

Model selection



Overview

Quick review of interaction effects and multicollinearity

Model selection

- Overfitting
- Statistics and methods useful for model selection

Review: Interaction terms

An ***interaction effect*** occurs when the response variable y is influenced by the levels of two or more predictors in a non-additive way

We can model this using an equation with an interaction term

$$y = \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_3 (x_1 \cdot x_2) + \epsilon$$

An interaction term between a quantitative and categorical variable corresponds to different slopes depending for the quantitative variable depending on the value of the categorical variable

Review: Interaction terms

If Full Professor:

$$\text{salary} \approx \beta_0 + \beta_1 \cdot \text{endowment}$$

If Assistant Professor:

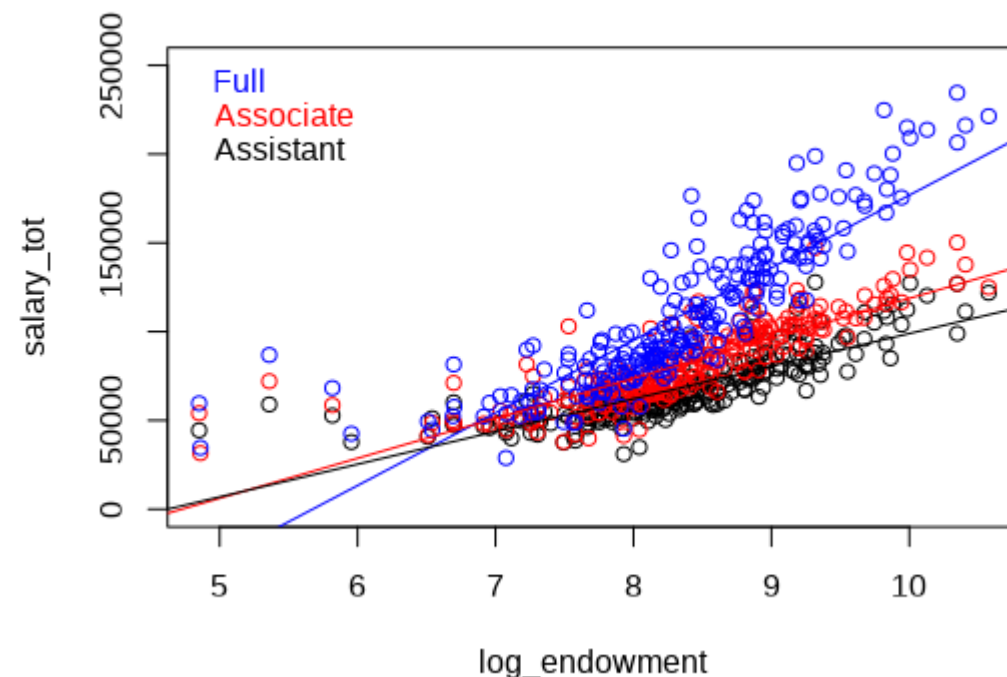
$$\text{salary} \approx (\beta_0 + \beta_2) + (\beta_1 + \beta_3) \cdot \text{endowment}$$

Modification to intercept if Assistant Professor

Modification to slope if Assistant Professor

$$x_{i2} = \begin{cases} 1 & \text{if assistant professor} \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \hat{\beta}_3 x_{i1} \cdot x_{i2}$$



Review: Interaction terms

```
Call:
lm(formula = salary_tot ~ log_endowment + rank_name + log_endowment:rank_name,
    data = IPED_2)
```

Residuals:

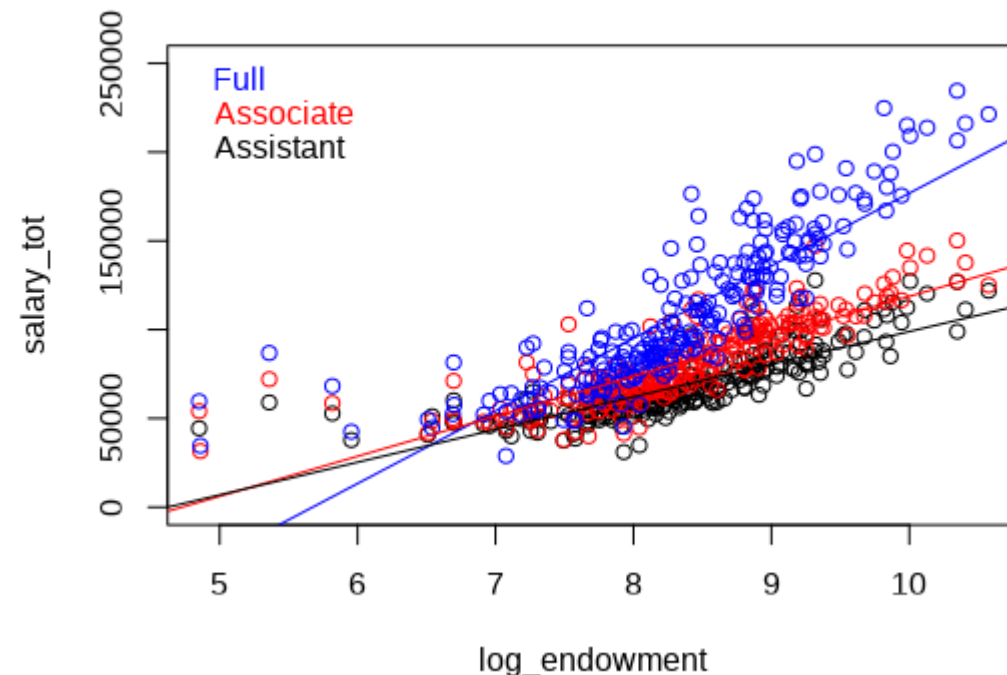
Min	1Q	Median	3Q	Max
-46914	-9554	-2263	6233	99678

