REPORT ON SR. DATA ANALYST CODE CHALLNEGE QUESTIONS

SAS and R programing are used to solve the four questions in this exercise. Proc SQL, a popular SQL procedure in SAS is used to solve question 1. SAS Macro procedure along with Proc SQL are used to solve question 2. R programming is used to solve question 3 and 4.

Question One: Data Access

An often-encountered step to data pre-processing usually involves querying data that exists within a relational database system. Typically, such data may be transactional in nature, and as a result, it is optimized for a relational data model. While this format is critical for performance, efficient storage, and quick updates, it is often not suitable for data science work. In this next question, you will be asked to access data from a fictitious relational database and perform query operations to get it into a suitable format for analysis. The required output for this question is the correct SQL code and query results as a markdown table, you may use any dialect of SQL you wish, but submissions in any other language will be considered incorrect.

You have been granted access to a database that contains two tables; Order_Detail and Product_Detail that are stored in a relational format.

Table Order_Detail Schema

invoice_id	invoice_line	store_id	time_stamp	product	units	sales	cogs
10000001	31215	3	9/3/17	3000	1	99.99	58.00
10000001	31216	3	9/3/17	3354H	3	32.99	18.00
10000002	91455	1	9/5/17	1234	2	24.99	20.00
10000003	59943	2	9/5/17	3000	1	19.99	14.99
10000004	95027	2	9/5/17	18518	1	9.99	3.00
10000005	73994	2	9/5/17	12HA12	15	3.99	1.99
10000006	98464	1	10/1/17	wh30000	1	199.99	75.00

Table Product_Detail Schema

product	cat_name	key_internal
3000	WidgetA	17375273
3354H	WidgetB	15477787
1234	WidgetC	97845235
18518	WidgetD	47813334
12HA12	WidgetD	48573988
wh30000	WidgetE	00000000

For Question 1, the data tables Order_Detail and Product_Detail are saved manually as an excel file first and then imported to SAS by using the procedure “Proc Import”.

```
libname hammad 'C:/Users/Hammad/Desktop/New Folder/New Folder';
proc import datafile="C:\Users\Hammad\Desktop\New Folder\New
Folder\Order_Detail.xlsx" dbms= xlsx out=Order_Detail;
run;
proc import datafile="C:\Users\Hammad\Desktop\New Folder\New
Folder\Product_Detail.xlsx" dbms=xlsx out=Product_Detail;
run;
```

1a- With the Order_Detail table, create a table that summarizes the total sales, total quantity, total profit per store. Note that profit can be calculated as sales - cost. Lastly, order the stores by profitability with the most profitable store at the top.

Proc Sql, a procedure in SAS to write query is used to get total sales, total quantity and total profit by each store. Proc Print, a procedure in SAS to print or generate reports is used to get the desired table.

```
Proc Sql;
create Table Order1 as
select store_id, sum(sales) 'Total Sales/Store', sum(units) 'Total
Quantity/Store', sum(sales) - sum(cogs) as TotalProfit
from Order_Detail
Group by Store_id
Order by TotalProfit desc;

Proc print data = Order1 label;
Label store_id = 'Store ID' _TEMA001 = 'Total Sales/Store' TotalProfit =
'Total Profit';
run;
```

Obs	Store ID	Total Sales/Store	Total Quantity/Store	Total Profit
1	1	224.98	3	129.98
2	3	132.98	4	56.98
3	2	33.97	17	13.99

Store 1 leading the sales and profit followed by store 3 and store 2. It will be interesting to see if we shift some inventory from store 2 to store 1 and 3 to see if the sales of shifted inventory are increasing due to good sales performance of store 1 and 3.

1b- Building on your query above, create a similar table that summarizes sales, quantity, and profit per store, per product (ie, sales for each product_id sold in each store_id)

```
Proc Sql;
create Table Order2 as
select store_id, product, sum(sales) 'Total Sales/Store/Product' , sum(units)
'Total Quantity/Store/Product', sum(sales) - sum(cogs) as TotalProfit
from Order_Detail
Group by Store_id, product
Order by TotalProfit desc;
```

```
Proc print data = Order2 label;
Label store_id = 'Store ID' product = 'Product' _TEMA001 = 'Total
Sales/Store/Product' TotalProfit = 'Total Profit';
run;
```

Obs	Store ID	Product	Total Sales/Store/Product	Total Quantity/Store/Product	Total Profit
1	1	wh30000	199.99	1	124.99
2	3	3000	99.99	1	41.99
3	3	3354H	32.99	3	14.99
4	2	18518	9.99	1	6.99
5	2	3000	19.99	1	5.00
6	1	1234	24.99	2	4.99
7	2	12HA12	3.99	15	2.00

Product “wh30000” is a high cost high markup product and is leading the sales and profit per store per product.

