Chapter 6: Linear Model Selection & Regularization

In the regression setting, the standard linear model is commonly used to describe the relationship between a response Y and a set of variables $X_1, \ldots, X_p$.

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \varepsilon$$

typically fit w/ least squares.

Upcoming: more flexible models (non-linear).

The linear model has distinct advantages in terms of inference and is often surprisingly competitive for prediction. How can it be improved?

replace least squares w/ alternative fitting procedures.

We can yield both better *prediction accuracy* and *model interpretability*:

- prediction accuracy: If true relationship is $\approx$ linear, least squares will have low bias.
  If $n \gg p \Rightarrow$ also low variance $\Rightarrow$ perform well on test data.
  But if n not much larger than $p \Rightarrow$ higher variability $\Rightarrow$ poor performance.
  If $p > n$: no longer a unique solution $\Rightarrow$ variance $= \infty \Rightarrow$ cannot be used at all!
  goal: reduce variance without adding too much bias.

- model interpretability: often many variables in regression are not in fact associated w/ response.
  By removing them (setting $\hat{\beta}_i = 0$), we can obtain a more easily interpretable model.
  Note: least squares will hardly ever result in $\hat{\beta}_i = 0$.
  $\Rightarrow$ need variable selection.

Same ideas apply to logistic regression.

1 Subset Selection

We consider methods for selecting subsets of predictors.

1.1 Best Subset Selection

To perform *best subset selection*, we fit a separate least squares regression for each possible combination of the p predictors.

↖ $\binom{p}{2} = \frac{p(p-1)}{2}$ models w/ exactly 2 predictors, etc.

Algorithm:

1. Let $\mathcal{M}_0$ denote the null model - no predictors.
2. For $k = 1, \ldots, p$
   (a) Fit all $\binom{p}{k}$ models that contain k predictors.
   (b) Pick the best of these (call $\mathcal{M}_k$). Best is defined by ↓RSS (↑R^2).
3. Select a single best model from $\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_p$ using CV error, C_p, AIC/BIC, or adjusted R^2

Note don't use R^2 for step 3 because as p↑, R^2↑ always. Why might we not want to do this at all? (more later)

We can perform something similar with logistic regression.

Fitting 2^p models. $p = 10 \Rightarrow 1000$ models.

1.2 Stepwise Selection

For computational reasons, best subset selection cannot be performed for very large p. * → impossible w/ $p \geq 40$.

Best subset may also suffer when p large because w/ a large search space can find models that work well on training data but poorly on test data ⇒ high variability & overfitting can occur.

Stepwise selection is a computationally efficient procedure that considers a much smaller subset of models.

Forward Stepwise Selection: Start w/ no predictors and add predictors one at a time until all predictors are in the model. Choose the "best" from these.

1. Let $\mathcal{M}_0$ denote the null model - no predictors.
2. For $k = 0, \ldots, p-1$
   (a) Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$ w/ 1 additional predictor.
   (b) Choose the best among $p-k$ and call it $\mathcal{M}_{k+1}$ (↑R^2, ↓RSS).
3. Select a single best model from $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using CV error, C_p, AIC/BIC, or adjusted R^2.

Now we fit $1 + \sum_{k=0}^{p-1} (p-k) = 1 + \frac{p(p+1)}{2}$ models.

Backward Stepwise Selection: Begin w/ full model and take predictors away one at a time until you get to null. Choose from sequence.

Similar algorithm to forwards stepwise selection.

* Neither forward nor backwards stepwise selection are guaranteed to find the best model containing a subset of the p predictors.

forward selection can be used when $p > n$ (but only up to $n-1$ predictors, not p!).

1.3 Choosing the Optimal Model

Best subset, forward, backward selection require a way to pick the "best model" – according to test error.

RSS & R^2 are proxies for training error ⇒ not good estimates of test error.

↳ either (1) estimate this directly or (2) adjust training errors for model size.

(2) $C_p = \frac{1}{n}(RSS + 2d\hat{\sigma}^2)$ ← estimate of variance of ε (full model). ↑ # predictors in subset model

adds a penalty to training error (RSS) to adjust for underestimation of test error.

