Statistical Computing: Solutions to Midterm Exam

Name & email:

Instructions:

1.The exam can be found in a pdf and Rmarkdown form in Github.

2.Use the Rmarkdown form to answer the questions of the exam, using R.

3.You will need to submit your Rmarkdown file and the knit PDF file as your solutions to this exam, in D2L->Assignments-> Midterm Exam.

4.The exam is prepared so that you can complete it by 4:30pm. Notice that you will have until 5 pm to upload your submission on D2L.

5.There are 3 questions in the exam, each in a different page. Each question has sub-questions. The exam is with open notes. Points for each subquestion can be found in a parenthesis.

Question 1: Maximum Likelihood Estimation for the parameters α, β of the Gamma distribution.

Let X be a random variable that follows the Gamma(α, β), where α is the shape parameter and β is the rate parameter. Then the density function of X is:

$f(X = x|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$.

If $X_1, X_2, X_3, ..., X_n$ are n independent and identically distributed random variables that follow the same $Gamma(\alpha, \beta)$,then the likelihood function of $X_1 = x_1, X_2 = x_2, X_3 = x_3, ..., X_n = x_n$ is:

$L(\alpha, \beta | x_1, x_2, x_3, ..., x_n) = \prod_{i=1}^{n} \frac{\beta^\alpha}{\Gamma(\alpha)} x_i^{\alpha-1} e^{-\beta x_i}$ and the loglikelihood is

$\ell(\alpha, \beta | x_1, x_2, x_3, ..., x_n) = \sum_{i=1}^{n} log(\frac{\beta^\alpha}{\Gamma(\alpha)} x_i^{\alpha-1} e^{-\beta x_i}) = nlog(\frac{\beta^\alpha}{\Gamma(\alpha)}) + (\alpha - 1) \sum_{i=1}^{n} log(x_i) - \beta \sum_{i=1}^{n} x_i$.

In R:

-You can use the R function gamma(x) to calculate values of $\Gamma(x)$.

-For your convenience you can find the R functions for the Gamma distribution here.

-Notice that for this problem we define a Gamma distribution only by the shape and rate parameter and we ignore the scale parameter.

1.1)(18 points) Write a function that evaluates the negative log-likelihood of a random sample of IID Gamma(α, β) distributed random variables.

```r
#METHOD 1
negLogLik=function(x,theta){
-sum( dgamma(x,shape=theta[1],rate=theta[2],log=TRUE))
}

# OR
```

```r
#METHOD 2
negLogLik2=function(x,theta){
n = length(x)
l = n*theta[1]*log(theta[2]) - n*log(gamma(theta[1])) + (theta[1]-1)*sum(log(x)) - theta[2]*sum(x)
return(-l)
}
```

1.2) (18 points)Using your function from 1.1) and the following data find the maximum likelihood estimators of α and β.

- Use $\alpha = 1, \beta = 1$ as starting parameters.

- Report your parameter estimates.

```r
x = as.matrix(read.table('https://raw.githubusercontent.com/gdlc/STAT_COMP/master/DATA/Gamma_data.txt')
head(x)
```

```
##           x
## 1 1.9215984
## 2 2.3419210
## 3 1.1312172
## 4 2.3851083
## 5 0.9982682
## 6 0.3641080
```

```r
fm1=optim(fn=negLogLik,x=x,par=c(1,1),hessian = TRUE)
fm1
```

```
## $par
## [1] 3.015348 2.064274
##
## $value
## [1] 563.0525
##
## $counts
## function gradient
##       71       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##           [,1]      [,2]
## [1,]  196.2914 -242.2160
## [2,] -242.2160  353.8124
```

OR

```
fm2=optim(fn=negLogLik2,x=x,par=c(1,1), hessian = TRUE)
fm2
```

```
## $par
## [1] 3.015348 2.064274
##
## $value
## [1] 563.0525
##
## $counts
## function gradient
##       71       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##           [,1]      [,2]
## [1,]  196.2914 -242.2160
## [2,] -242.2160  353.8124
```

Based on the maximization of the log likelihood the estimated $\alpha = 3.015348$ and the estimated $\beta = 2.064274$.Notice that the function optimize() can not be used here, because it is used for the maximization of a function with respect to only one parameter. Here we maximize the loglikehood with respect to two parameters.

