

Math 390.4 / 650.3 Spring 2019

Midterm Examination Two

Solutions

Professor Adam Kapelner

Thursday, April 16, 2019

Full Name _____

Code of Academic Integrity

Since the college is an academic community, its fundamental purpose is the pursuit of knowledge. Essential to the success of this educational mission is a commitment to the principles of academic integrity. Every member of the college community is responsible for upholding the highest standards of honesty at all times. Students, as members of the community, are also responsible for adhering to the principles and spirit of the following Code of Academic Integrity.

Activities that have the effect or intention of interfering with education, pursuit of knowledge, or fair evaluation of a student's performance are prohibited. Examples of such activities include but are not limited to the following definitions:

Cheating Using or attempting to use unauthorized assistance, material, or study aids in examinations or other academic work or preventing, or attempting to prevent, another from using authorized assistance, material, or study aids. Example: using an unauthorized cheat sheet in a quiz or exam, altering a graded exam and resubmitting it for a better grade, etc.

I acknowledge and agree to uphold this Code of Academic Integrity.

signature

date

Instructions

This exam is 110 minutes and closed-book. You are allowed **one** page (front and back) of a "cheat sheet." You may use a graphing calculator of your choice. Please read the questions carefully. If the question reads "compute," this means the solution will be a number otherwise you can leave the answer in *any* widely accepted mathematical notation which could be resolved to an exact or approximate number with the use of a computer. I advise you to skip problems marked "[Extra Credit]" until you have finished the other questions on the exam, then loop back and plug in all the holes. I also advise you to use pencil. The exam is 100 points total plus extra credit. Partial credit will be granted for incomplete answers on most of the questions. Box in your final answers. Good luck!

Problem 1 Below are some theoretical questions related to OLS.

- (a) [5 pt / 5 pts] Let $\mathbf{w} \in \mathbb{R}^{p+1}$ be a vector, let $\mathbf{y} \in \mathbb{R}^n$ be a vector constant with respect to \mathbf{w} and let $\mathbf{X} \in \mathbb{R}^{n \times (p+1)}$ be a full-rank matrix constant with respect to \mathbf{w} . Find the \mathbf{w} that solves the following equation by showing all steps:

$$\frac{\partial}{\partial \mathbf{w}} [\mathbf{y}^\top \mathbf{y} - 2\mathbf{w}^\top \mathbf{X}^\top \mathbf{y} + \mathbf{w}^\top \mathbf{X}^\top \mathbf{X} \mathbf{w}] = 0$$

$$\Rightarrow -\cancel{2\mathbf{X}^\top \mathbf{y}} + \cancel{2\mathbf{X}^\top \mathbf{X}} \mathbf{w} = 0$$

$$\Rightarrow \mathbf{X}^\top \mathbf{X} \mathbf{w} = \mathbf{X}^\top \mathbf{y} \quad \text{multiply both sides by } (\mathbf{X}^\top \mathbf{X})^{-1}$$

$$\Rightarrow \underbrace{(\mathbf{X}^\top \mathbf{X})^{-1} (\mathbf{X}^\top \mathbf{X})}_{\mathbf{I}} \mathbf{w} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

which exists since \mathbf{X} is full rank

$$\Rightarrow \mathbf{w} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

Problem 2 We continue now with questions related to OLS. Let $\mathbf{b} \in \mathbb{R}^{p+1}$ be the vector found in 1(a), let $\mathbf{y} \in \mathbb{R}^n$ be a constant vector, let $\mathbf{X} \in \mathbb{R}^{n \times (p+1)}$ be a constant full-rank matrix where the first column equals $\mathbf{1}_n$ and let \mathbf{H} be the orthogonal projection matrix that we spoke about in class. Let $\hat{\mathbf{y}} \in \mathbb{R}^n$ be the orthogonal projection of \mathbf{y} using \mathbf{H} and let $\mathbf{e} \in \mathbb{R}^n$ be the difference of \mathbf{y} and its orthogonal projection using \mathbf{H} . Further, let $\text{SST} := \|\mathbf{y} - \bar{y}\|^2$, $\text{SSR} := \|\hat{\mathbf{y}} - \bar{y}\|^2$ and $\text{SSE} := \|\mathbf{e}\|^2$.