Choose model w/ lowest value.

(2) AIC & BIC Can compute for maximum likelihood fits.

proportional ⇒ same answer.

$AIC = \frac{1}{n\hat{\sigma}^2}(RSS + 2d\hat{\sigma}^2)$

$BIC = \frac{1}{n\hat{\sigma}^2}(RSS + \log(n)\, d\hat{\sigma}^2)$

Since $\log(n) > 2$ for $n > 7$ ⇒ heavier penalty on models w/ many variables ⇒ results in smaller models.

Choose model w/ lowest BIC.

(2) Adjusted R^2 (only for least squares).

$R^2 = 1 - \frac{RSS}{TSS}$ always ↑ as d ↑

$\text{Adj } R^2 = 1 - \frac{RSS/(n-d-1)}{TSS/(n-1)}$

Choose model w/ highest adj. R^2.

* (1) Validation and Cross-Validation

- Directly estimate test error w/ validation or CV and choose model w/ lowest estimated error.
- Very general (can be used for any model) even when it not clear how many "predictors" we have.

Now have fast computers ⇒ these are preferred.

2 Shrinkage Methods

The subset selection methods involve using least squares to fit a linear model that contains a subset of the predictors. As an alternative, we can fit a model with all p predictors using a technique that constrains (*regularizes*) the estimates.

↳ shrinks estimates towards zero.

Shrinking the coefficient estimates can significantly reduce their variance!

Helps us to avoid overfitting.

2.1 Ridge Regression

Recall that the least squares fitting procedure estimates $\beta_1, \ldots, \beta_p$ using values that minimize

$$RSS = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

residual sum of squares.

Ridge Regression is similar to least squares, except that the coefficients are estimated by minimizing $\hat{\beta}^R$

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = RSS + \lambda \boxed{\sum_{j=1}^{p} \beta_j^2}$$

note we do not penalize β_0 we want to penalize the relationships not the intercept (mean value of response when $x_{i1} = \ldots = x_{ip} = 0$).

$\lambda \geq 0$ tuning parameter (determined separately of the fitting procedure).

trades off 2 criteria: minimize RSS to fit the data well.

$\lambda \sum_{j=1}^{p} \beta_j^2$ shrinkage penalty small when β_j close to zero $\Rightarrow$ shrinks estimates towards zero.

The tuning parameter λ serves to control the impact on the regression parameters.

When $\lambda = 0$ penalty has no effect and ridge regression = least squares.

As $\lambda \to \infty$, impact of penalty grows and $\hat{\beta}^R \to 0$.

Ridge regression will produce a different set of coefficients for each penalty λ $\left(\hat{\beta}_\lambda^R\right)$

Selecting a good λ is critical! How to choose? CV!

The standard least squares coefficient estimates are scale invariant.

Multiplying X_j by a constant c leads to a scaling of least squares coefficients by a factor of $\frac{1}{c}$.

$\Rightarrow$ regardless of how X_j is scaled, $X_j \hat{\beta}_j$ will remain the same.

In contrast, the ridge regression coefficients $\hat{\beta}_\lambda^R$ can change substantially when multiplying a given predictor by a constant.

e.g. say we have an income variable in (1) dollars and (2) thousands of dollars.

(1) = (2) $\times$ 1000

due to the sum of squared coef. term, this will simply scale the coefficient estimate by a factor of 1000.

$\Rightarrow X_j \hat{\beta}^R_{j,\lambda}$ depends not only on λ but also on the scaling of X_j.

(may even depend on scaling of other predictors!).

Therefore, it is best to apply ridge regression *after standardizing the predictors* so that they are on the same scale:

i.e. have standard deviation of 1.

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}}$$

estimate of st. deviation of j^{th} predictor.

scale() or in workflow() using step_ in recipe.

(1) standardize data

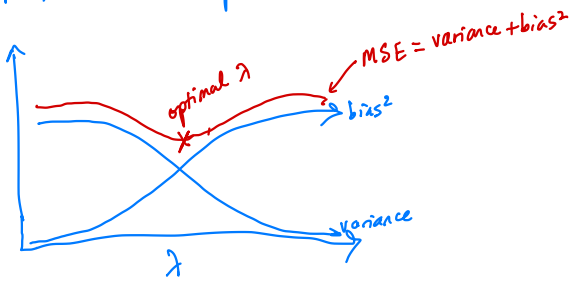
(2) tune model to choose λ via CV.

