Final

Due: Wednesday, April 28, 2021 6:45 pm (Pacific Daylight Time)

Thanks for your submission!

Your assignment has been received and is waiting to be graded.

Review your submission

HC (0 points)

1 page submitted

Honour Code

STAT 240: Introduction to Data Science

Final Exam Spring 2021

This final exam consists of 5 questions. The last question is choose your own adventure (you must upload only one of the two options). Aspects of this final exam must be handed in through crowdmark. This final exam open book and take home and due Wednesday April 28th at 6:30PM PST. You may work on this final exam for any amount of time between Wednesday April 21st at 3:30PM PST and the deadline. You may access any texts, notes or lectures while completing this final exam. You may access resources on the internet provided that you don't communicate any aspect of this final exam, distribute the final exam material in any way, or confer with other students or third parties regarding this final exam; as formalized below. This final exam is out of 40 marks.

Honour Code

In taking this final exam you are required to affirm your willingness to abide with the course policies. By signing your name below, you affirm that you are abiding by the following honour code:

I understand that the following activities are prohibited and will be considered cheating. I agree that I will not participate in any of the following activities:

- *Looking at or copying from another student's exam or materials while writing the exam.*
- *Conferring with other students.*
- *Having someone else take the exam in your place.*
- *Distributing the exam materials in any way or discussing final exam materials with anyone in any form or media.*

The above honour code is an undertaking for students to abide by both individually and collectively. You must uphold both the spirit and letter of this honour code. Please sign this honour code and upload it to crowdmark.

Signature: 

Full Name: Anshal Chopra

Student Number: 301384760

Q1 (10 points)

1 page submitted

Question 1

Q1

a)

Java Script Object Notation

b.)

Yes, JSON is considered to be a NoSQL database format. It is a database format where we do not formally arrange things into rows and columns.

c.)

Advantage and Disadvantages of NoSQL

Advantages: NoSQL data model has fewer restrictions for the smallest changes with the data model. It easily allows for relative flexibility like an addition of new columns with no major breakdowns.

Disadvantages: Does not have the reliability that relational databases have and can lead to really messy databases.

d.)

Advantages and Disadvantages of Relational Database Systems (MySQL)

Advantages: The table format can be easily understood by the users, which makes it simpler for them to use it. The data access and data organization are arranged using a natural structure. Matching entries can be located with ease using database queries.

Disadvantages: The expense of maintaining and even setting up a database system is relatively high and one of the drawbacks of relational databases. A special software is required for setting up a relational database and this could cost a fortune. For non-programmers, they would need to implement several products to set up this database. It might not be an easy task to update all the information and finally get the program running.

e.)

file1.json is invalid.

file2.jason is invalid.

file3.jason is valid.

file4.jason is invalid.

Q2 (5 points)

1 page submitted

Question 2

Q2

```
library(rjson)
library(sqldf)
library(dplyr)
library(RSQLite)
library(DBI)
convert = function (infile, outfile, name = "test") {
data = fromJSON(file = infile)
if (! name)
{
   name = "default"
}
rd_data = as.data.frame(data, row.names = data[[1]])
tf_data = as.data.frame(t(rd_data), row.names = 1 : ncol(rd_data) - 1)
result = tf_data[2:nrow(tf_data),]
sqldf(dbname = "outfile.sqlite")
db = dbConnect(SQLite(), dbname = outfile)
dbWriteTable(conn = db,
        name = name,
        value = result,
        row.names = FALSE,
        overwrite = TRUE) }
```

Q3 (5 points)

1 page submitted

Question 3

Q3

The 21st century has seen several breakthroughs in terms of technologies and various industries have benefited from them. To make the most out of these technologies, companies hire new employees with the right skill and motivation to work for them. Now the motivation is much higher when people get to work in the industries they want to work for. For me that industry is the Finance industry. Finance is one of the most critical sectors in the world and is the third largest private sector contributor to the GDP of Canada. Earlier financial management, used to take a lot of time and effort but now, using Data Science, one can quickly and efficiently make better decisions to manage finances. There are a few applications of Data Science in the Finance industry, including but not limited to risk analytics, real-time analytics, customer data management, financial fraud detection and Algorithmic Trading. I will elaborate on Algorithmic Trading since that sparked my interest in the industry the most. One of the essential parts of financial institutions is Algorithmic Trading which is used to compute complex mathematical formulas at lightning speed which helps in formulating new trading strategies by financial institutions. Big Data has wholly revolutionized Data Science and Algorithmic Trading in a huge way. By the understanding of massive datasets in a better way, financial institutions can make better predictions for the future market, and that is the aim of the analytical engine.