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-231986	9989	-23.224	<2e-16 ***
log_endowment	40888	1190	34.357	<2e-16 ***
rank_nameAssociate	125551	14289	8.786	<2e-16 ***
rank_nameAssistant	146880	14429	10.180	<2e-16 ***
log_endowment:rank_nameAssociate	-18369	1701	-10.800	<2e-16 ***
log_endowment:rank_nameAssistant	-22482	1717	-13.094	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16260 on 705 degrees of freedom
 Multiple R-squared: 0.7806, Adjusted R-squared: 0.7791
 F-statistic: 501.7 on 5 and 705 DF, p-value: < 2.2e-16



Intercept for full professor

Slope for full professor

Modification to intercept for assistant prof

Modification to slope for assistant prof

x_{i1} : Log endowment (continuous)

x_{i2} : Assistant prof (indicator/dummy variable)

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \hat{\beta}_3 x_{i1} \cdot x_{i2}$$

Review: Multicollinearity

Multicollinearity occurs when two or more variables are closely related to each other

- e.g., if they have a high correlation

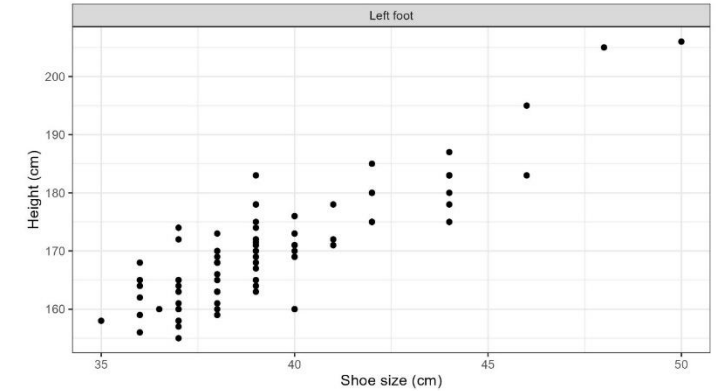
Multicollinearity can make our estimate of the regression coefficients unstable

- e.g., standard error of coefficients become large

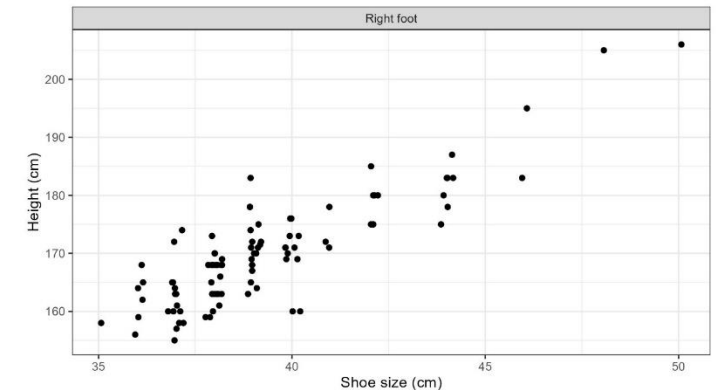
The **variance inflated factor** is a statistic that can be computed to test for multicollinearity

- Rule of thumb: suspect multicollinearity for $VIF > 5$

Left foot



Right foot



`car::vif(lm_fit)`

Polynomial regression

Polynomial regression

Polynomial regression extends linear regression to non-linear relationships by including nonlinear transformations of predictors

$$\begin{aligned}\text{salary} = & \beta_0 + \beta_1 \cdot \text{endowment} \\ & + \beta_2 \cdot (\text{endowment})^2 + \\ & + \beta_3 \cdot (\text{endowment})^3 + \varepsilon\end{aligned}$$

Still a linear equation but non-linear in original predictors

Polynomial regression

Polynomial regression extends linear regression to non-linear relationships by including nonlinear transformations of predictors

We can compare model fits by:

- Assessing if the coefficients on the higher order terms are statistically significant
- Looking at the R^2 values
- Running hypothesis tests comparing nested models
- Etc.

Let's try it in R...



Living life on the edge

[Tweet übersetzen](#)



Model selection

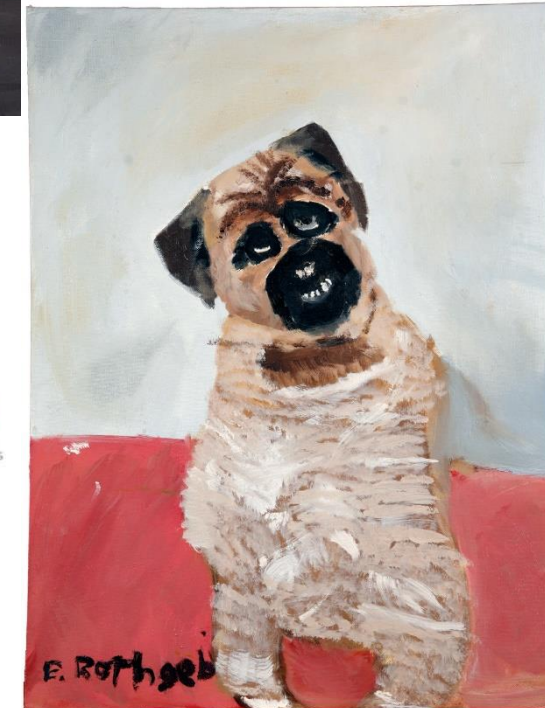
Model selection

Model selection is the process of selecting a statistical model from a set of candidate models