- (a) [3 pt / 8 pts] Prove \mathbf{H} is symmetric.

$$\begin{aligned} \mathbf{H} &= \mathbf{X} (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top, \quad \mathbf{H}^\top = (\mathbf{X} (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top)^\top = (\mathbf{X}^\top)^\top (\mathbf{X}^\top \mathbf{X})^{-1}^\top \mathbf{X}^\top \\ &= \mathbf{X} ((\mathbf{X}^\top \mathbf{X})^\top)^{-1} \mathbf{X}^\top = \mathbf{X} (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top = \mathbf{H} \checkmark \end{aligned}$$

(b) [3 pt / 11 pts] Compute $\|Proj_{\text{colsp}[\mathbf{X}]}(\mathbf{1}_n)\|^2$. Justify each non-trivial step.

$$\begin{aligned}
 &= \|\mathbf{H} \vec{\mathbf{1}}_n\|^2 \\
 &= \|\vec{\mathbf{1}}_n\|^2 \text{ since } \vec{\mathbf{1}}_n \in \text{colsp}[\mathbf{X}] \\
 &= 1^2 + 1^2 + \dots + 1^2 = n
 \end{aligned}$$

(c) [2 pt / 13 pts] Let θ be the angle between \mathbf{y} and $\hat{\mathbf{y}}$. As the number of columns grows larger and \mathbf{X} remains full rank, what value does $\cos(\theta)$ converge to?

$$\text{Since } R^2 \rightarrow 1, \cos(\theta) \rightarrow 1$$

(d) [2 pt / 15 pts] Let \mathbf{Q} denote \mathbf{X} orthogonalized using the Gram-Schmidt algorithm. What is the dimension of \mathbf{Q} ?

$$\dim(\mathbf{Q}) = \dim(\mathbf{X}) = n \times (p+1)$$

(e) [3 pt / 18 pts] Let \mathbf{q}_j denote the j th column of \mathbf{Q} . Find \mathbf{q}_1 .

$\vec{\mathbf{q}}_1$ is $\vec{\mathbf{x}}_1$ normalized

$$\vec{\mathbf{x}}_1 = \vec{\mathbf{1}}_n \Rightarrow \vec{\mathbf{q}}_1 = \frac{\vec{\mathbf{1}}_n}{\|\vec{\mathbf{1}}_n\|} = \frac{\vec{\mathbf{1}}_n}{\sqrt{n}} = \frac{1}{\sqrt{n}} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

(f) [3 pt / 21 pts] Find $\|\mathbf{q}_3^T \mathbf{H}\|^2$.

$$\|\vec{\mathbf{q}}_3^T \mathbf{H}\|^2 = \|\mathbf{H}^T \vec{\mathbf{q}}_3\|^2 = \|\mathbf{H} \vec{\mathbf{q}}_3\|^2 = \|\vec{\mathbf{q}}_3\|^2 = 1$$

↑
since \mathbf{H} is
of col vector

= norm row
vector

(its transpose)

$$\mathbf{H} = \mathbf{H}^T$$

↑
since $\vec{\mathbf{q}}_3 \in \text{colsp}(\mathbf{Q}) = \text{colsp}(\mathbf{X})$

due to gram-schmidt

from the
def. of orthogonal matrix:
each col has
normalized
columns

(g) [3 pt / 24 pts] If you add one column to \mathbf{X} and it remains full rank and recompute $\hat{\mathbf{y}}$, circle all quantities below that change:

- i) n
- ii) p
- iii) b
- iv) SST
- v) SSR
- vi) SSE
- ☐ dim $[\mathbf{H}]$
- viii) rank $[\mathbf{H}]$
- ix) $Proj_{\text{colsp}[\mathbf{X}]}(\mathbf{y})$

