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. $\boxed{\text{Box}}$ in your final answers. Good luck!

Problem 1 Below are some theoretical questions related to OLS.

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

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

$$\Rightarrow -\cancel{2}X^T\vec{y} + \cancel{2}X^TX\vec{w} = 0$$

$$\Rightarrow X^TX\vec{w} = X^T\vec{y}$$

multiply both sides by $(X^TX)^{-1}$ which exists since X is full rank

$$\Rightarrow \underbrace{\cancel{(X^TX)^{-1}}\cancel{(X^TX)}}_{I}\vec{w} = (X^TX)^{-1}X^T\vec{y}$$

$$\Rightarrow \vec{w} = (X^TX)^{-1}X^T\vec{y}$$

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

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

$$H = X(X^TX)^{-1}X^T, \quad H^T = \left(X(X^TX)^{-1}X^T\right)^T = (X^T)^T\left((X^TX)^{-1}\right)^TX^T$$

$$= X\left((X^TX)^T\right)^{-1}X^T = X(X^TX)^{-1}X^T = H \checkmark$$

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

$= ||H \vec{1}_n||^2$

$= ||\vec{1}_n||^2$ since $\vec{1}_n \in \text{Colsp}[X]$

$= 1^2 + 1^2 + \ldots + 1^2 = n$

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

Since $R^2 \to 1$, $\cos(\theta) \to 1$

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

$\dim[Q] = \dim[X] = n \times (p+1)$

(e) [3 pt / 18 pts] Let $\boldsymbol{q}_{\cdot j}$ denote the jth column of $\boldsymbol{Q}$. Find $\boldsymbol{q}_{\cdot 1}$.

$\vec{q}_{\cdot 1}$ is $\vec{X}_{\cdot 1}$ normalized

$$\vec{X}_{\cdot 1} = \vec{1}_n \implies \vec{q}_{\cdot 1} = \frac{\vec{1}_n}{||\vec{1}_n||} = \frac{\vec{1}_n}{\sqrt{n}} = \frac{1}{\sqrt{n}} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

(f) [3 pt / 21 pts] Find $||\boldsymbol{q}_{\cdot 3}^\top \boldsymbol{H}||^2$.

$$\underbrace{||\vec{q}_{\cdot 3}^\top H||^2 = ||H^\top \vec{q}_{\cdot 3}||^2}_{\text{since norm of col vector = norm row vector (its transpose)}} \underbrace{=}_{H = H^\top} ||H \vec{q}_{\cdot 3}||^2 \underbrace{=}_{\text{since } \vec{q}_{\cdot 3} \in \text{colsp}[Q] = \text{colsp}[X] \text{ due to gram-schmidt}} ||\vec{q}_{\cdot 3}||^2 \underbrace{=}_{\text{from the def. of orthogonal matrix: each col has normalized columns}} 1^2 = 1$$