Q4 (10 points)

3 pages submitted

Question 4

Untitled1* × Lab4.R ×

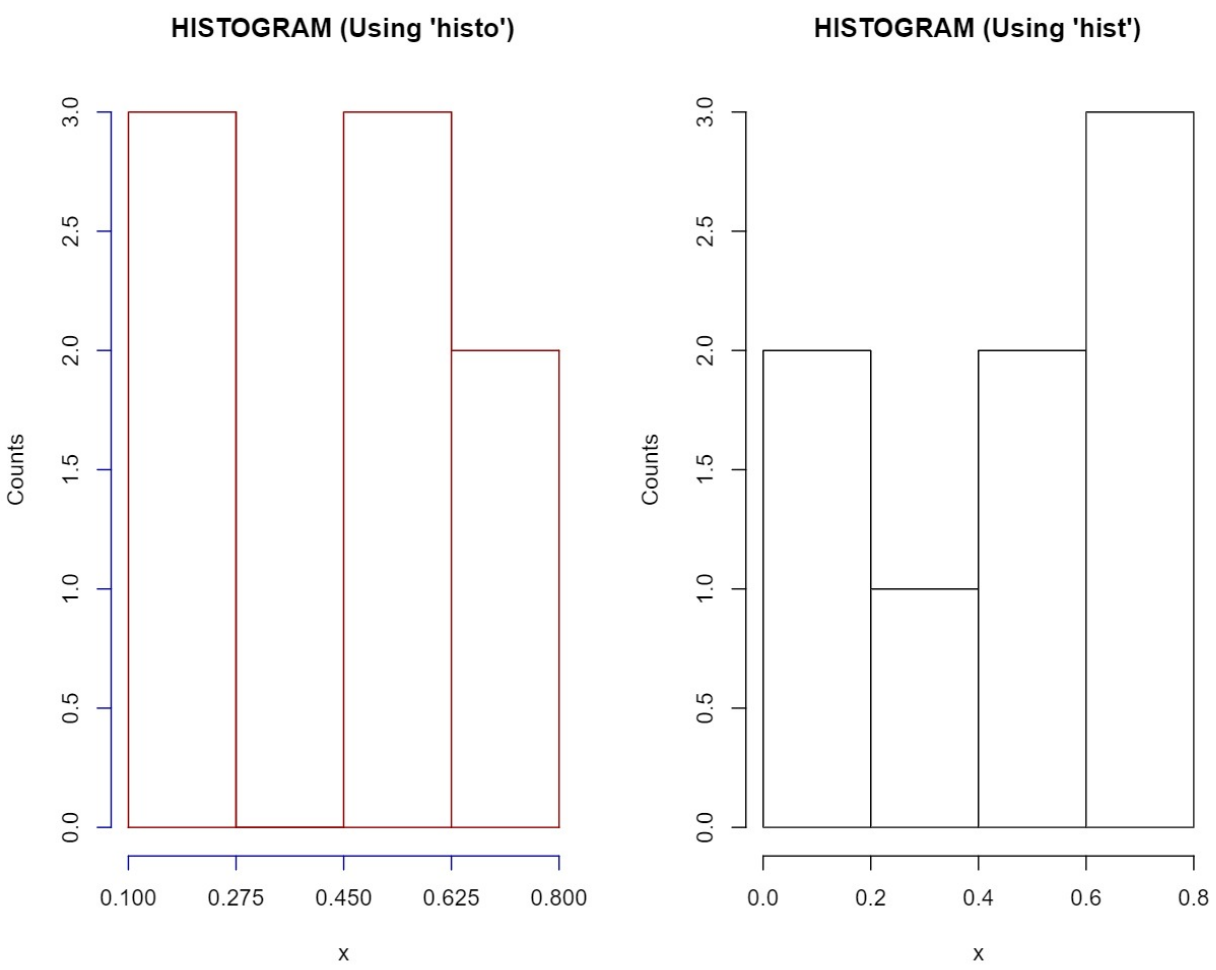
Source on Save | Run | Source

```r


#Anshal Chopra
#301384760

#Q4

#########################################################################################################

x = c(0.2,0.25,0.1,0.7,0.6,0.6,0.61,0.8)
n = 4

#########################################################################################################
counts = function(x, n)
{
  # Creating Intervals
  width = (max(x)-min(x))/n   # width of the bins
  freq = c(min(x))            # first interval([min(x), i))

  for(i in seq_len(n+1))
  {
    freq[i] = min(x) + (i-1)*width
  }

  num = c(0)

  # Number of integers within the intervals
  for(i in 1:n)
  {
    count = 0;
    for(j in seq_along(x))
    {
      if(i<n)
      {
        if(x[j] >= freq[i] && x[j] < freq[i+1])
        {
          count = count+1
        }
      }
      else  # Case when counting the max(x) i.e [i, max(x)]
      {
        if(x[j] >= freq[i] && x[j] <= freq[i+1])
        {
          count = count+1
        }
      }
    }
    num[i+1] = count
  }
  num
}

#########################################################################################################