1.3) (18 points)Provide Standard Errors, and 95% Confidence Intervals for each of the parameters.

For the standards errors we use the inverse of the Fisher information matrix, which is the inverse of the hessian of the likelihood computed at the MLE estimators.

```
VAR = solve(fm1$hessian)
SE = sqrt(diag(VAR))
SE
```

```
## [1] 0.1811515 0.1349292
```

```
VAR2 = solve(fm2$hessian)
SE2 = sqrt(diag(VAR2))
SE2
```

```
## [1] 0.1811515 0.1349292
```

#A 95% CI for α will be

```
lower = fm1$par[1] - 1.96*SE[1]
upper = fm1$par[1] + 1.96*SE[1]

lower
```

```
## [1] 2.660291
```

```
upper
```

```
## [1] 3.370405
```

#and for β

```
lower = fm1$par[2] - 1.96*SE[2]
upper = fm1$par[2] + 1.96*SE[2]

lower
```

```
## [1] 1.799812
```

```
upper
```

```
## [1] 2.328735
```

The 95% CI for α is (2.660291, 3.370405) and for β is (1.799812, 2.328735).

Question 2: Logistic regression

Below you are provided with a simulated data set containing information on ten thousand customers. Specifically, for each customer we are given information about the following variables:

1.*default* : A factor with levels No and Yes indicating whether the customer defaulted on their debt. Default usually happens after six months in a row of not making at least the minimum payment due.

2.*student* : A factor with levels No and Yes indicating whether the customer is a student

3.*balance*: The average balance that the customer has remaining on their credit card after making their monthly payment

4.*income* : Income of customer

```
library(ISLR2)
```

```
## Warning: package 'ISLR2' was built under R version 4.1.3
```

```
Default_data = Default
head(Default_data)
```

```
##   default student   balance    income
## 1      No      No  729.5265 44361.625
## 2      No     Yes  817.1804 12106.135
## 3      No      No 1073.5492 31767.139
## 4      No      No  529.2506 35704.494
## 5      No      No  785.6559 38463.496
## 6      No     Yes  919.5885  7491.559
```

2.1) (18 points)We model the probability of defaulting based on the variables *student*, *balance* and *income*. Using the logistic regression model default~student + balance + income, estimate the probability that a student with median balance and median income will default. To do that follow the steps below:

-Fit a logistic regression model and

-Find the median of the variable balance and the median of the variable income.

-Find the predicted probability of defaulting for a student with balance equal with the median balance of the data and income equal with the median income of the data.

-Construct 95% Confidence interval for the prediction of the probability of defaulting for a student with median balance and median income.

-Based on your results would you expect a student with median income and balance to default?

```
Default_data$default <- ifelse(Default_data$default == 'Yes', 1, 0)
head(Default_data)
```

```
##   default student   balance    income
## 1       0      No  729.5265 44361.625
## 2       0     Yes  817.1804 12106.135
## 3       0      No 1073.5492 31767.139
## 4       0      No  529.2506 35704.494
## 5       0      No  785.6559 38463.496
## 6       0     Yes  919.5885  7491.559
```

```r
fm = glm(default ~ student + balance + income, data = Default_data, family = 'binomial')
summary(fm)
```

```
## 
## Call:
## glm(formula = default ~ student + balance + income, family = "binomial", 
##     data = Default_data)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.4691  -0.1418  -0.0557  -0.0203   3.7383  
## 
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -1.087e+01  4.923e-01 -22.080  < 2e-16 ***
## studentYes  -6.468e-01  2.363e-01  -2.738  0.00619 ** 
## balance      5.737e-03  2.319e-04  24.738  < 2e-16 ***
## income       3.033e-06  8.203e-06   0.370  0.71152    
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1571.5  on 9996  degrees of freedom
## AIC: 1579.5
## 
## Number of Fisher Scoring iterations: 8
```

```r
md_balance = median(Default_data$balance)
md_income = median(Default_data$income)

newData = data.frame(student = 'Yes',balance=md_balance, income = md_income)

pred_prob = predict(fm,type = 'response', se.fit = TRUE, newdata = newData)
pred_prob
```

```
## $fit
##           1 
## 0.001246449 
## 
## $se.fit
##            1 
## 0.0003421971 
## 
## $residual.scale
## [1] 1
```

```r
CI = c(pred_prob$fit-1.96*pred_prob$se.fit,pred_prob$fit+1.96*pred_prob$se.fit)
CI
```

```
##            1            1 
## 0.0005757425 0.0019171553
```

Conclusion, we would not expect a student with median balance and median income to NOT default because:

1. the predicted probability of defaulting for such a student is 0.001246449, which is very low,
2. and based on the produced confidence interval, we are 95% confident that the probability of defaulting will be within the range 0.00057 and 0.0019, which includes very small probabilities of defaulting.