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

i) n

ii) p

iii) $\boldsymbol{b}$

iv) SST

v) SSR

vi) SSE

vii) $\dim[\boldsymbol{H}]$

viii) $\text{rank}[\boldsymbol{H}]$

ix) $Proj_{\text{colsp}[\boldsymbol{X}]}(\boldsymbol{y})$

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

```
> dim(cars)
[1] 93 27
> summary(cars)
     Manufacturer        Model          Type        Min.Price          Price
 Chevrolet: 8      100     : 1    Compact:16   Min.   : 6.70    Min.   : 7.40
 Ford     : 8      190E    : 1    Large  :11   1st Qu.:10.80    1st Qu.:12.20
 Dodge    : 6      240     : 1    Midsize:22   Median :14.70    Median :17.70
 Mazda    : 5      300E    : 1    Small  :21   Mean   :17.13    Mean   :19.51
 Pontiac  : 5      323     : 1    Sporty :14   3rd Qu.:20.30    3rd Qu.:23.30
 Buick    : 4      535i    : 1    Van    : 9   Max.   :45.40    Max.   :61.90
 (Other)  :57      (Other):87
   Max.Price        MPG.city        MPG.highway                  AirBags
 Min.   : 7.9    Min.   :15.00   Min.   :20.00    Driver & Passenger:16
 1st Qu.:14.7    1st Qu.:18.00   1st Qu.:26.00    Driver only       :43
 Median :19.6    Median :21.00   Median :28.00    None              :34
 Mean   :21.9    Mean   :22.37   Mean   :29.09
 3rd Qu.:25.3    3rd Qu.:25.00   3rd Qu.:31.00
 Max.   :80.0    Max.   :46.00   Max.   :50.00

 DriveTrain   Cylinders     EngineSize       Horsepower          RPM
 4WD  :10    3      : 3   Min.   :1.000   Min.   : 55.0   Min.   :3800
 Front:67    4      :49   1st Qu.:1.800   1st Qu.:103.0   1st Qu.:4800
 Rear :16    5      : 2   Median :2.400   Median :140.0   Median :5200
             6      :31   Mean   :2.668   Mean   :143.8   Mean   :5281
             8      : 7   3rd Qu.:3.300   3rd Qu.:170.0   3rd Qu.:5750
             rotary: 1    Max.   :5.700   Max.   :300.0   Max.   :6500

  Rev.per.mile   Man.trans.avail Fuel.tank.capacity   Passengers
 Min.   :1320    No :32          Min.   : 9.20      Min.   :2.000
 1st Qu.:1985    Yes:61          1st Qu.:14.50      1st Qu.:4.000
 Median :2340                    Median :16.40      Median :5.000
 Mean   :2332                    Mean   :16.66      Mean   :5.086
 3rd Qu.:2565                    3rd Qu.:18.80      3rd Qu.:6.000
 Max.   :3755                    Max.   :27.00      Max.   :8.000

     Length          Wheelbase          Width          Turn.circle     Rear.seat.room
 Min.   :141.0   Min.   : 90.0   Min.   :60.00   Min.   :32.00   Min.   :19.00
 1st Qu.:174.0   1st Qu.: 98.0   1st Qu.:67.00   1st Qu.:37.00   1st Qu.:26.00
 Median :183.0   Median :103.0   Median :69.00   Median :39.00   Median :27.50
 Mean   :183.2   Mean   :103.9   Mean   :69.38   Mean   :38.96   Mean   :27.83
 3rd Qu.:192.0   3rd Qu.:110.0   3rd Qu.:72.00   3rd Qu.:41.00   3rd Qu.:30.00
 Max.   :219.0   Max.   :119.0   Max.   :78.00   Max.   :45.00   Max.   :36.00
```

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

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

$\boxed{\text{Model 1}}$

(Intercept)	TypeLarge	TypeMidsize	TypeSmall	TypeSporty	TypeVan
18.212500	6.087500	9.005682	-8.045833	1.180357	0.887500

$\boxed{\text{Model 2}}$

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?

lm(price ~ 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

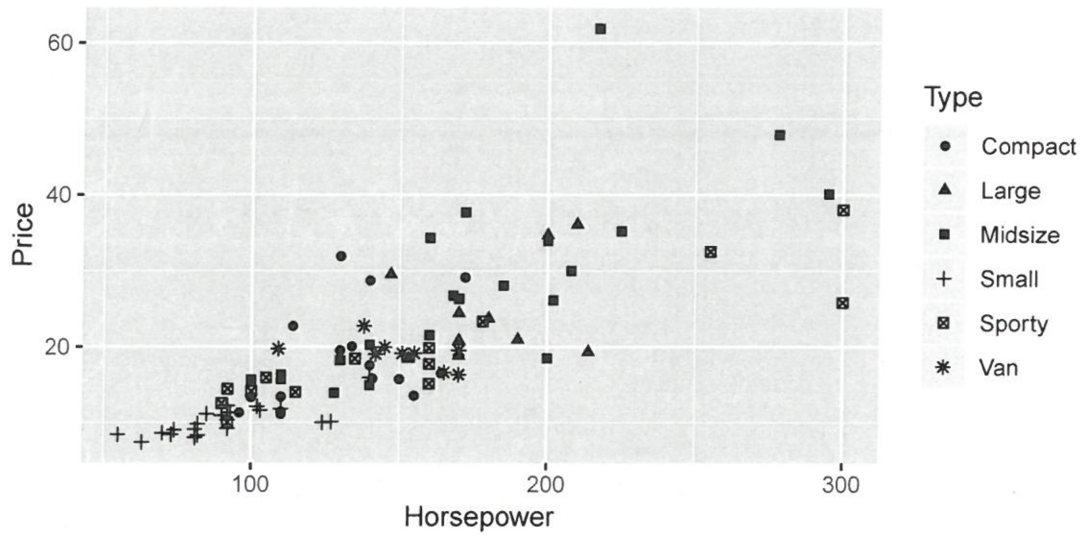
(c) [2 pt / 30 pts] Which model *most likely* has higher oosR^2?

i) Model 1
ii) Model 2
iii) They have equal oosR^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 $\boldsymbol{X}^\top \boldsymbol{X}$ explicitly for Model 2.

