Semester 1 Examinations 2020 / 2021
ONLINE EXAMINATION

Exam Code(s)	1CSD1, 1CSD2
Exam(s)	M.Sc. in Computer Science (Data Analytics)
Module Code(s)	CT5102
Module(s)	Programming for Data Analytics
Paper No.	I
External Examiner(s)	Professor P. L. Lanzi
Internal Examiner(s)	Professor Michael Madden *Prof. Jim Duggan
Instructions:	Answer any 3 questions. All questions carry equal marks. Answers must be hand-written and a pdf of the answers uploaded via Blackboard.
Duration	2 hrs with 30 minutes for uploading at the end of the exam
No. of Pages	7 (including cover page)
Department(s)	School of Computer Science

Disclaimer

This is an "open book" style exam. In submitting this work I confirm that it is entirely my own. I acknowledge that I may be invited to online interview if there is any concern in relation to the integrity of my exam, and I am aware that any breach will be subject to the University's Procedures for dealing with breaches of Exam Regulations: https://www.nuigalway.ie/media/registry/exams/QA230---Procedures-for-Dealing-with-Breaches-of-Examination-Regulations.pdf

By sitting this exam, you agree to the above terms.

1. (a) Consider the following code snippet:

```
library(aimsir17)
library(ggplot2)
library(dplyr)

mean <- function(x)x^2
```

When the code is loaded into R, draw a diagram of the environments in the search path. Explain what the call to mean(1:5) will return, and indicate how to avoid the scenario where the intended target function (in this case the R function mean) is not called.

[5]

(b) Visualise the following code, and show the result of the call to f1(). Explain the mechanism by which the value is calculated.

```
f1 <- function (x,y){
  function (z){
    x+y+z
  }
}

y <- f1(2,3)(4)
```

[8]

(c) Implement a closure that acts as a stopwatch.

Use the function Sys.time() to record the time, as this function returns an object that supports date arithmetic.

Create a stopwatch variable as follows

```
st = stopwatch()
```

Function Name	**Details**
start()	Records the start time
stop()	Records the finish time
get_duration()	Returns the time elapsed between the start and stop time

Visualise the resulting closure state after calls to start() and stop().

[12]

2. (a) The following are two tables (t1, t2).

```
> t1
# A tibble: 5 x 2
  StudentID Name
      <int> <chr>
1         1 AA
2         2 BB
3         3 CC
4         4 DD
5         5 EE
```

```
> t2
# A tibble: 5 x 3
  StudentID Subject Grade
      <int> <chr>   <int>
1         1 CX101      63
2         3 CX101      91
3         1 CX103      77
4         3 CX101      87
5         3 CX102      83
```

Show what **dplyr** functions can be used to create the following results, and explain how the process works.

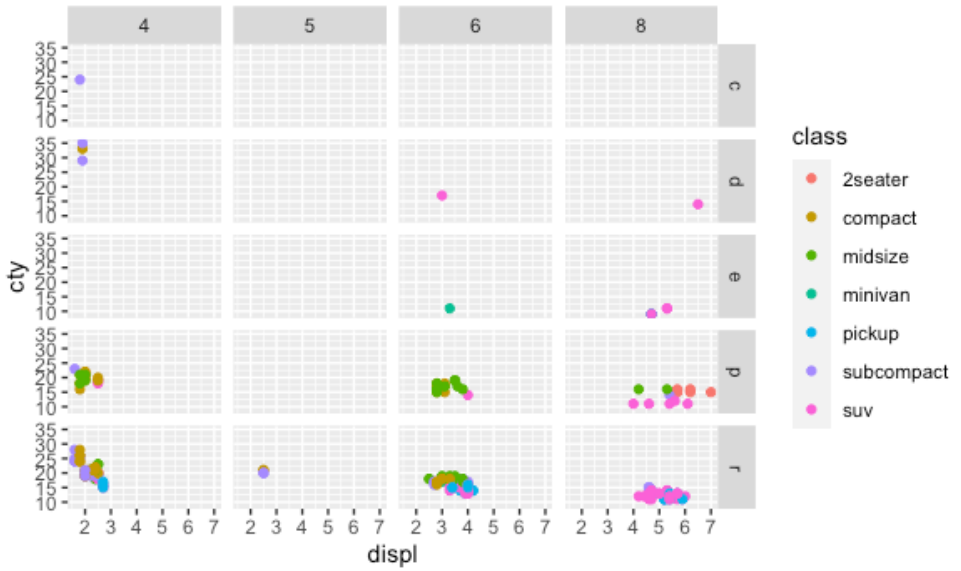
```
> r1
# A tibble: 2 x 2
  StudentID Name
      <int> <chr>
1         1 AA
2         3 CC
```

```
> r2
# A tibble: 3 x 2
  StudentID Name
      <int> <chr>
1         2 BB
2         4 DD
3         5 EE
```

[6]

(b) Summarise the main idea behind exploratory data analysis.

