Ollscoil na Gaillimhe
University of Galway

Semester 1 Examinations 2022 / 2023

Course Instance 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)	Dr. John Woodward
Internal Examiner(s)	Professor Michael Madden *Prof. Jim Duggan
Instructions:	Answer any 3 questions. All questions carry equal marks.
Duration	2 hrs
No. of Pages	6 (including cover page)
Department(s)	School of Computer Science
Course Co-ordinator	Dr. Frank Glavin

Requirements	Release in Exam Venue	Yes [X] No [ ]
	MCQ Answersheet	Yes [ ] No [X]
	Handout	None
	Statistical/ Log Tables	None
	Cambridge Tables	None
	Graph Paper	None
	Log Graph Paper	None
	Other Materials	None
	Graphic material in colour	Yes [X] No [ ]

(1) (a) Describe the difference between an atomic vector and a list. Show how you could filter a list to return every second element.

[3]

(b) Predict the data types and values of the following atomic vectors, and explain your answers.

```
(1)     c(10, 20, TRUE, 123.45)
(2)     c(T,T,F,0)
(3)     unlist(list(10, 20, TRUE, "TRUE"))
```

[3]

(c) Describe the workings of this function, and explain how each line of code contributes to the output. What will the output data type of this function be?

```
my_func <- function(x, f, ...) {
  out <- vector(mode = "list", length = length(x))
  for (i in seq_along(x)) {
    out[[i]] <- f(x[[i]], ...)
  }
  unlist(out)
}
```

[6]

(d) Consider the following list:

```
l <- list(el1=1:3, el2="Test", el3=list(n1=10, n2=2:5))
```

Visualise the outputs from the following commands, and explain each solution.

```
l[3]
l[1:2]
l[[1]]
l[[3]][[2]][3]
```

[6]

(e) Consider the following code.

```
x <- 10
y <- 1

f <- function(a,b){
  x <- 200
  c(First=a+b+x,
    Second=a+b+y)
}
```

What will the following function call return. Explain your answer.

f(1,2)

[7]

2. (a) Consider the following two tibbles (acc, tx).

```
> acc
# A tibble: 3 × 2
  Account Balance
  <fct>     <dbl>
1 12345       100
2 45678       300
3 67891       400
```

```
> tx
# A tibble: 4 × 4
     TX Account Type   Amount
  <int> <fct>   <chr>   <dbl>
1     1 12345   Debit     100
2     2 12345   Credit    300
3     3 67891   Credit    100
4     4 12345   Credit     50
```

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

```
> r1
# A tibble: 4 × 5
  Account Balance    TX Type   Amount
  <fct>     <dbl> <int> <chr>   <dbl>
  12345       100     1 Debit     100
  12345       100     2 Credit    300
  12345       100     4 Credit     50
  67891       400     3 Credit    100
```

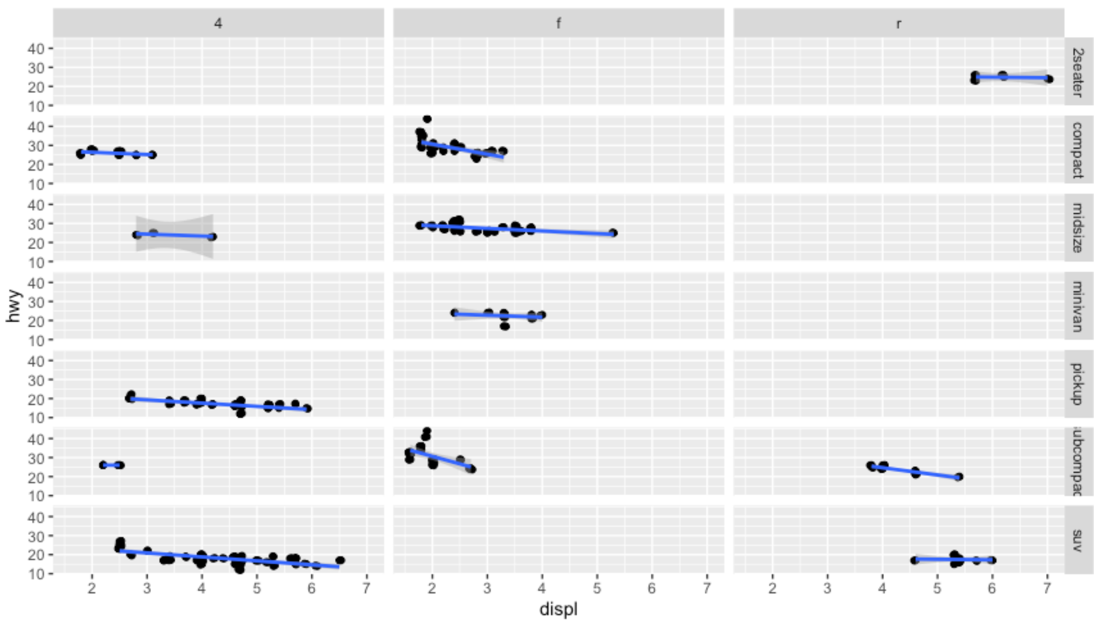
```
> r2
# A tibble: 1 × 2
  Account Balance
  <fct>     <dbl>
  45678       300
```

[5]

(b) Briefly list the main ideas behind exploratory data analysis.