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

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 ~~x~~ * 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 oosR^2?

i) Model 2
ii) Model 3
iii) They have equal oosR^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 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:

```
n = 100
x = runif(n, 0, 1)
X = cbind(1, x)
beta = c(1, 1)
delta = rnorm(n, mean = 0, sd = 0.1)
y = X %*% beta + delta

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) $\boldsymbol{b}$ will be very close to β

ii) $\boldsymbol{b}$ 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:

```
x_prime = x + rnorm(n, mean = 0, sd = 1e-6)
X = cbind(X, x_prime)
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) $\boldsymbol{b}$ will be very close to β

ii) $\boldsymbol{b}$ 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:

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

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

i) $\boldsymbol{b}$ will be very close to β

ii) $\boldsymbol{b}$ 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

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

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 $g_1, g_2, \ldots, g_M$. Which of the following are recommended strategies of doing so?

i) Fitting $g_1, g_2, \ldots, g_M$ 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 $g_1, g_2, \ldots, g_M$ to $\mathbb{D}_{\text{train}}$ and then testing on $\mathbb{D}$ and then choosing the model with lowest error on $\mathbb{D}$.

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

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

i) Fitting $g_1, g_2, \ldots, g_M$ 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.

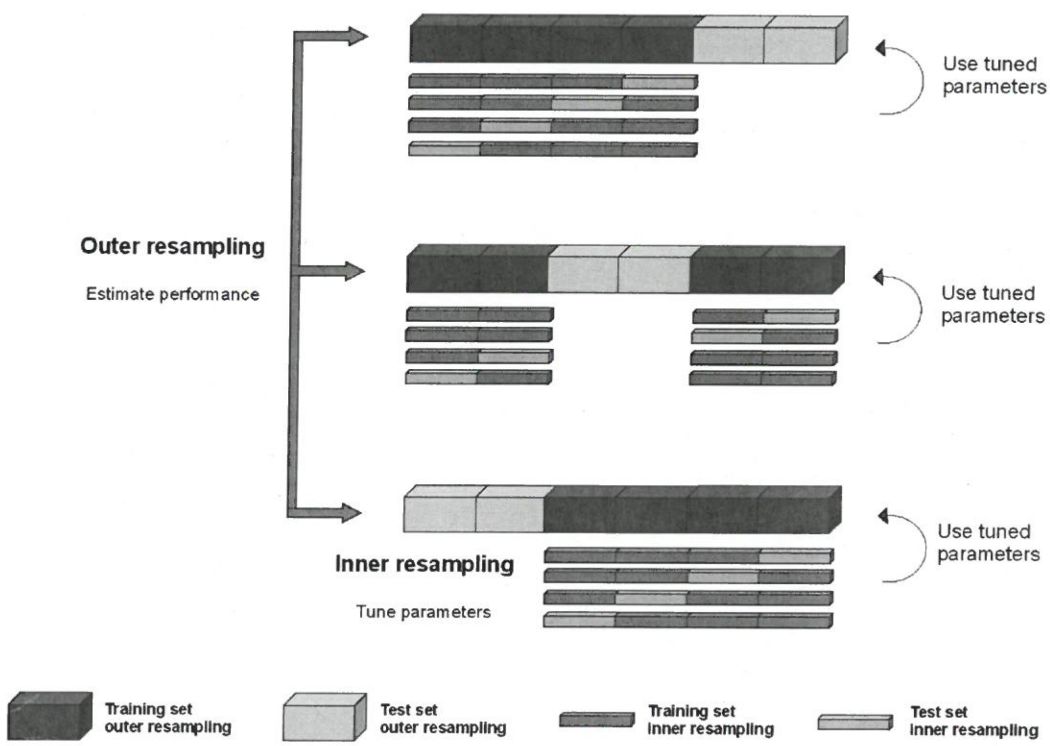
ii) Fitting $g_1, g_2, \ldots, g_M$ to $\mathbb{D}_{\text{train}}$ and then testing on $\mathbb{D}_{\text{select}}$ and then choosing the model with lowest error on $\mathbb{D}_{\text{select}}$ and then testing on $\mathbb{D}$ and providing the estimate using that error.

