

ETC3250/5250: Introduction to Machine Learning

Resampling for model development and choice

Lecturer: *Professor Di Cook*

Department of Econometrics and Business Statistics

✉ ETC3250.Clayton-x@monash.edu

CALENDAR
Week 3b



Model development and choice

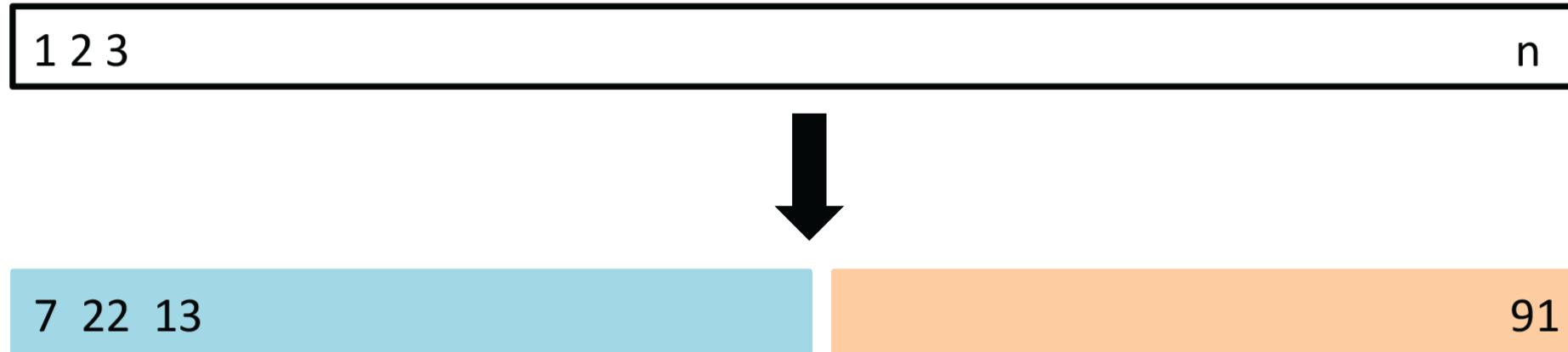


How do you get new data?

Resampling

- **Training/test split:** make one split of your data, keeping one purely for assessing future performance.
- **Leave-one-out:** make n splits, fitting multiple models and using left-out observation for assessing variability.
- **k -fold:** break data into k subsets, fitting multiple models with one group left out each time.
- **Bootstrap:** make many samples, with replacement, using out-of-bag observations for testing.

Training and test sets



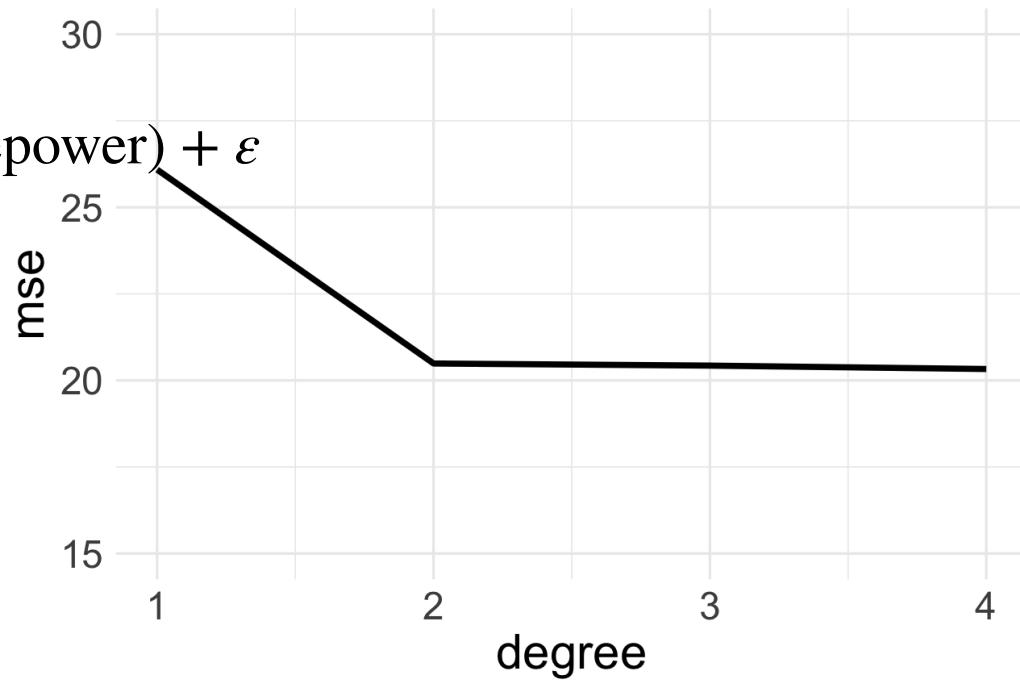
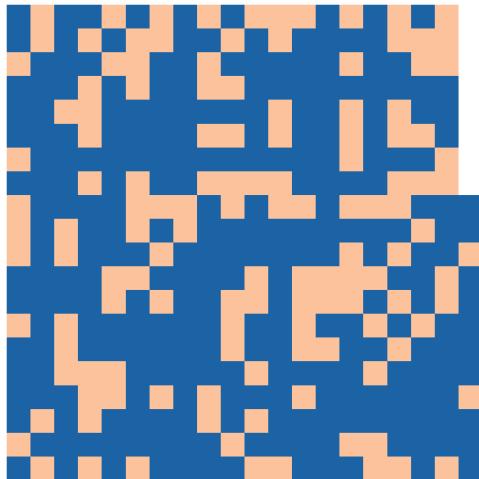
A set of n observations are randomly split into a training set (blue, containing observations 7, 22, 13, ...) and a test set (yellow, all other observations not in training set).

