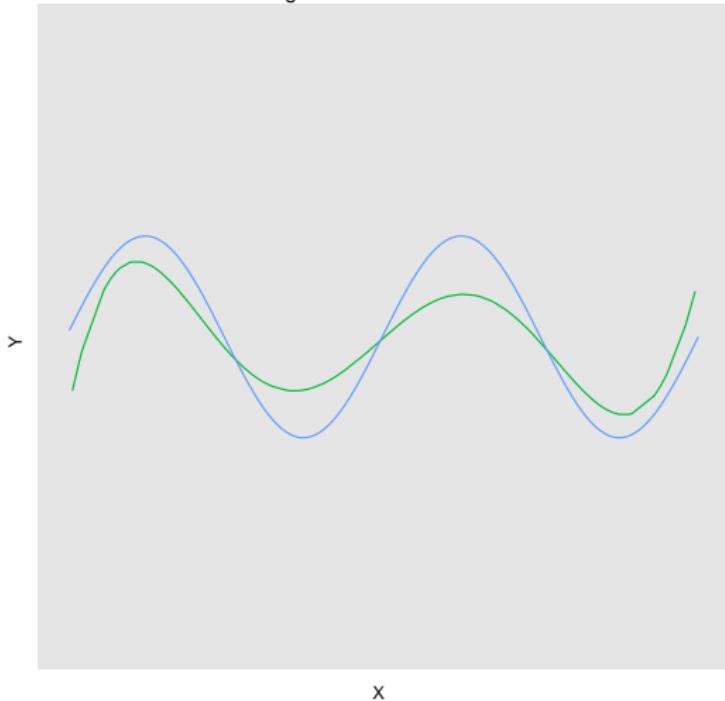
Lasso Regression with Lambda = 0.156



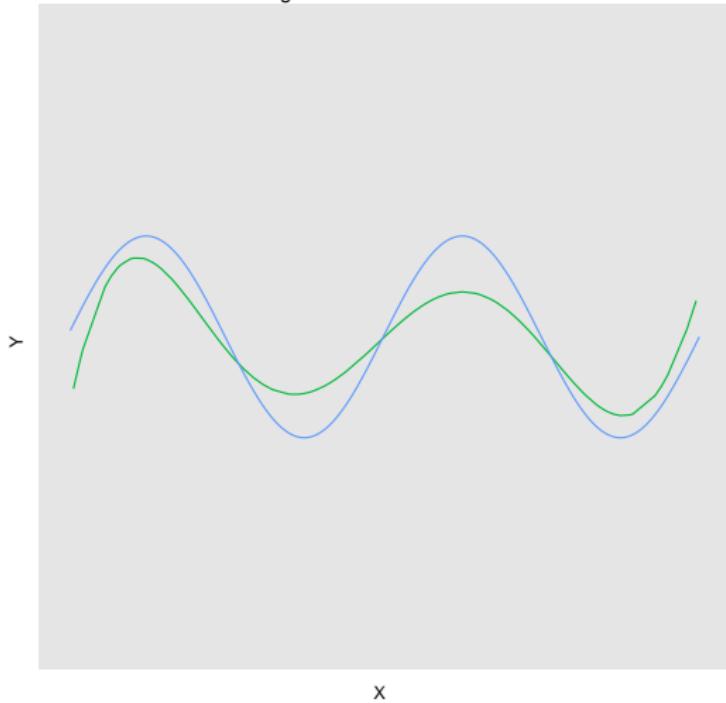
Lasso Regression with Lambda = 0.142



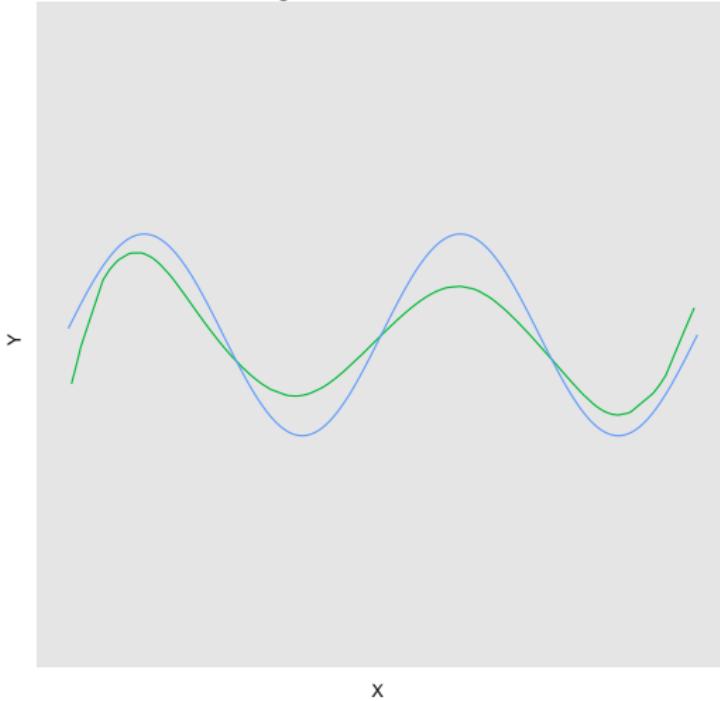
Lasso Regression with Lambda = 0.13



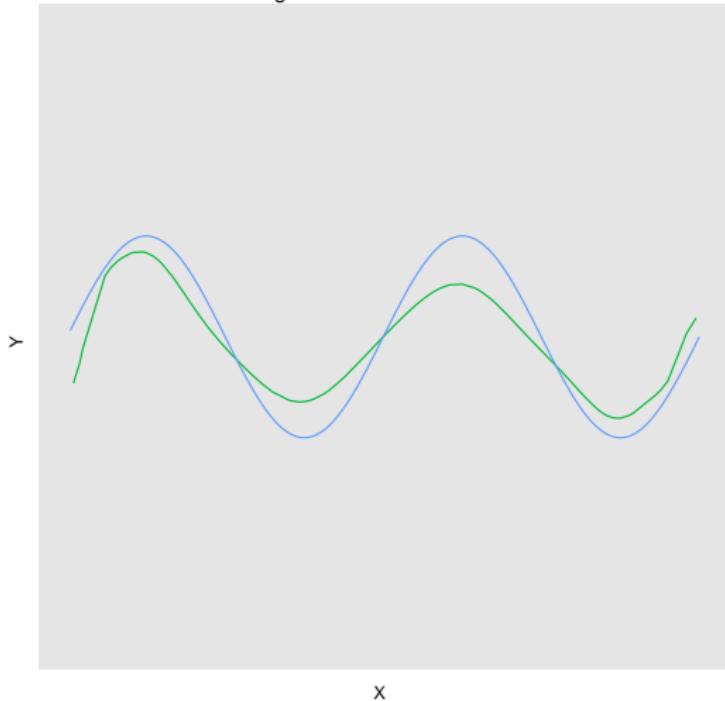
Lasso Regression with Lambda = 0.118



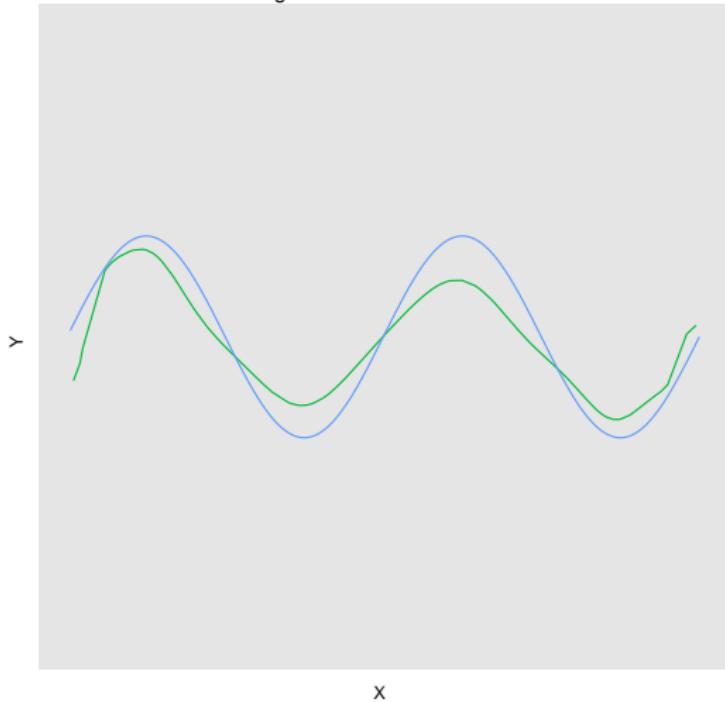
Lasso Regression with Lambda = 0.108



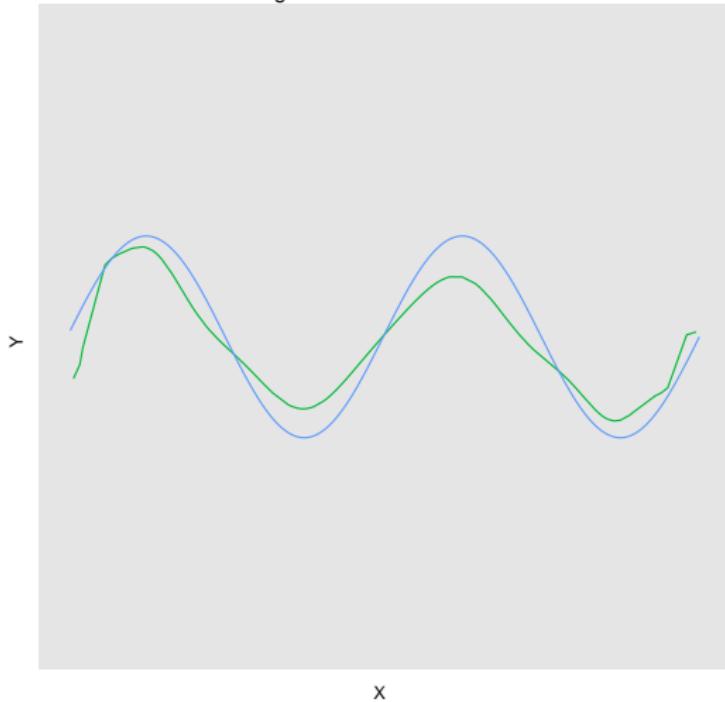
Lasso Regression with Lambda = 0.0982



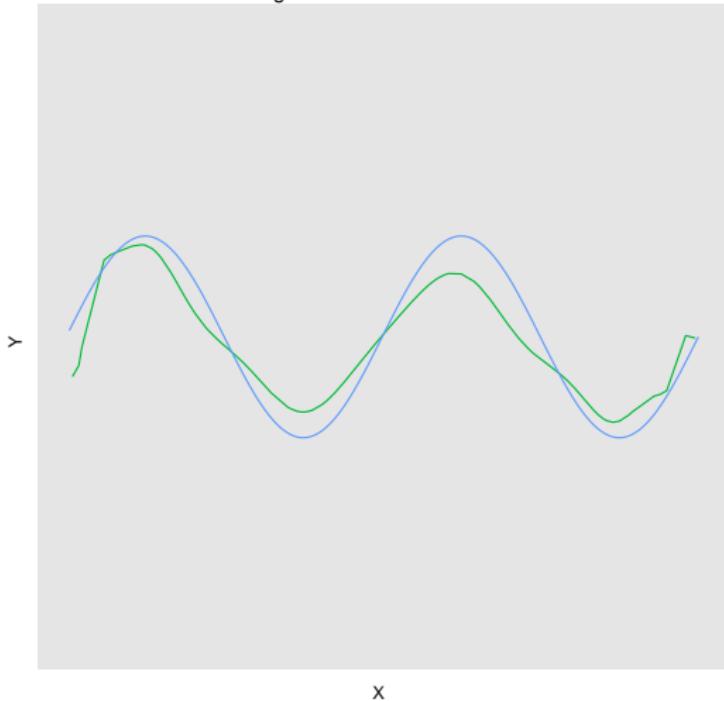
Lasso Regression with Lambda = 0.0895



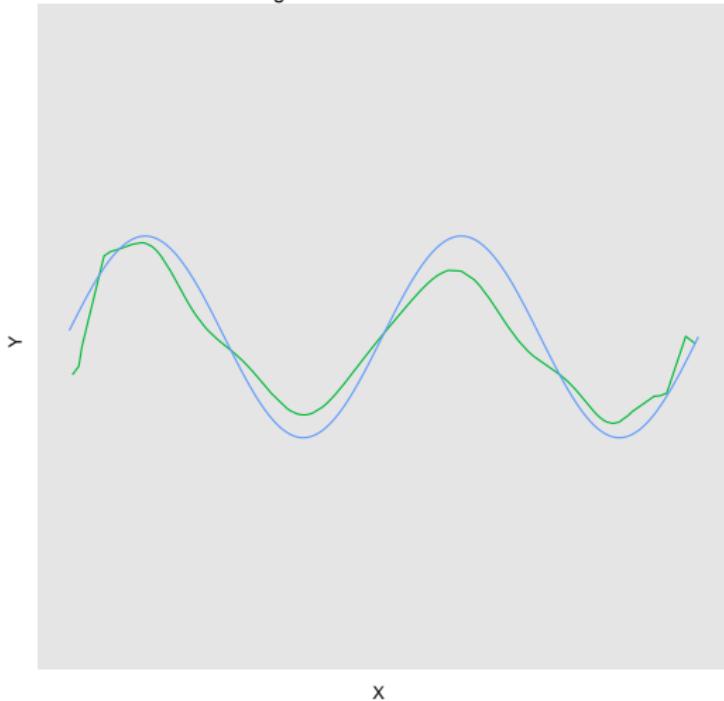
Lasso Regression with Lambda = 0.0815



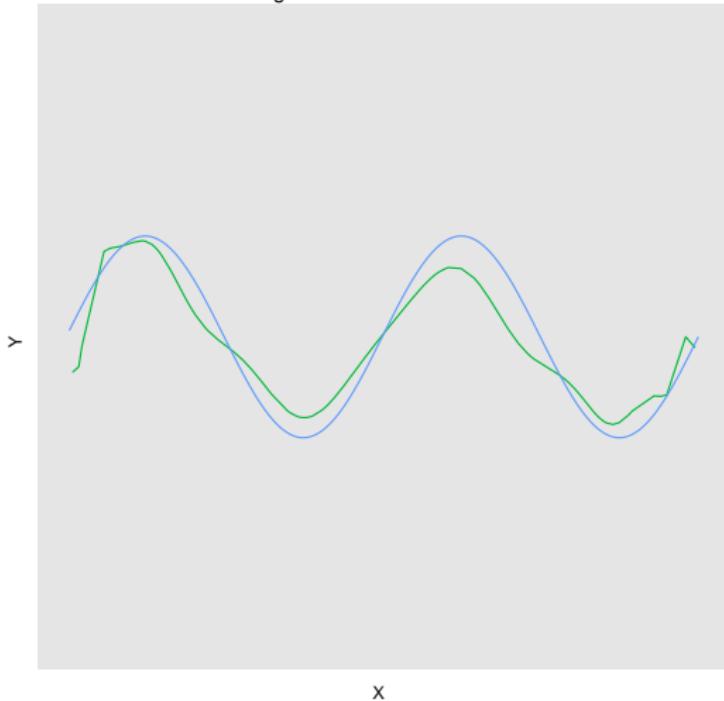
Lasso Regression with Lambda = 0.0743



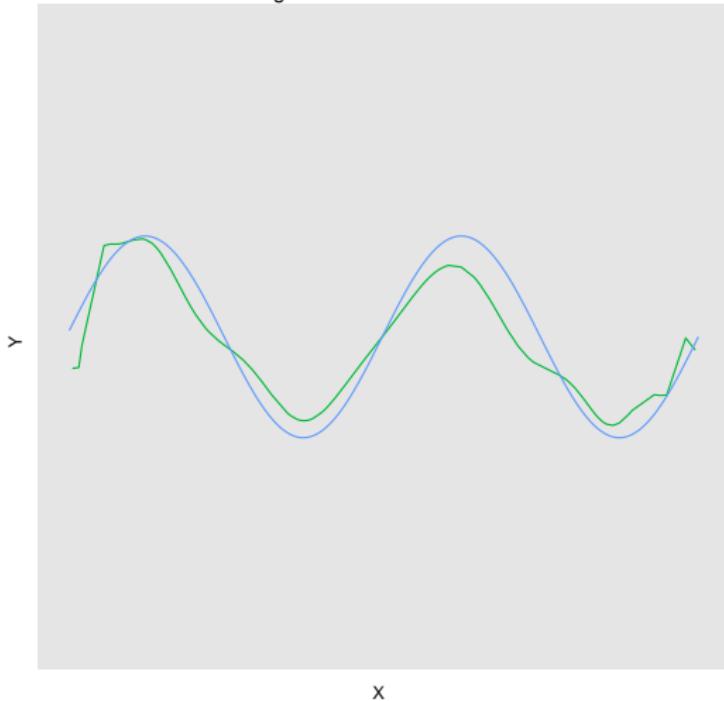
Lasso Regression with Lambda = 0.0677



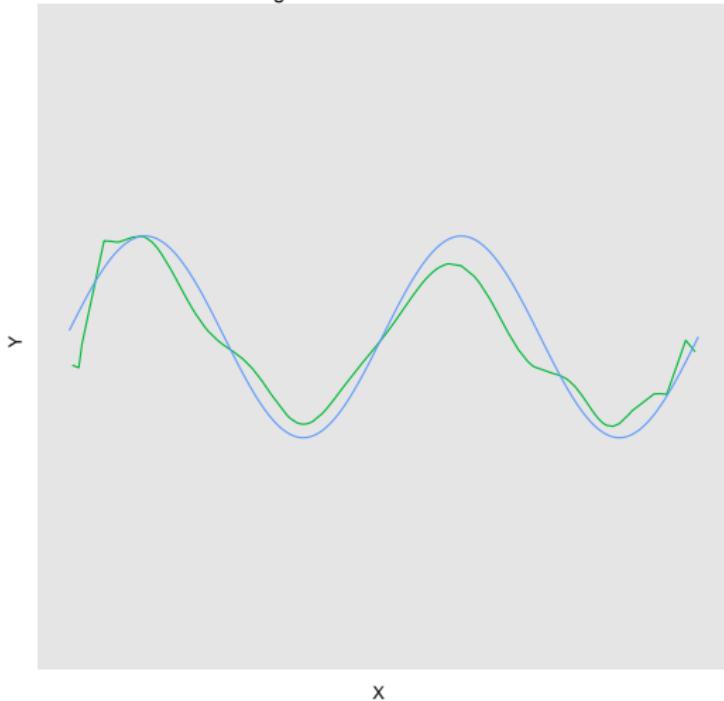
Lasso Regression with Lambda = 0.0617



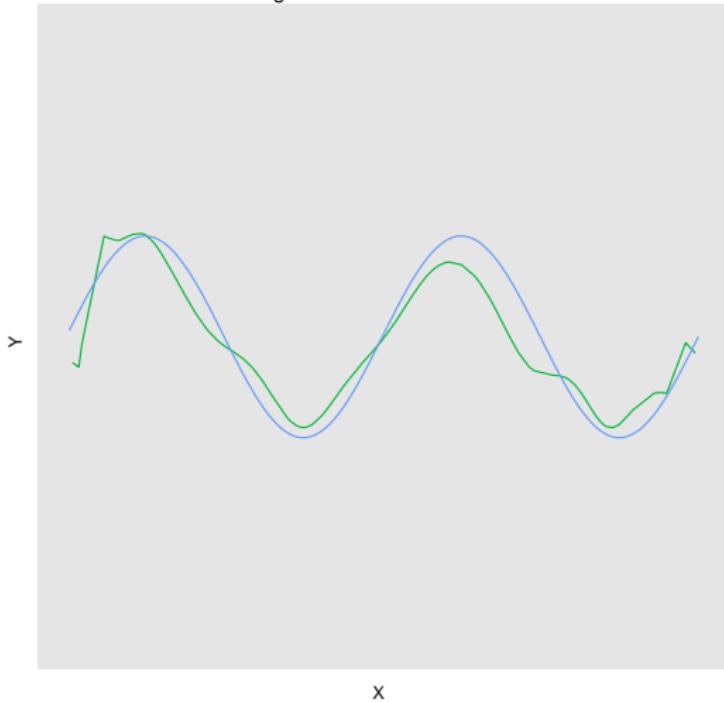
Lasso Regression with Lambda = 0.0562



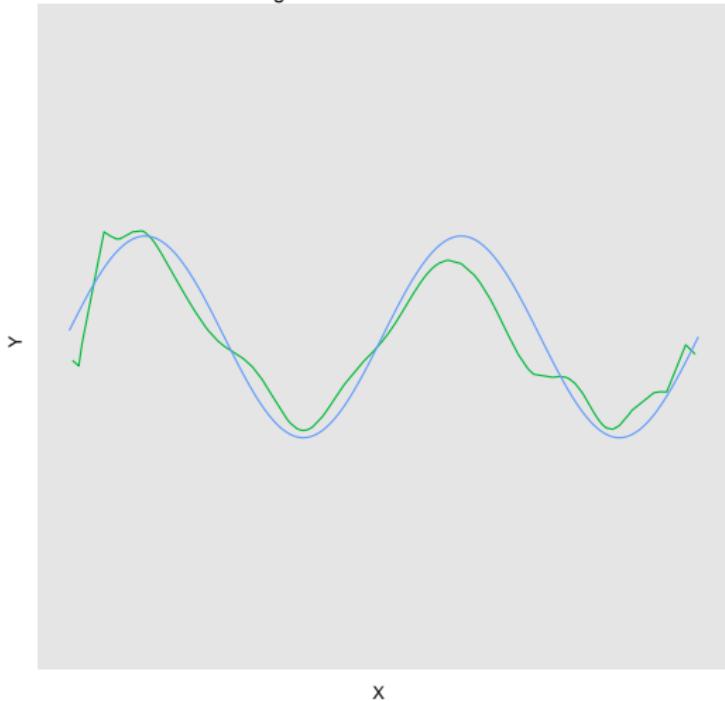
Lasso Regression with Lambda = 0.0512



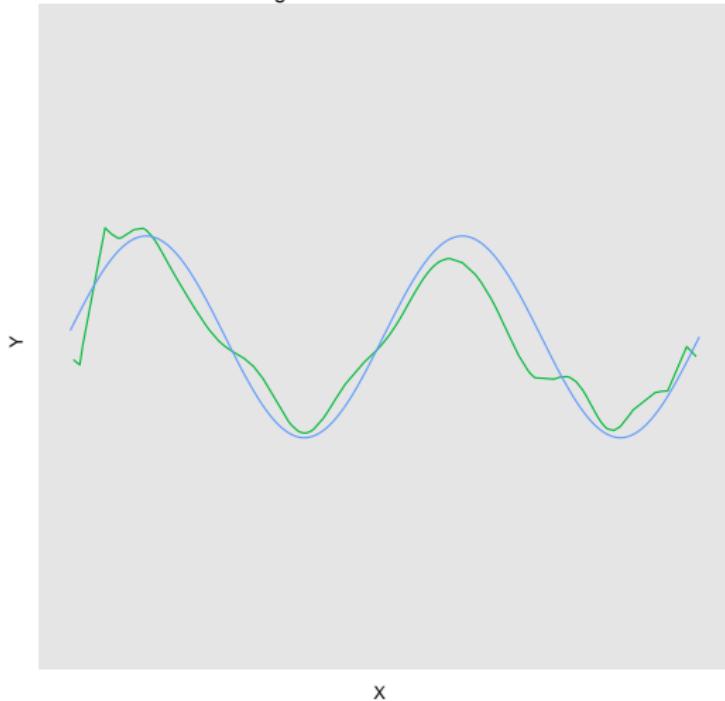
Lasso Regression with Lambda = 0.0467



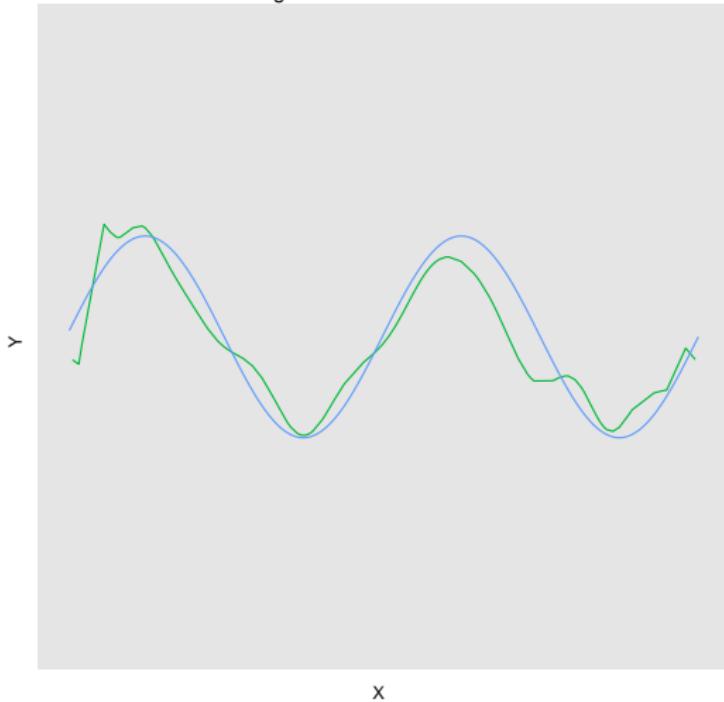