(3) fit ridge regression model.

Why does ridge regression work?

Because of the bias-variance tradeoff!

As $\lambda \uparrow$, the flexibility of the ridge regression $\downarrow \Rightarrow \downarrow$ variance and $\uparrow$ bias.



In situations where relationship between response and predictors is $\approx$ linear, least squares will have low bias.

When p almost as large as $n \Rightarrow$ least squares has high variability!
if $p > n$ least squares doesn't even have a unique solution.

$\Rightarrow$ ridge regression can still perform well in these scenarios by trading off a small amount of bias for a decrease in variance.

$\Rightarrow$ Ridge works best in high variance scenarios.

Also
Cost advantage over subset selection.
b/c for a fixed λ, only fit one model. (very fast model to fit).

Ridge improves predictive performance.

Does it also help us w/ interpretation? No.

2.2 The Lasso

Ridge regression does have one obvious disadvantage.

Unlike subset selection methods (generally select model w/ a subset of variables), ridge regression will include all p variables in the final model.

penalty $\lambda \sum \beta_j^2$ will shrink $\beta_j \to 0$ but $\beta_j \neq 0$ (unless $\lambda = \infty$)!

This may not be a problem for prediction accuracy, but it could be a challenge for model interpretation when p is very large.

We will always have all variables in model, whether they have a relationship w/ response Y or not.

least absolute shrinkage and selection operator.

The *lasso* is an alternative that overcomes this disadvantage. The lasso coefficients $\hat{\beta}_\lambda^L$ minimize

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda \sum_{j=1}^{p}|\beta_j| = RSS + \lambda \underbrace{\sum_{j=1}^{p}|\beta_j|}_{= \|\beta\|_1,\ \ell_1\text{-norm.}}$$

ℓ_1 penalty

(ridge uses ℓ_2 penalty)

// regularization

As with ridge regression, the lasso shrinks the coefficient estimates towards zero.

ℓ_1 penalty also has the effect of forcing some coefficients to be exactly zero when λ is sufficiently large!

$\Rightarrow$ performs variable selection!

As a result, lasso models are generally easier to interpret.

The lasso yields sparse models - models w/ only a subset of the variables.

Again, selecting a good λ is critical. $\Rightarrow$ CV.

variable selection

Why does the lasso result in estimates that are exactly equal to zero but ridge regression does not? One can show that the lasso and ridge regression coefficient estimates solve the following problems

lasso: $\underset{\beta_0,\dots,\beta_p}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 \right\}$ subject to $\sum_{j=1}^{p} |\beta_j| \le s$

ridge: minimize $\left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 \right\}$ subject to $\sum_{j=1}^{p} \beta_j^2 \le s$

constraints.

constrained optimization

$\Leftrightarrow$ equivalent to formulation w/ λ

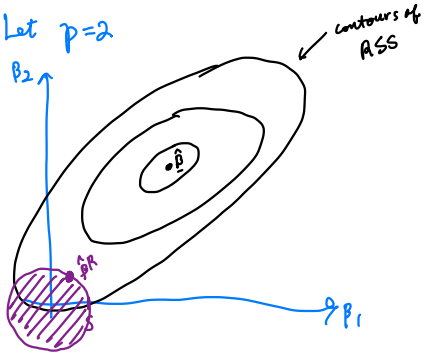
In other words, when we perform the lasso we are trying to find the set of coefficient estimates that lead to the smallest RSS, subject to the contraint that there is a budget s for how large $\sum_{j=1}^{p} |\beta_j|$ can be.

When s is very large, this is not much of a constrain $\Rightarrow$ coef. estimates can be large.