Problem 3 This question is about modeling price of cars in the cars dataset:

```

1 > dim(cars)
2 [1] 93 27
3 > summary(cars)
4   Manufacturer      Model      Type      Min. Price      Price
5 Chevrolet: 8      100      : 1      Compact:16      Min.      : 6.70      Min.      : 7.40
6 Ford      : 8      190E      : 1      Large :11      1st Qu.:10.80      1st Qu.:12.20
7 Dodge     : 6      240      : 1      Midsize:22      Median :14.70      Median :17.70
8 Mazda     : 5      300E      : 1      Small :21      Mean :17.13      Mean :19.51
9 Pontiac    : 5      323      : 1      Sporty :14      3rd Qu.:20.30      3rd Qu.:23.30
10 Buick     : 4      535i      : 1      Van : 9      Max. :45.40      Max. :61.90
11 (Other) :57      (Other):87
12   Max. Price      MPG. city      MPG. highway      AirBags
13 Min.      : 7.9      Min.      :15.00      Min.      :20.00      Driver & Passenger:16
14 1st Qu.:14.7      1st Qu.:18.00      1st Qu.:26.00      Driver only :43
15 Median :19.6      Median :21.00      Median :28.00      None :34
16 Mean :21.9      Mean :22.37      Mean :29.09
17 3rd Qu.:25.3      3rd Qu.:25.00      3rd Qu.:31.00
18 Max. :80.0      Max. :46.00      Max. :50.00
19
20   DriveTrain  Cylinders      EngineSize      Horsepower      RPM
21 4WD :10      3      : 3      Min. :1.000      Min. : 55.0      Min. :3800
22 Front:67      4      :49      1st Qu.:1.800      1st Qu.:103.0      1st Qu.:4800
23 Rear :16      5      : 2      Median :2.400      Median :140.0      Median :5200
24      6      :31      Mean :2.668      Mean :143.8      Mean :5281
25      8      : 7      3rd Qu.:3.300      3rd Qu.:170.0      3rd Qu.:5750
26      rotary: 1      Max. :5.700      Max. :300.0      Max. :6500
27
28   Rev. per. mile  Man. trans. avail  Fuel. tank. capacity  Passengers
29 Min. :1320      No :32      Min. : 9.20      Min. :2.000
30 1st Qu.:1985      Yes:61      1st Qu.:14.50      1st Qu.:4.000
31 Median :2340      Median :16.40      Median :5.000
32 Mean :2332      Mean :16.66      Mean :5.086
33 3rd Qu.:2565      3rd Qu.:18.80      3rd Qu.:6.000
34 Max. :3755      Max. :27.00      Max. :8.000
35
36   Length      Wheelbase      Width      Turn. circle      Rear. seat. room
37 Min. :141.0      Min. : 90.0      Min. :60.00      Min. :32.00      Min. :19.00
38 1st Qu.:174.0      1st Qu.: 98.0      1st Qu.:67.00      1st Qu.:37.00      1st Qu.:26.00
39 Median :183.0      Median :103.0      Median :69.00      Median :39.00      Median :27.50
40 Mean :183.2      Mean :103.9      Mean :69.38      Mean :38.96      Mean :27.83
41 3rd Qu.:192.0      3rd Qu.:110.0      3rd Qu.:72.00      3rd Qu.:41.00      3rd Qu.:30.00
42 Max. :219.0      Max. :119.0      Max. :78.00      Max. :45.00      Max. :36.00

```


43					NAs :2
44	Luggage.room	Weight	Origin	Make	
45	Min. : 6.00	Min. :1695	USA :48	Acura Integra: 1	
46	1st Qu.:12.00	1st Qu.:2620	non-USA:45	Acura Legend : 1	
47	Median :14.00	Median :3040		Audi 100 : 1	
48	Mean :13.89	Mean :3073		Audi 90 : 1	
49	3rd Qu.:15.00	3rd Qu.:3525		BMW 535i : 1	
50	Max. :22.00	Max. :4105		Buick Century: 1	
51	NAs :11			(Other) :87	

Below are the outputs for a few different OLS models for variable price:

Model 1	(Intercept)	TypeLarge	TypeMidsize	TypeSmall	TypeSporty	TypeVan
	18.212500	6.087500	9.005682	-8.045833	1.180357	0.887500
Model 2	TypeCompact	TypeLarge	TypeMidsize	TypeSmall	TypeSporty	TypeVan
	18.21250	24.30000	27.21818	10.16667	19.39286	19.10000

(a) [2 pt / 26 pts] What is the R code used to fit Model 1?