1c- Now, create a similar table that summarizes the same metrics (sales, quantity, profit) per store, per product, per week day (ie, products sold for each day of the week, for each store)

```
Proc Sql;
create Table Order3 as
select store_id, product,time_stamp,WeekDay(time_stamp) as Weekday,
sum(sales) 'Total Sales/Store/Product/day' , sum(units) 'Total
Quantity/Store/Product/day', sum(sales) - sum(cogs) as TotalProfit
from Order_Detail
Group by Store_id, product, WeekDay
Order by TotalProfit desc;

Proc print data = Order3 label;
Label store_id = 'Store ID' product = 'Product' time_stamp = 'Time Stamp'
_TEMA001 = 'Total Sales/Store/Product/day' TotalProfit = 'Total Profit';
run;
```

Obs	Store ID	Product	Time Stamp	Weekday	Total Sales/Store/ Product/day	Total Quantity/Store/ Product/day	Total Profit
1	1	wh30000	10/01/2017	1	199.99	1	124.99
2	3	3000	09/03/2017	1	99.99	1	41.99
3	3	3354H	09/03/2017	1	32.99	3	14.99

Obs	Store ID	Product	Time Stamp	Weekday	Total Sales/Store/ Product/day	Total Quantity/Store/ Product/day	Total Profit
4	2	18518	09/05/2017	3	9.99	1	6.99
5	2	3000	09/05/2017	3	19.99	1	5.00
6	1	1234	09/05/2017	3	24.99	2	4.99
7	2	12HA12	09/05/2017	3	3.99	15	2.00

High cost high markup products such as “wh30000” are high on Sunday.

1d- Your task is to construct a single SQL query that returns the following results:

Summarized total sales, total quantity sold, and total profit (which can be calculated as total sales less cogs) by the week number, store id, product category name. It is important to note that in this business, the week begins on a Tuesday.

```
Proc Sql;
create Table Order4 as
select Week(time_stamp) as weeknum, Store_id, cat_name 'Product Category',
sum(sales) 'Total Sales/weeknum/StoreID/Prod-Cat' , sum(units) 'Total
Quantity/weeknum/StoreID/Prod-Cat', sum(sales) - sum(cogs) as TotalProfit
from Order_Detail as o
inner join Product_Detail as p
on o.product = p.product
Group by Week(time_stamp), Store_id, o.product;

Proc print data = Order4 label;
Label weeknum = 'Week Number' store_id = 'Store ID' cat_name = 'Product
Category'  _TEMA001 = 'Total Sales/Weeknum/StoreID/Prod-Cat' TotalProfit =
'Total Profit';
run;
```

Obs	Week Number	StoreID	Product Category	Total Sales/Weeknum/ StoreID/Prod-Cat	Total Quantity/ weeknum/ StoreID/Prod-Cat	Total Profit
1	36	1	WidgetC	24.99	2	4.99
2	40	1	WidgetE	199.99	1	124.99
3	36	2	WidgetD	3.99	15	2.00

Obs	Week Number	StoreID	Product Category	Total Sales/Weeknum/ StoreID/Prod-Cat	Total Quantity/ weeknum/ StoreID/Prod-Cat	Total Profit
4	36	2	WidgetD	9.99	1	6.99
5	36	2	WidgetA	19.99	1	5.00
6	36	3	WidgetA	99.99	1	41.99
7	36	3	WidgetB	32.99	3	14.99

1e- There was an error in one of the records in the database... can you find it?

invoice_id	invoice_line	store_id	time_stamp	product	units	sales	cogs
10000001	31215	3	9/3/17	3000	1	99.99	58.00
10000001	31216	3	9/3/17	3354H	3	32.99	18.00
10000002	91455	1	9/5/17	1234	2	24.99	20.00
10000003	59943	2	9/5/17	3000	1	19.99	14.99
10000004	95027	2	9/5/17	18518	1	9.99	3.00
10000005	73994	2	9/5/17	12HA12	15	3.99	1.99
10000006	98464	1	10/1/17	wh30000	1	199.99	75.00

Product ID 3000 is not unique for Product in Table Order_Detail. The same product has two different prices which confirms product ID entry error.

Recommendations from question 1:

- Store 2 sales need to be improved
- Consumer are likely to buy expensive products (i.e., Product Category E) on Sunday
- Unit of analysis by week number does not give any new insight

Question Two: Data Transformation

Another common task is to take data that may not be in a usable format and 'wrangle' it into a better representation. This next question will test your ability to clean and order data.