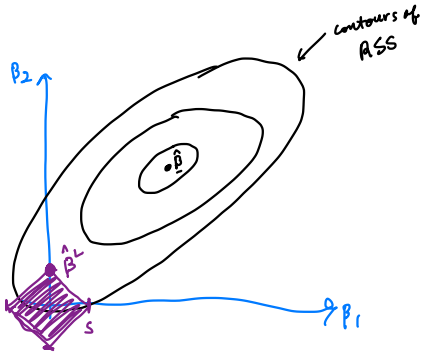
Similar to ridge.

But why does the lasso result in coefficient estimates exactly equal to zero?

Let $p=2$



$\beta_1^2 + \beta_2^2 \le s$



$|\beta_1| + |\beta_2| \le s$

Solution for lasso or ridge is first point where RSS surface contacts constraint region.

Ridge has a circular constraint region $\Rightarrow$ no sharp points $\Rightarrow$ intersection can occur anywhere

Lasso has corners on each axis $\Rightarrow$ RSS surface often have first contact at an axis $\Rightarrow$ one of the coefficients will equal zero!

If we believe there are predictors that do not have a relationship w/ response (we just don't know which), lasso will perform better (bias + variance).

If not (everything is important), ridge will perform better.

2.3 Tuning

We still need a mechanism by which we can determine which of the models under consideration is "best".

For subset C_p, AIC/BIC, adjusted R^2, CV error

For both the lasso and ridge regression, we need to select λ (or the budget s).

How? CV.

regularization parameter.

(0) Scale the "predictors" data to have st. dev. = 1

(1) Choose a grid of λ values.

(2) Compute CV error for each λ (k-fold).

(3) Select λ for which CV error is smallest (or return to step (1)).

(4) Fit model using all training data and selected λ.

Note: still important to scale "predictor" variables for lasso.

3 Dimension Reduction Methods

So far we have controlled variance in two ways:

(1) Using a subset of original variables
- best subset, forward selection, lasso

(2) Shrinkage of coefficients towards zero
- ridge, lasso.

These methods all defined using original predictor variables $X_1, \ldots, X_p$.

We now explore a class of approaches that

(1) transform predictors
(2) then fit least squares using transformed variables.

We refer to these techniques as *dimension reduction* methods.

(1) Let $Z_1, \ldots, Z_M$ represent $M < p$ linear combinations of our original predictors.

$$Z_m = \sum_{j=1}^{p} \phi_{jm} X_j$$

for constants $\phi_{1m}, \ldots, \phi_{pm}$, $m = 1, \ldots, M$.

(2) Fit the linear regression model using least squares

$$y_i = \theta_0 + \sum_{m=1}^{M} \theta_m z_{im} + \varepsilon_i \quad i = 1, \ldots, n$$

($\theta_0, \ldots, \theta_M$: regression coefficients)

If ϕ_{jm} chosen well, this can outperform least squares (on $X_1, \ldots, X_p$).

The term *dimension reduction* comes from the fact that this approach reduces the problem of estimating $p+1$ coefficients to the problem of estimating $M+1$ coefficients where $M<p$.

$\beta_0, \beta_1, \ldots, \beta_p$ $\quad$ $\theta_0, \theta_1, \ldots, \theta_M$

Note:

$$\sum_{m=1}^{M} \theta_m z_{im} = \sum_{m=1}^{M} \theta_m \sum_{j=1}^{p} \phi_{jm} x_{ij} = \sum_{j=1}^{p} \left[\sum_{m=1}^{M} \theta_m \phi_{jm} \right] x_{ij}$$

(def'n of z_{im})

$$= \sum_{j=1}^{p} \beta_j x_{ij}$$

Dimension reduction serves to constrain β_j, since now they must take a particular form.

$$\beta_j = \sum_{m=1}^{M} \theta_m \phi_{jm}$$

$\Rightarrow$ special case of original linear regression model. (with β_j constrained).

- ↳ can bias coefficient estimates
- ↳ if $p > n$ (or $p \approx n$) selecting $M << p$ can reduce variance.

All dimension reduction methods work in two steps.

1. transform predictors
2. fit model using M transformed predictors from (1) (OLS).

The selection of ϕ_{jm}'s can be done in multiple ways. We will talk about 2.