$\text{lm}(\text{price} \sim \text{Type})$

(b) [2 pt / 28 pts] Which model *most likely* has higher R^2 ?

- i) Model 1
- ii) Model 2
- iii) They have equal R^2
- iv) Not enough information to tell

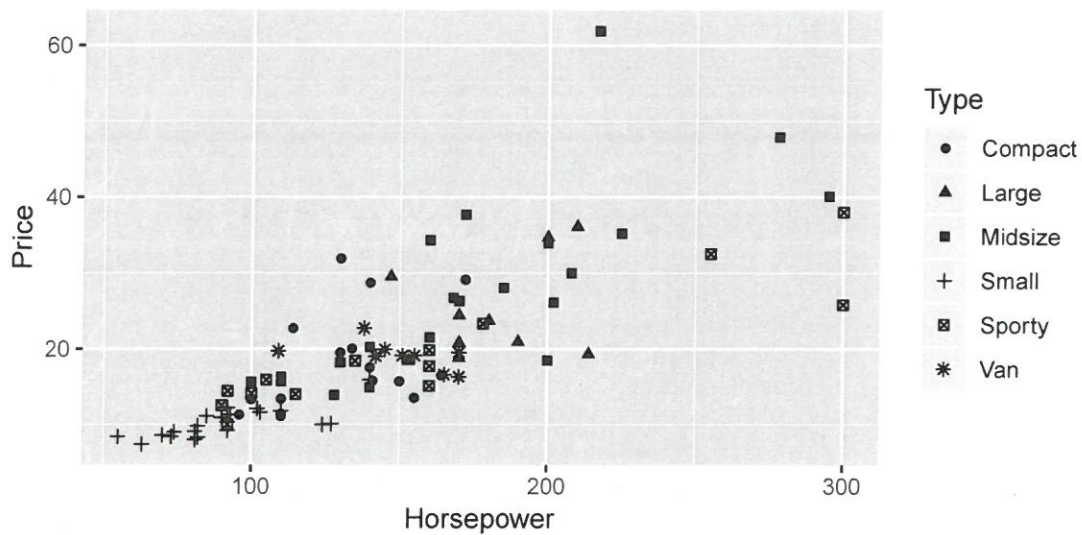
(c) [2 pt / 30 pts] Which model *most likely* has higher oos R^2 ?

- i) Model 1
- ii) Model 2
- iii) They have equal oos R^2
- iv) Not enough information to tell

(d) [4 pt / 34 pts] Assume the dataframe `cars` is sorted by variable `Type` in ascending alphabetical order of the factor level name. Find $\mathbf{X}^T \mathbf{X}$ explicitly for Model 2.

$$\mathbf{X}^T \mathbf{X} = \begin{bmatrix} n_{\text{compact}} & 0 & 0 & 0 & 0 & 0 \\ 0 & n_{\text{large}} & 0 & 0 & 0 & 0 \\ 0 & 0 & n_{\text{midsize}} & 0 & 0 & 0 \\ 0 & 0 & 0 & n_{\text{small}} & 0 & 0 \\ 0 & 0 & 0 & 0 & n_{\text{sporty}} & 0 \\ 0 & 0 & 0 & 0 & 0 & n_{\text{van}} \end{bmatrix} = \begin{bmatrix} 16 & 0 & 0 & 0 & 0 & 0 \\ 0 & 11 & 0 & 0 & 0 & 0 \\ 0 & 0 & 22 & 0 & 0 & 0 \\ 0 & 0 & 0 & 31 & 0 & 0 \\ 0 & 0 & 0 & 0 & 14 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9 \end{bmatrix}$$

Consider the following plot:



(e) [4 pt / 38 pts] Write ggplot code (as best as you can) to generate this figure.