Lasso Regression with Lambda = 0.0425



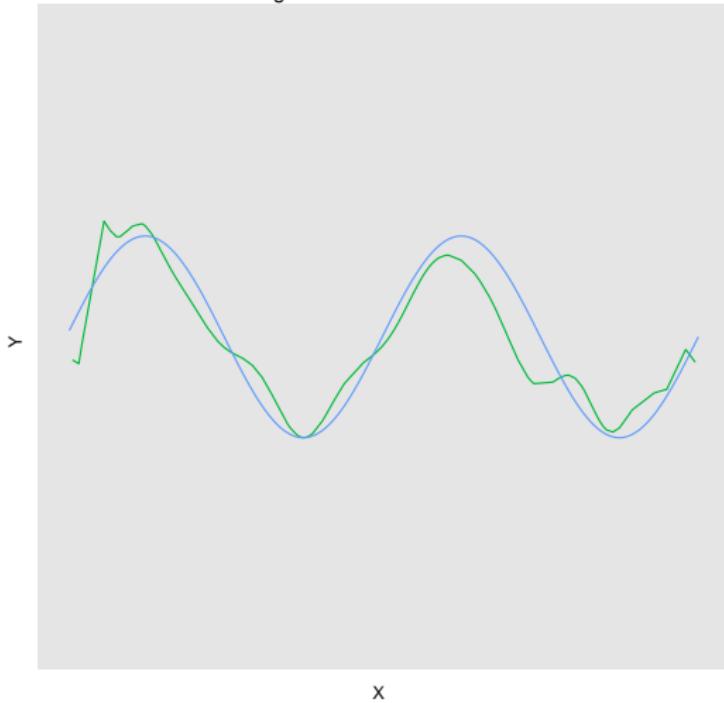
Lasso Regression with Lambda = 0.0387



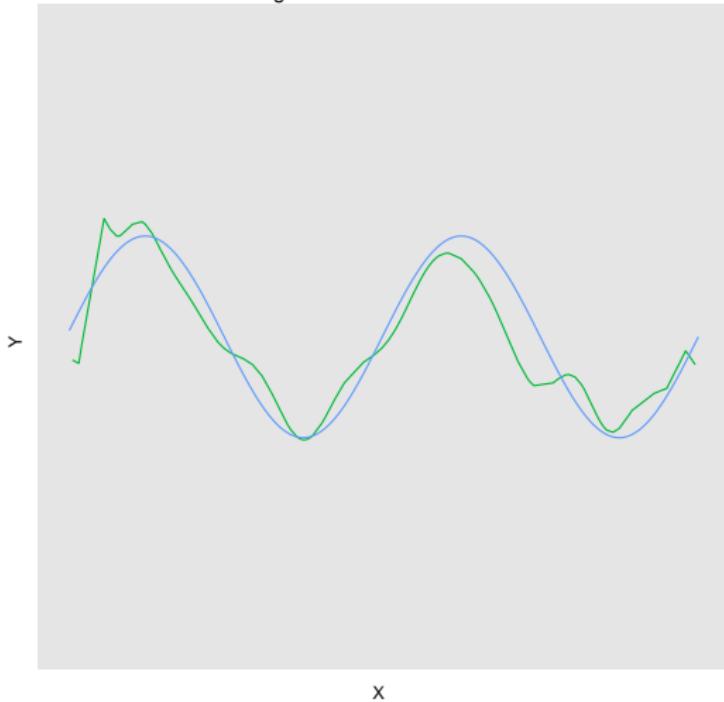
Lasso Regression with Lambda = 0.0353



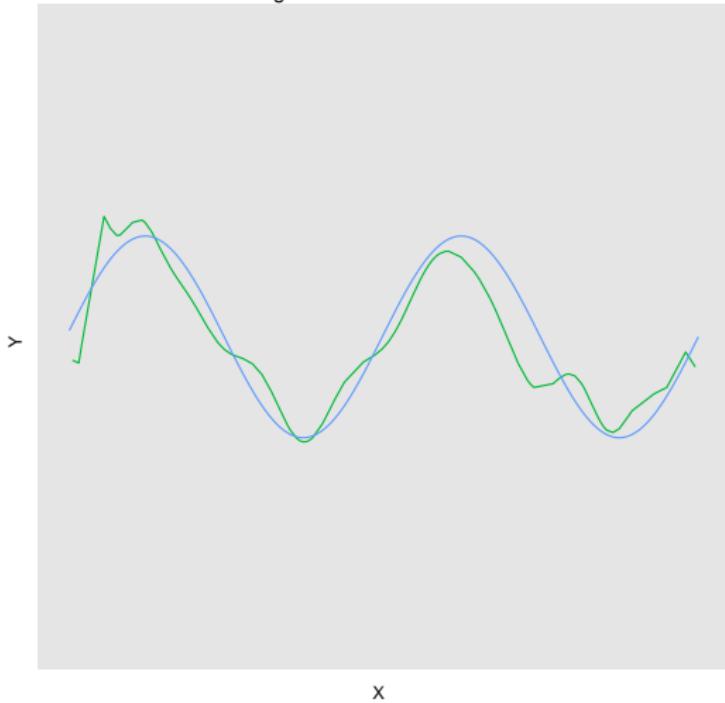
Lasso Regression with Lambda = 0.0322



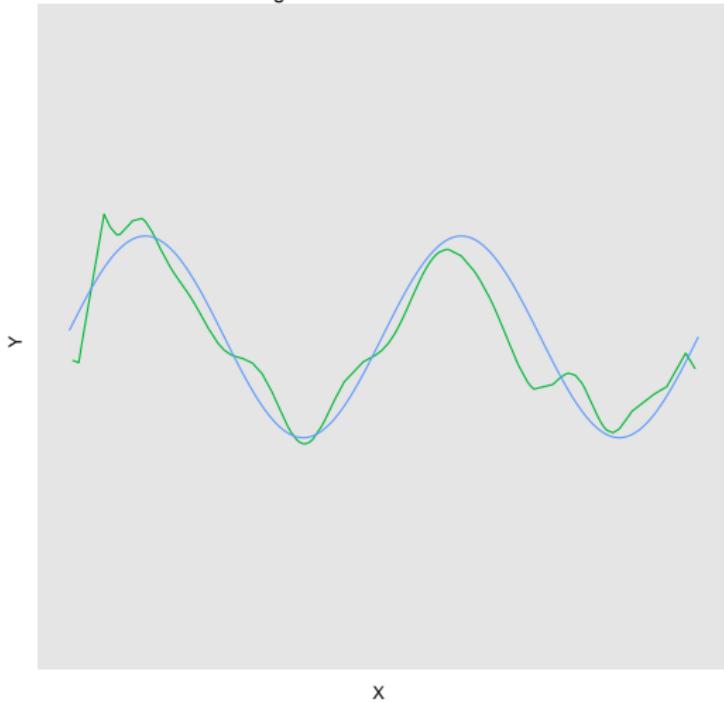
Lasso Regression with Lambda = 0.0293



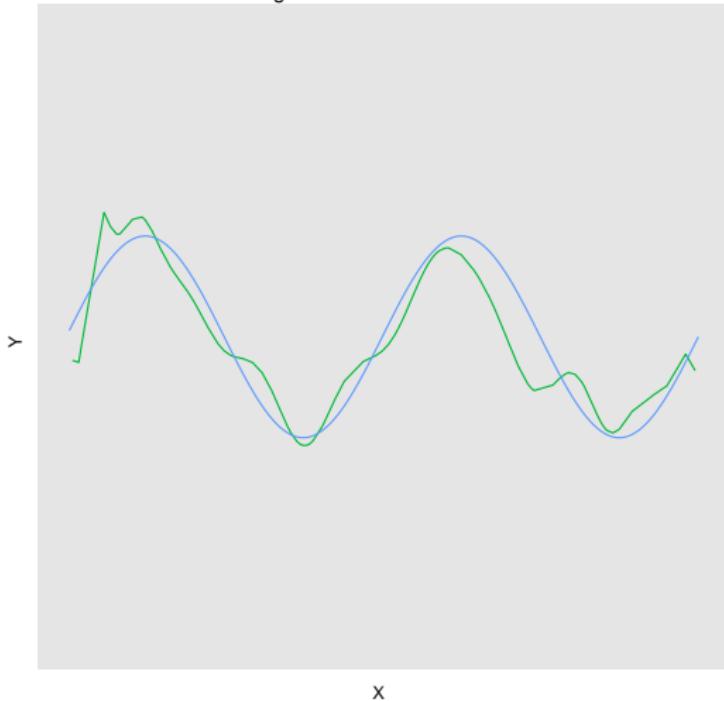
Lasso Regression with Lambda = 0.0267



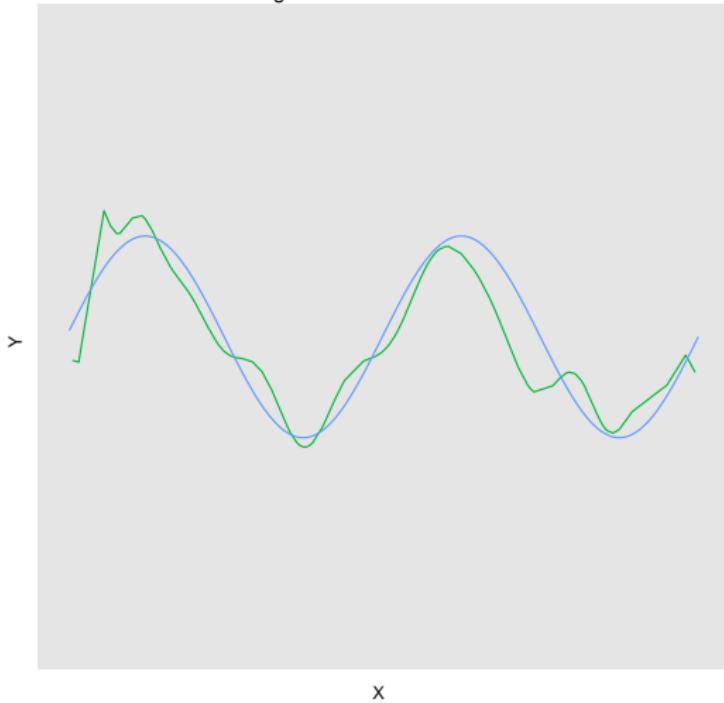
Lasso Regression with Lambda = 0.0243



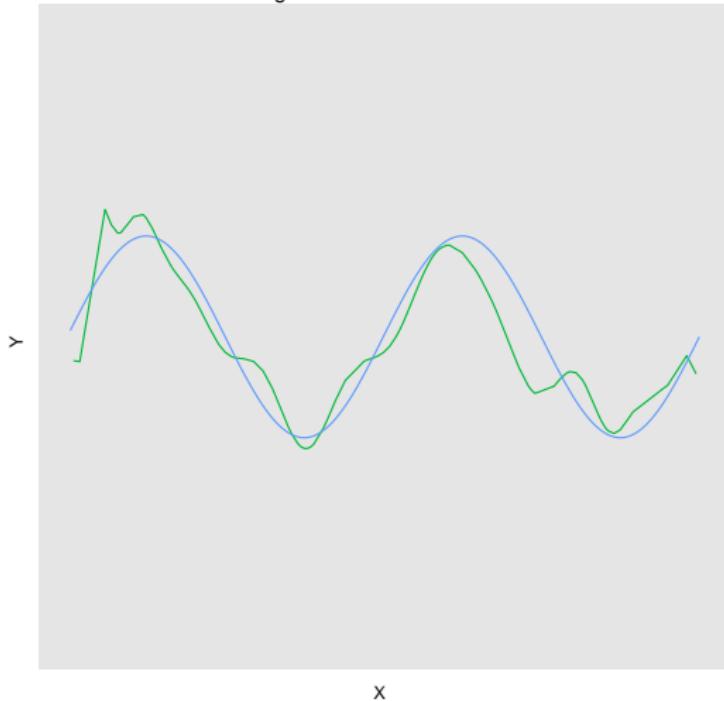
Lasso Regression with Lambda = 0.0222



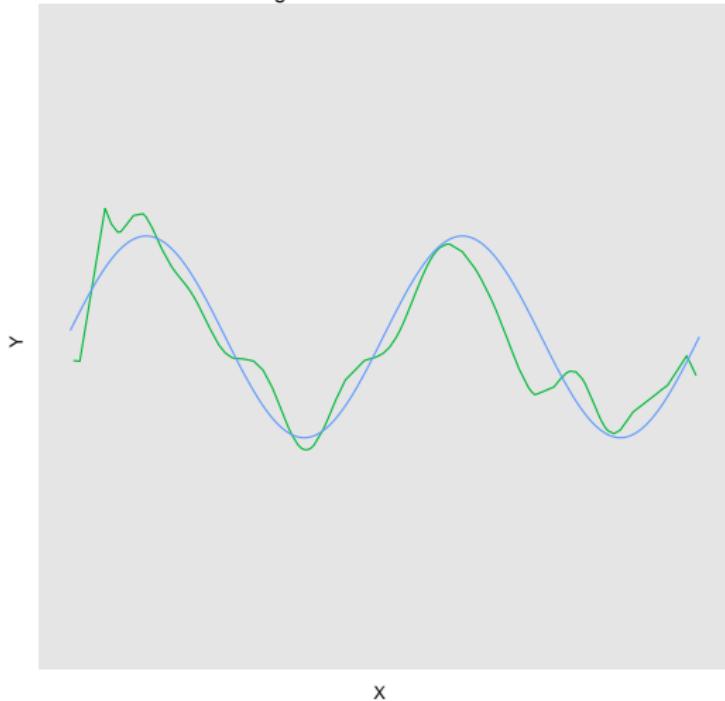
Lasso Regression with Lambda = 0.0202



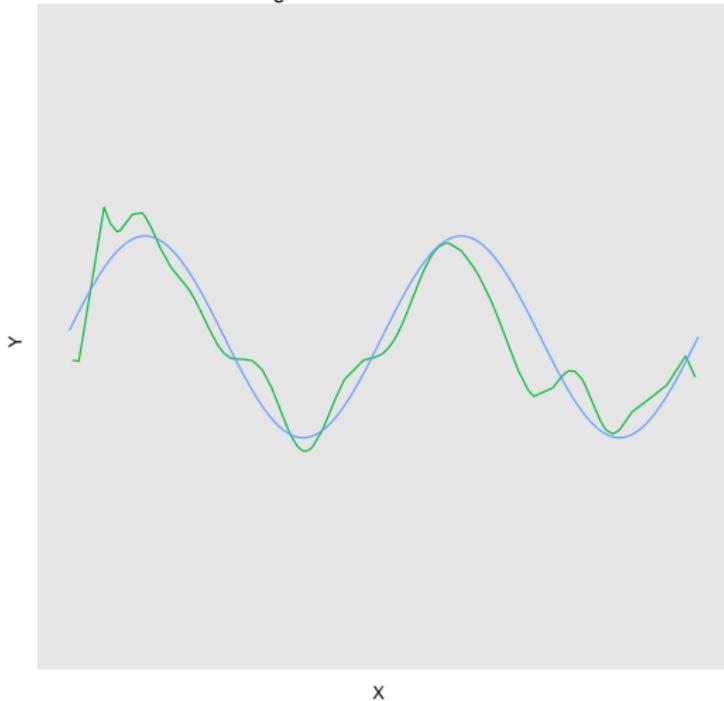
Lasso Regression with Lambda = 0.0184



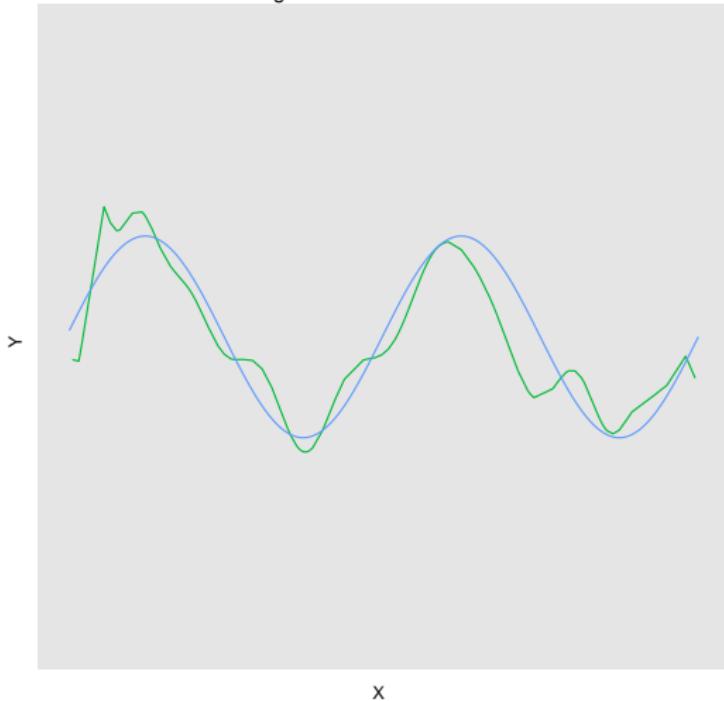
Lasso Regression with Lambda = 0.0168



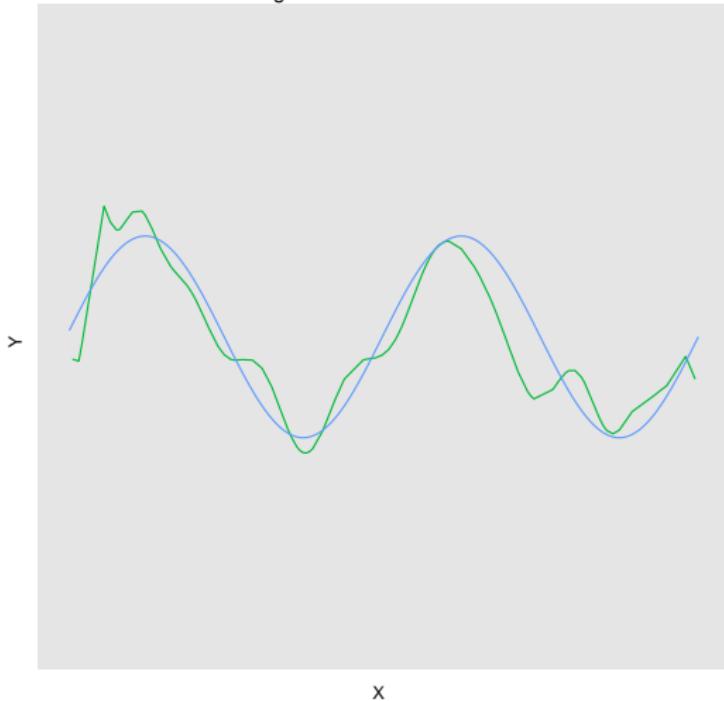
Lasso Regression with Lambda = 0.0153



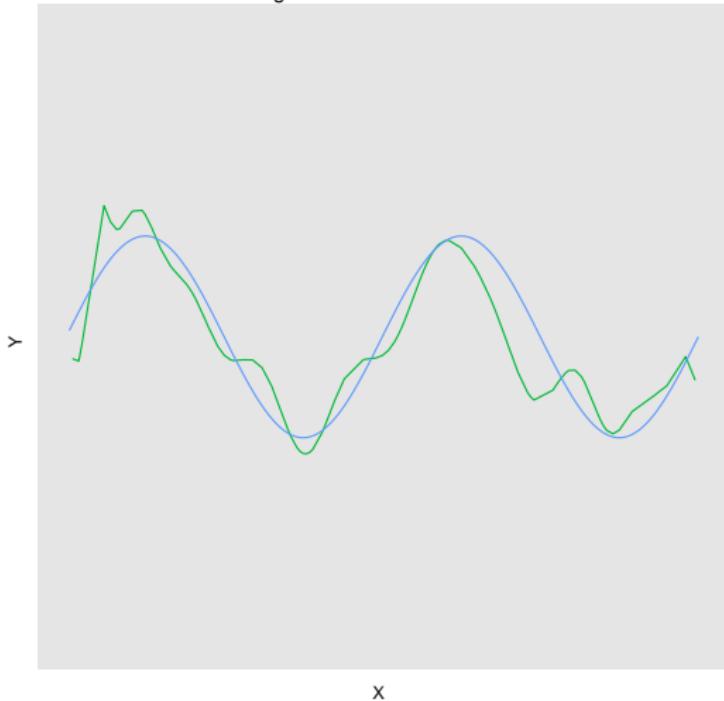
Lasso Regression with Lambda = 0.0139



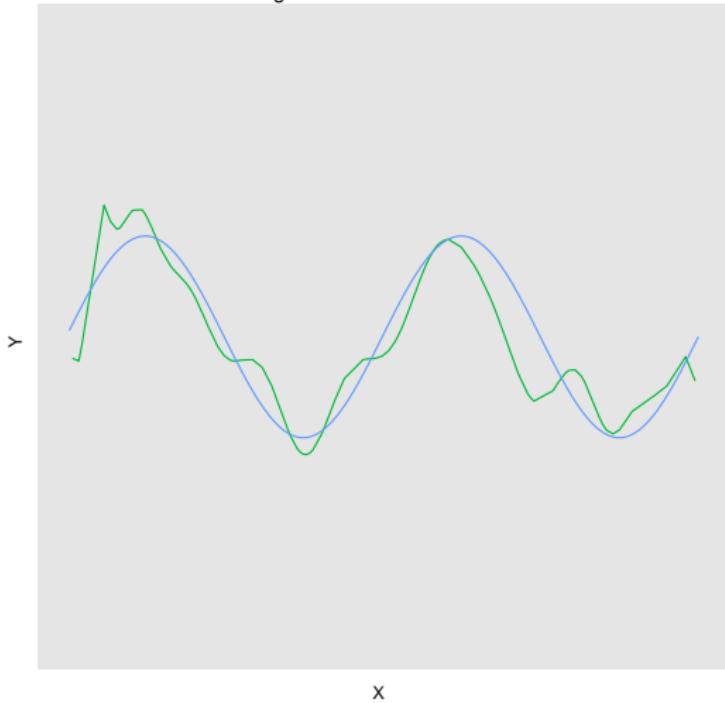
Lasso Regression with Lambda = 0.0127



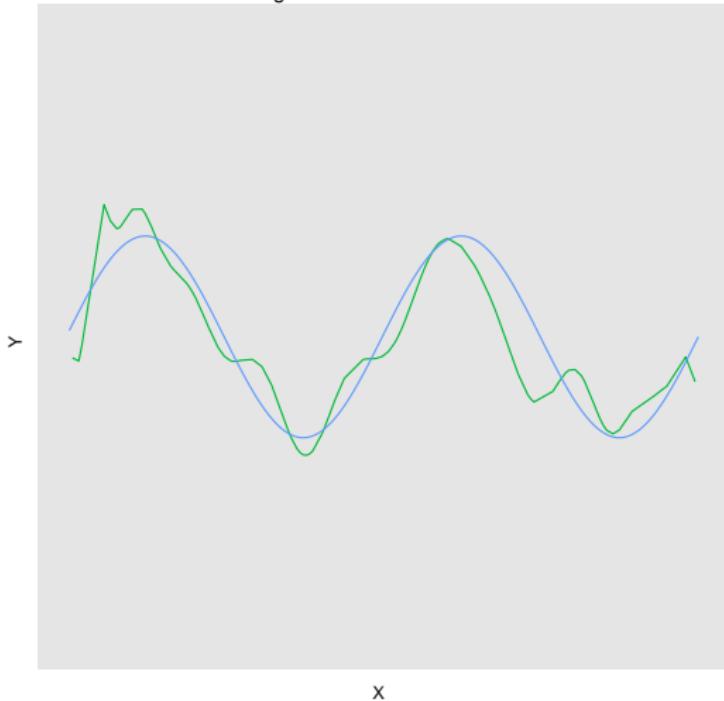
Lasso Regression with Lambda = 0.0116



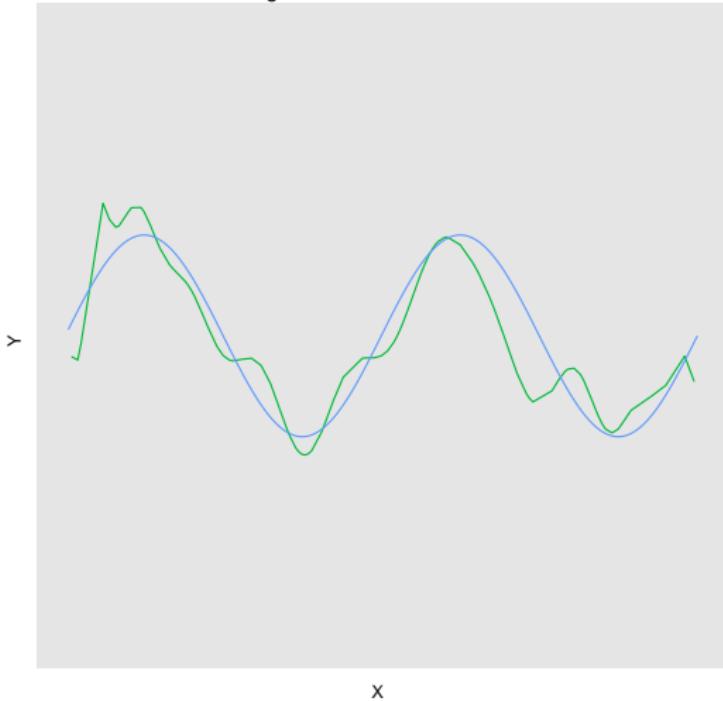