Drawback: Only one split of data made, may have a lucky or unlucky split, accurately estimating test error relies on the one sample.

Training/test set split and choosing polynomial degree (1/4)

$$\text{mpg} = \beta_0 + \beta_1 f(\text{horsepower}) + \varepsilon$$

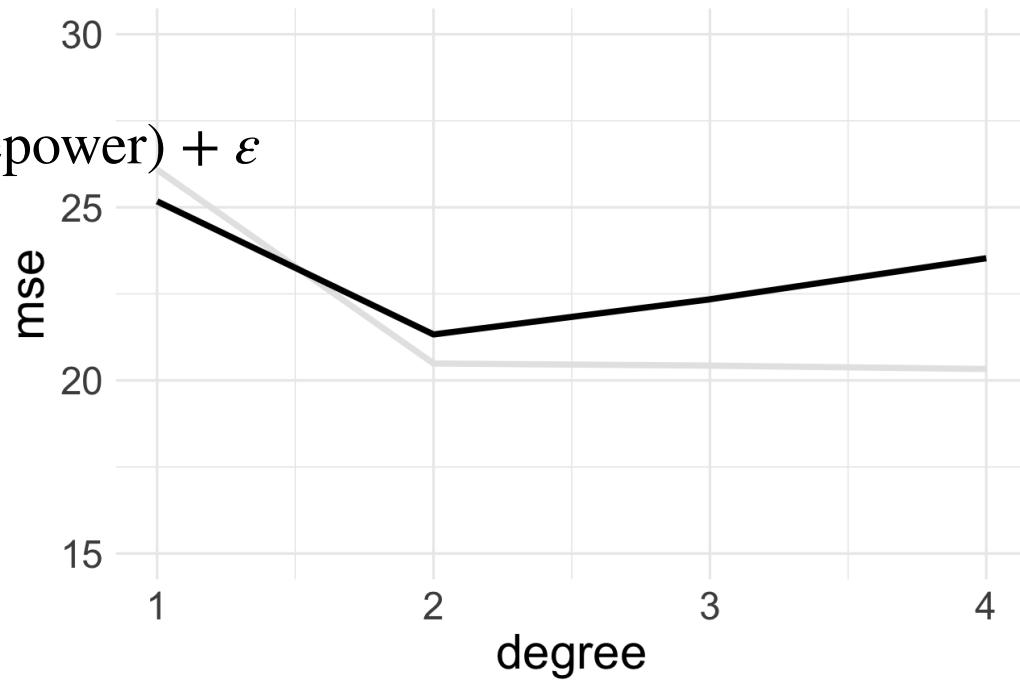
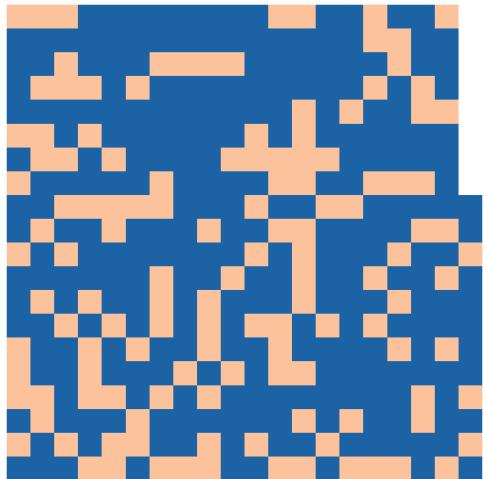
Split into 2/3 training and 1/3 test sets.



Training/test set split and choosing polynomial degree (2/4)

