**Section 1: SAS/SQL**

| **#** | **Question** |
| --- | --- |
| Q1 | What is the following code doing?    Masproductid eq ‘9001’ and approach eq ‘SA’  Masproductid eq ‘9001’ and approach eq ‘SA’ |
| A1 |  |
| Q2 | 1. What does the code below do? 2. Assume that there 2 values of the name variable; “GDP” and “UE”. What are the values of the macro variables defined? |
| A2 | GDP, GDP,UE,UE,UE |
| Q3 | You would like to create a format using the table below as input, to label the industry code as a full description.   |  |  | | --- | --- | | **Industry\_Code** | **Industry\_Desc** | | 01239 | Oil & Gas | | 11111 | Media | | 33333 | NBFI | | … | ….. | | … | ….. |   Write the code to generate the format **using the SAS dataset as an input**, and apply the format within a datastep. |
| A3 | # import module  **from industry import industry**    # assign data  mydata = [      ["01239", " Oil & Gas "],      ["11111", " Media "],      ["33333", " NBFI "]    ]    # create header  head = ["**Industry\_Code** ", "**Industry\_Desc**"] |
| Q4 | With reference to the code below:  if import\_source in ('SG\_RT2','HK\_RT') then  entity = ifc(import\_source eq 'SG\_RT2', 'DBSSG','DBSHK');   1. What is the value of entity when import\_source = “SG”? ANS -- DBSSG 2. What is the value of entity when import\_source = “HK\_RT”? AND -- DBSHK |
| A4 | What is the value of entity when import\_source = “SG”?  ANS – DBSSG  What is the value of entity when import\_source = “HK\_RT”?  ANS-- DBSHK |
| Q5 | You have a table containing a mix of character and numeric values e.g.   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | Entity | EAD | EAD\_PRE | EAD\_POST | RWA | NPL | SP | | DBSHK | 10000 | 10000 | 10000 | 2000 | 100 | 50 |   You also have a macro variable containing the exchange rate from SGD to HKD:  %let SGDTOHKD = 5.56;  Write a data step to convert all of the numeric values from SGD to HKD in as few lines of code as possible. |
| A5 | def perform(self,):  amount = float(self. EAD.get())  SGD = self. EAD.get()  SGDTOHKD = self. EAD.get()    CONAMT= self.currency\_converter.convert(EAD, SGDTOHKD,amount)  CONAMT= round(SGDTOHKD, 5.56) |

**Section 2: Python**

| **#** | **Question** |
| --- | --- |
| Q1 | Write a function for checking the outstanding of customers. This function should have one parameter: outstanding.   1. If outstanding is less than 20K, it should print “Ok”. 2. Otherwise, for every 5K above the cutoff of 20K, it should give the customer 1% probability of default and print the total probability of default. For example, it outstanding is 30K, it should print: “Probability of default is 2%” 3. If the customer gets more than 10%, the function should print: “Customer to be under high risk segment” |
| A1 | **def cust(outstanding):**  **if outstanding <20000:**  **print("ok")**  **elif outstanding >5 and 20000> outstanding :**  **print("total prob",1%)**  **else:**  **print("high risk segment.")** |
| Q2 | * Import the HDB resale prices dataset from Jan 2017 onwards from <https://data.gov.sg/dataset/resale-flat-prices> * Convert the dataset into a dataframe * Convert the remaining\_lease field from string to a number * Find out the mean and standard deviation of resale price * Create a table that maps Town to Region (North/North-east/East/West/Centrail) from <https://www.hdb.gov.sg/about-us/history/hdb-towns-your-home> * Merge both datasets * Find out the mean and standard deviation of resale price by Region and by flat\_type (2 room/3 room) * Find out how strong is the correlation between floor\_area\_sqm and resale\_price |
| A2 |  |
| Q3 | Write a function to achieve the below.   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Input:   |  |  |  | | --- | --- | --- | | Rep\_Month | Customer | Default | | Jan-19 | A | 0 | | Feb-19 | A | 1 | | Mar-19 | A | NaN | | Apr-19 | A | 0 | | May-19 | A | 1 | | Jun-19 | A | 1 | | Jul-19 | A | 1 | | Aug-19 | A | NaN | | Sep-19 | A | NaN | | Oct-19 | A | NaN | | Nov-19 | A | NaN | | Dec-19 | A | NaN | | Jan-20 | A | NaN | | Feb-20 | A | NaN | |  |  |  | | Mar-21 | A | NaN | | Apr-21 | A | NaN | | … | | | | Apr-21 | Z | 1 | | Output:   |  |  |  |  | | --- | --- | --- | --- | | Rep\_Month | Customer | Default\_12m | Excl | | Jan-19 | A | 1 | 0 | | Feb-19 | A | 0 | 1 | | … | | | | | Jan-19 | B | 0 | 0 | | … | | | | |  1. Identify valid customer at reporting month by flagging Excl=1 if Default =1 2. Tracking valid customer for next 12 months (e.g. for Jan-19 reporting, tracking Default flag from Feb-19 to Jan-20) to identify default 3. Summarize the table as per the output |
| A3 | max\_val = 0  llist1 = []  Excl=1  for j in range(len(Rep\_Month)):  try:  for i in range(Rep\_Month):  val = (Rep\_Month [' Default\_12m’]. Rep\_Month [i+1])  if Excl > max\_val:  max\_v = val  llist1.append(max\_v)  except IndexError:  pass  Excl.append(max(llist1)) |
| Q4 | Write a function to compute K below. You are given parameters PD, LGD, b and M. |
| A4 |  |
| Q5 | Convert the following SAS code to Python using pandas:  data cust\_inner cust\_left\_a cust\_outer; merge cust\_a (in = a)cust\_b (in = b); by descending custid; if a and b then output cust\_inner; if a then output cust\_left\_a; if a or b then output cust\_outer; run; |
| A5 | cust\_inner.merge(‘cust\_a, ‘a’,"inner", on=[' cust\_b ',’b'])  cust\_left\_a.merge(‘cust\_a, ‘a’,"left", on=[' cust\_b ',’b'])  cust\_outer.merge(‘cust\_a, ‘a’,"left", on=[' cust\_b ',’b'])  cust\_outer .sort(custid)  #if if a and b then output cust\_inner;  if a & b  print("cust\_inner ", cust\_inner)  # if a then output cust\_left\_a;  If a  Print (“cust\_left\_a”, cust\_left\_a)  # cust\_outer  If a || b  Print (“cust\_left\_a”, cust\_left\_a)    , |