*ggplot(cars) +
geom_point(aes(x = Horsepower, y = Price, shape = Type))*

linear OLS

Consider the following model for target variable price:

Model 3	(Intercept)	Horsepower
	1.45938866	0.12788635
	TypeLarge	TypeMidsize
	5.13487179	-4.98652796
	TypeSmall	TypeSporty
	2.42815602	2.23460382
	TypeVan	Horsepower:TypeLarge
	25.53605395	-0.02922214
	Horsepower:TypeMidsize	Horsepower:TypeSmall
	0.04973893	-0.05888501
	Horsepower:TypeSporty	Horsepower:TypeVan
	-0.02985597	-0.1807183

(f) [2 pt / 40 pts] What is the R code used to fit Model 3?

lm(price ~ Horsepower + Type)

(g) [1 pt / 41 pts] Which model *most likely* has higher R^2 ?

i) Model 2

☒ ii) Model 3

iii) They have equal R^2

iv) Not enough information to tell

(h) [2 pt / 43 pts] Which model *most likely* has higher $\text{oos}R^2$?

i) Model 2

☒ ii) Model 3

iii) They have equal $\text{oos}R^2$

iv) Not enough information to tell

(i) [3 pt / 46 pts] Interpret the number -0.1807183 for term Horsepower:TypeVan in Model 3.

the slope of the Horsepower variable ~~decreases~~ if the car type is Van is 0.18 lower than the slope of the Horsepower variable if car type is Compact.

Problem 4 This question is about OLS again. For the questions concerned with out of sample, consider running the code using split-sample or gathering future data under stationarity. Consider the following code:

```
1 n = 100
2 x = runif(n, 0, 1)
3 X = cbind(1, x)
4 beta = c(1, 1)
5 delta = rnorm(n, mean = 0, sd = 0.1)
6 y = X % * % beta + delta
7
8 mod1 = lm(y ~ 0 + X)
```

(a) [2 pt / 48 pts] What is $f(x)$ in this case? f is defined as we did in class.

$$f(x) = 1 + x$$

(b) [3 pt / 51 pts] Circle all the following that are true for `mod1`.

- ☒ i) b will be very close to β
- ☐ ii) b will not be very close to β
- ☒ iii) s_e will be very small
- ☐ iv) s_e will not be very small
- ☒ v) ooss_e will be very small
- ☐ vi) ooss_e will not be very small

Now consider running the following code after running the first chunk of code:

```
1 x_prime = x + rnorm(n, mean = 0, sd = 1e-6)
2 X = cbind(X, x_prime)
3 mod2 = lm(y ~ 0 + X)
```

(c) [1 pt / 52 pts] In the case of model 2, what is p ?

$$1+1 = 2$$

(d) [4 pt / 56 pts] Circle all the following that are true for `mod2`.

- ☐ i) b will be very close to β
- ☒ ii) b will not be very close to β
- ☒ iii) s_e will be very small
- ☐ iv) s_e will not be very small
- ☒ v) ooss_e will be very small
- ☐ vi) ooss_e will not be very small

Now consider running the following code after running the two previous chunks of code:

```
1 mod3 = lm(y ~ poly(x, 6))
```


(e) [4 pt / 60 pts] Circle all the following that are true for mod3.

- i) b will be very close to β
- ii) b will not be very close to β *(it won't even be the same dimension!)*
- iii) s_e will be very small
- iv) s_e will not be very small
- v) ooss_e will be very small
- vi) ooss_e will not be very small

Problem 5 This question is about the concept of model validation and the strategy we discussed in class. Let's say we divide scramble the rows of \mathbb{D} then create a partition

$$\mathbb{D} = \begin{bmatrix} \mathbb{D}_{\text{train}} \\ \text{---} \\ \mathbb{D}_{\text{select}} \\ \text{---} \\ \mathbb{D}_{\text{test}} \end{bmatrix}$$

in a 3:1:1 ratio train : select : test (in number of rows).