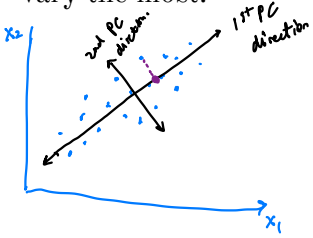
3.1 Principle Component Regression (PCR).

How to choose $Z_1, \dots, Z_M$ (one way).

Principal Components Analysis (PCA) is a popular approach for deriving a low-dimensional set of features from a large set of variables.

PCA is an unsupervised approach for reducing the dimension of an $n \times p$ data matrix X.

The *first principal component* direction of the data is that along which the observations vary the most.



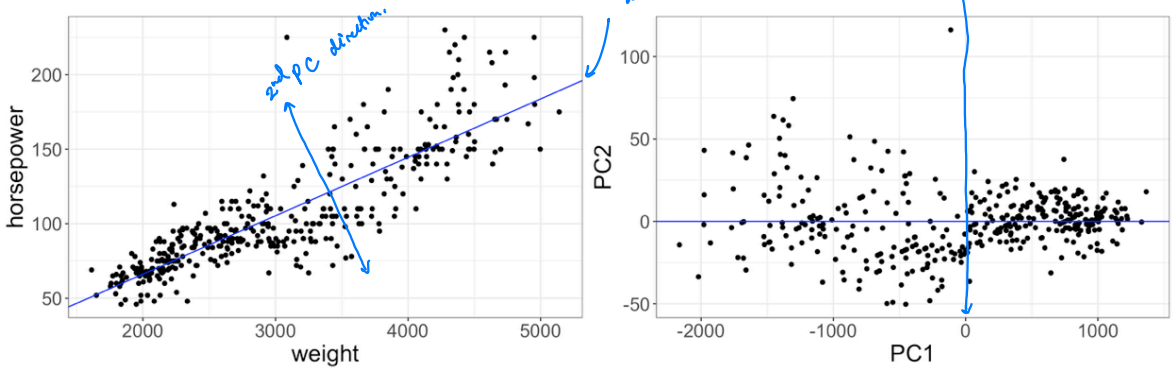
The 1st principal components are obtained by projecting the data onto the 1st PC direction.

a point is projected onto a line by finding the point on the line that is closest to the point.

Out of every possible linear combination of X_1 and X_2 such that $\phi_{11}^2 + \phi_{21}^2 = 1$, choose so that $\text{Var}\left[\phi_{11}(x_1 - \bar{x}_1) + \phi_{21}(x_2 - \bar{x}_2)\right]$ is maximized. $\Rightarrow z_{i1} = \phi_{11}(x_{i1} - \bar{x}_1) + \phi_{21}(x_{i2} - \bar{x}_2)$ for $i = 1, \dots, n$.

↑ are principal component "scores".

We can construct up to p principal components, where the 2nd principal component is a linear combination of the variables that are uncorrelated to the first principal component and has the largest variance subject to this constraint. ⇒ perpendicular to 1st PC direction!



1st PC = direction along which data vary the most.

↳ projected onto PC directions.

The 1st PC contains the most information → p^{th} PC contains the least.

The Principal Components Regression approach (PCR) involves

1. Construct first M principal components $Z_1, \ldots, Z_M$

2. Fit a linear regression model w/ $Z_1, \ldots, Z_M$ as predictors in OLS.

Key idea: Often a small # of PC suffice to explain most of the variability in the data, as well as the relationship w/ the response. (we hope).

In other words, we assume that the directions in which $X_1, \ldots, X_p$ show the most variation are the directions that are associated with Y.

This is not guaranteed to be true, but often works well in practice.

If the assumption holds, PCR will lead to better results than OLS on $X_1, \ldots, X_p$ because we are mitigating overfitting.

How to choose M, the number of components?

M can be thought of as a tuning parameter $\Rightarrow$ use CV to choose!

as $M \uparrow p$, PCR $\longrightarrow$ OLS $\Rightarrow$ $\downarrow$ bias but $\uparrow$ variance, will see the U-shape in the test MSE.

Note: PCR is not feature selection!

each of the M principal components is a linear combination of all p original features!