$$\text{mpg} = \beta_0 + \beta_1 f(\text{horsepower}) + \varepsilon$$

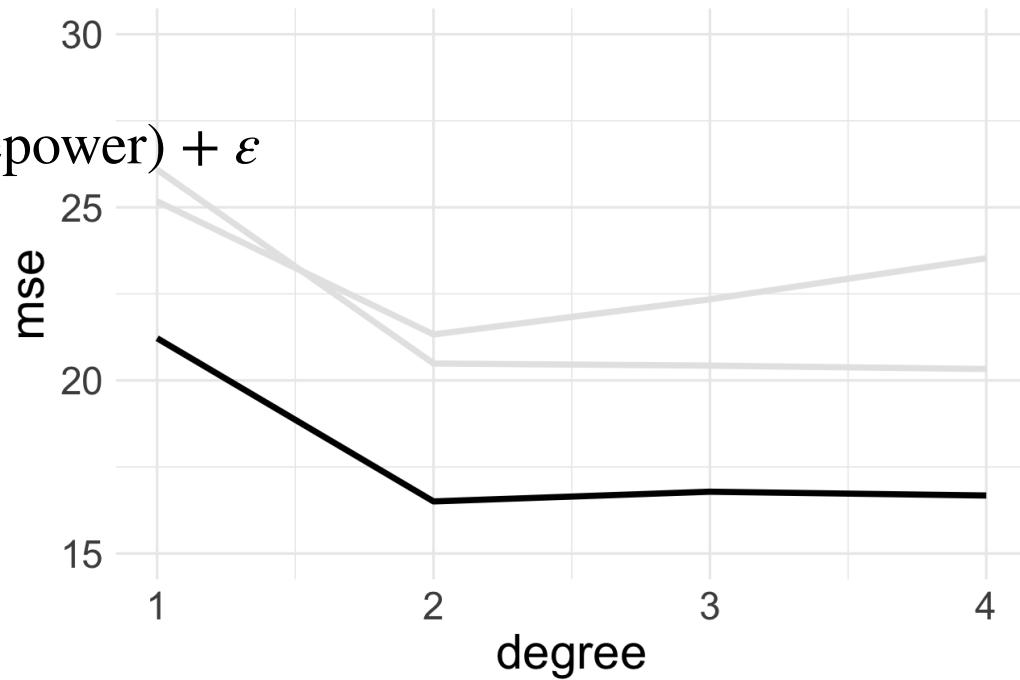
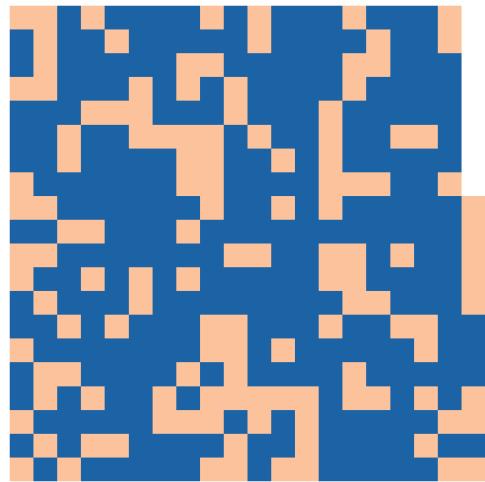
Split into 2/3 training and 1/3 test sets.



Training/test set split and choosing polynomial degree (3/4)

$$\text{mpg} = \beta_0 + \beta_1 f(\text{horsepower}) + \varepsilon$$

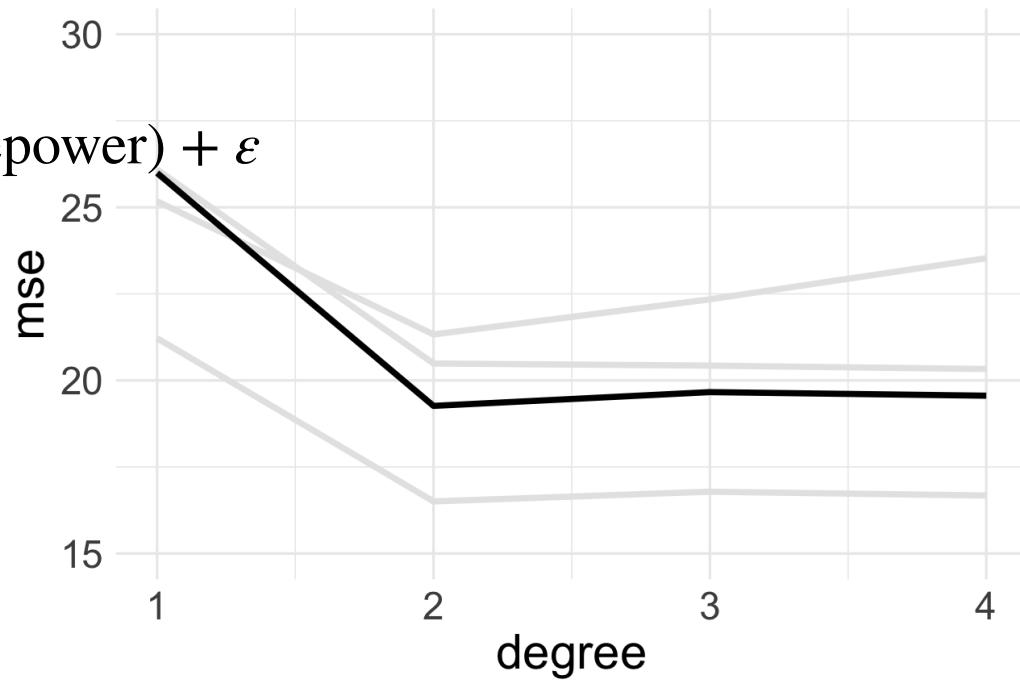
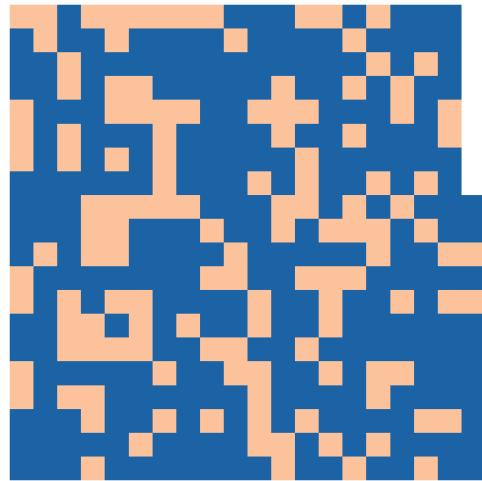
Split into 2/3 training and 1/3 test sets.



Training/test set split and choosing polynomial degree (4/4)

$$\text{mpg} = \beta_0 + \beta_1 f(\text{horsepower}) + \varepsilon$$

Split into 2/3 training and 1/3 test sets.



One split may not be enough

Test MSE changes if a different split is made. For this example, the choice would be degree = 2, though, regardless of the split.



If the **variability** between different draws of test sets is **large**, it indicates a model with high variance. This would be a poor choice of model, or provide a poor error estimate.