```

Untitled1* × Lab4.R ×

Source on Save | Run | Source

```
      {
        if(x[j] >= freq[i] && x[j] <= freq[i+1])
        {
          count = count+1
        }
      }
    }
    num[i+1] = count
  }
  num
}

#################################################################################################

histo = function(x, n)
{
  magic = counts(x, n) # calling the function counts

  # making intervals for the x axis
  width = (max(x)-min(x))/n
  freq = c(min(x))
  baseline = c(0)
  for(i in seq_len(n+1))
  {
    freq[i] = min(x) + (i-1)*width
    baseline[i] = 0
  }

  #plotting the intervals and counts
  plot(0:max(magic),xaxt = "n",type = "n"
       , main = "HISTOGRAM (Using 'histo')",xlab = "x"
       , ylab = "Counts"
       , xlim = c(min(x), max(x))
       , mgp = c(3,3,3)
       , axes = FALSE )

  axis(1, at = seq(min(x), max(x), by = width), col = "Dark Blue") #to get the desired bins
  axis(2, at = seq(min(magic), max(magic), by = 0.5), col = "Dark Blue")

  lines(freq, magic, type = "h", col = "Dark Red") #makes vertical lines

  lines(freq, magic, type = "S", col = "Dark Red") #makes stair like lines

  lines(freq, baseline, col = "Dark Red") #makes the base line

}

#################################################################################################

par(mfrow = c(1,2))
histo(x,n)

hist(x,n, main = "HISTOGRAM (Using 'hist')", xlab = "x", ylab = "Counts", col = "White")

#################################################################################################

```



Q5 OA (10 points)

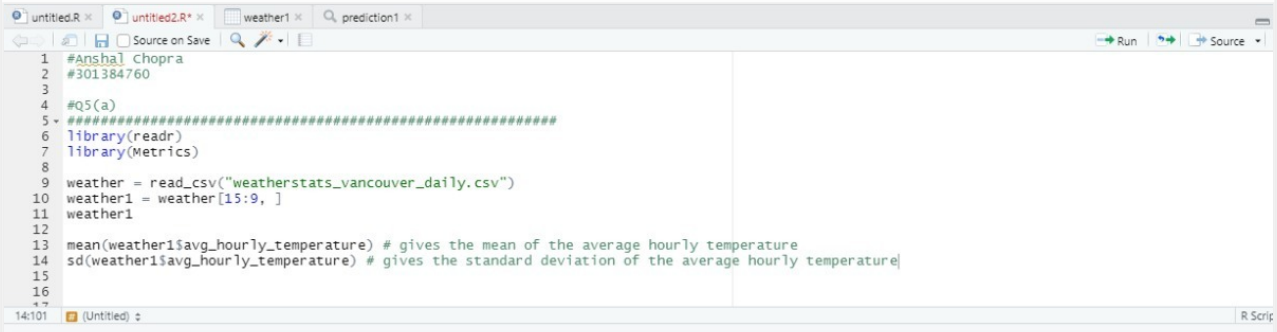
Not submitted

Question 5, Option A

Q5 OB (10 points)