- E.g., which explanatory variables, interaction terms, transformations of variables, etc. to include in a final model

Model selection is a bit of an art

- “All models are wrong but some are useful”
 - But there is definitely some bad art out there



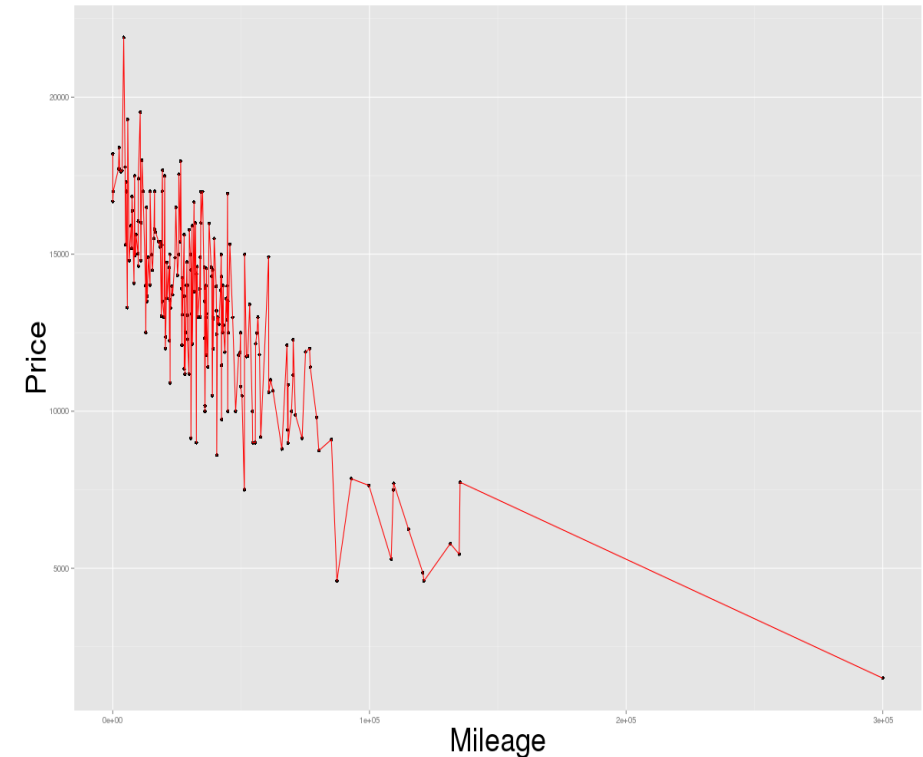
Model selection

Model selection depends on our goal, which usually is either:

- Making accurate predictions
- Understanding relationships in our data

If we fit our model too closely to the data sample we have, we are likely to fail in both of these goals

- i.e., it will be hard to understand relationships between explanatory and response variables
- and model will not make good predictions on new data



Model selection

The first year I taught 230 I had students use the Edmunds data to create a multiple regression models that could explain as much of the variability in a car's sale price as possible.

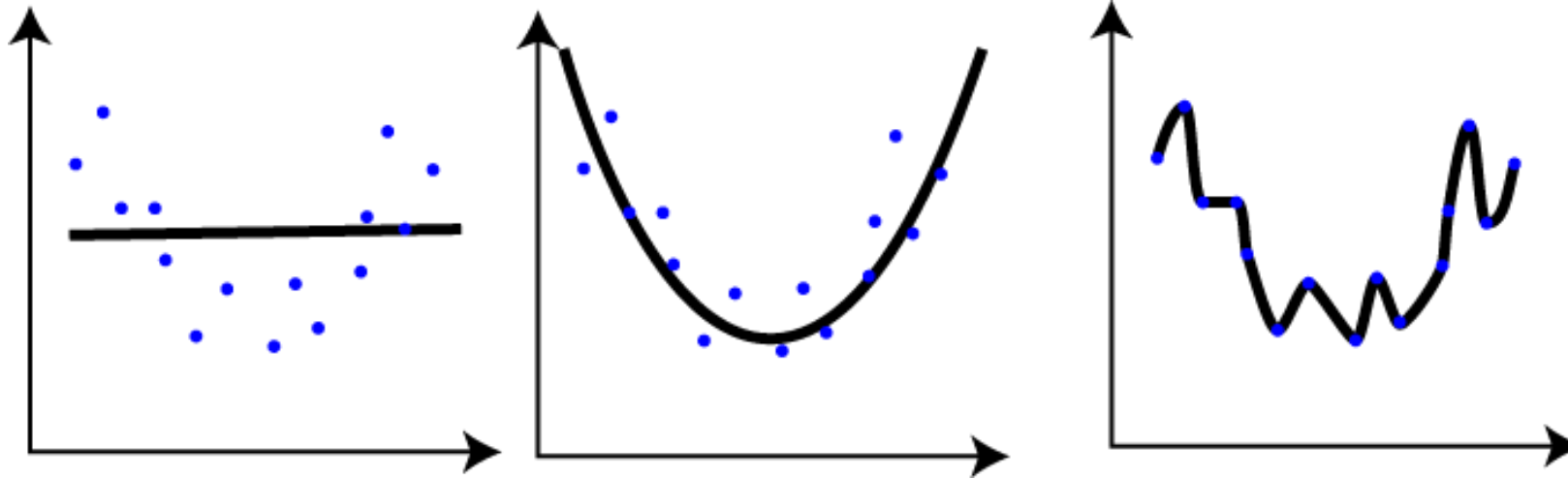
One student came up with a model had $R^2 = 0.9434$

