UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

WST 212 – Semester Test 3 (60 marks)

Instructions:

- All your code should be saved in a single **R** script, named ST3.R
- Both submissions are due on the **4th of June at 12:00 PM.**
- Submission 1 consists of 18 marks, while Submission 2 consists of 42 marks.

Submission 1: Code 💻

- Unlimited code submissions are allowed for the semester test
- The autograder consists of both **visible** and **hidden** tests. Results for visible tests will be available shortly after submission. Results for hidden tests will be available after publishing your marks.
- **Ensure all variables are named correctly, as incorrectly named variables will not be awarded any marks.** (Remember variable names are case sensitive.)
- **Ensure your code does not consist of any syntax errors. If your code produces errors when run, the autograder will not be able to mark it.**
- Any code commented out (code is commented out when # is typed in front of it) will be considered rough work and will not be marked.
- Once you have completed your submission, submit your script with the correct filename onto **Gradescope.**

Submission 2: Theory/Interpretation 📖

- **After completing the code submission**, attempt the Theory/Interpretation submission.
- Only **one attempt** is allowed, which must be completed in one sitting. Even if you close the browser, the timer will continue counting down.
- You will have **60 minutes** to complete this submission.
- If you accidentally submit your test but you still have time left, simply click on the resubmit button to continue. The resubmit button will disappear when the timer runs out.

Guidelines:

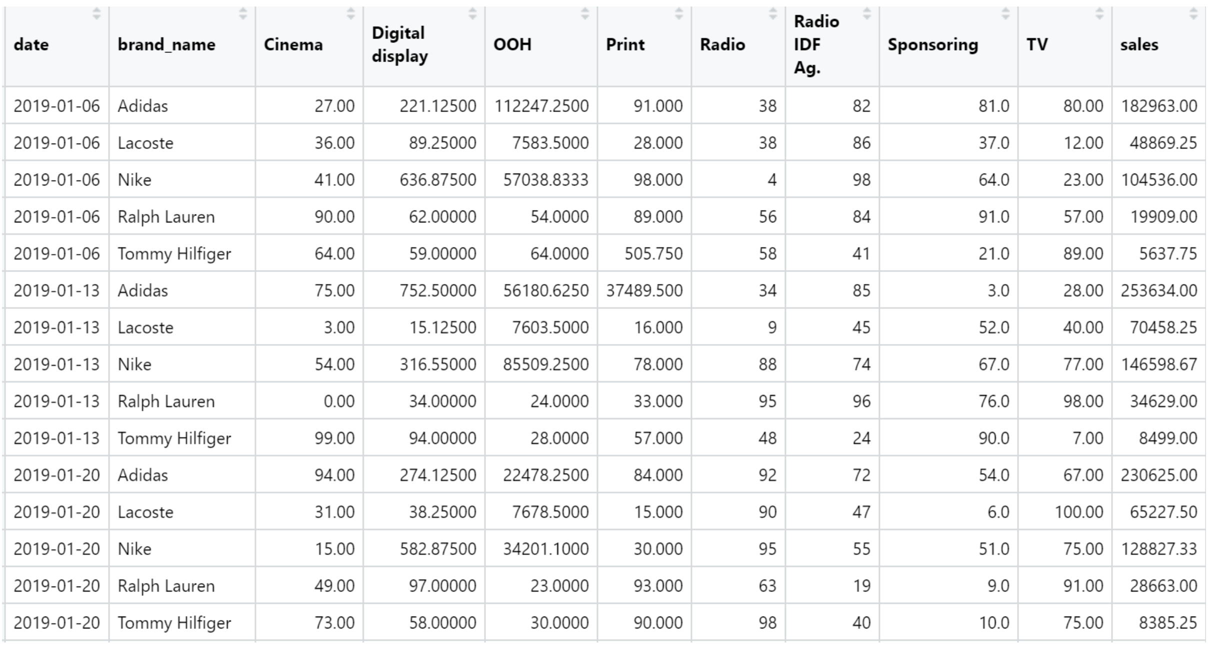
- A template, where students should fill their code in, has been provided on ClickUP. **Do not make any changes to the code that has been provided in the template.**
- For this Semester Test, seven packages are required and loaded in the template, while the necessary data has also been provided within the template.

Questions:

Question 1 (24 marks):

As a recent hire at LEGO Analytics, you've been tasked with some data wrangling for a marketing analytics project. You have been provided with a dataset of weekly spend volumes from 2019 to 2021. As well as weekly sales volumes also for the year 2019 to 2020. Your task is to put together a dataset that contains media spend per channel and append the resulting sales per week for each of the five brands (Nike, Adidas, Lacoste, Ralph Lauren and Tommy Hilfiger).

The resulting dataset should look something like this.