We then fit $g_1 = \mathcal{A}(\mathcal{H}, \mathbb{D}_{\text{train}})$, $g_2 = \mathcal{A}(\mathcal{H}, \mathbb{D}_{\text{test}})$ and $g_{\text{final}} = \mathcal{A}(\mathcal{H}, \mathbb{D})$. Which of the following statement(s) can be employed as a means of *honest* model validation?

(a) [3 pt / 63 pts] We wish to select a model out of M candidate models g_1, g_2, \dots, g_M . Which of the following are recommended strategies of doing so?

- i) Fitting g_1, g_2, \dots, g_M to $\mathbb{D}_{\text{train}}$ and then testing on $\mathbb{D}_{\text{train}}$ and then choosing the model with lowest error on $\mathbb{D}_{\text{train}}$.
- ii) Fitting g_1, g_2, \dots, g_M to $\mathbb{D}_{\text{train}}$ and then testing on \mathbb{D} and then choosing the model with lowest error on \mathbb{D} .
- iii) Fitting g_1, g_2, \dots, g_M to $\mathbb{D}_{\text{train}}$ and then testing on $\mathbb{D}_{\text{select}}$ and then choosing the model with lowest error on $\mathbb{D}_{\text{select}}$.
- iv) Fitting g_1, g_2, \dots, g_M to $\mathbb{D}_{\text{train}}$ and then testing on $\mathbb{D}_{\text{select}}$ and then \mathbb{D}_{test} and then choosing the model with lowest error on \mathbb{D}_{test} .

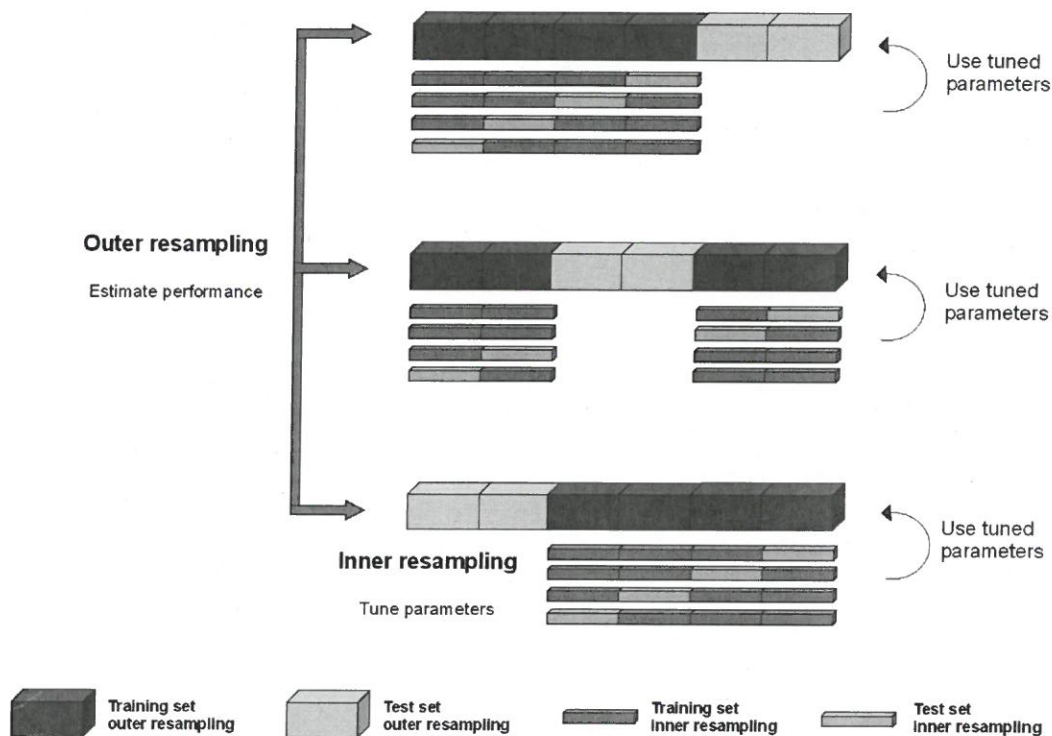
(b) [3 pt / 66 pts] We wish to select a model out of M candidate models g_1, g_2, \dots, g_M and then provide an estimate of model generalization error. Which of the following are recommended strategies of doing so?