LOOCV

Leave-one-out (LOOCV) cross-validation: n test sets, each with **ONE** observation left out.



LOOCV

Leave-one-out (LOOCV) cross-validation: n test sets, each with **ONE** observation left out. For each set, i compute the MSE_i

The LOOCV estimate for the test MSE is the average of these n test error estimates:

$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^n MSE_i$$

There is a computational shortcut, for linear or polynomial models, where not all n models need to be fitted.

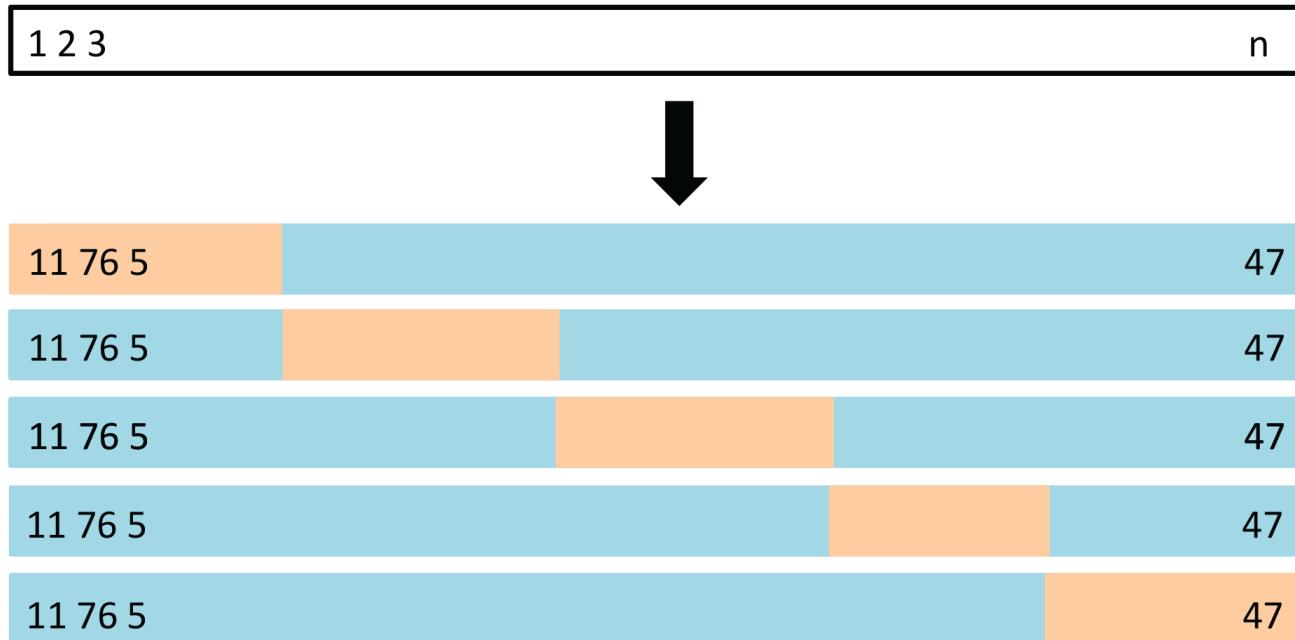
$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^n \frac{(y_i - \hat{y})^2}{1 - h_i}$$

where h_i (known as leverage from regression diagnostics).

LOOCV is a special case of k-fold cross-validation

k-fold cross validation

1. Divide the data set into k different parts.
2. Remove one part, fit the model on the remaining $k-1$ parts, and compute the MSE on the omitted part.
3. Repeat k times taking out a different part each time



(Chapter 5/ 5.5)

k-fold cross validation

1. Divide the data set into k different parts.
2. Remove one part, fit the model on the remaining $k-1$ parts, and compute the MSE on the omitted part.
3. Repeat k times taking out a different part each time

Cross-validation MSE is:

$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^n MSE_k$$

No shortcut for calculation.



Choice of $k=5-10$ is typically reasonable

Try it

```
set.seed(2021)
auto_folds <- vfold_cv(data = Auto)
auto_folds

## # 10-fold cross-validation
## # A tibble: 10 × 2
##   splits          id
##   <list>        <chr>
## 1 <split [352/40]> Fold01
## 2 <split [352/40]> Fold02
## 3 <split [353/39]> Fold03
## 4 <split [353/39]> Fold04
## 5 <split [353/39]> Fold05
## 6 <split [353/39]> Fold06
## 7 <split [353/39]> Fold07
## 8 <split [353/39]> Fold08
## 9 <split [353/39]> Fold09
```

Data is split into 10 subsets

Next, write a function for the four polynomial models:

1. Fit the model to `training(split)`
-> `analysis(split)`
2. Assess the model fit with
`testing(split)` ->
`assessment(split)`
3. Report the MSE

Recommended reading: Alison Hill's
[Take a Sad Script & Make it Better: Tidymodels Edition](#)

```
# Model specification
lm_mod <-
  linear_reg() %>%
  set_engine("lm")

# We want to fit the four polynomial models
compute_fold_mse <- function(split){
  auto_train <- analysis(split)
  auto_test <- assessment(split)

  # Set up polynomials
  auto_rec1 <-
    recipe(mpg ~ horsepower,
           data = auto_train) %>%
    step_poly(horsepower, degree = 1)
  auto_rec2 <-
    recipe(mpg ~ horsepower,
```

Check the calculation for one fold

```
compute_fold_mse(auto_folds$splits[[1]])
```