iii) Fitting $g_1, g_2, \ldots, g_M$ 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 $\mathbb{D}_{\text{test}}$ and providing the estimate using that error.

iv) Fitting $g_1, g_2, \ldots, g_M$ to $\mathbb{D}_{\text{train}}$ and then testing on $\mathbb{D}_{\text{select}}$ and then $\mathbb{D}_{\text{test}}$ and then choosing the model with lowest error on $\mathbb{D}_{\text{test}}$ 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 $g_1, g_2, \ldots, g_M$ 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 $g_1, \ldots, g_m$" 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:

```
compute_distance_matrix = function(X){
  n = nrow(X)
  D = matrix(NA, n, n)
  for (i_1 in 1 : (n - 1)){
    for (i_2 in (i_1 + 1) : n){
      D[i_1, i_2] = sqrt(sum((X[i_1, ] - X[i_2, ])^2))
    }
  }
  D
}

pacman::p_load(Rcpp)
cppFunction('
  NumericMatrix compute_distance_matrix_cpp(NumericMatrix X) {
    int n = X.nrow();
    int p = X.ncol();
    NumericMatrix D(n, n);
    std::fill(D.begin(), D.end(), NA_REAL);

    for (int i_1 = 0; i_1 < (n - 1); i_1++){
      for (int i_2 = i_1 + 1; i_2 < n; i_2++){
        int sqd_diff = 0;
        for (int j = 0; j < p; j++){
          sqd_diff += pow(X(i_1, j) - X(i_2, j), 2);
        }
        D(i_1, i_2) = sqrt(sqd_diff);
      }
    }
    return D;
  }
')
```

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

```
system.time({
  D = compute_distance_matrix(X)
})
system.time({
  D = compute_distance_matrix_cpp(X)
})
```

(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 ≈ 10.

(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 the world over. You won't be able to beat it easily.

Problem 7 Consider the following dataset:

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

```
                              Mean    :2007    Mean    : 6.406
                              3rd Qu.:2011     3rd Qu.: 9.000
                              Max.    :2015    Max.    :12.000

     sales              volume              median
 Min.   :   6.0   Min.   :8.350e+05   Min.   : 50000
 1st Qu.:  86.0   1st Qu.:1.084e+07   1st Qu.:100000
 Median : 169.0   Median :2.299e+07   Median :123800
 Mean   : 549.6   Mean   :1.069e+08   Mean   :128131
 3rd Qu.: 467.0   3rd Qu.:7.512e+07   3rd Qu.:150000
 Max.   :8945.0   Max.   :2.568e+09   Max.   :304200
 NAs    :568      NAs    :568         NAs    :616
    listings         inventory          date
 Min.   :    0   Min.   : 0.000   Min.   :2000
 1st Qu.:  682   1st Qu.: 4.900   1st Qu.:2004
 Median : 1283   Median : 6.200   Median :2008
 Mean   : 3217   Mean   : 7.175   Mean   :2008
 3rd Qu.: 2954   3rd Qu.: 8.150   3rd Qu.:2012
 Max.   :43107   Max.   :55.900   Max.   :2016
 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.

D %<>%
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**.

D %<>%
mutate(month_date = paste(month, "/", 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.

D %<>%
filter(volume ≥ quantile(volume, 0.05) & volume ≤ quantile(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:

```
> D %<>% na.omit
> pacman::p_load(mlr)
> modeling_task = makeRegrTask(data = D, target = "volume")
> algorithm = makeLearner("regr.lm")
> validation = makeResampleDesc("CV", iters = 5)
> resample(algorithm, modeling_task, validation, measures = list(rmse))$
   aggr
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 ~~on~~ 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 outcome variable that is skewed right yields better predictive performance.

Consider the following code:

```
> 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`?

forward stepwise linear regression / OLS.