- i) Fitting g_1, g_2, \dots, g_M to $\mathbb{D}_{\text{train}}$ and then testing on $\mathbb{D}_{\text{select}}$ and then choosing the model with lowest error on $\mathbb{D}_{\text{select}}$ and providing the estimate of that error.
- ii) Fitting g_1, g_2, \dots, g_M to $\mathbb{D}_{\text{train}}$ and then testing on $\mathbb{D}_{\text{select}}$ and then choosing the model with lowest error on $\mathbb{D}_{\text{select}}$ and then testing on \mathbb{D} and providing the estimate using that error.

- iii) Fitting g_1, g_2, \dots, g_M to $\mathbb{D}_{\text{train}}$ and then testing on $\mathbb{D}_{\text{select}}$ and then choosing the model with lowest error on $\mathbb{D}_{\text{select}}$ and then testing on \mathbb{D}_{test} and providing the estimate using that error.
- iv) Fitting g_1, g_2, \dots, g_M to $\mathbb{D}_{\text{train}}$ and then testing on $\mathbb{D}_{\text{select}}$ and then \mathbb{D}_{test} and then choosing the model with lowest error on \mathbb{D}_{test} and providing the estimate using that error.

(c) [2 pt / 68 pts] Would your answer in (b) be able to provide an estimate of the variability in the generalization error? Yes / **No**.

Consider the selection of the model g_1, g_2, \dots, g_M to be termed "tuning". Imagine we used the protocol pictured below.



(d) [3 pt / 71 pts] What are the number of folds in the inner loop and the outer loop in our problem respecting the ratio given in the problem description?

Inner loop: 4 folds
Outer loop: 5 folds

(e) [4 pt / 75 pts] What are the two main advantages of the protocol above over the answer you gave in (b)?

I It will yield an estimate of the error in the algorithm "select the best model from $\mathcal{J}_1, \dots, \mathcal{J}_m$ " and not only the error of one of those models.

II The generalization error ^{estimate} will be more stable.

Problem 6 Consider the following code:

```

1 compute_distance_matrix = function(X){
2   n = nrow(X)
3   D = matrix(NA, n, n)
4   for (i_1 in 1 : (n - 1)){
5     for (i_2 in (i_1 + 1) : n){
6       D[i_1, i_2] = sqrt(sum((X[i_1, ] - X[i_2, ])^2))
7     }
8   }
9   D
10 }
11
12 pacman::p_load(Rcpp)
13 cppFunction('
14   NumericMatrix compute_distance_matrix_cpp(NumericMatrix X) {
15     int n = X.nrow();
16     int p = X.ncol();
17     NumericMatrix D(n, n);
18     std::fill(D.begin(), D.end(), NA_REAL);
19
20     for (int i_1 = 0; i_1 < (n - 1); i_1++){
21       for (int i_2 = i_1 + 1; i_2 < n; i_2++){
22         int sqd_diff = 0;
23         for (int j = 0; j < p; j++){
24           sqd_diff += pow(X(i_1, j) - X(i_2, j), 2);
25         }
26         D(i_1, i_2) = sqrt(sqd_diff);
27       }
28     }
29     return D;
30   }
31 ')

```

We now profile both functions using a matrix X that has n in the 100's via the code:

```
1 system.time({  
2   D = compute_distance_matrix(X)  
3 })  
4 system.time({  
5   D = compute_distance_matrix_cpp(X)  
6 })
```

- (a) [2 pt / 77 pts] Which function registers a faster profiling time and by how much? Provide a multiple.

the cpp function by a factor of 2/0.

- (b) [2 pt / 79 pts] Explain why this should be.

R has poor performance for loops.

- (c) [2 pt / 81 pts] You wish to recode the R function `sort` using Rcpp. Assume your C++ code is bug-free. Is this endeavor fruitful? Why or why not?