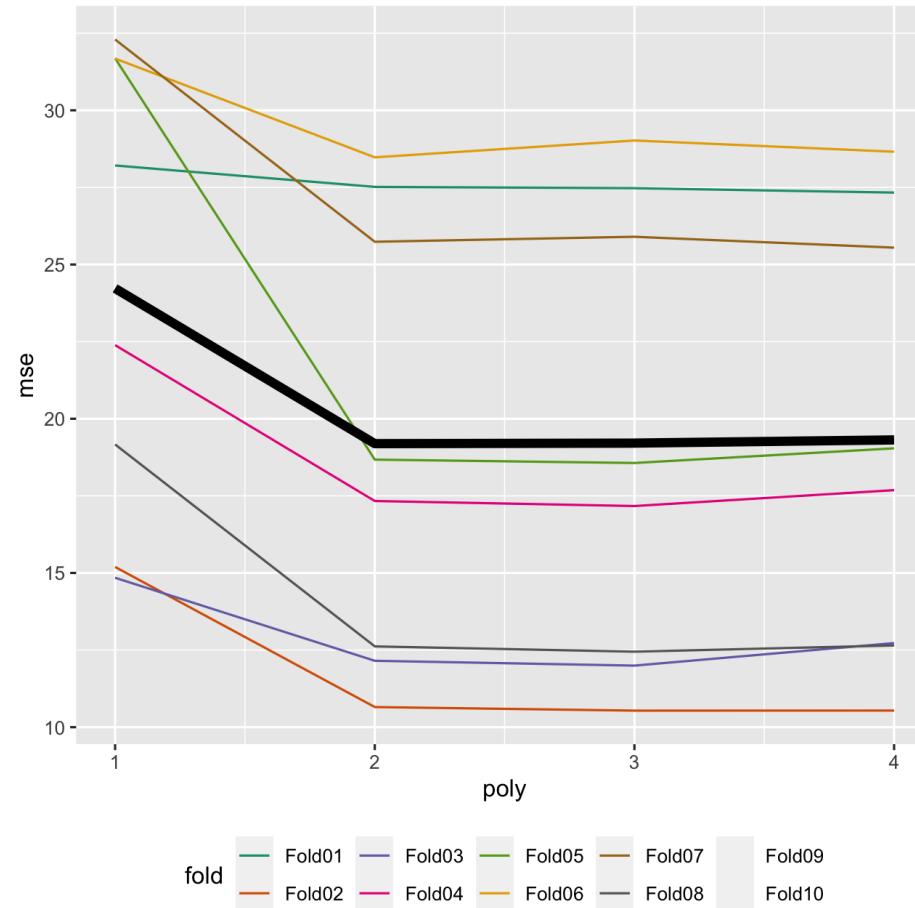
```
##   poly      mse     id
## 1     1 28.21266 Fold01
## 2     2 27.51733 Fold01
## 3     3 27.47538 Fold01
## 4     4 27.33215 Fold01
```



Compute across all folds

```
kfold_results <-
  map_df(
    auto_folds$splits,
    ~compute_fold_mse(.x))
kfold_results
```

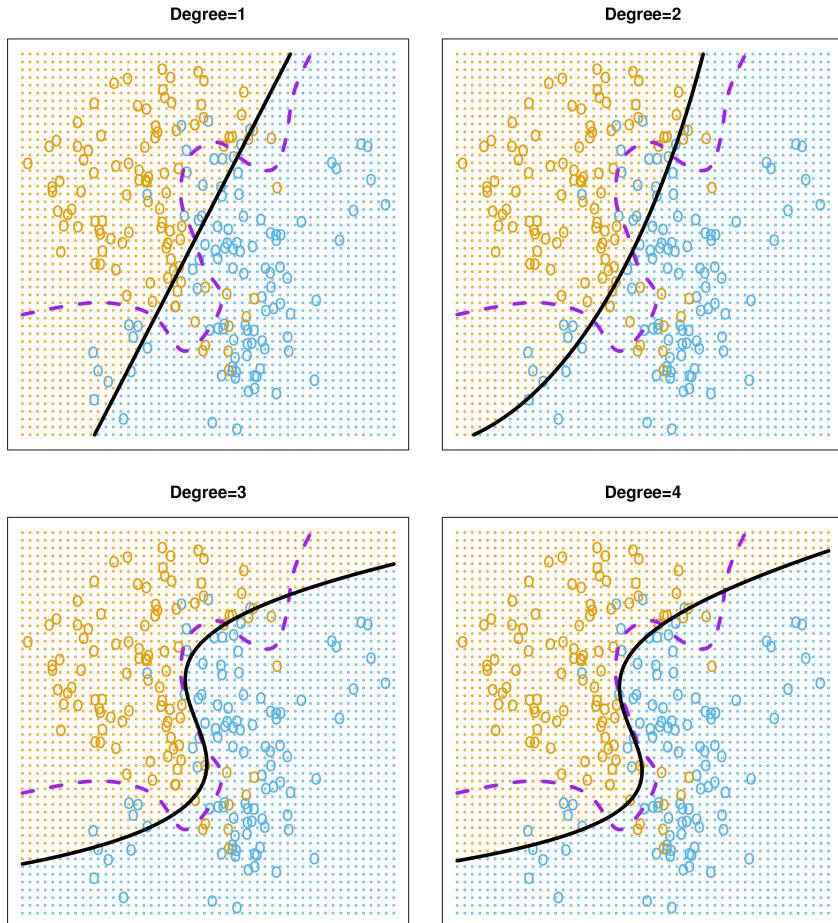
##	poly	mse	id
## 1	1	28.21266	Fold01
## 2	2	27.51733	Fold01
## 3	3	27.47538	Fold01
## 4	4	27.33215	Fold01
## 5	1	15.19469	Fold02
## 6	2	10.65360	Fold02
## 7	3	10.53691	Fold02
## 8	4	10.53989	Fold02
## 9	1	14.84492	Fold03
## 10	2	12.15279	Fold03