$$\begin{aligned}\hat{y} = & 24919.66 + 101976420913.07MS - 8801176308.15MS^2 + 1241642151.11MS^3 + -294845652.56MS^4 + \\ & 81081110.14MS^5 - 19486206.21MS^6 + 4627457.40MS^7 - 962171.87MS^8 + 159411.19MS^9 + 4758.12MS^{10} - \\ & 215919.33YO + 61629.11YO^2 - 46883.91YO^3 + 27616.00YO^4 + -41.24YO^5 + 986604.79MB - \\ & 35074977.23MB^2 - 65070922.16MB^3 - 80975553.98MB^4 - 39704725.17MB^5 - 13029640.26MB^6 - \\ & 4181962.33MB^7 - 1234301.18MB^8 - 444859.17MB^9 - 130775.72MB^{10} + 10404.48MPY - 3884.70MPY^2 + \\ & 2328.80MPY^3 + 13563.89MPY^4 - 5277.89MPY^5 - 15338.18MPY^6 + 12851.91MPY^7 + 4899.51MPY^8 + \\ & 383.03MPY^9 - 68.52MPY^{10} - 82637866000.10(\log(MS)) - 41355241329.94(\log(MS))^2 - 32425545929.48(\log(MS))^3 - \\ & 24481236288.29(\log(MS))^4 - 14974293674.01(\log(MS))^5 - 7624035557.33(\log(MS))^6 - 2264871890.73\log(MS)^7 - \\ & 411050984.04\log(MS)^8 - 54557306.69\log(MS)^9 - 2606070.61\log(MS)^{10} + 172377.75(YO * MS) - 85365.06(YO * \\ & MS)^2 + 20333.05(YO * MS)^3 + 38388.69(YO * MS)^4 - 11361.46(YO * MS)^5 + 37428.48(YO * MS)^6 + \\ & 5120.32(YO * MS)^7 + 15285.89(YO * MS)^8 + 9300.88(YO * MS)^9 + 11552.40(YO * MS)^{10} + 35723937.44w^2 + \\ & 66215674.21w^3 + 80347987.00w^4 + 38445702.35w^5 + 12868623.19w^6 + 4105787.54w^7 + 1161535.74w^8 + \\ & 426310.43w^9 + 133553.84w^{10}\end{aligned}$$

Do you think this model would do well making predictions on new data?

Overfitting

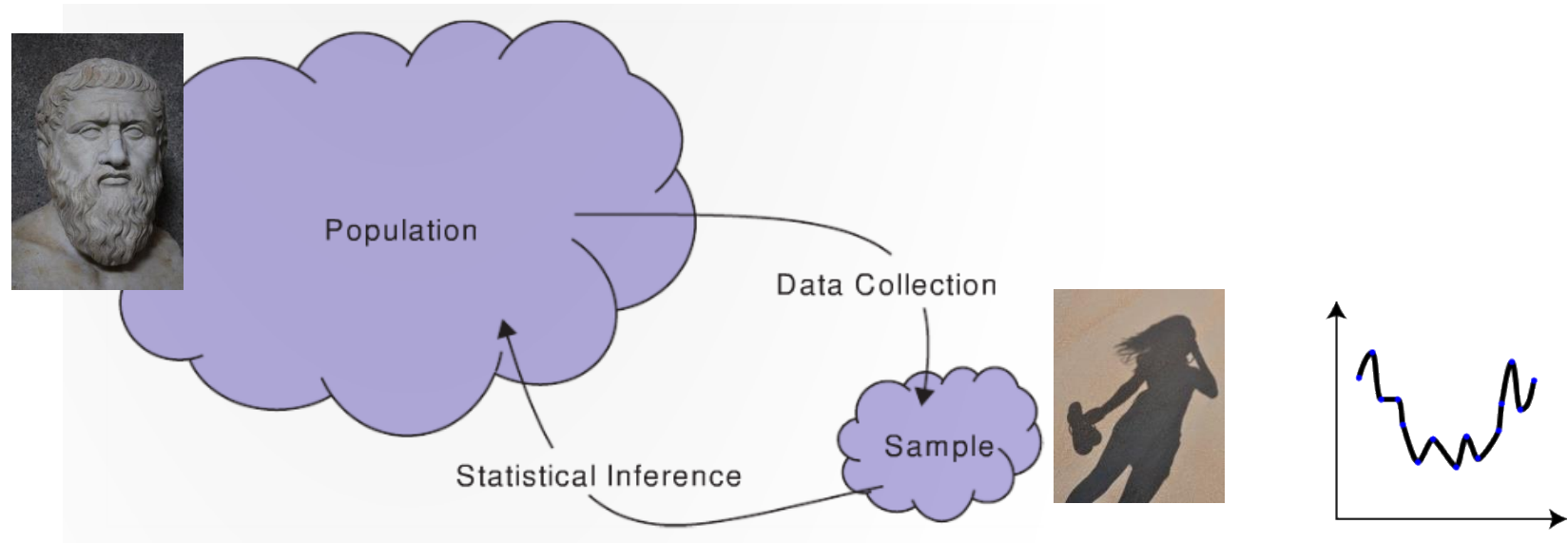
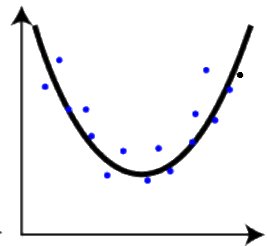
Overfitting occurs when we generate a function that too closely matches random sample we have, but does not generalize to the full probability distribution