No. sort is a base R function already written in C++/Fortran and optimized over decades of contributions by brilliant people. He would over. You won't be able to beat it easily.

Problem 7 Consider the following dataset:

```
1 > pacman::p_load(ggplot2, dplyr, magrittr)  
2 > D = ggplot2::txhousing  
3 > dim(D)  
4 [1] 8602 9  
5 > summary(D)  
6      city      year      month  
7 Length:8602      Min.   :2000      Min.   : 1.000  
8 Class :character 1st Qu.:2003      1st Qu.: 3.000  
9 Mode  :character Median :2007      Median : 6.000
```


10		Mean :2007	Mean : 6.406
11		3rd Qu.:2011	3rd Qu.: 9.000
12		Max. :2015	Max. :12.000
13			
14	sales	volume	median
15	Min. : 6.0	Min. :8.350e+05	Min. : 50000
16	1st Qu.: 86.0	1st Qu.:1.084e+07	1st Qu.:100000
17	Median : 169.0	Median :2.299e+07	Median :123800
18	Mean : 549.6	Mean :1.069e+08	Mean :128131
19	3rd Qu.: 467.0	3rd Qu.:7.512e+07	3rd Qu.:150000
20	Max. :8945.0	Max. :2.568e+09	Max. :304200
21	NAs :568	NAs :568	NAs :616
22	listings	inventory	date
23	Min. : 0	Min. : 0.000	Min. :2000
24	1st Qu.: 682	1st Qu.: 4.900	1st Qu.:2004
25	Median : 1283	Median : 6.200	Median :2008
26	Mean : 3217	Mean : 7.175	Mean :2008
27	3rd Qu.: 2954	3rd Qu.: 8.150	3rd Qu.:2012
28	Max. :43107	Max. :55.900	Max. :2016
29	NAs :1424	NAs :1467	

- (a) [2 pt / 83 pts] Write **dplyr** code below to update D to convert the city variable into a nominal factor variable.

① `%.<>%`
`mutate(city = factor(city))`

- (b) [5 pt / 88 pts] Write **dplyr** code below to update D to create a new character variable called **month_date** which has a string timestamp with format MM/YYYY, then sort by date (earliest first) and then drop columns month, year and date.

① `%.<>%`
`mutate(month_date = paste(month, the "/", year, sep = "/")) %>%`
`select(-c(month, year, date))`

- (c) [2 pt / 90 pts] Write **dplyr** code below to "windSORize" D on the **volume** variable. This means it will only contain rows that are between the 5%ile and 95%ile of volumes.

① `%.<>%`
`filter(volume >= qvolume(volume, 0.05) & volume <= qvolume(volume, 0.95))`

- (d) [3 pt / 93 pts] Write `dplyr` code below to summarize the data in `D` by providing the average volume in each month.

*D %>%
group_by(month_date) %>%
summarize(avg_volume = mean(volume))*

We now wish to predict the target `volume` based on the other variables as features. Consider the following code after the first chunk has been executed:

```
1 > D %>% na.omit
2 > pacman::p_load(mlr)
3 > modeling_task = makeRegrTask(data = D, target = "volume")
4 > algorithm = makeLearner("regr.lm")
5 > validation = makeResampleDesc("CV", iters = 5)
6 > resample(algorithm, modeling_task, validation, measures = list(rmse))$
  aggr
7 34120325
```

- (e) [3 pt / 96 pts] Interpret the output, 34120325, as best as you can.

An estimate of the generalization error (se) of the linear model of volume ^{explained by} all other features.

- (f) [2 pt / 98 pts] What simple transformation can be done to one of the variables in the dataset that would likely increase predictive performance?

log the volume. The previous page shows the summary of all variables. It is clear that volume is skewed right. Typically logging the skewed variable that is skewed right yields better predictive performance.

Consider the following code:

```
1 > X = model.matrix(volume ~ . * . * ., D)
```

- (g) [2 pt / 100 pts] In one sentence (or less) answer the following: which procedure could you use to build a model predicting `volume` based on the features now found in the design matrix `X`?

formal stepwise linear regression / OLS.