Lasso Regression with Lambda = 0.0105



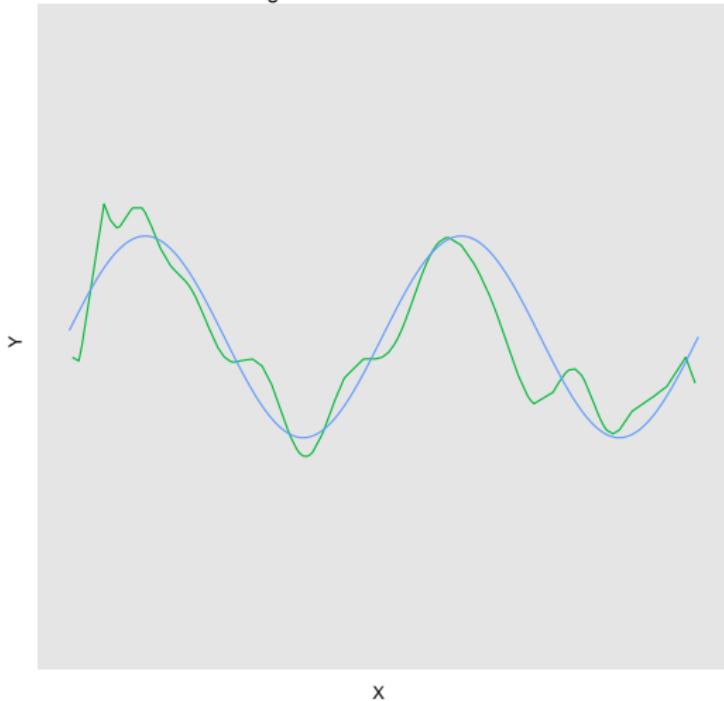
Lasso Regression with Lambda = 0.0096



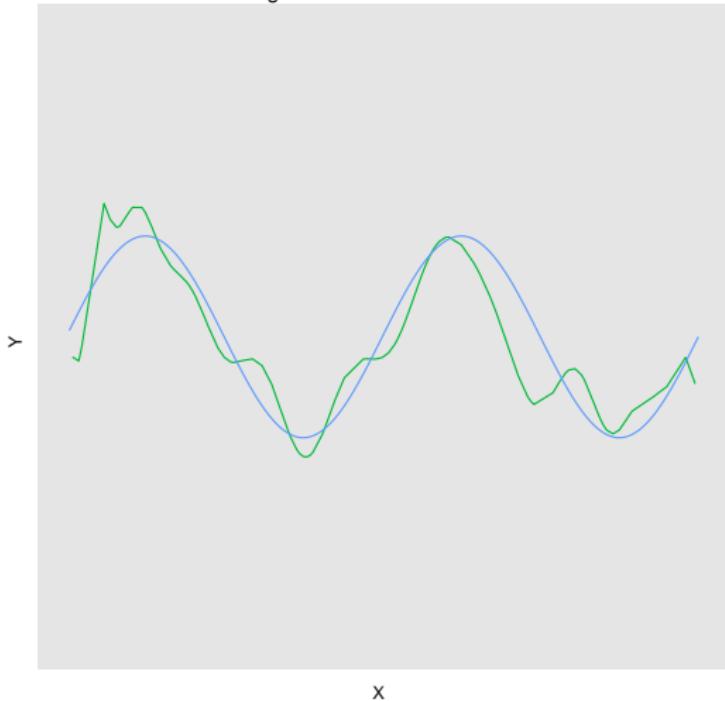
Lasso Regression with Lambda = 0.00874



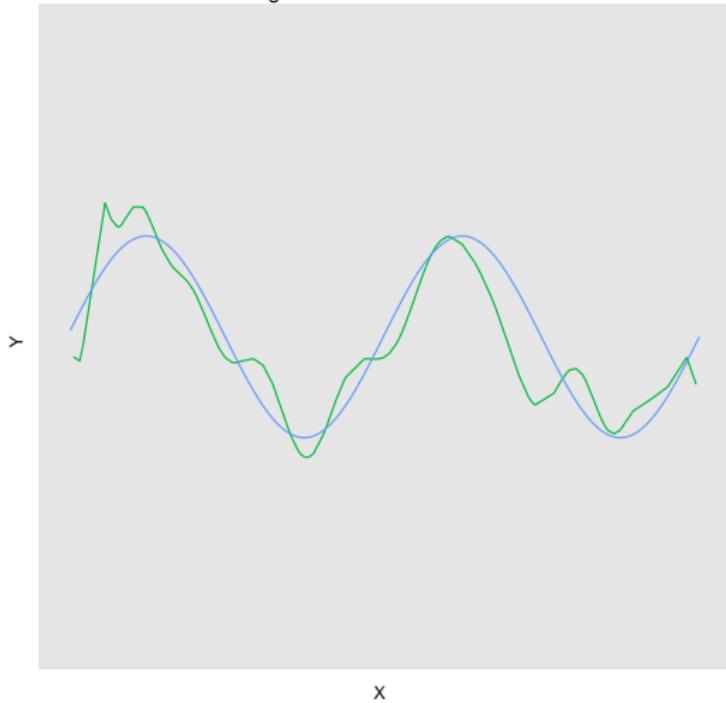
Lasso Regression with Lambda = 0.00797



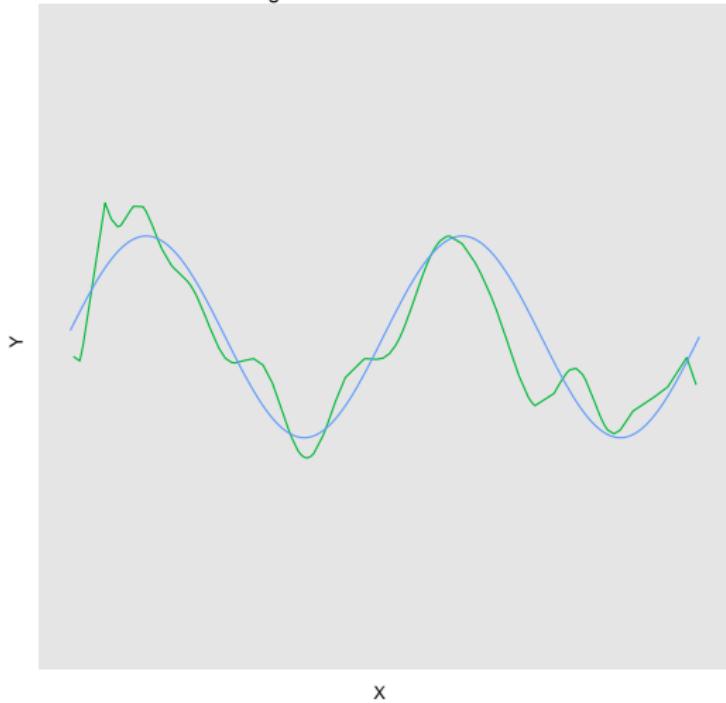
Lasso Regression with Lambda = 0.00726



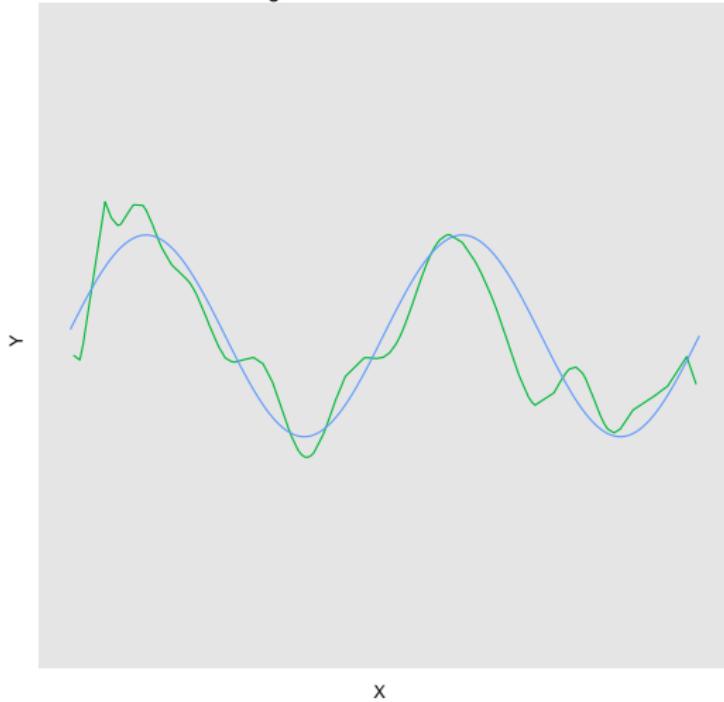
Lasso Regression with Lambda = 0.00661



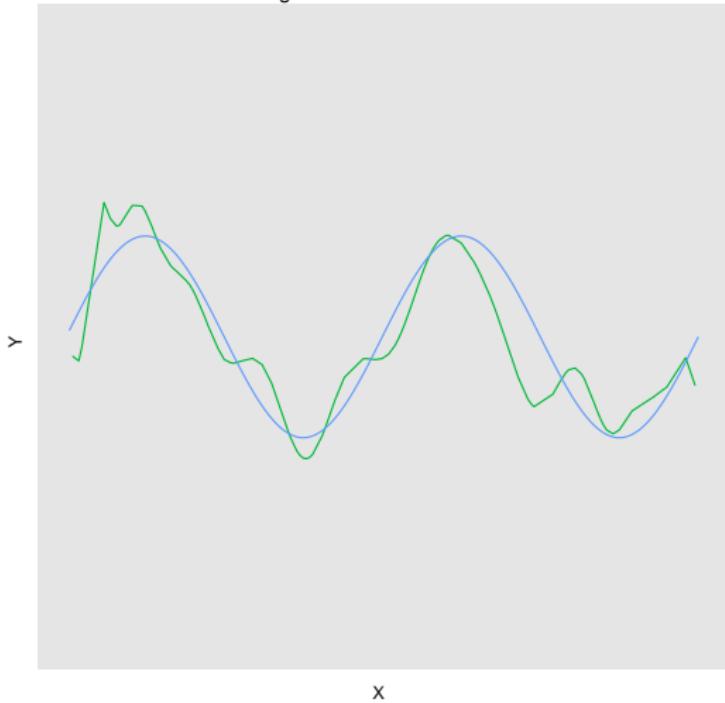
Lasso Regression with Lambda = 0.00603



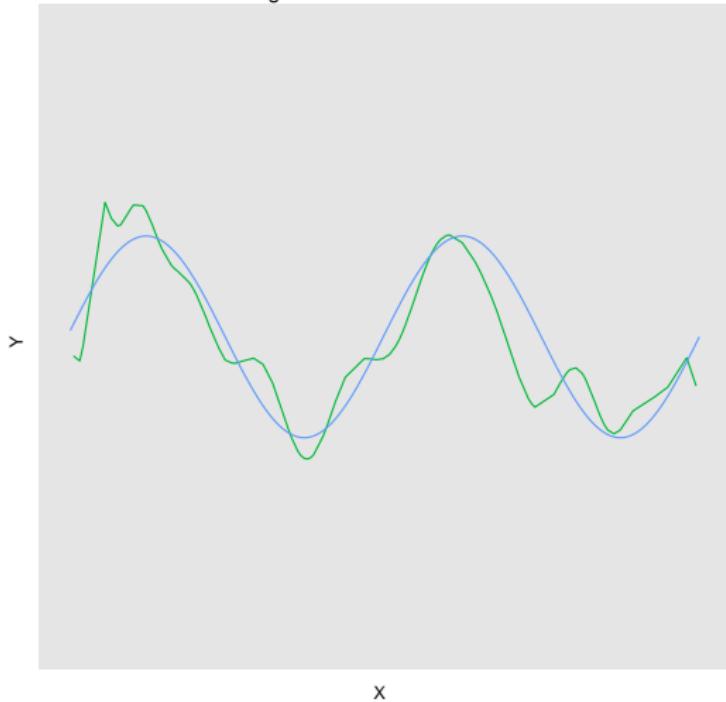
Lasso Regression with Lambda = 0.00549



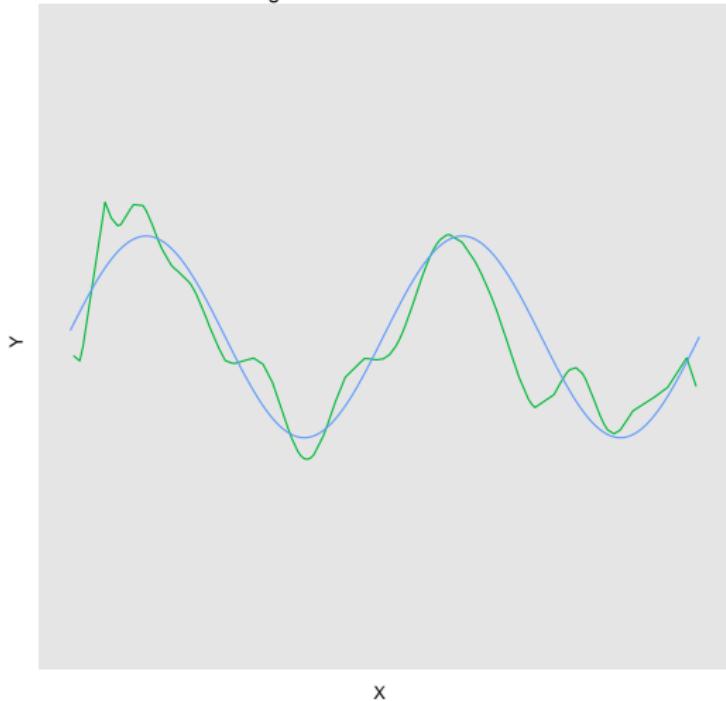
Lasso Regression with Lambda = 0.005



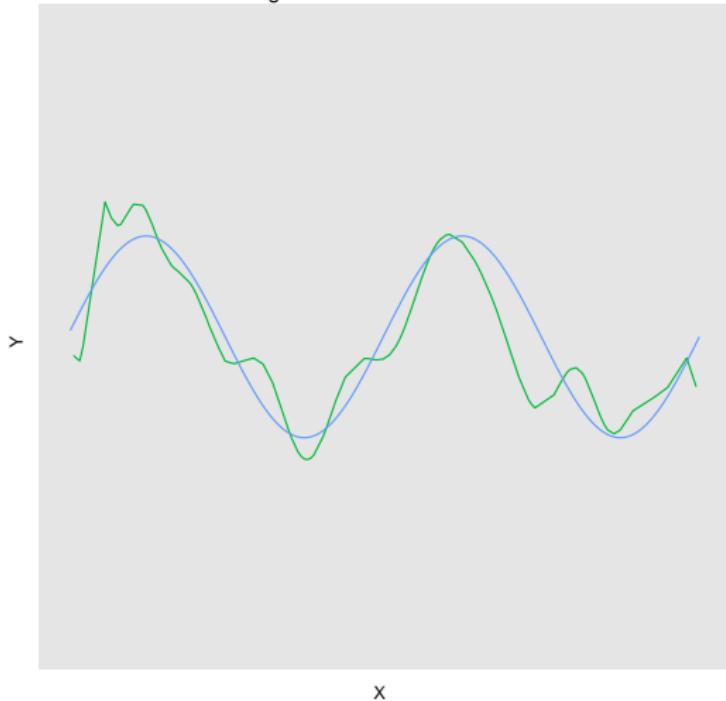
Lasso Regression with Lambda = 0.00456



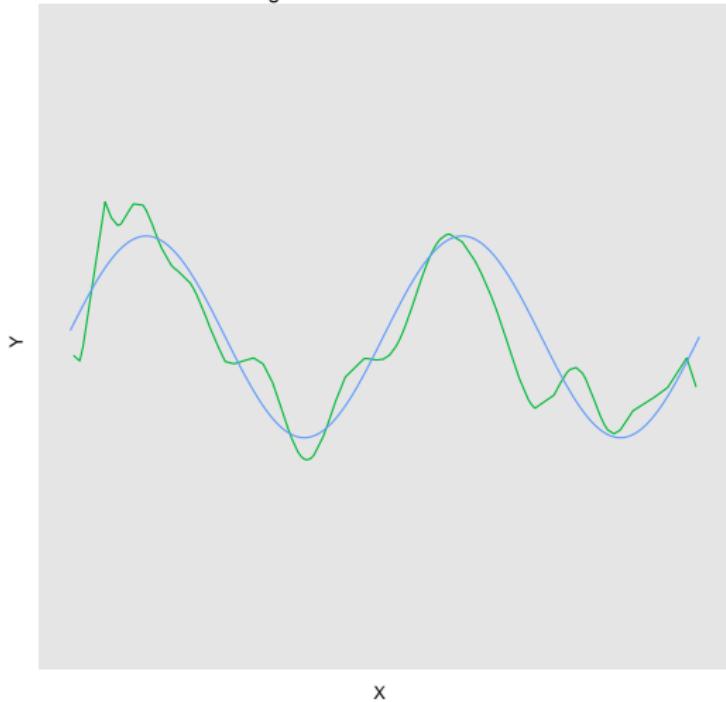
Lasso Regression with Lambda = 0.00415



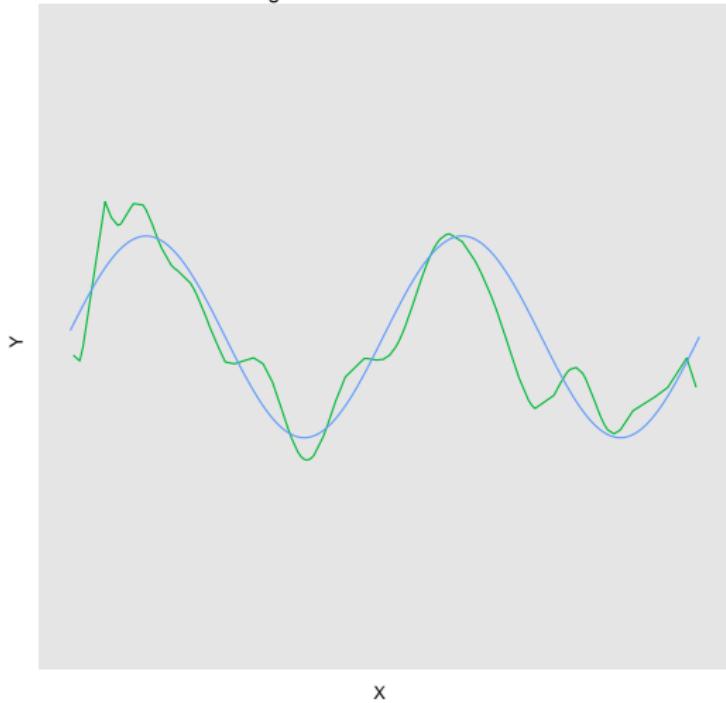
Lasso Regression with Lambda = 0.00378



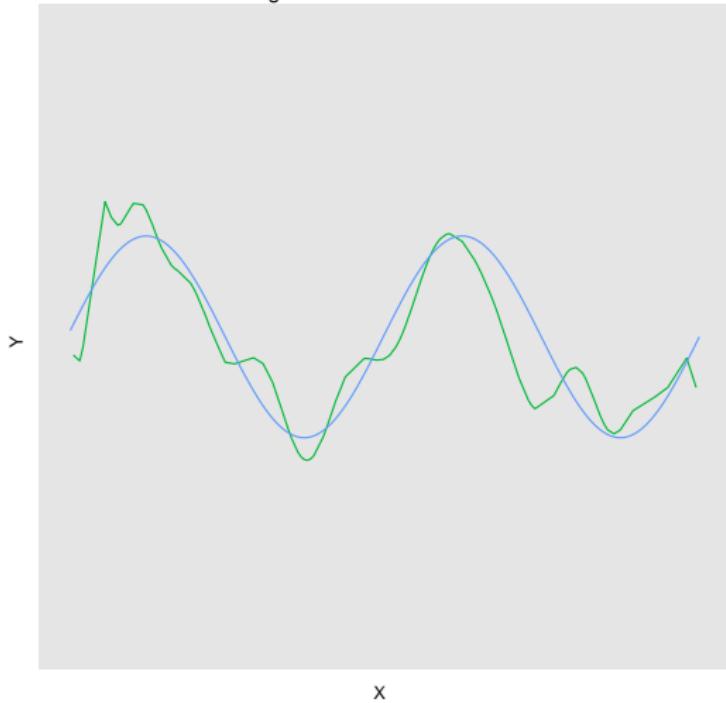
Lasso Regression with Lambda = 0.00345



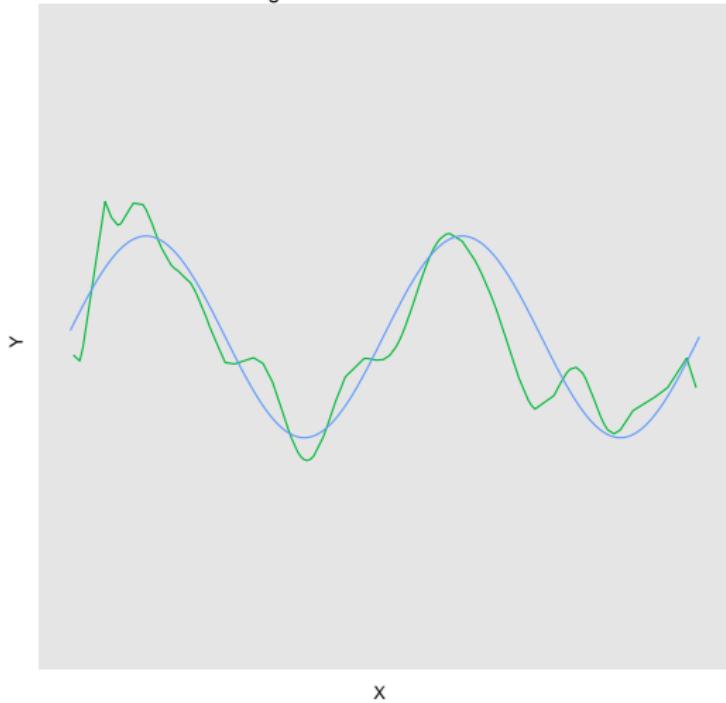
Lasso Regression with Lambda = 0.00314



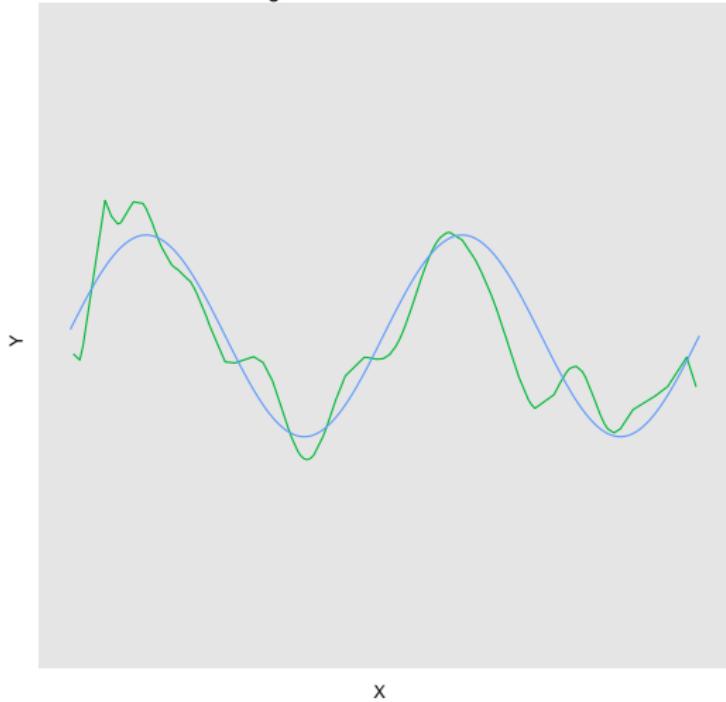
Lasso Regression with Lambda = 0.00286