Based on the mpg data set, show the code that generates the following plot. The variables used include class, fl, cyl, cty and displ.




```
> colnames(mpg)
 [1] "manufacturer" "model"        "displ"        "year"
 [5] "cyl"          "trans"        "drv"          "cty"
 [9] "hwy"          "fl"           "class"
```

[6]

(c) Based on the observations tibble in the **aimsir17** data set, perform the following data science pipeline activity to model the effect of the minimum daily sea level pressure on the total daily rainfall.

```
> colnames(observations)
 [1] "station" "year"    "month"   "day"     "hour"
 [6] "date"    "rain"    "temp"    "rhum"    "msl"
[11] "wdsp"    "wddir"
```

(1) For the three stations, create a grouped tibble that contains the summarised data of total daily rainfall and minimum msl for each station.

```
> tb
# A tibble: 3 x 2
# Groups:   station [3]
  station        data
  <chr>          <list>
1 BELMULLET      <tibble [365 × 3]>
2 DUBLIN AIRPORT <tibble [365 × 3]>
3 ROCHES POINT   <tibble [365 × 3]>

pluck(pull(tb,data),3) %>% slice(1:3)
# A tibble: 3 x 4
    day month daily_rain min_pressure
  <int> <dbl>      <dbl>        <dbl>
1     1     1        3.2        1019.
2     1     2        7.2         984.
3     1     3        4           996
```

(2) Use the purrr library to create a linear model for each weather station, where the independent variable is min_pressure.

```
> tb
# A tibble: 3 x 3
# Groups:   station [3]
  station        data              LM
  <chr>          <list>            <list>
1 BELMULLET      <tibble [365 × 4]> <lm>
2 DUBLIN AIRPORT <tibble [365 × 4]> <lm>
3 ROCHES POINT   <tibble [365 × 4]> <lm>
```

(3) Add a new column (descending order) that captures the R Squared value from each linear model.

```
> tb
# A tibble: 3 x 4
# Groups:   station [3]
  station        data              LM      R.Sq
  <chr>          <list>            <list> <dbl>
1 ROCHES POINT   <tibble [365 × 4]> <lm>   0.202
2 BELMULLET      <tibble [365 × 4]> <lm>   0.182
3 DUBLIN AIRPORT <tibble [365 × 4]> <lm>   0.11
```

[13]

(3) (a) Given the following vector x, show how the names can be added to the vector using the **attr()** function, and using the **structure()** function.

```
> x
[1] 1 2 3 4 5
```

```
> x
a b c d e
1 2 3 4 5
```

[2]

(b) Distinguish between base objects and OO objects in R. For a data frame object such as **datasets::mtcars**, what is (1) its base type and (2) its S3 class?

Show what functions you could use to gather this information.

[3]

(c) Use the code example below to clearly explain how the S3 object system works. Use diagrams where appropriate to clarify your explanation.

```
> y1 <- 1:5
>
> y1
[1] 1 2 3 4 5
>
> summary(y1)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      1       2       3       3       4       5
```

```
> y2 <- as.factor(sample(c("A","B"),5,repl=T))
>
> y2
[1] B A A B B
Levels: A B
>
> summary(y2)
A B
2 3
```

[10]

(d) Use the S3 system to create a class called “my_df”. A constructor should create the class as follows:

```
> d <- my_df(mtcars)
>
> class(d)
[1] "my_df"      "data.frame"
>
> summary(d)
The column names are  mpg cyl disp hp drat wt qsec vs am gear
carb
The number of rows are  32
Here is a summary of the columns
      mpg             cyl             disp             hp
 Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0
 1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5
 Median :19.20   Median :6.000   Median :196.3   Median :123.0
 Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7
 3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0
 Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0
     drat             wt             qsec             vs
 Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000
 1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000
 Median :3.695   Median :3.325   Median :17.71   Median :0.0000
 Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375
 3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000
 Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000
       am              gear            carb
 Min.   :0.0000   Min.   :3.000   Min.   :1.000
 1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000
 Median :0.0000   Median :4.000   Median :2.000
 Mean   :0.4062   Mean   :3.688   Mean   :2.812
 3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000
 Max.   :1.0000   Max.   :5.000   Max.   :8.000
```

[10]

4. (a) When the following R code gets executed, determine the final values for A and B. Explain what determines the values of A and B.

```
A <- 100; B <- 20

f1<-function(a){
  B <- 100
  f2<-function(b){
    A <<- 200+b
    B <<- 1000-b
  }
  f2(a)
}

f1(B)
```

[5]

(b) For the list **`l <- list(c(T,F),1:5,list(1:2,6:7))`**

- Visualise the list
- Explain the difference between **`l[3]`** and **`l[[3]]`**
- Visualise the results of the following, and clearly show their type.
  - **`l[1]`**
  - **`l[[2]]`**
  - **`l[[2]][1]`**

[5]

(c) Write a factory function power(x) that returns a function that will raise an input number to that power. Visualise this function.

```
> p2 <- pow(2)
>
> is.function(p2)
[1] TRUE
>
> p2(5)
[1] 25
```

[5]

(d) Consider the formula $f(x) = ax^3+bx^2+c$x + d

Use **sapply()** to transform an input vector in the range [-100,+100] using this formula, where the parameters a, b, c and d are provided as additional inputs to the transformation.

Use the corresponding function from purrr to generate the same answer, using the shortcut notation of purrr.

[5]