STSCI/BTRY 3520: Statistical Computing
FINAL EXAM - Spring 2016

Instructions:

- Answer any *four* questions. Each question is worth *20 points*. Up to *5 points* extra credit will be given for each questions answered in addition to the four required.
- When code is requested, make sure you include comments so that it is easy for other people (the graders) to interpret.
- You must complete all work on your own. Unlike with the homework, you are not permitted to discuss the problems with your classmates or refer to outside sources. However, you are permitted to post questions on *Piazza*.
- Submit your solutions via CMS by 5pm on Friday, May 20.

Files to submit to CMS:

Problem 1:

- writeup1.html or writeup1.pdf

Problem 2:

- writeup2.html or writeup2.pdf, and LPmodel.r

Problem 3:

- writeup3.html or writeup3.pdf

Problem 4:

- writeup4.html or writeup4.pdf, and Mygibbs.r

Problem 5:

- writeup5.html or writeup5.pdf, and FindHn.r

Problem 6:

- writeup6.html or writeup6.pdf, and Findx.r and FindLambda.r

Problem 1: The file *ovarian.csv* contains survival times (in days) for six ovarian cancer patients who took vitamin C supplements and the mean survival times of 10 individually matched controls. You can read about this data in Cameron and Pauling (1978, PNAS 75:4538-4542).

Consider a model in which the survival times of these individuals are independent exponential random variables with different rate parameters for the vitamin C group and the controls. Let y_{i1}, $i = 1, \ldots, 6$, denote the survival times in the vitamin C group, and let y_{i2} denote the *total* survival time of the ith group of controls. Denote the two rate parameters by β_1 and β_2 respectively.

Note that the gamma distribution with shape $\alpha > 0$ and rate $\beta > 0$ has density function

$$f(x) = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)}, \quad x > 0\,.$$

Denote this distribution by Gamma(α, β). Note that the mean and variance are given by α/β and α/β^2 respectively, and that the exponential distribution with rate β is the special case in which $\alpha = 1$.

1a: (4 points) Show that the sum of two independent gamma variables with the same rate parameter but different shape parameters is also gamma. Hence determine the distribution of the total survival time for 10 matched controls.

sum of 10 patients

1b: (4 points) Write down the likelihood function for the ovarian survival data. Show that $(\sum y_{i1}, \sum y_{i2})$ are sufficient statistics for the two rate parameters.

factorization theorem

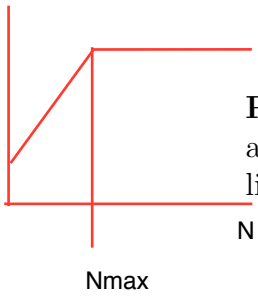
1c: (4 points) Consider a Bayesian model with independent Gamma(a, b) priors on the two rate parameters (for the vitamin C and control groups). Determine the posterior distributions of β_1 and β_2.

prior for the parameter
likelihood*prior

1d: (6 points) Simulate the posterior distribution of the ratio of mean survival times (β_1/β_2) for the vitamin C and control groups when $a = b = 0.1$. Construct a histogram of this distribution and determine a 95% posterior credible interval for the ratio.

Not Done

1e: (2 points) Does there appear to be significant difference between the survival rates for the two groups? Briefly explain your answer.



Problem 2: Plateau models are used in agriculture to estimate the optimal amount of nitrogen (N) to apply to soil to increase crop yields (Y). The linear plateau model is given by

$$Y = \beta_0 + \beta_1 \min(N, N_{max}) + E$$

where E denotes random error, and $N_{max} > 0$ is the nitrogen content beyond which there is no improvement (increase) in yield.

The following data was collected on nitrogen amount and yield of a particular crop.

```
data.frame("nitrogen"= c(rep(0,4),rep(30,4),rep(60,4),rep(90,4),rep(120,4)),
"yield"=c(1.41,1.75,2.02,2.13,1.93,2.24,2.29,2.35,2.12,2.38,
          2.49,2.57,2.16,2.20,2.28,2.49,2.34,2.45,2.59,2.62))
```

The first four yield values correspond to zero nitrogen application, the second four correspond to nitrogen values of 30, etc..

2a: (3 points) Fit a simple linear regression model to the data. Display the fitted model on a scatterplot of the data.