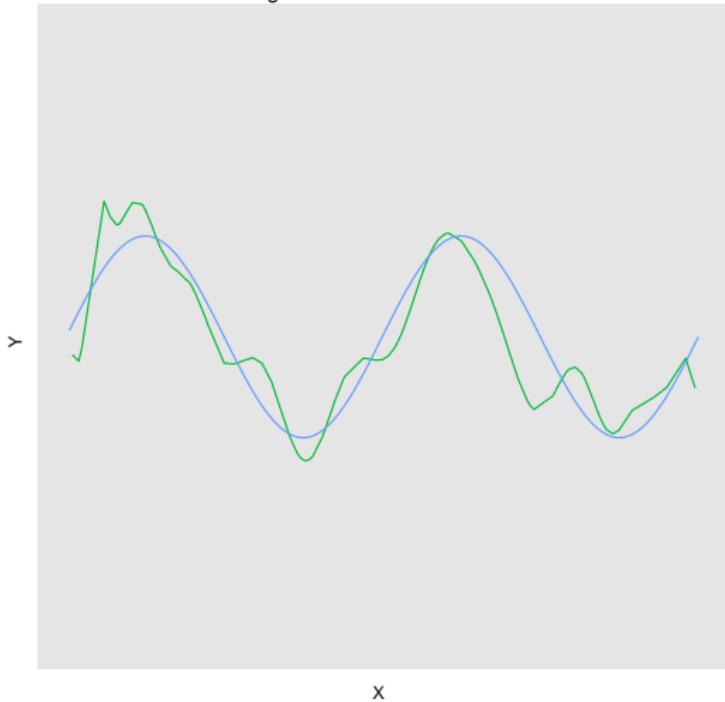
Lasso Regression with Lambda = 0.00261



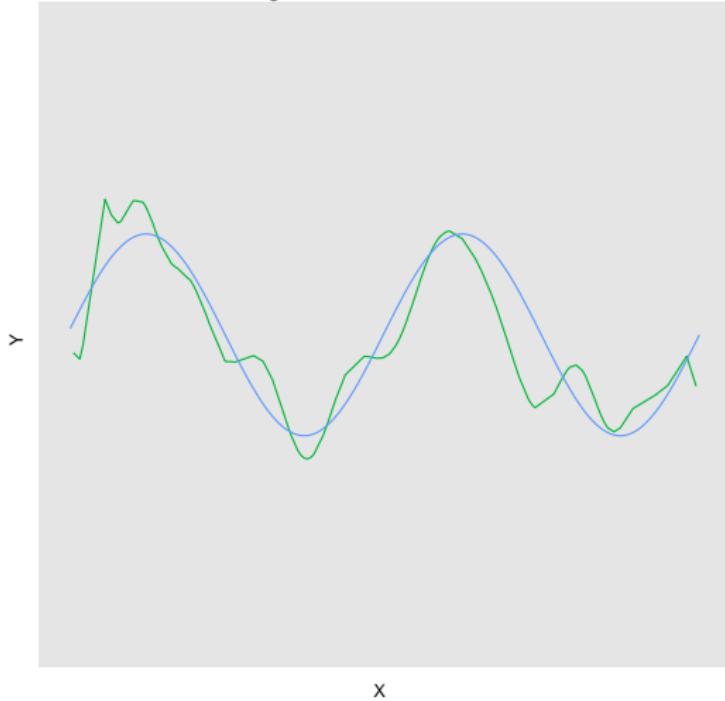
Lasso Regression with Lambda = 0.00238



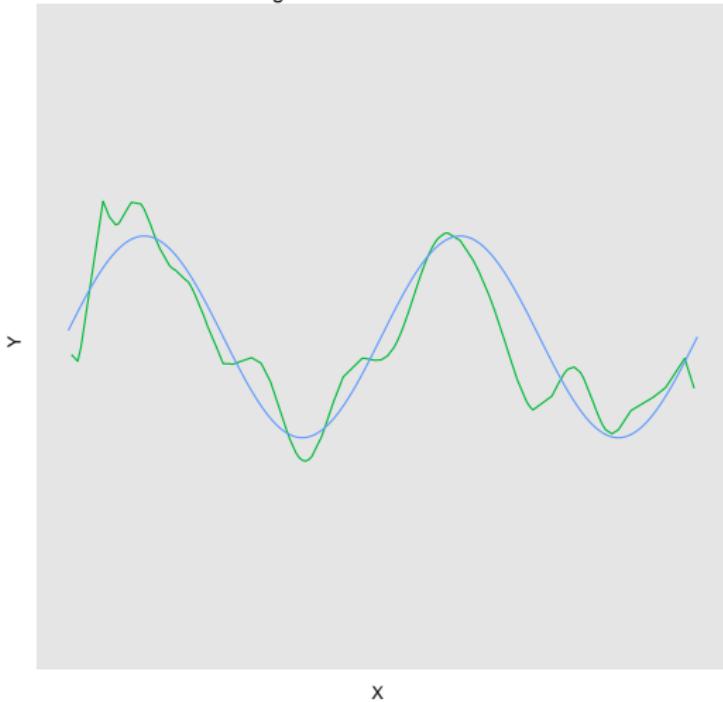
Lasso Regression with Lambda = 0.00217



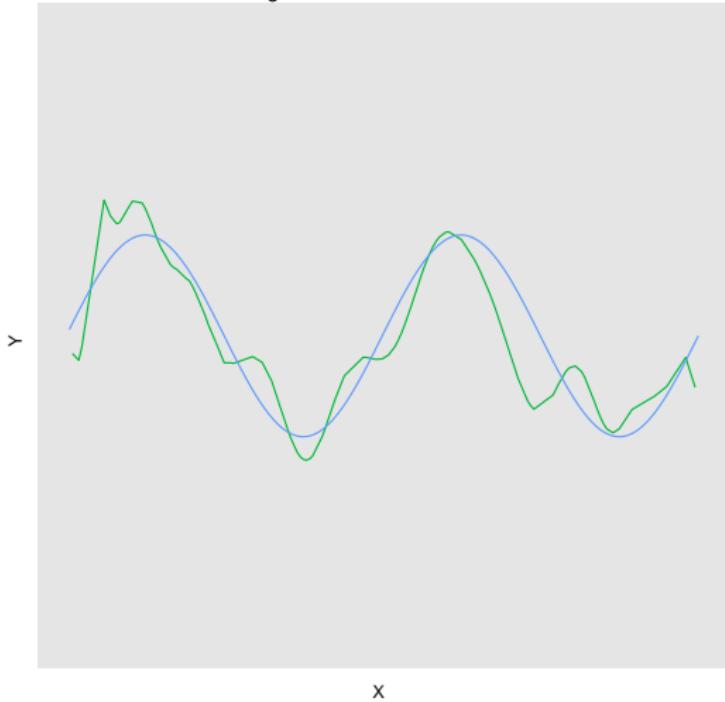
Lasso Regression with Lambda = 0.00197



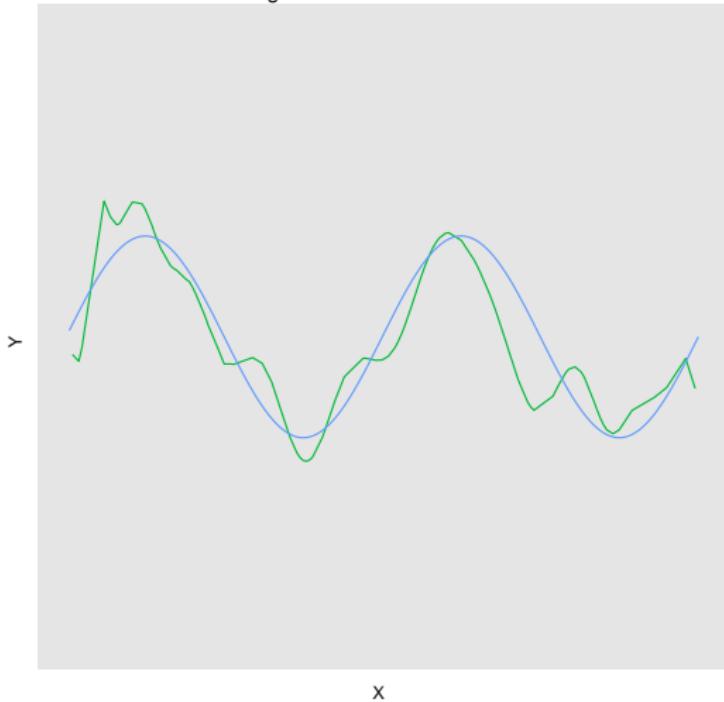
Lasso Regression with Lambda = 0.0018



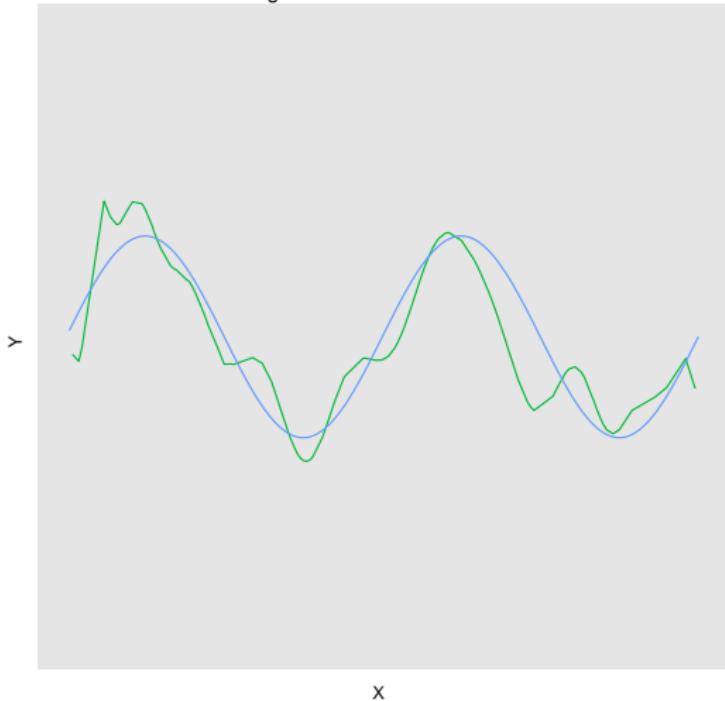
Lasso Regression with Lambda = 0.00164



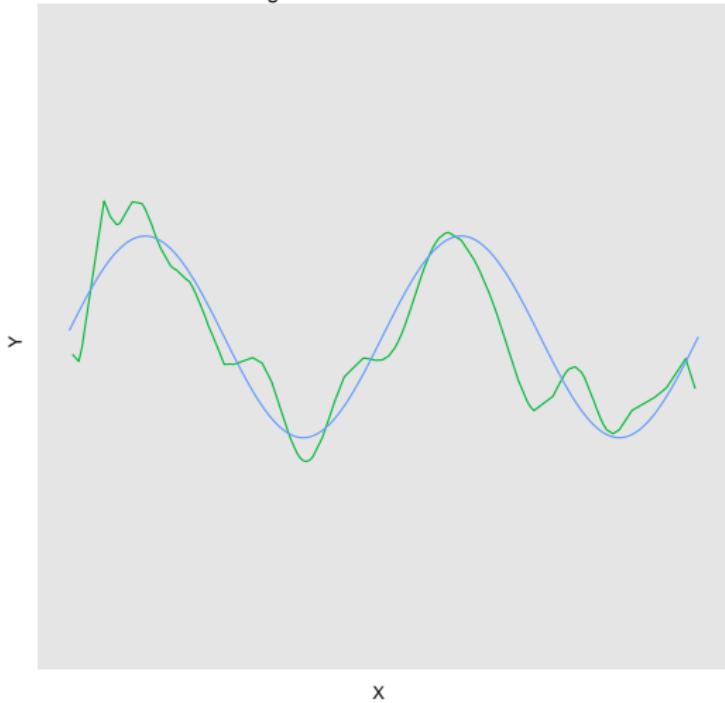
Lasso Regression with Lambda = 0.00149



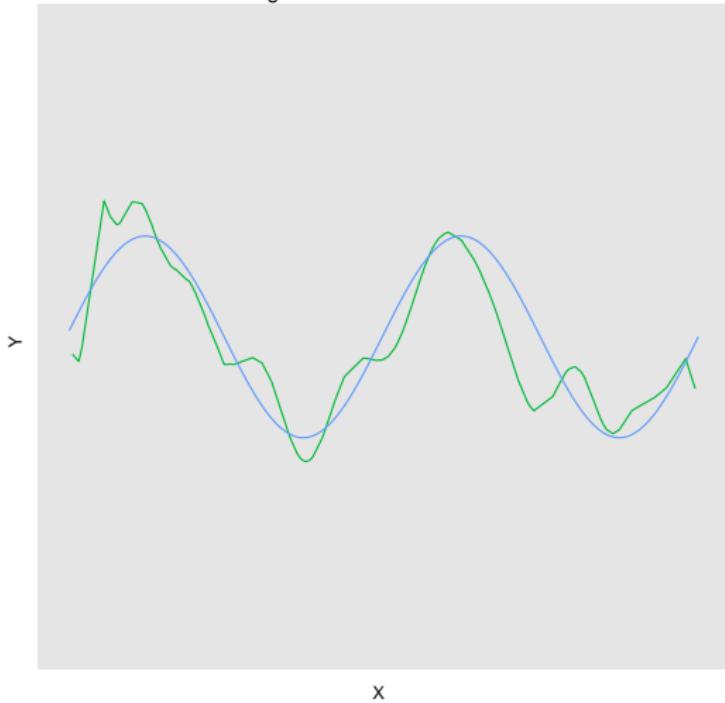
Lasso Regression with Lambda = 0.00136



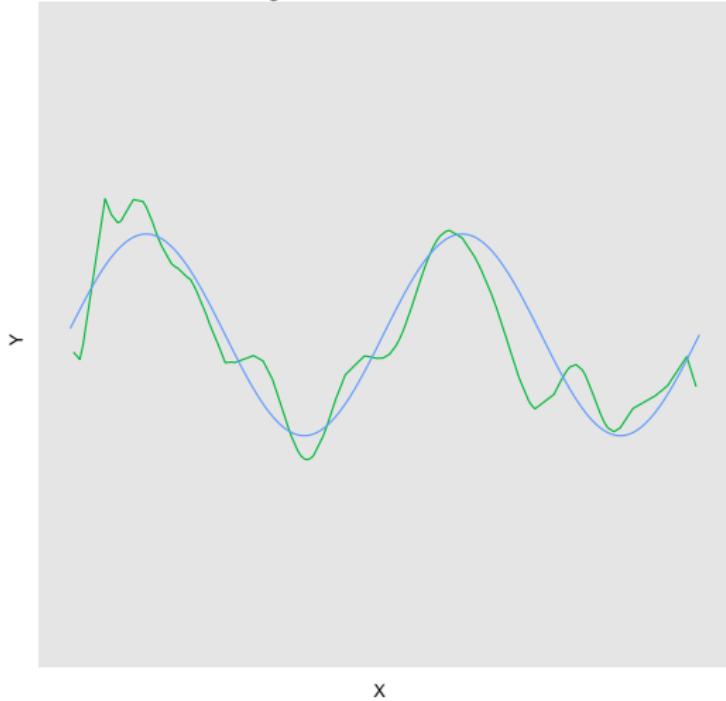
Lasso Regression with Lambda = 0.00124



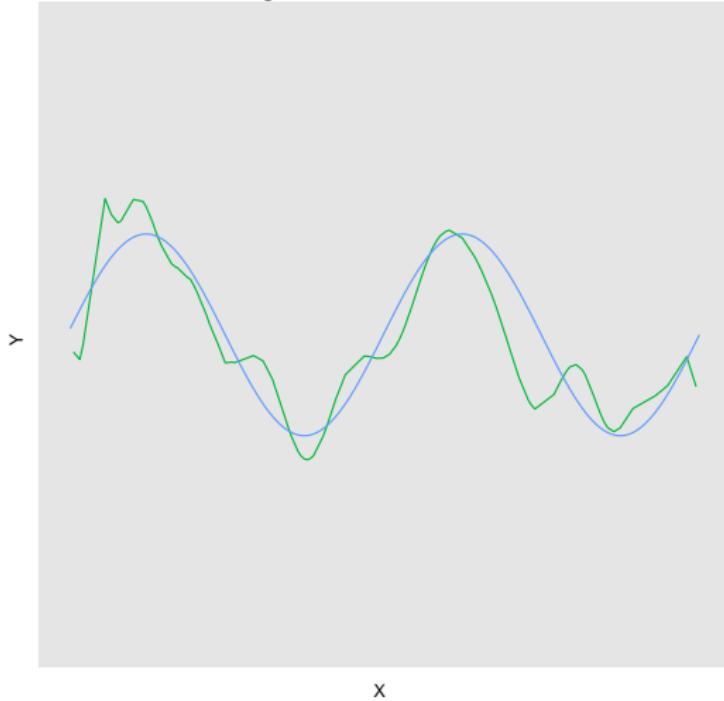
Lasso Regression with Lambda = 0.00113



Lasso Regression with Lambda = 0.00103

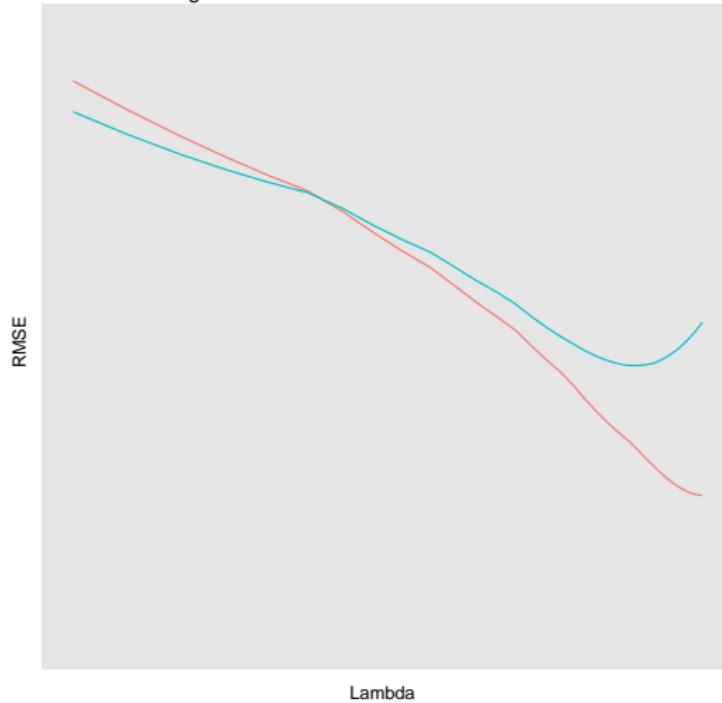


Lasso Regression with Lambda = 0.000937



At the start the Lasso works better, but eventually it breaks down

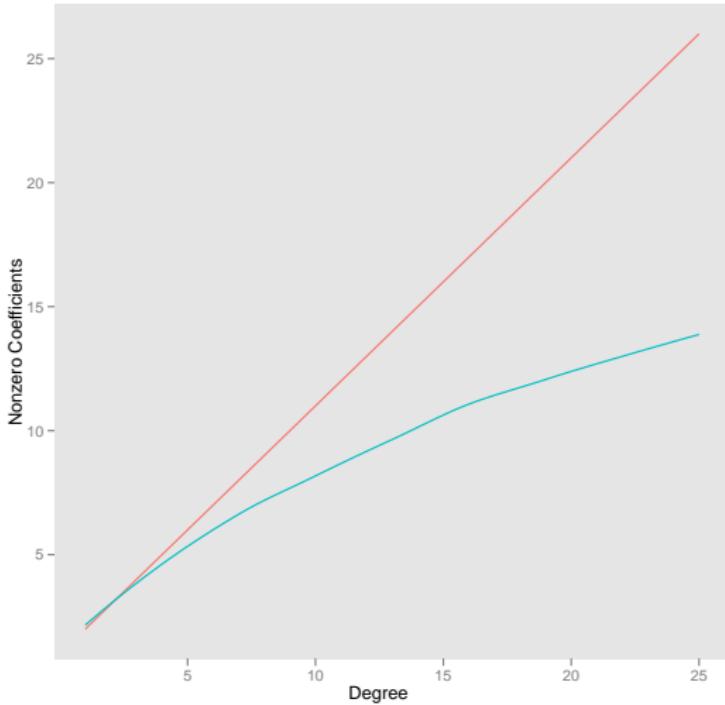
Training Set Performance versus Test Set Performance



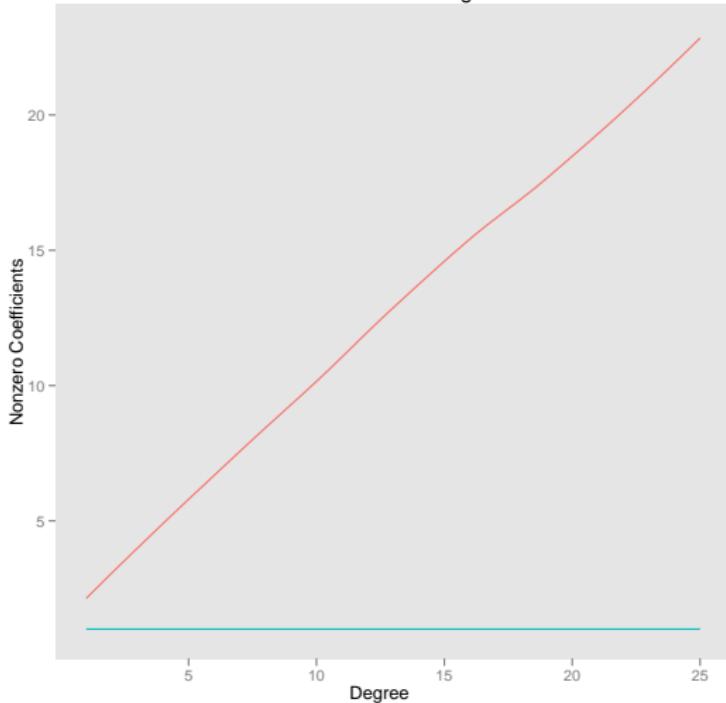
Why does the Lasso produce smoother models?