You have been presented with two tables:

Table A: Product Attributes

This table contains two columns; the first one is a unique product ID represented by an integer, the second is a string containing a collection of attributes assigned to that product.

product	tags
100	shoes, hats
101	shoes, socks
102	accessories

Table B: Purchase History

The second table contains two columns as well; the first one is a string that contains a customer name, the second is an integer that contains a product number. The product IDs from column two are the same as the product IDs from column one of dataframe A.

customer	product
A	100
A	101
B	101
C	100
C	102
B	101
A	100
C	102

2a- You are asked to create a query matching this format, where the contents of the cells represent the count of occurrences of product attribute by customer.

customer	shoes	hats	socks	accessories
A	?	?	?	?

customer	shoes	hats	socks	accessories
B	?	?	?	?
C	?	?	?	?

After you have completed your code, describe how you would evaluate it for performance bottlenecks and determine how you would improve the code.

Data step in SAS is used to write the data manually. Macro (%) is used to perform this task even the observations are in millions. Macro is a powerful simulation function in SAS which is used to automate the repeated task. Scan function is used to get the text string. Proc Sql is used to get the desired table and Proc print is used to generate the report.

```
data product;
      length product tags $20.;
      input product $ tags $;
      datalines;
100 shoes,hats
101 shoes,socks
102 accessories
;
run;

data history;
      length customer product $20.;
      input customer $ product $;
      datalines;
A 100
A 101
B 101
C 100
C 102
B 101
A 100
C 102
;
run;

proc sql noprint;
      select tags into: tags separated by ","
      from product
;
run;
%put &tags;

%macro create_columns;
data tags_col;
      customer_tag = "";
      %let i = 1;
      %do %while (%scan("&tags",&i,",") ne);
            %scan("&tags",&i,",") = "";
```

```
            %let i = %eval(&i+1);
      %end;
run;
%mend create_columns;
%create_columns;

proc sql;
      create table history_product (drop=customer_tag) as
      select distinct history.customer, tags_col.*
      from history left join tags_col on history.customer =
tags_col.customer_tag;
;
quit;

Proc Print data = history_product;
run;
```

Obs	customer	shoes	hats	socks	accessories
1	A				
2	B				
3	C				

2b- If the two starting tables were in a Hadoop cluster and each had a 100 million rows, how might your approach change?

Hadoop is an open source distributed processing framework that manages data processing and storage for big data applications running in clustered systems. The data is stored in HDFS and we can use MapReduce or Hive to access or process the data to accomplish the same task in part a.

3a- This question will test your statistical and reasoning abilities. You have been asked to analyze the results of a randomized, controlled experiment on a fictitious website and provide a recommendation. For this experiment, each visitor to the site is randomly exposed to one of four different product images; visitors are then tracked to see which ones make a purchase or not. Based on the data provided, which version of the image should be selected to be presented to all visitors and why?

image	visitors	purchases
A	21	3
B	180	30
C	250	50
D	100	15

Probability test is used to answer question 3 for testing the null that the proportions (probabilities of success which in this case is more number of visitors and purchases) in several groups (i.e, image A, B, C, and D) are the same, or that they equal certain given values. The alternative hypothesis implies that

at least one of the proportions differs across images. Probability test is done by using prop.test in R programming.

```
purchases=c(3,30,50,15)

visitors=c(21,180,250,100)

prop.test(purchases, visitors)
```

```
4-sample test for equality of proportions without continuity correction

data:  purchases out of visitors
X-squared = 1.6991, df = 3, p-value = 0.6371
alternative hypothesis: two.sided
sample estimates:
   prop 1    prop 2    prop 3    prop 4
0.1428571 0.1666667 0.2000000 0.1500000
```

As you can see in the outcome above, the p-value is 0.6371 which means that we fail to reject the null hypothesis. This implies that there is a statistical evidence that the proportion of visitors and purchases across different images are same. So, selection of any image will not give advantage over selection of any other image.

3b- How would your analysis change if the visitors and purchase counts numbered in the millions?

Now if the visitors and purchase counts numbered in the millions and we again apply the probability test, we will get a different result.

```
purchases=c(3000000,30000000,50000000,15000000)

visitors=c(21000000,180000000,250000000,100000000)

prop.test(purchases,visitors)
```