2b: (4 points) Determine the form of the gradient vector of the model function

$$f(x, \theta) = \beta_0 + \beta_1 \min(x, N_{max})$$

with respect to the parameter $\theta = (\beta_0, \beta_1, N_{max})$. (Hint: For N_{max} consider the two cases $x < N_{max}$ and $x > N_{max}$.)

2c: (10 points) Write an R function to fit the linear plateau model using the Gauss-Newton method discussed in Lecture 10. Use the estimated coefficients from your simple linear regression fit as starting values for β_0 and β_1, and $\max(N) - 5$ as the starting value for $N_{\max}$. Submit your code on CMS as LPmodel.r

```
LPmodel=function(Y,N,tol=1e-5,maxiter=100)
{
  ## compute starting values inside the function
  ## del=change in sum of squares between iterations
  return(list(b0,b1,Nmax,iter,del)
}
```

(Note: you can check your answer using the *nls* function in R.)

2d: (3 points) Display the fitted model on the scatterplot. Use different colors for the simple linear regression and linear plateau model fits and indicate which is which using a legend.

chapter 8, bootstrap
lab 6

open and close book

Problem 3: The data in *ocbook.csv* contains the scores on five first year exams for 88 mathematics students.

3a: (2 points) Calculate the largest eigenvalue of the variance-covariance matrix for the five exam scores. Hence determine, $\hat{\pi}_1$, the proportion of variance explained by the first principle component. (Note that this is an estimate based on a sample of 88 mathematics students of a theoretical/hyperthetical quantity, π_1.) 88: sample from larger population

3b: (8 points) For $N = 200$ and $B = 50$, 10000 times calculate an $N \times B$ matrix of bootstrapped versions of $\hat{\pi}_1$. (Each version is based on resampling the students with replacement.) Use each row of the matrix to approximate the bootstrap variance for $\hat{\pi}_1$. Let $\hat{\sigma}_i^2$ denote the ith variance estimate, for $i = 1, \ldots, N$. Determine the mean and variance of $X_i = (B-1)\hat{\sigma}_i^2/\hat{\sigma}^2$, where $\hat{\sigma}^2$ is the mean of all N estimates, and can be regarded as the "true" bootstrap variance.

50 bootstraps
do 200 times

how to approximate?

3c: (6 points) Construct a histogram of the X_i's using the *probability=TRUE* option. What standard probability density can by used to approximate this empirical histogram. Briefly explain why? Overlay the density on the histogram.

The percent error is the relative error expressed in terms of per 100.

3d: (4 points) Construct a histogram of the percent relative error in the bootstrap standard errors, $100 * (\hat{\sigma}_i/\sigma_{\text{true}} - 1)$. Do you think that $B = 50$ is enough resamples to estimate the standard error? Briefly explain.

Not written

known: Y,N, Pl_se, Pl_sp

prevalence P

Y #positive tests : true positive + false positive
N-Y #negative tests: true negative + false negative

Problem 4: Antimicrobial resistance (AMR) is becoming a significant problem due to overuse of antimicrobial drugs. The level of AMR can be estimated by testing isolates obtained from samples (say from retail chicken). Suppose that Y isolates test positive for AMR in N samples. A Bayesian model for the population prevalence of resistance, P, is $Y|P \sim B(N, P)$, combined with a uniform prior distribution on P. This implies a beta posterior distribution for P. (See the notes from Lecture 14.) Suppose, however, that the test has known sensitivity and specificity values,

beta((k+1)-1, (n-k+1)-1)

$$\pi_{se} = P(\text{positive test result}|\text{resistant})$$

and

(Ntp,Ntn) I P,Y

$$\pi_{sp} = P(\text{negative test result}|\text{not resistant})\,.$$

Pl(Ntp,Ntn),Y

A perfect test would have $\pi_{se} = \pi_{sp} = 1$ but this is rarely the case in practice.

Let TP denote a "true positive" test result event; that is, a resistant isolate that tests positive. Let N_{TP} denote the number of TP events in the sample. Let FP denote a "false positive" test result, a non-resistant isolate that tests positive, and let N_{FP} denote the number of such events. Similarly, let TN and FN denote "true negative" and "false negative" test results, and let N_{TN} and N_{FN} denote the number of times these events occur in the sample. Note that $Y = N_{TP} + N_{FP}$ and $N - Y = N_{TN} + N_{FN}$.