$\Rightarrow$ while PCR works well to reduce variance, it does not produce a sparse model.

More like ridge than lasso.

* NOTE: * recommend standardize predictors $X_1, \ldots, X_p$ to each have St. dev = 1 before getting PCs.

3.2 Partial Least Squares (PLS).

The PCR approach involved identifying linear combinations (directions) that best represent the predictors $X_1, \ldots, X_p$.

We identified these directions in an unsupervised way (Y not used to find directions).

Consequently, PCR suffers from a drawback

There is no guarantee the directions that best explain the predictors will also be best directions to explain relationship w/ response!

Alternatively, *partial least squares (PLS)* is a supervised version. – still dimension reduction

(1) identify new features $Z_1, \ldots, Z_M$ linear combinations of features

(2) Fit OLS using transformed predictors.

PLS also uses Y (not just X) to find linear combinations of $X_1, \ldots, X_p$ (i.e. find $\phi_{1m}, \ldots, \phi_{pm}$ $m = 1, \ldots, M$).

Roughly speaking, the PLS approach attempts to find directions (linear combinations.) that help explain both the reponse and the predictors.

The first PLS direction is computed,

(1) standardize the p predictors (all have st. dev. = 1).

(2) Set each ϕ_{j1} equal to slope coefficient from simple linear regression $Y \sim X_j$

Since the coefficient from SLR of $Y \sim X_j$ is ("proportional to") $\propto Cor(Y, X_j)$, PLS places highest weight on variables most strongly related to response.

To identify the second PLS direction,

(1) regress each predictor $X_1, \ldots, X_p$ on Z_1 ($X_j \sim Z_1$) and take residuals ($r_{ij} = x_{ij} - \hat{x}_{ij}$, $i = 1, \ldots, n$, $j = 1, \ldots, p$)

(2) Compute Z_2 by setting each ϕ_{j2} equal to the slope coefficient from SLR $Y \sim r_j$ ← residuals from step (1).

The residuals $r_1, \ldots, r_p$ ≈ remaining information not explained by 1st PLS direction.

← can repeat to get M directions.

As with PCR, the number of partial least squares directions is chosen as a tuning parameter. ⇒ CV! M

Generally, standardize the predictors and response before performing PLS.

In practice PLS usually performs no better than ridge or PCR.

↳ Supervised nature of problem does reduce bias, but also often increases variance ⇒ not always better.

4 Considerations in High Dimensions

Most traditional statistical techniques for regression and classification are intended for the low-dimensional setting. $n \gg p$

This is because throughout the history of the field, the bulk of scientific problems requiring statistics have been low dimensional.

In the past 25 years, new technologies have changed the way that data are collected in many fields. It is now commonplace to collect an almost unlimited number of feature measurements. p very large.

But n can still be small due to cost, sample availability, etc..

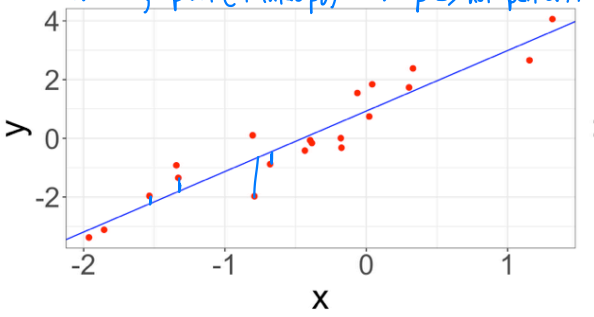
e.g. consider predicting crop yield but now you can sequence the genome of the corn species you are planting.

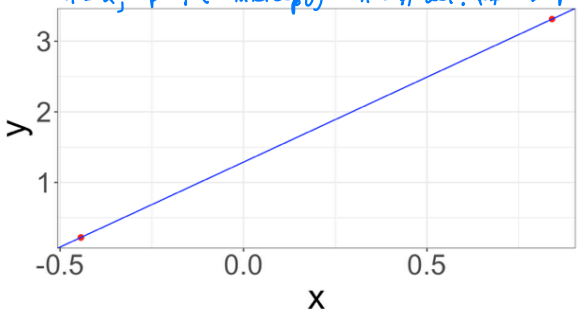
Data sets containing more features than observations are often referred to as *high-dimensional*.

classical approaches (like OLS) are not appropriate in this setting.
Why? bias-variance trade-off $\Rightarrow$ overfitting!
$\Rightarrow$ we need to be extra careful when $n \approx p$ or $n > p$.