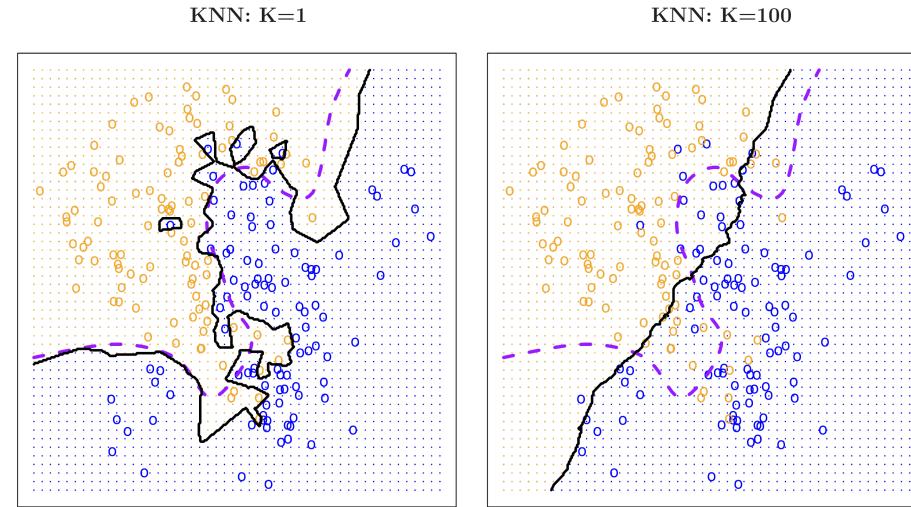
Black is the average MSE across folds.

Classification

Polynomial



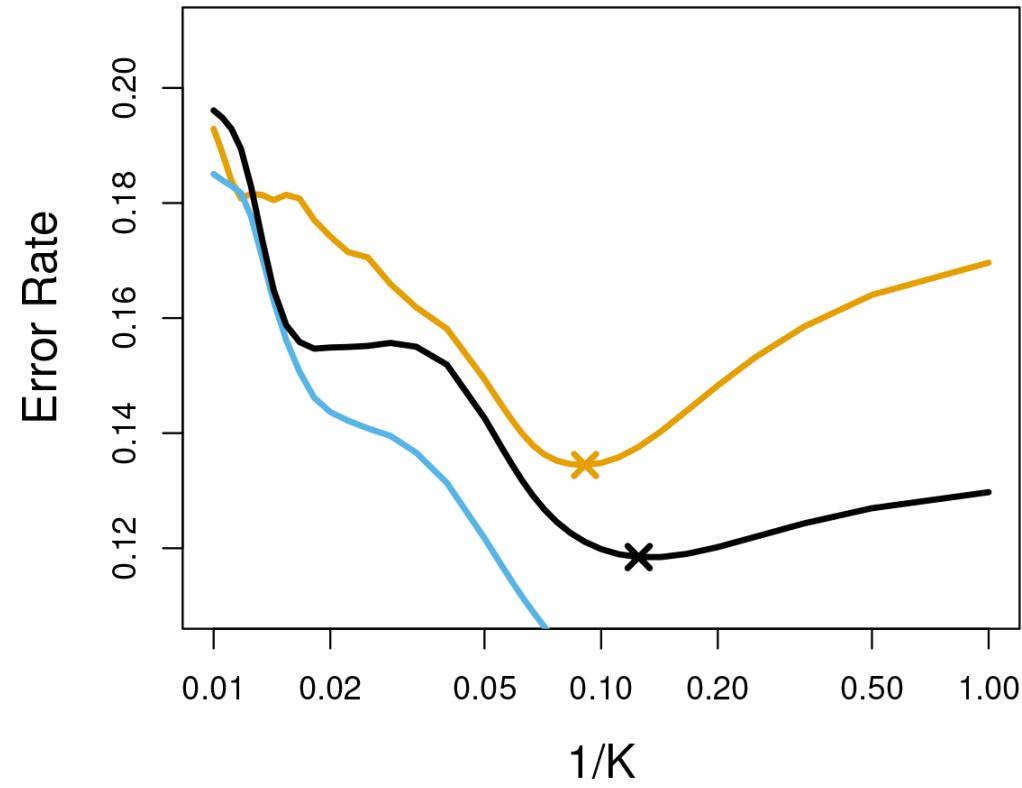
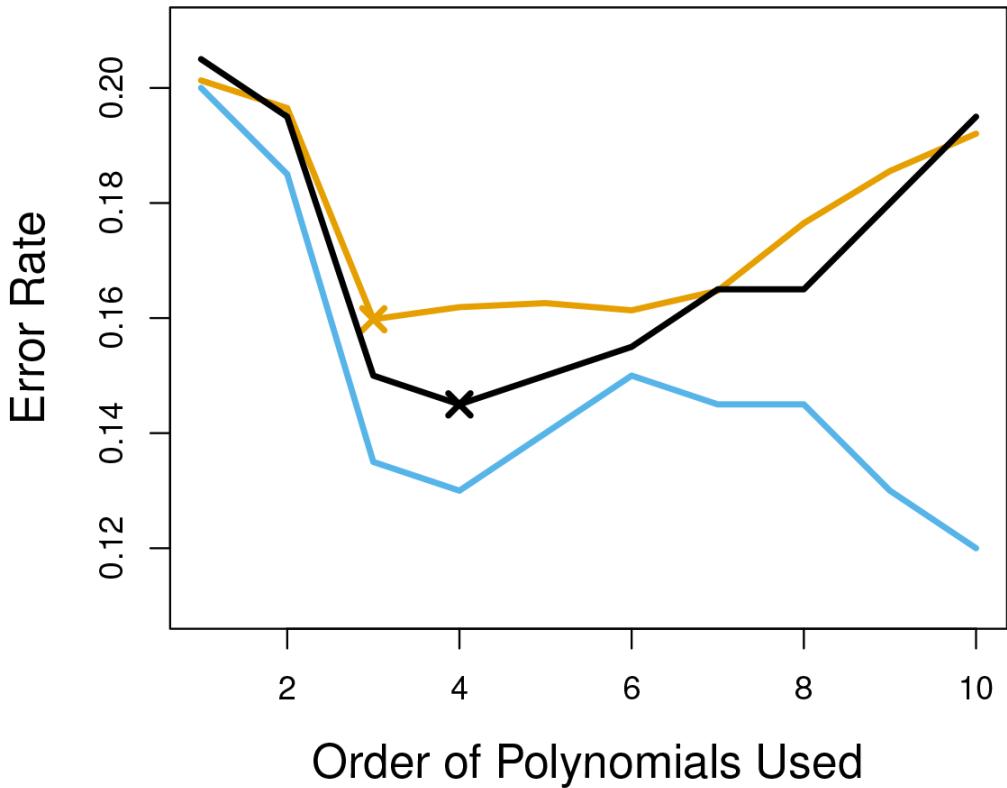
kNN