4a: (2 points) Determine the conditional probabilities of the events TP, FP, TN an FN given P.

P is simply a parameter

4b: (4 points) What is the distribution of N_{TP} given P and Y? What is the distribution of N_{TN} given P and Y? Hence, what is the distribution of (N_{TP}, N_{TN}) given P and Y?

4c: (4 points) What is the distribution of P given (N_{TP}, N_{TN}) and Y? Note that $N_{TP} + N_{FN}$ is the true number of resistance isolates in the sample.

4d: (6 points) Based on your answers to problems to 4a-c write an R function to generate values from the posterior distribution of P given Y using a Gibbs sampler.

L: length of the chain

```
Mygibbs=function(N,Y,sens,spec,L)
{
```

sensitivity, specificity

```
  # L=chain length
  your code
  return(Pchain)
}
```

Submit your function on CMS as *Mygibbs.r*.

4e: (4 points) Use your function to estimate the posterior distribution of P based on $Y = 30$ positive test results out of $N = 100$ if $\pi_{se} = 0.9$ and $\pi_{sp} = 0.8$. Report estimates of the mean, median, and 2.5 and 97.5 percentiles of the posterior distribution.

Problem 5: Consider the following series:

$$S_N = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \ldots + (-1)^{N+1}\frac{1}{N} = \sum_{n=1}^{N}(-1)^{n+1}\frac{1}{n}$$

Four ways to sum this series are:

1. Sum the terms from $n = 1$ to $n = N$, i.e. compute $1 - \frac{1}{2} + \ldots$
2. Sum the terms from $n = N$ to $n = 1$, i.e. compute $(-1)^{N+1}\frac{1}{N} + (-1)^{N}\frac{1}{N-1} + \ldots$
3. Sum *each* paired term, i.e. compute $(1 - \frac{1}{2}) + (\frac{1}{3} - \frac{1}{4}) + \ldots$
4. Sum *each* paired term written as $\sum_{k=1}^{K} \frac{1}{(2k-1)(2k)}$

5a: (4 points) Write code to compare the performance of these methods. Use *signif* to round numbers to 6 significant figures (i.e. assume we are living in a 6 sf world instead of a 16 sf world). Present your comparisons using a table or graph. accuracy

5b: (2 points) Now, suppose you are given any *arbitrary sequence* a_n, and we want to find the partial sum $S_N = \sum_{i=1}^{N} a_i$. Describe a heuristic to compute this sum with the least error, explaining your reasoning.

5c: (4 points) Plot the following functions over a sequence of points very close to zero (e.g. -4e-8 to 4e-8). For each function, derive and plot a Taylor series approximation which better represents the function values over this range.

(i) $f(x) = 1 - \sin(x)/x$

(ii) $f(x) = \exp\{-2x^2\} - \exp\{-8x^2\}$

(iii) $f(x) = \log(1 + x)/x$

(iv) $f(x) = (1 - e^{x^2})/x^2$

5d: (10 points) In this course we've learned that the harmonic series converges to a constant C in R. The goal of this subproblem is to find this value C. One way to do this is to write this code:

computer has limited accuracy

```
FindHn<-function(k){
  HPrev = -1
  Hnow = 0
  n = 1
  while(HPrev != Hnow){
    HPrev = Hnow
    Hnow = Hnow + round(1/n,k)
    n = n + 1
  }
  n = n-1
  return(Hnow)
}
FindHn(16)  # Recall we work with 16 significant figures!
```

Unfortunately, running this code will take a few months - and we want to beat this time. However, consider the case where we set $k = 1$ (round to 1 decimal place). If we denote $r_1(x)$ as rounding x to 1 decimal place, then we have:

actually has two answers

1/26 rounds to 0

1/4=0.25

$$\begin{aligned} H_n &= \sum_{i=1}^{\infty} r_1(\frac{1}{i}) \\ &= 1 + 0.5 + 0.3 + 0.2 + 0.2 + 0.2 + 0.1 + \ldots + 0.1 \\ &= 1 \times 1 \ + \ 1 \times 0.5 \ + \ 1 \times 0.3 \ + \ 3 \times 0.2 \ + \ 14 \times 0.1 \end{aligned}$$

Thus, instead of adding, we simply add multiples of the rounded terms. Therefore, the goal is to find these multiples for a general k. Submit your function to CMS as *FindHn.r*, and give the value C.