Overfitting

Overfitting occurs when we generate a function that too closely matches random sample we have, but does not generalize to the full probability distribution

- The model is fit too closely to the shadows and not getting at the Truth



Overfitting song



<https://www.youtube.com/watch?v=DQWI1kvmwRg>

Selecting models methods

There are a number of different methods for selecting models. Four we will briefly discuss are:

1. Creating measures of fit (statistics) that penalize models with more predictors
2. Creating simpler models by removing predictors
3. Evaluating models using cross-validation
4. If there is time: methods that shrink regression coefficients

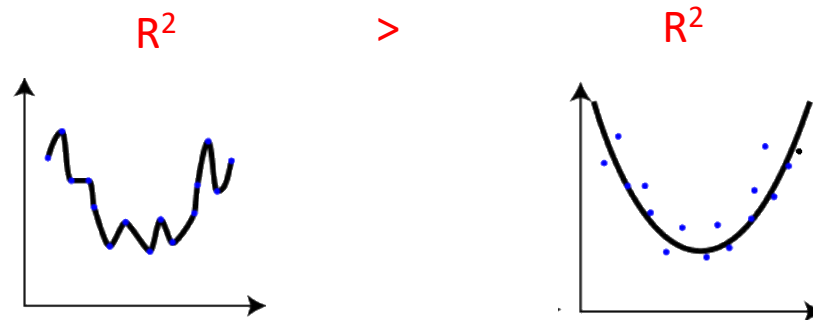
Model method selection 1: Selected models using statistics that penalize larger models

R^2 as a measure of model fit

We have used the coefficient of multiple determination (R^2) to determine how well our model is fitting the data:

$$R^2 = \frac{SS_{Model}}{SS_{Total}} = 1 - \frac{SS_{Residual}}{SS_{Total}} = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

R^2 always increases with more predictors x_i because the response variable y can always fit more closely with more predictors



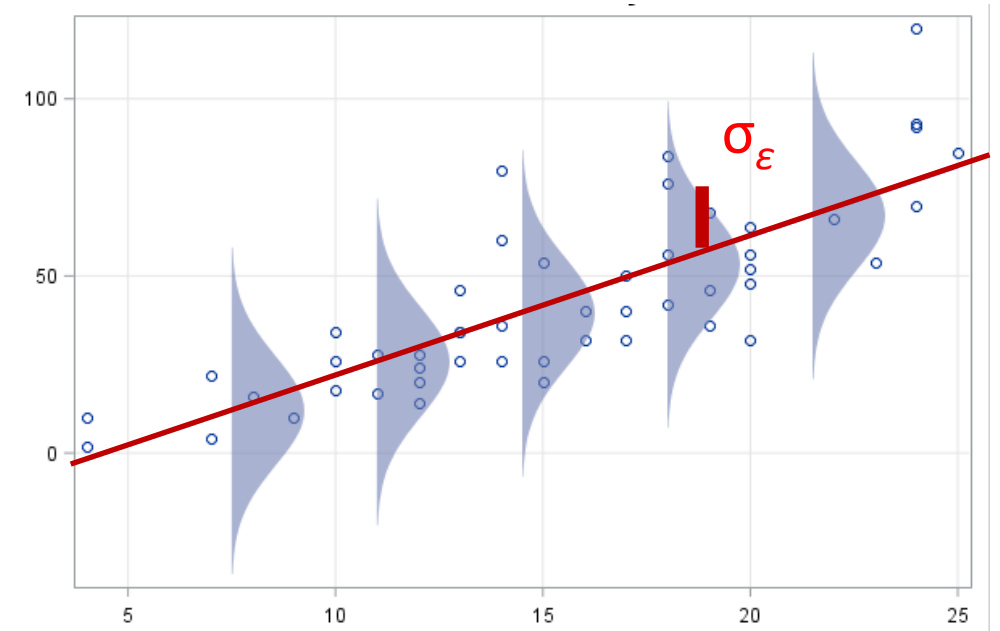
Recall: the standard deviation of the errors: σ_ε

Recall for simple linear regression, the standard deviation of the errors is denoted σ_ε and shows how far the points fall off the true regression line.