(Chapter2/2.16.pdf)

(Chapter 5/ 5.7)

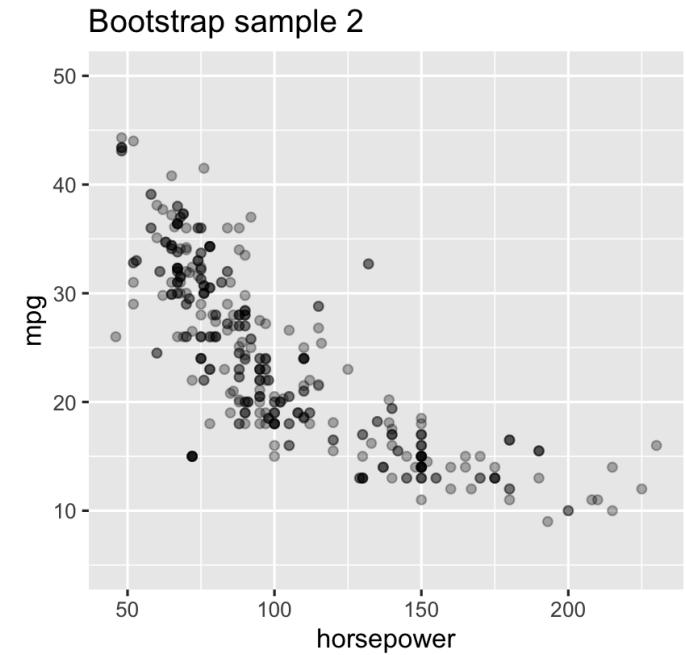
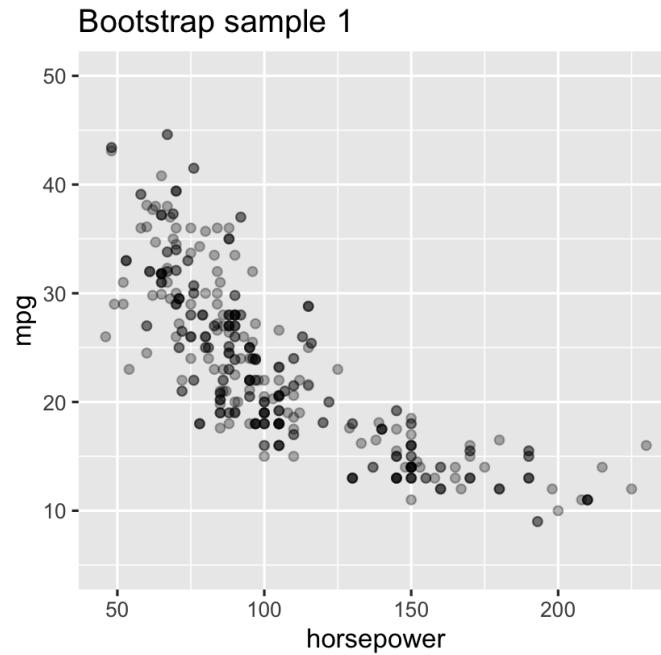
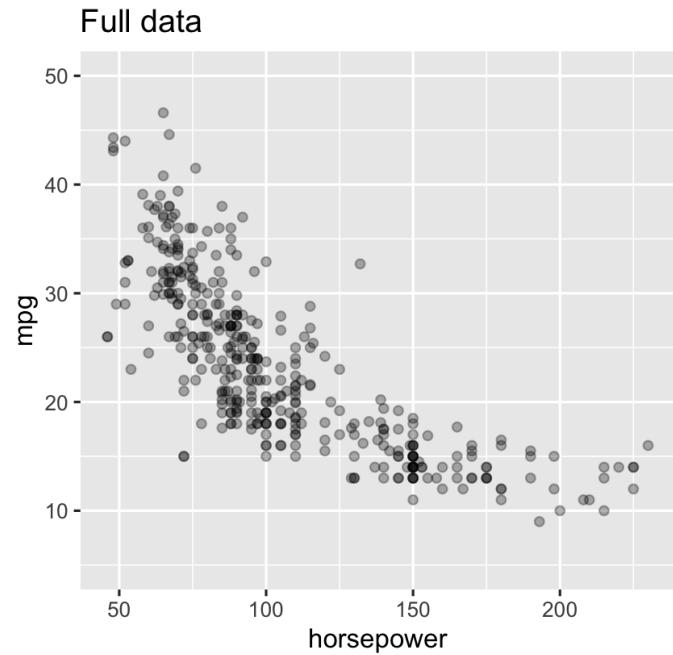
Classification