6 pages submitted

Question 5, Option B



```r
#Anshal Chopra
#301384760

#Q5(a)
##########################################################
library(readr)
library(metrics)

weather = read_csv("weatherstats_vancouver_daily.csv")
weather1 = weather[15:9, ]
weather1

mean(weather1$avg_hourly_temperature) # gives the mean of the average hourly temperature
sd(weather1$avg_hourly_temperature) # gives the standard deviation of the average hourly temperature
```

```
  avg_hourly_cloud_cover_4 = col_logical(),
  avg_cloud_cover_4 = col_logical(),
  min_cloud_cover_4 = col_logical(),
  max_cloud_cover_10 = col_logical(),
  avg_hourly_cloud_cover_10 = col_logical(),
  avg_cloud_cover_10 = col_logical(),
  min_cloud_cover_10 = col_logical()
)
i Use `spec()` for the full column specifications.

> weather1 = weather[15:9, ]
> weather1
# A tibble: 7 x 70
  date       max_temperature avg_hourly_temp~ avg_temperature min_temperature max_humidex min_windchill max_relative_hu~ avg_hourly_rela~
  <date>             <dbl>            <dbl>           <dbl>           <dbl> <lgl>               <dbl>            <dbl>            <dbl>
1 2021-04-12          11.9              7.3            5.85            -0.2 NA                  NA                  90             64.4
2 2021-04-13          13.8              9.38           8.45             3.1 NA                  NA                  84             55.4
3 2021-04-14          14.4              9.41           7.9              1.4 NA                  NA                  84             67.8
4 2021-04-15          17.1             11.6           10.6              4   NA                  NA                  93             68.4
5 2021-04-16          17.9             12.2           11.4              4.8 NA                  NA                  92             70.5
6 2021-04-17          18.7             12.7           11.9              5.1 NA                  NA                  98             68.7
7 2021-04-18          18.1             13.0           11.4              4.8 NA                  NA                  97             65.7
# ... with 61 more variables: avg_relative_humidity <dbl>, min_relative_humidity <dbl>, max_dew_point <dbl>, avg_hourly_dew_point <dbl>,
#   avg_dew_point <dbl>, min_dew_point <dbl>, max_wind_speed <dbl>, avg_hourly_wind_speed <dbl>, avg_wind_speed <dbl>, min_wind_speed <dbl>,
#   max_wind_gust <dbl>, wind_gust_dir_10s <dbl>, max_pressure_sea <dbl>, avg_hourly_pressure_sea <dbl>, avg_pressure_sea <dbl>,
#   min_pressure_sea <dbl>, max_pressure_station <dbl>, avg_hourly_pressure_station <dbl>, avg_pressure_station <dbl>, min_pressure_station <dbl>,
#   max_visibility <dbl>, avg_hourly_visibility <dbl>, avg_visibility <dbl>, min_visibility <dbl>, max_health_index <dbl>,
#   avg_hourly_health_index <dbl>, avg_health_index <dbl>, min_health_index <dbl>, heatdegdays <dbl>, cooldegdays <dbl>, growdegdays_5 <dbl>,
#   growdegdays_7 <dbl>, growdegdays_10 <dbl>, precipitation <dbl>, rain <dbl>, snow <dbl>, snow_on_ground <dbl>, sunrise <time>, sunset <time>,
#   daylight <dbl>, sunrise_f <dbl>, sunset_f <dbl>, min_uv_forecast <dbl>, max_uv_forecast <dbl>, min_high_temperature_forecast <dbl>,
#   max_high_temperature_forecast <dbl>, min_low_temperature_forecast <dbl>, max_low_temperature_forecast <dbl>, solar_radiation <lgl>,
#   max_cloud_cover_4 <lgl>, avg_hourly_cloud_cover_4 <lgl>, avg_cloud_cover_4 <lgl>, min_cloud_cover_4 <lgl>, max_cloud_cover_8 <dbl>,
#   avg_hourly_cloud_cover_8 <dbl>, avg_cloud_cover_8 <dbl>, min_cloud_cover_8 <dbl>, max_cloud_cover_10 <lgl>, avg_hourly_cloud_cover_10 <lgl>,
#   avg_cloud_cover_10 <lgl>, min_cloud_cover_10 <lgl>
> mean(weather1$avg_hourly_temperature)
[1] 10.80143
> sd(weather1$avg_hourly_temperature)
[1] 2.134225
```

```r
#Q5(b)
##########################################################
df1 = data.frame(
  x = 1:3,
  y = weather1$avg_hourly_temperature[1:3])

df2 = data.frame(
  x = 1:3,
  y = weather1$avg_hourly_temperature[2:4])

df3 = data.frame(
  x = 1:3,
  y = weather1$avg_hourly_temperature[3:5])

df4 = data.frame(
  x = 1:3,
  y = weather1$avg_hourly_temperature[4:6])

x = 4
pred = c()

model1 = lm(y ~ x, df1)
pred1 = x*model1$coefficients[2] + model1$coefficients[1] #prediction of temperature for 15th April 2021

model2 = lm(y ~ x, df2)
pred2 = x*model2$coefficients[2] + model2$coefficients[1] #prediction of temperature for 16th April 2021

model3 = lm(y ~ x, df3)
pred3 = x*model3$coefficients[2] + model3$coefficients[1] #prediction of temperature for 17th April 2021

model4 = lm(y ~ x, df4)
pred4 = x*model4$coefficients[2] + model4$coefficients[1] #prediction of temperature for 18th April 2021

pred = c(pred1, pred2, pred3, pred4)
prediction1 = list(April15 = pred1, April16 = pred2,
                   April17 = pred3, April18 = pred4)

rmse(weather1$avg_hourly_temperature[4:7], pred) #rmse value for 3-day method

prediction1$April15 #prediction of temperature for 15th April 2021
prediction1$April16 #prediction of temperature for 16th April 2021
prediction1$April17 #prediction of temperature for 17th April 2021
prediction1$April18 #prediction of temperature for 18th April 2021
```

```
Console C:/Users/chopr/OneDrive/Desktop/SPRING 21/STAT 240/
> rmse(weather1$avg_hourly_temperature[4:7], pred) #rmse value for 3-day method
[1] 0.7068946
> prediction1$April15 #prediction of temperature for 15th April 2021
       x
10.80667
> prediction1$April16 #prediction of temperature for 16th April 2021
       x
12.36333
> prediction1$April17 #prediction of temperature for 17th April 2021
       x
13.83667
> prediction1$April18 #prediction of temperature for 18th April 2021
    x
13.28
>
```