The Lasso finds sparse solutions

Nonzero Coefficients



Nonzero Coefficients with Larger Lambda



Bayesian interpretations:

- ▶ β is normally distributed with variance $f(\lambda)$
- ▶ β is Laplace distributed with variance $f(\lambda)$

Let's switch gears

How can we use regularization to solve real world problems?

If you work with text, you probably already use regularization

Pseudocounts are a form of regularization

Document Term Matrix:

Document	I	Want	Talk	Jobs	Week
A	1	2	2	3	2
B	0	0	0	0	1

Total word counts:

I	Want	Talk	Jobs	Week
1	2	2	3	3

How frequent is the word “Jobs”?

- ▶ k_i = number of times that word i occurs (i.e. “Jobs”)
- ▶ W = total number of words in corpus

We can calculate empirical frequencies

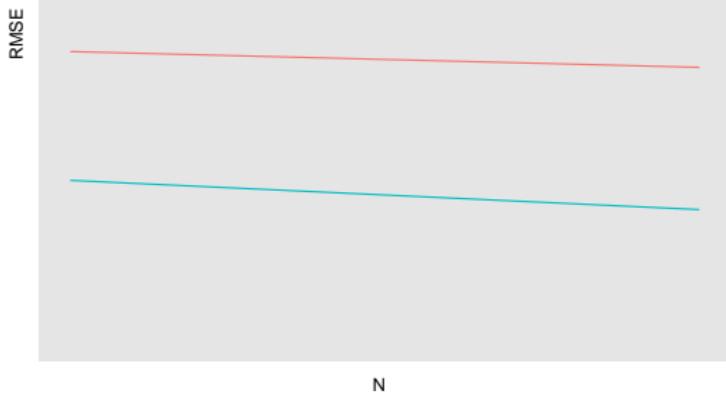
$$\frac{k_{jobs}}{\sum_{i=1}^W k_i} = \frac{3}{1+2+2+3+3} = \frac{3}{11}$$

Or we can add pseudocounts to every word

$$\frac{k_{jobs} + 1}{\sum_{i=1}^W k_i + 1} = \frac{(3+1)}{(1+1) + (2+1) + (2+1) + (3+1) + (3+1)} = \frac{4}{16}$$

In “small” samples, adding pseudocounts gives better estimates

Pseudocounts versus Empirical Frequencies



Adding pseudocounts is also called Laplace smoothing

Bayesian interpretations:

- ▶ $\text{Beta}(1, 1)$ prior
- ▶ $\text{Dirichlet}(1, \dots, 1)$ prior

There are other uses for regularization in text analysis

The text regression problem:

- ▶ N documents
- ▶ D words
- ▶ Predict continuous value for each document from word counts

Examples:

- ▶ From IPO notices, predict stock volatility
- ▶ From HTML, predict the number of pageviews per day
- ▶ From press releases, predict Congress member's politics

We need regularization because we:

- ▶ Observe more words than documents ($D > N$)
- ▶ Want sparse solutions, e.g. a few words that matter a lot
- ▶ Want to perform well on unseen data

From text, how can we find words that signal political views?

*Hey, does anybody notice this crazy thing that we're on
the road to socialism? I'm just saying. Wow. We got —
we got the SCHIPs thing going for us. That's great.*

How about that McDonalds two blocks from Ground Zero? That's killed more people than the nineteen hijackers.

Who thinks:

1. Text A was pro-Democrat and Text B was pro-Republican?
2. Text A was pro-Republican and Text A was pro-Democrat?

Example corpus statistics:

- ▶ $N = 1,408$ unique documents
- ▶ $D = 20,521$ unique words

Document A:

i want to talk about jobs lately it seems that everyone says they want to talk about jobs and that we'll get around to tackling jobs next week or the week after

Document B:

*there was a major legislative accomplishment in
washington last week and it's getting less attention than
it deserves because it isn't national health care reform*

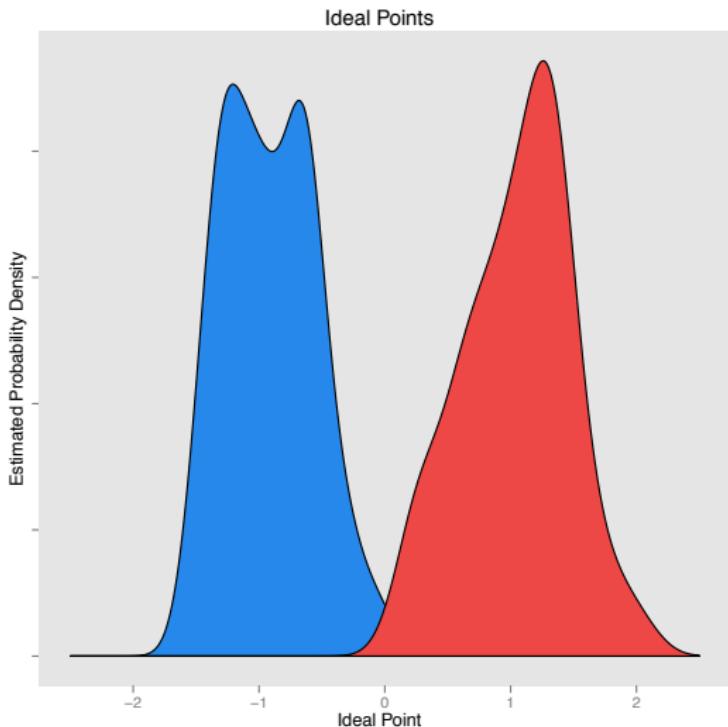
Document Term Matrix:

Document	I	Want	Talk	Jobs	Week
A	1	2	2	3	2
B	0	0	0	0	1

- ▶ Fit Lasso regression to document word counts
- ▶ Predict ideal points for the senators who wrote text

Example ideal points:

Senator	Ideal Point
Demint (R SC)	1.79
Snowe (R ME)	0.23
Bayh (D IN)	-0.15
Durbin (D IL)	-1.50



Which words are selected by the Lasso?

Top 10 Most Republican Terms:

Term	Value
okla	1.23
bailey	0.647
johnny	0.588
administering	0.561
neb	0.556
sam	0.542
986	0.532
texans	0.493
patriotism	0.466
demint	0.417

Top 10 Most Democratic Terms:

Term	Value
sherrod	-0.367
sheldon	-0.249
dec	-0.196
possess	-0.168
salaries	-0.158
tom	-0.152
debbie	-0.151
dark	-0.148
lautenberg	-0.133
fought	-0.106

Debugging:

- ▶ Too many names of senators in our list
- ▶ Strip out all the names from corpus