Based on the mpg tibble covered in the lectures, show the code that generates the following plot. The variables used include displ, hwy, class (rows) and drv (columns). On the x-axis is the variable displ, and the y axis contains the variable hwy.

[8]



(c) Perform the following analysis, based on the tibble flights. Make use of any pipe operator for the calculations.

```
> slice(flights,1:4)
# A tibble: 4 × 19
   year month   day dep_time sched_de…¹ dep_d…² arr_t…³
  <int> <int> <int>    <int>      <int>   <dbl>   <int>
   2013     1     1      517        515       2     830
   2013     1     1      533        529       4     850
   2013     1     1      542        540       2     923
   2013     1     1      544        545      -1    1004
# … with 12 more variables: sched_arr_time <int>,
#   arr_delay <dbl>, carrier <chr>, flight <int>,
#   tailnum <chr>, origin <chr>, dest <chr>,
#   air_time <dbl>, distance <dbl>, hour <dbl>,
#   minute <dbl>, time_hour <dttm>, and abbreviated
#   variable names ¹sched_dep_time, ²dep_delay,
#   ³arr_time
```

a) Calculate the average departure delay for each airport

```
# A tibble: 3 × 2
  origin AvrDepDelay
  <chr>        <dbl>
  EWR           15.1
  JFK           12.1
  LGA           10.3
```

b) Calculate the evening breakdown of departure delays in all three airport, which include the hours from

```
# A tibble: 3 × 4
  origin Q05Distance MedianDistance Q95Distance
  <chr>        <dbl>          <dbl>       <dbl>
1 EWR            200            872        2454
2 JFK            187           1069        2586
3 LGA            198            762        1416
```

c) Calculate the following summaries for the departure delays for the hours {18, 19, 20 and 21}

```
# A tibble: 12 × 4
# Groups:   origin [3]
   origin  hour MeanDelay SDDelay
   <chr>  <dbl>     <dbl>   <dbl>
 1 EWR       19      31.1    53.2
 2 EWR       20      27.5    49.1
 3 EWR       21      26.1    44.8
 4 JFK       21      26.1    48.6
 5 EWR       18      25.1    50.1
 6 LGA       20      22.5    51.8
 7 LGA       19      22.4    54.2
 8 JFK       19      22.3    51.0
 9 JFK       20      21.7    45.9
10 LGA       18      20.1    51.1
11 LGA       21      19.2    46.8
12 JFK       18      18.4    46.8
```

[12]

3. (a) Describe the key differences between the S3 class system and message-passing OO systems such as Java and C++. Show an example of S3 using the generic function summary() from base R.

[5]

(b) Write a constructor function called new_account(), which creates an S3 object ("account" class), as follows.

```
a1 <- new_account(1234,200)
str(a1)
List of 2
 $ number : num 1234
 $ balance: num 200
 - attr(*, "class")= chr "account"
```

Next, create generic functions and associated methods that will implement the following calls.

```
(1) a1 <- credit(a1, 100)
    str(a1)
    List of 2
     $ number : num 1234
     $ balance: num 300
      - attr(*, "class")= chr "account"

(2) a1 <- debit(a1, 99)
    str(a1)
    List of 2
     $ number : num 1234
     $ balance: num 201
     - attr(*, "class")= chr "account"
```

[15]

(c) Implement a method that will achieve the following result.

```
a1
Acc# = 1234   Balance = 201.01
```

[5]

4. (a) Describe the following functions from the package purrr.

- **map**(.x, .f)
- **map_df**(.x, .f)

[4]

(b) Describe the two ways of defining an anonymous function using purrr.

Show how the tilde-dot shorthand notation can be used to generate values for the equation $\mathbf{y = 4x^3 - 3x^2 - 5x + 10}$, assuming an input range of [-100, +100], in steps of 0.1

[4]

(c) Use the functions group_split() and map_df() to generate the following tibble showing the total rainfall for the year. Note that the summarise() function cannot be used.

```
# A tibble: 5 × 2
  Station                TotalRain
  <chr>                      <dbl>
  NEWPORT                    1752.
  VALENTIA OBSERVATORY       1598.
  KNOCK AIRPORT              1343.
  BELMULLET                  1243.
  FINNER                     1222.
```

[7]

(d) Based on the flights tibble, create the following nested tibble

```
  origin data
  <chr>  <list>
1 EWR    <tibble [120,835 × 18]>
2 LGA    <tibble [104,662 × 18]>
3 JFK    <tibble [111,279 × 18]>
```

Based on this nested tibble, add two new columns that show the slope and the intercept of a linear model lm(arr_delay~dep_delay), where the intercept is coef(mod)[1] and the slope is ceof(mod)[2].

```
  origin data                    Intercept Slope
  <chr>  <list>                      <dbl> <dbl>
  EWR    <tibble [120,835 × 18]>      7.37 0.839
  LGA    <tibble [104,662 × 18]>      5.48 0.831
  JFK    <tibble [111,279 × 18]>      7.63 0.791
```

[10]