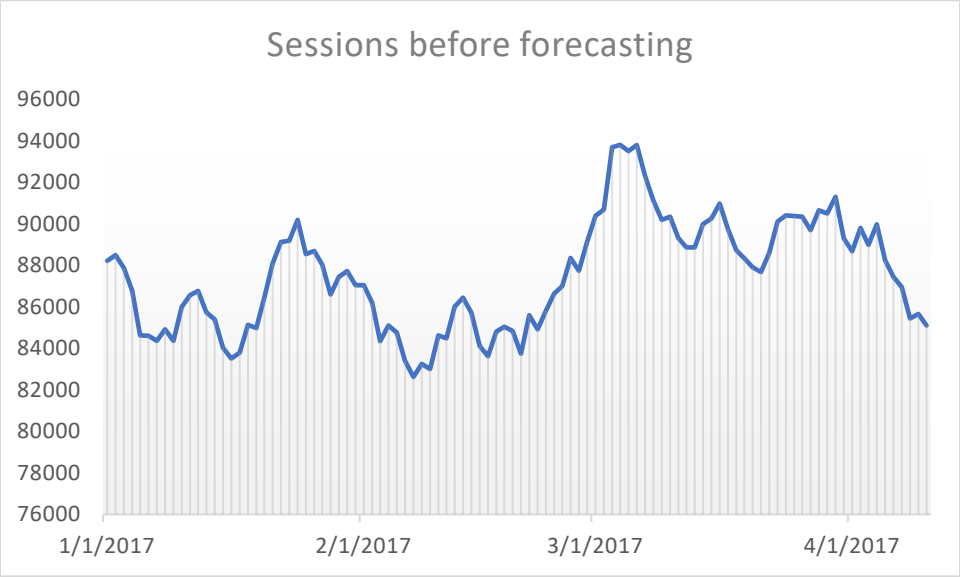
```
4-sample test for equality of proportions without continuity correction

data:  purchases out of visitors
X-squared = 1699100, df = 3, p-value < 2.2e-16
alternative hypothesis: two.sided
sample estimates:
   prop 1    prop 2    prop 3    prop 4
0.1428571 0.1666667 0.2000000 0.1500000
```

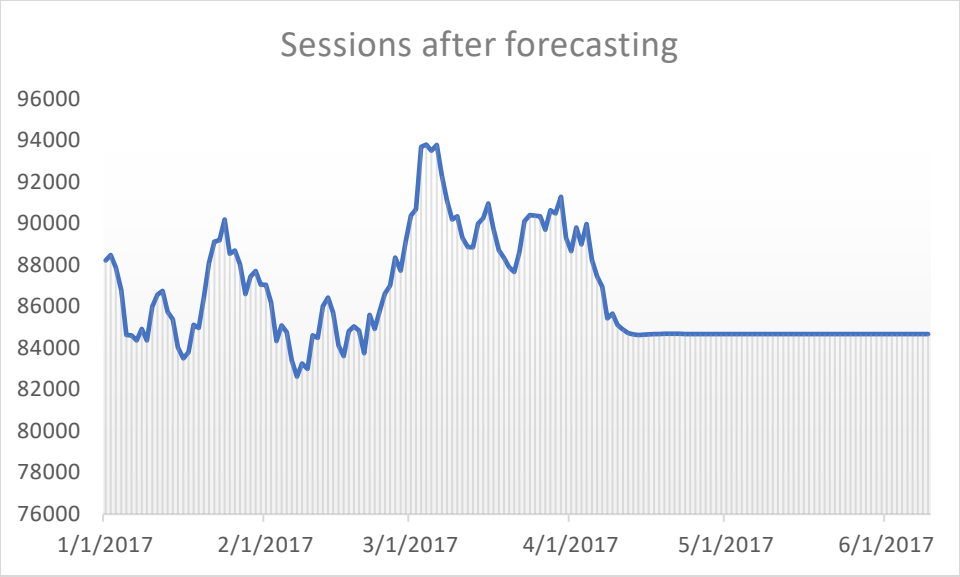
Now we get the p value which is 2.2e-16 (<0.05). This means that we reject the null hypothesis and can conclude that at least one of the images is differ in terms of proportion of visitors and purchases. This happened because of the change in the magnitude of sample size. The bigger the sample size, the more the power of the statistical technique is to detect the main effect, and the more likely it will be a true representative of our population from which our sample is drawn. Image C will be picked and presented to visitors because of high visitors and better conversation rate (purchases over visitors). Number of visitors and conversion rate are the two widely used KPIs in the business of online shopping.

4- For this question, you will be provided with data related to the count of website sessions by day for the past one hundred days. You are now asked to create a forecast for the next sixty days using this data. The data for this problem is included in as a .csv file in the attached packet.

ARIMA time series modeling is used by using R programming to solve question. ARIMA is often suitable for historic data.

```
install.packages('forecast', dependencies = TRUE)

library(forecast)

require(ggplot2)

websessions <- read.csv(file.choose(), header = T)

plot.ts(websessions$sessions)

y=tsclean(ts(websessions$sessions,start = 2017,frequency = 365))

fit=ets(y)

forecast(y,h=60)

forecast(auto.arima(y),h=60)

fit <- nnetar(y, lambda=0)

forecast(fit,h=60)

autoplot(y)

plot.ts(y)
```





As you can see, the ARIMA models gives the aggregated forecasting trend based on historic data. This often occurs when you have less historic data and want to forecast farther future.