If the predicted probability was close to 0.5 or larger, or if the confidence interval included 0.5, then there would be a chance that a student with median balance and median income could default.

Question 3: Linear regression

A statistics professor regularly wears a smart watch to keep track of their steps and calories burnt within a day. The smart watch records not only the number of steps taken each day, but also the number of minutes walked at a moderate pace, and the number of miles total that they walked.

The professor created a data set of the above information for each day and added the variable Rain to record whether it was a rainy, sunny, or cold day.

The following dataset has 68 observations on the following 8 variables.

1.*Steps*: Total number of steps for the day

2.*Moderate*: Number of steps at a moderate walking speed

3.*Min*: Number of minutes walking at a moderate speed

4.*kcal*: Number of calories burned walking at a moderate speed

5.*Mile*: Total number of miles walked

6.*Rain*: Type of weather (rain or shine)

7.*Day*: Day of the week (U=Sunday, M=Monday, T=Tuesday, W=Wednesday, R=Thursday, F=Friday, S=Saturday

8.*DayType*: Coded as Weekday or Weekend

```r
path='https://vincentarelbundock.github.io/Rdatasets/csv/Stat2Data/Pedometer.csv'

walk_data = read.csv(path, row.names = 1)
```

Solutions are written based on a significance level of 0.001. If you used a different level, grading was adjusted to your significance level.

3.1) (10 points)Use the *walk_data* to fit linear regression model that predicts calories *kcal* burnt on a day from walking as function of *Steps* and *Min.* Summarize in no more than 2 sentence your conclusions regarding the fit of this model.

```r
walk_model = lm(kcal~Steps + Min, data = walk_data)

summary(walk_model)
```

```
##
## Call:
## lm(formula = kcal ~ Steps + Min, data = walk_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21.1173  -4.4282  -0.9327   3.2941  17.2841
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.106855   3.971302  -1.790   0.0782 .
## Steps        0.007161   0.001187   6.034 8.48e-08 ***
## Min          4.415472   0.155877  28.327  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.214 on 65 degrees of freedom
## Multiple R-squared:  0.9922, Adjusted R-squared:  0.992
## F-statistic:  4138 on 2 and 65 DF,  p-value: < 2.2e-16
```

We observe that both Min and Steps are useful in predicting calories burnt, since the F-test, reported in the summary, for H0: kcal~1 and Ha: kcal~ Steps+Min, has a small p-value, less than 0.01.

In addition, the parametric test for the coefficients of Steps and the parametric test for the coefficients of Mins have also small p-values (less than 0.01), implying that coefficients are non-zero.

Finally, the adjusted R squared and R squared are close to 1 meaning that there is a good fit and a big percentage of the kcal data can be explained/described by the model.

3.2) (18 points)The professor would typically bike on sunny days, so they would walk less on those days. At the same time, if it was raining they would choose to walk instead of biking. Based on that, the variable *Rain* could possibly affect the variables *Steps* and the variable *kcal*.

-Test if adding the variable *Rain* and the interaction of *Rain* with *Steps* to the above model, would help the prediction of *kcal*. What is your conclusion?

```
walk_modelf = lm(kcal~Steps + Min + Rain + Rain*Steps, data = walk_data)

anova(walk_model, walk_modelf)
```

```
## Analysis of Variance Table
##
## Model 1: kcal ~ Steps + Min
## Model 2: kcal ~ Steps + Min + Rain + Rain * Steps
##   Res.Df    RSS Df Sum of Sq     F   Pr(>F)
## 1     65 3382.4
## 2     61 2651.7  4    730.66 4.202 0.004534 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value of the F-test, H0: kcal ~ Steps + Min vs Ha: kcal ~ Steps + Min + Rain + Rain * Steps, has a p-value= 0.004534 < 0.01.

So, we have significant evidence against the model which only includes the variables Min and Step. Including the interaction term is helpful in predicting calories burnt.