untitled.R | untitled2.R* | weather1 | prediction1

Source on Save | Run | Source

```
63
64
65  #Q5(c)
66  ########################################################
67
68  df5 = data.frame(
69    x = 1:2,
70    y = weather1$avg_hourly_temperature[1:2])
71
72
73  df6 = data.frame(
74    x = 1:3,
75    y = weather1$avg_hourly_temperature[1:3])
76
77
78  df7 = data.frame(
79    x = 1:4,
80    y = weather1$avg_hourly_temperature[1:4])
81
82
83  df8 = data.frame(
84    x = 1:5,
85    y = weather1$avg_hourly_temperature[1:5])
86
87
88  predn = c()
89
90  x = 3
91  model5 = lm(y ~ x, df5)
92  pred5 = x*model5$coefficients[2] + model5$coefficients[1] #prediction of temperature for 15th April 2021
93
94  x = 4
95  model6 = lm(y ~ x, df6)
96  pred6 = x*model6$coefficients[2] + model6$coefficients[1] #prediction of temperature for 16th April 2021
97
98  x = 5
99  model7 = lm(y ~ x, df7)
100 pred7 = x*model7$coefficients[2] + model7$coefficients[1] #prediction of temperature for 17th April 2021
101
102 x = 6
103 model8 = lm(y ~ x, df8)
104 pred8 = x*model8$coefficients[2] + model8$coefficients[1] #prediction of temperature for 18th April 2021
105
106 predn = c(pred5, pred6, pred7, pred8)
107 prediction2 = list(April15 = pred5, April16 = pred6,
108                    April17 = pred7, April18 = pred8)
109
110 rmse(weather1$avg_hourly_temperature[4:7], predn) #rmse for martingale assumption
111
112 prediction2$April15 #prediction of temperature for 15th April 2021
113 prediction2$April16 #prediction of temperature for 16th April 2021
114 prediction2$April17 #prediction of temperature for 17th April 2021
115 prediction2$April18 #prediction of temperature for 18th April 2021
116
117 ################################################################################
118
```

```
> rmse(weather1$avg_hourly_temperature[4:7], predn) #rmse for martingale assumption
[1] 0.7464145
> prediction2$April15 #prediction of temperature for 15th April 2021
    x
11.46
> prediction2$April16 #prediction of temperature for 16th April 2021
       x
10.80667
> prediction2$April17 #prediction of temperature for 17th April 2021
     x
12.665
> prediction2$April18 #prediction of temperature for 18th April 2021
      x
13.573
>
```

Q5

d.) Although the RMSE values of both predictions i.e., the 3-day method and the martingale assumption are pretty close which are 0.706 and 0.746 respectively, predictions using the 3 day method are much better over here because the temperatures are changing constantly everyday and we do not have a lot of data for the predictions although if the amount of data is huge and there is a constant change, previous day method or martingale assumption would have been much better.

Back to top