- ▶ Run Lasso from scratch on cleaner corpus

Top 10 Most Republican Terms excluding Names:

Term	Value
okla	1.13
neb	0.726
bailey	0.674
2415	0.638
986	0.578
kansans	0.543
administering	0.516
texans	0.467
profoundly	0.459
patriotism	0.430

Top 10 Most Democratic Terms excluding Names:

Term	Value
cedar	-0.224
chaired	-0.197
dec	-0.158
dark	-0.146
blocked	-0.138
reverses	-0.134
1960s	-0.125
insurers	-0.0958
fought	-0.0926
possess	-0.0923

Regularization lets us find signal in text data

Concluding remarks:

- ▶ When you fit many parameters, use regularization
- ▶ “Many parameters” starts at 3 (cf. James-Stein Theorem)
- ▶ Always test your model for overfitting on held out data
- ▶ Bayesian methods automatically regularize parameters

References:

- ▶ Pattern Recognition and Machine Learning
- ▶ Elements of Statistical Learning
- ▶ Bayesian Data Analysis
- ▶ Noah Smith's CMU web page

Useful resources:

- ▶ `poly` - R function for polynomial regression
- ▶ `glmnet` - R function for regularized regression
- ▶ `ggplot2` - R package for generating graphics
- ▶ `tm` - R package for text analysis

Thanks:

- ▶ David Blei
- ▶ Drew Conway
- ▶ Lindsey Cormack
- ▶ Thomas Zeitzoff
- ▶ Kevin Collins
- ▶ Joey Markowitz