Black line is 10-fold CV; [training](#) and [test](#) error (based on knowing the true boundary) for different choices of polynomial (left) and KNN classifier (right).

(Chapter 5/ 5.8)

Bootstrap samples



Can you see the differences between the data in each plot?

Bootstrap MSE

- Each of these bootstrap data sets is created by sampling with replacement, and is the same size as our original dataset. As a result some observations may appear more than once in a given bootstrap data set and **some not at all**.
- Fit the model on each set of bootstrap samples, and calculate MSE for the observations left out of each sample (out-of-bag), call this $MSE_{(b)}$

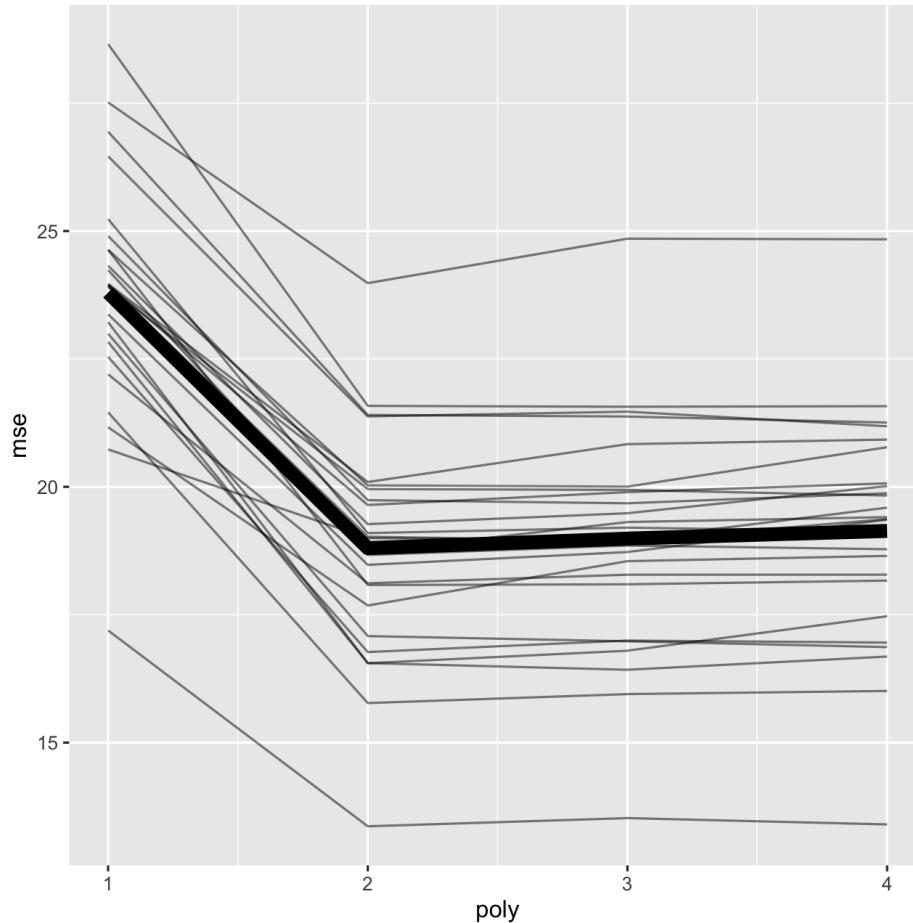
$$MSE_{boot} = \frac{1}{B} \sum_{b=1}^B MSE_{(b)}$$

```

auto_boots <- bootstraps(Auto)
boot_results <-
  map_df(
    auto_boots$splits,
    ~compute_fold_mse(.x))
boot_results

```

	poly	mse	id
## 1	1	24.39040	Bootstrap01
## 2	2	17.66788	Bootstrap01
## 3	3	17.86563	Bootstrap01
## 4	4	18.47839	Bootstrap01
## 5	1	23.54878	Bootstrap02
## 6	2	18.76633	Bootstrap02
## 7	3	18.90653	Bootstrap02
## 8	4	18.89178	Bootstrap02
## 9	1	24.05195	Bootstrap03
## 10	2	18.87384	Bootstrap03
## 11	3	18.87026	Bootstrap03



Best model is quadratic polynomial

All the resampling suggest the same decision

Summary

Re-sampling provides robust estimate of future error, and the variation you are likely to see, in the statistic being calculated, in future samples.



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#).

Lecturer: *Professor Di Cook*

Department of Econometrics and Business Statistics

✉ ETC3250.Clayton-x@monash.edu

CALENDAR
Week 3b