date	brand_name	Cinema	Digital display	OOH	Print	Radio	Radio IDF Ag.	Sponsoring	TV	sales
2019-01-06	Adidas	27.00	221.12500	112247.2500	91.000	38	82	81.0	80.00	182963.00
2019-01-06	Lacoste	36.00	89.25000	7583.5000	28.000	38	86	37.0	12.00	48869.25
2019-01-06	Nike	41.00	636.87500	57038.8333	98.000	4	98	64.0	23.00	104536.00
2019-01-06	Ralph Lauren	90.00	62.00000	54.0000	89.000	56	84	91.0	57.00	19909.00
2019-01-06	Tommy Hilfiger	64.00	59.00000	64.0000	505.750	58	41	21.0	89.00	5637.75
2019-01-13	Adidas	75.00	752.50000	56180.6250	37489.500	34	85	3.0	28.00	253634.00
2019-01-13	Lacoste	3.00	15.12500	7603.5000	16.000	9	45	52.0	40.00	70458.25
2019-01-13	Nike	54.00	316.55000	85509.2500	78.000	88	74	67.0	77.00	146598.67
2019-01-13	Ralph Lauren	0.00	34.00000	24.0000	33.000	95	96	76.0	98.00	34629.00
2019-01-13	Tommy Hilfiger	99.00	94.00000	28.0000	57.000	48	24	90.0	7.00	8499.00
2019-01-20	Adidas	94.00	274.12500	22478.2500	84.000	92	72	54.0	67.00	230625.00
2019-01-20	Lacoste	31.00	38.25000	7678.5000	15.000	90	47	6.0	100.00	65227.50
2019-01-20	Nike	15.00	582.87500	34201.1000	30.000	95	55	51.0	75.00	128827.33
2019-01-20	Ralph Lauren	49.00	97.00000	23.0000	93.000	63	19	9.0	91.00	28663.00
2019-01-20	Tommy Hilfiger	73.00	58.00000	30.0000	90.000	98	40	10.0	75.00	8385.25

In order to track your progress in the project, your manager has decided to break your tasks into subtasks, where you're required to write single functions that perform a specific task. All functions should take **df** as an input. These functions can then be used as verbs to achieve the final result.

e.g `result <- spend %>% round_up_dates`

For all functions, please keep the Function definitions the same as what you have in the template. Only fill in the function body.

Tasks:

a) Amongst other issues with the spend and sales data. You will find that some weeks may fall on different days when recorded. Write a function to round up all dates to the nearest week. This function will need to be applied to both the sales volume data and the spend volumes data in 1f for an inner join to be possible. [4 marks]

- **Function definition**

```
round_up <- function(df){

  }
```

b) You'll also see that there are some missing sales volumes. Write a function to impute any missing value with the average sales of that brand across the entire 2019 – 2021 interval. [5 marks]

- **Function definition**

```
impute <- function(df){

  }
```

c) It appears some brand names are in the sales volumes data are not capitalised as in the spend volumes data. Write a function to ensure brand names in the sales volumes are similar to those in the spend volumes data. [4 marks]

- **Function definition**

```
fix_brand_names <- function(df){

  }
```

Hint: For this function if using dplyr::case_when, omit the alternative statement `TRUE ~ SOME_Value`.

d) To remove unused brands and improve our data processing speeds. Write a function to filter out any brand from the spend volumes dataset other than Nike, Adidas, Lacoste, Ralph Lauren and Tommy Hilfiger. [2 marks]

- **Function definition**

```
filter_brands <- function(df){

  }
```

e) Some measures of spend volumes were recorded twice. In such a case, we need to average over the two or more values for that week. Write a function to average out spend volumes by per date, per media channel and per brand. [4 marks]

- **Function definition**

```
aggregate_spend <- function(df){

  }
```

Finally use all the functions you've created above and other necessary dplyr/tidyr functions to bring it all together and create the final dataset. Assign this object to Q1. [5 marks]

Question 2 (6 marks):

As your second project in LEGO Analytics. You've been given a task to extract all clothing items as well as their new and old prices from their website. The html webpage is called sale_website and can be read into R using the readr package:

```
page <- readr::read_file('sale_website')
```

To view the file you can just click on it in your r-studio file pane. It should open in R-studio.

Your team-lead has broken the task into 3 functions which require you to make use of the stringr package (namely: **`extract_items`**, **`extract_newprices`**, **`extract_oldprices`**). Be sure to set `simplify = TRUE` inside your stringr functions. eg. str_extract(obj, pattern, simplify = TRUE).

Tasks

a) Complete the function below that takes in the page as an input and returns the clothing items as characters. [2 marks]

- **Function definition**

```
extract_items <- function(page){

  }
```

- **Expected output**

```
extract_items(page)
```

```
## [1] "Hoodie" "Shorts" "Dress" "Shirt"
```

b) Complete the function below that takes in the page as an input and ouputs all the new prices as characters. [2 marks]

- **Function definition**

```
extract_newprices <- function(page){

}
```

- **Expected output**

```
extract_newprices(page)
## [1] "R50" "R20" "R40" "R80"
```

c) Complete the function below that takes in the page as an input and ouputs all the old prices as characters. [2 marks]

- **Function definition**

```
extract_oldprices <- function(page){

}
```

- **Expected output**

```
extract_oldprices(page)
## [1] "R60" "R40" "R70" "R100"
```

Question 3 (12 marks):

Load the **BreastCancer** train and test data named **trainSet.csv** and **testSet.csv**:

```
trainSet <- read.csv("trainSet.csv")
testSet <- read.csv("testSet.csv")
```

a) Fit a K-means model with 2-clusters on the training set on all but the diagnosis variable. Assign this fitted model to **q3a** [2 marks]

b) Fit a logistic regression model on the training set (all variables) with diagnosis as a target variable. Assign this fitted model to **q3b** [2 marks]

c) Use the fitted logistic regression model to make predictions on the **testSet**. Assign this fitted model to **q3c** [1 marks]

d) Use **`ifelse`** to assign Benign to a predicted 0 and Malignant to a predicted 1. Assign this fitted model to **q3d** [1 marks]

e) Write a function to make predictions on your test set from the k-means model fit. Assign this function to **predict.kmeans** [4 marks]

- **Function definition**

```
predict.kmeans<- function(newdata, object){

  }
```

f) Use your function to make predictions on your test set. And rename the 1 to Benign and 2 to Malignant. Assign these predictions to **q3f**. [2 marks]

... END