Problem 6: LU Factorization splits up a square matrix into the product of a lower triangular matrix, and an upper triangular matrix. For example, we would write $A = LU$ as: There is built in functions that can be used

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{pmatrix}$$

Suppose we wanted to find $\mathbf{y}$ satisfying $L\mathbf{y} = \mathbf{b}$, where we are given L and $\mathbf{b}$. For example, we might have:

$$\begin{pmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} l_{11}y_1 \\ l_{21}y_1 + l_{22}y_2 \\ l_{31}y_1 + l_{32}y_2 + l_{33}y_3 \\ l_{41}y_1 + l_{42}y_2 + l_{43}y_3 + l_{44}y_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}$$

Starting from the top, we know that $l_{11}y_1 = b_1$, and thus $y_1 = b_1/l_{11}$. We can work our way "down" to get the values of y_2, y_3, and y_4 easily by substituting in values we already know.

6a: (2 points) Given any square lower triangular matrix L and vector $\mathbf{b}$, write code to find $\mathbf{y}$ such that $L\mathbf{y} = \mathbf{b}$. Under what conditions can we not get a solution?

6b: (2 points) Using the same idea as the above, given any square upper triangular matrix U and vector $\mathbf{y}$, write code to find $\mathbf{x}$ such that $U\mathbf{x} = \mathbf{y}$.

6c: (3 points) Explain how you can use the code in parts 6a and 6b to find $A\mathbf{x} = \mathbf{b}$ given that you have the factorization $A = LU$. Thus, write a function: can avoid inversing

```
Findx<-function(A,b){
  # A is a n by n square matrix
  # b is a n by 1 column vector
  # x is a n by 1 column vectors such that A*x = b
  # You may use the package {\tt matrixcalc} or similar
  # but only to find the LU decomposition of a square matrix}.
  # code
  return(x)
}
```

Your function should output a warning message if there is no solution.

6d: (3 points) Explain how you could modify our code if you wanted to find $AX = B$, where X and B are now $n \times 2$ matrices instead of $n \times 1$ vectors? Thus, explain how you would modify your code to find A^{-1}.

Ridge regression is a modification of least squares in which large regression coefficients are penalized. Specifically, the coefficients in the linear model, $y = X\beta + e$, are estimated by minimizing the cost function:

$$C := \|\mathbf{y} - X\beta\|^2 + \lambda\|\beta\|^2$$

or equivalently:

$$C := \sum_{i=1}^{n}(y_i - x_i^T\beta)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

where prediction of X is very large
prediction is bigger than the sample size

where $\lambda > 0$ is an additional "tuning" parameter. It is easily verified that, for fixed λ, the value of β that mimimizes C is given by

$$\hat{\beta} = (X^TX + \lambda I_p)^{-1}X^T\mathbf{y}$$

The least squares solution corresponds to $\lambda = 0$.

6e: (2 points) Explain why it would be a bad idea to minimize C jointly with respect to λ and β.

6f: (8 points) A standard way to choose a value of λ is to find the value that leads to the best predictions via cross-validation. Write the following function to find λ by this method. Clearly document your code.

```
FindLambda<-function(X,y,lambda_vec){
  # Inputs: X, a n by p matrix of n observations and p variables
  #         Y, corresponding Y where Xbeta = Y
  #         lambda_vec, a vector of lambdas to cycle through

  # The function should:
  # 1. Randomly split X and Y into an 80-20% split
  #    You may call them X_Train, Y_Train, and X_Test, Y_Test if you
  #    want, where X_Train, Y_Train correspond to the 80% split
  #    and Y_Test, Y_Test to the 20% split.
```

```
  #    Round the values if needed.
  # 2. For every value of lambda in lambda_vec
  #      + Compute beta_hat by using X_Train and Y_Train
  #      + Compute the squared sum of residuals by using this
  #        beta_hat on X_Test and Y_Test
  # 3. Return lambda which gives the least squared sum of residuals
  return(lambda)
}
```

A bottleneck is computing the matrix inverse, $(X^TX + \lambda I_p)^{-1}$. A solution that gets full credit is one that deals with the bottleneck smartly. (HINT: Note that $X^TX = PDP^T$ for some orthogonal matrix P and diagonal matrix D.)