We can use the **standard deviation of residuals** ($\hat{\sigma}_e$) as an estimate for σ_ε

For simple linear regression we had:

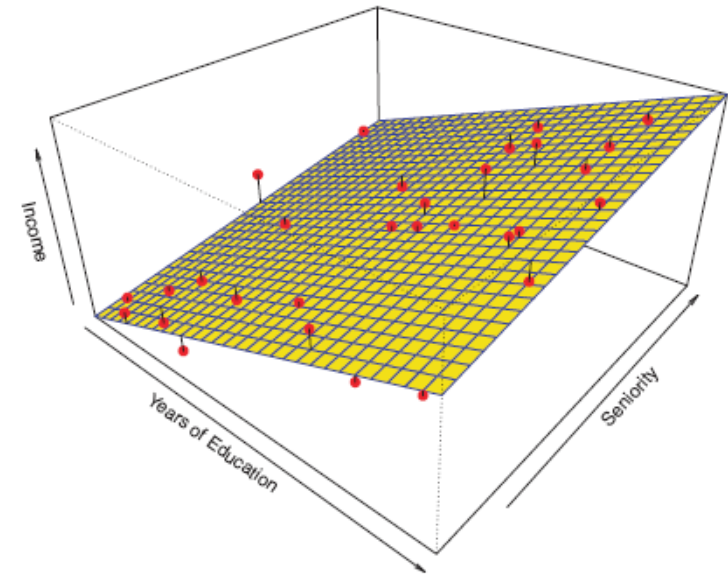
$$\begin{aligned}\hat{\sigma}_e &= \sqrt{\frac{1}{n-2} SSRes} \\ &= \sqrt{\frac{1}{n-2} \sum_{i=1}^n (y_i - \hat{y}_i)^2}\end{aligned}$$



Recall: the standard deviation of the errors: σ_ε

For multiple regression, we with k parameters (i.e., $k - 1$ predictors) an (almost) unbiased estimate of $\hat{\sigma}_\varepsilon$ is:

$$\begin{aligned}\hat{\sigma}_e &= \sqrt{\frac{1}{n-k} SSRes} \\ &= \sqrt{\frac{1}{n-k} \sum_{i=1}^n (y_i - \hat{y}_i)^2}\end{aligned}$$



The residual standard deviation $\hat{\sigma}_\varepsilon$ corrects for bias by dividing the SSResidual by $1/(n - k)$
This estimate does not always decrease with more predictors x

Adjusted R^2

The **adjusted R^2** helps account for the number of predictors in the model by using $\hat{\sigma}_\epsilon^2$

$$R_{adj}^2 = 1 - \frac{SSRes / (n - k)}{SSTotal / n - 1} = 1 - \frac{\hat{\sigma}_e^2}{s_y^2}$$

The adjusted R^2 does not always give a higher values to the model with the more predictors

- i.e., using this statistic, we will not always say that a model with the most predictors is a “better” fit to the data

Other statistics that penalize larger models

There are several other statistics that also ***penalize models that have more predictors***

- These statistics are only meaningful for within data set comparisons

Akaike information criterion: $AIC \propto 2 \cdot k + n \cdot \ln(SSRes/n)$ R: `AIC(lm_fit)`

Bayesian information criterion: $BIC \propto k \cdot \ln(n) + n \cdot \ln(SSRes/n)$ R: `BIC(lm_fit)`

One should select the model with the lowest value on these statistics

Let's try it in R...

Model method selection 2: Using algorithms
to select a subset of variables

Brief mention: Variable selection

Variable selection refers to finding models that rely on a small subset of predictors

- This can help make the regression model more interpretable as well

We could use individual feature p-values to determine which predictors to use, however...

- Some of these will be spuriously significant
 - i.e., if H_0 is true for all predictors, ~5 will be significant at $\alpha = .05$ level
- The p-values change as predictors are added and removed
 - Due to multicollinearity

```
lm_fit_mult <-  
  lm(log(endowment) ~  
    salary_tot,  
    salary_men,  
    salary_women  
  )
```

Feature selection: deciding which variables to use

Ideally we would like to try all combinations of predictors, however, if there are k features, there are 2^k possible models which can be intractable

A few heuristic methods exist for selecting smaller models

- Forward selection: start with a model with no predictors and add predictors (until you have enough)
- Backward selection: Start with the full model and delete predictors
- Mixed selection: Use a combination of forward and backward selection