What can go wrong in high dimensions? going to talk about least squares, but same issues arise for logistic regression, LDA, etc.

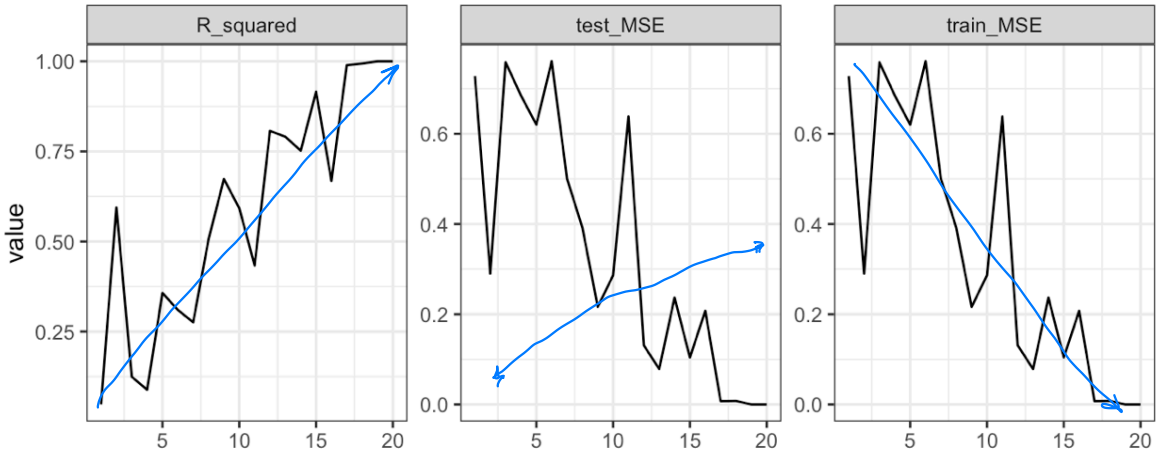
If p is as large or larger than n, regardless of if there is a relationship btw Y and $\underline{X}$, OLS will yield a set of coefficients that result in a perfect fit to training data $\Rightarrow$ residuals $= 0$.





the resulting model will most likely perform poorly on test data!

Simulated data w/ $n = 20$ and performed regression w/ between 1 and 20 features. features were generated w/ NO relationship to response!



increases even though no relationship.

This plot is wrong.

training MSE decreases even though no relationship.

test MSE never good because not a good predictive fit.

Note: we saw ways to adjust training RSS to better reflect test RSS: C_p, BIC/AIC, adjusted R^2. in high dimensional setting ($p \geq n$), we cannot compute these.

$\Rightarrow$ We must be very careful when analyzing data w/ many predictors.

Always consider performance on independent test data set.

Many of the methods that we've seen for fitting *less flexible* models work well in the high-dimension setting.

Key points:

1. regularization or shrinkage important in high-dimensional problems.

2. tuning parameter selection is critical for good predictive performance.

3. The test error tends to increase as $p\uparrow$, unless the additional features are actually associated w/ response.
   ↳ due to curse of dimensionality.

   adding additional signal features can improve model performance.

When we perform the lasso, ridge regression, or other regression procedures in the high-dimensional setting, we must be careful how we report our results.

In the high dimensional setting, it is more likely that predictors will be correlated.

⇒ some variables could be written as linear combinations of other variables.

This means we can never really know if any are truly predictive of the response.

⇒ we can never identify the best variables to include

at best, we can only hope to assign large regression coefficients to variable that are highly correlated to variables that are truly predictive of the response.

⁂ ⇒ when we use lasso/feature selection, we should be clear we have identified one of many possible models for predicting the response.

should be validated w/ independent replication.

⁂ also important to report test errors not training errors / R^2.