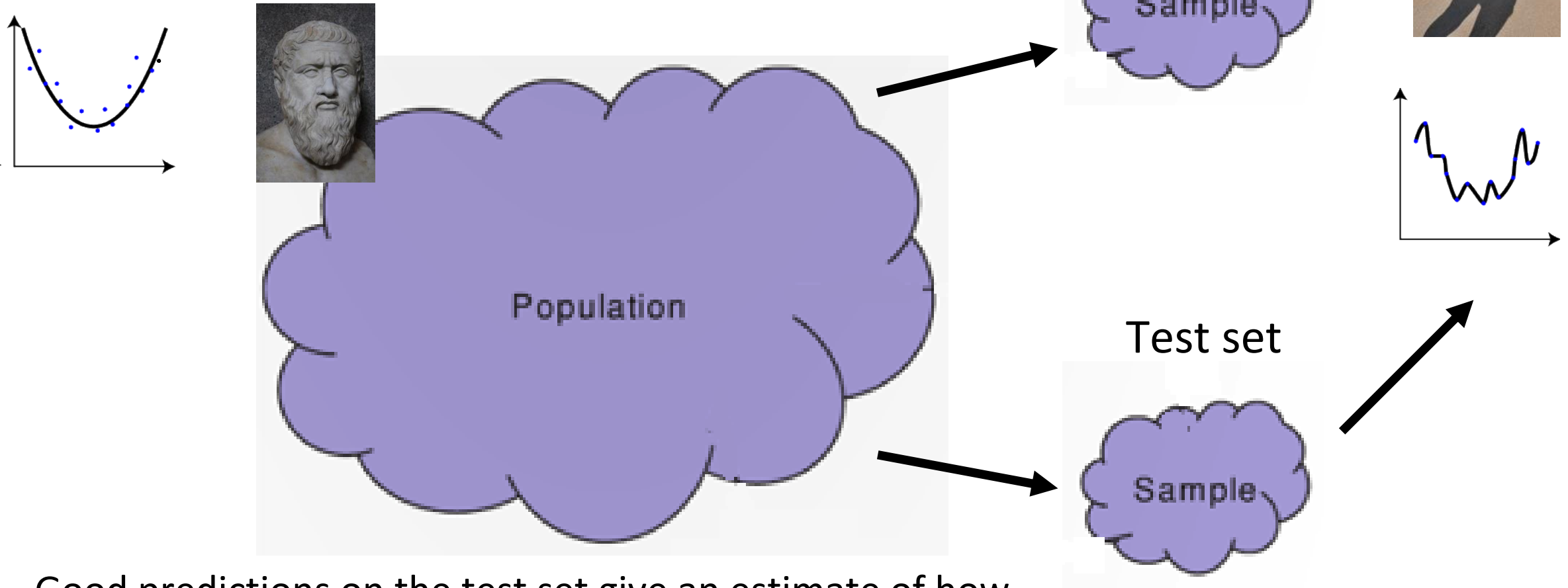
Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	12.85948900	0.17913330	71.787	< 0.00000000000000002 ***
salary_tot	0.00013686	0.00003950	3.465	0.000551 ***
salary_men	-0.00001569	0.00002019	-0.777	0.437160
salary_women	-0.00004304	0.00002156	-1.997	0.046099 *

In R: `leaps::regsubsets()`

Model method selection 3: Choosing a model through cross-validation

Cross-validation



Good predictions on the test set give an estimate of how accurate the model will be on new data from the population

Cross-validation

We run cross-validation by splitting data into two sets:

A training set in which the parameters of a regression model are fit (estimated)

A test set in which the prediction accuracy of our model is assessed



Mean squared prediction error

To evaluate how effective a model is, we can use the mean squared prediction error (MSPE) using the following steps:

1. Fit a model using the training data
2. Make predictions on the test data
3. We can use the MSPE to assess how accurate the predictions are:

$$MSPE = \frac{1}{n_t} \sum_{i=1}^{n_t} (y_i - \hat{y}_i)^2 = \frac{1}{n_t} \sum_{i=1}^{n_t} (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_1 + \dots + \hat{\beta}_k x_k))^2$$

Actual y values in the **test set**

Predicted y values on the **test set**

Parameters estimated on the training set

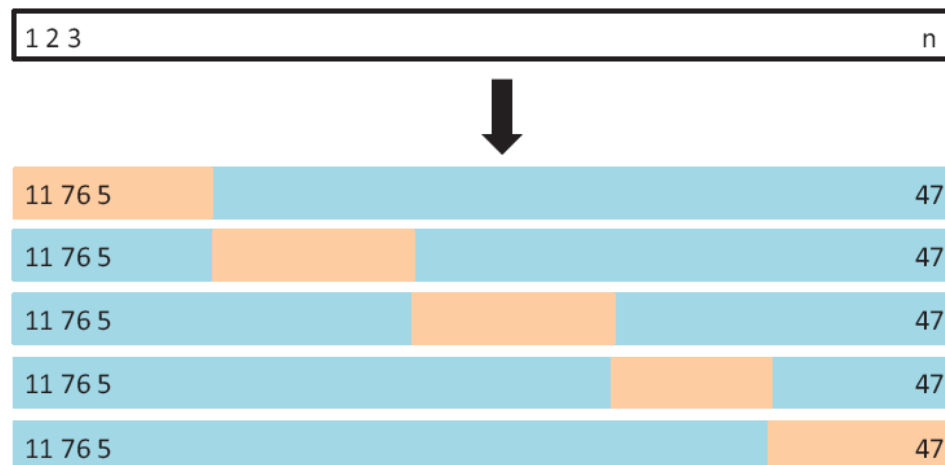
Predictions assessed on the test set

n_t is the number of points in the test set

K-fold cross-validation

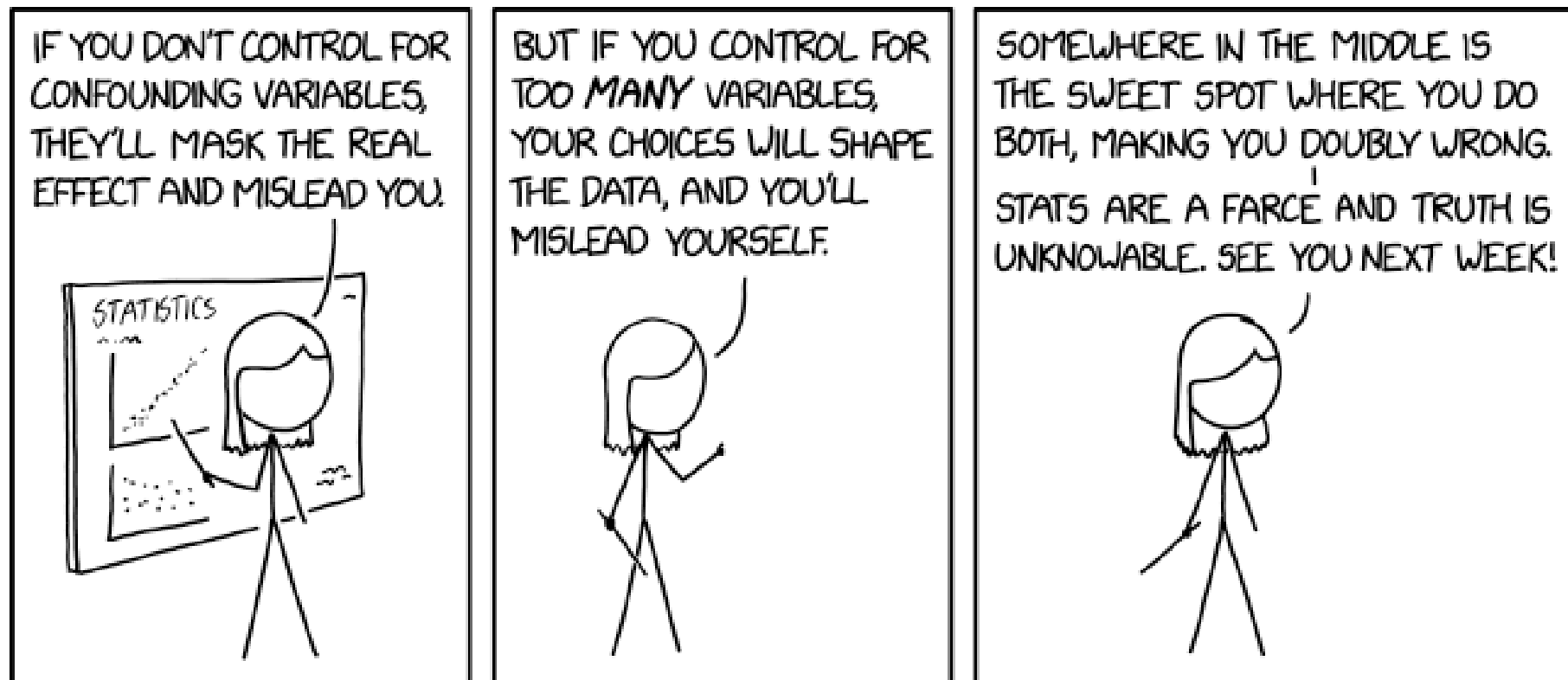
K-fold cross-validation

- Split the data into k parts
- Train on $k-1$ of these parts and test on the left out part
- Repeat this process for all k parts
- Average the prediction accuracies to get a final estimate of the generalization error



Leave-one-out (LOO)
cross-validation: $k = n$

Let's try it in R...



Side note: controlling model complexity with regularization

Brief mention: shrinkage methods

Rather than finding the coefficients $\hat{\beta}_i$ by just minimizing the SSRes, one can also add penalties in the fitting procedure to find simpler models



We will very briefly discuss two techniques:

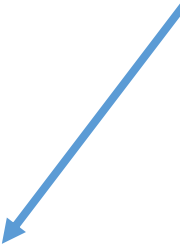
- Ridge regression (l_2 norm penalty)
- The lasso (l_1 norm penalty)

Brief mention: Ridge regression

Ridge regression finds the coefficients $\hat{\beta}_i$ that minimize:

$$\sum_{i=1}^n (y_i - \underbrace{\hat{\beta}_0 - \sum_{j=1}^{k-1} \hat{\beta}_j x_{ij}}_{\text{SSRes}})^2 + \lambda \underbrace{\sum_{j=1}^{k-1} \hat{\beta}_j^2}_{\text{shrinkage penalty}}$$

Tuning parameter



What happens if:

- $\lambda = 0$
- $\lambda \rightarrow \text{infinity}$
- (the coefficients depend on the tuning parameter value)

see the glmnet package

Brief mention: The Lasso

The Lasso finds the coefficients $\hat{\beta}_i$ that minimize:

$$\sum_{i=1}^n (y_i - \underbrace{\hat{\beta}_0 - \sum_{j=1}^{k-1} \hat{\beta}_j x_{ij}}_{\text{SSRes}})^2 + \lambda \underbrace{\sum_{j=1}^{k-1} |\hat{\beta}_j|}_{\text{shrinkage penalty}}$$

Tuning parameter

Similar to ridge regression but penalizes $|\hat{\beta}_i|$ instead of $\hat{\beta}_i^2$

- i.e., uses the L_1 penalty instead of the L_2 penalty

Advantages

- Final model will often have many set $\hat{\beta}_i$ to 0
- i.e., does variable selection and creates a 'sparse' model

see the glmnet package

Shrinkage/regularization methods

Regularization methods often work very well when we care about making accurate predictions on new data

Theory suggests that these methods work by minimizing the MSPE through a bias-variance tradeoff

- Average MSPE = $\text{bias}^2 + \text{variance}$
- We use a biased method (via regularization) to get models that vary less from one random data set to the next, reducing the average MSPE

To learn more about regularization methods, take a class on Machine Learning!