Data Science 202300200 – 2023/2024-1A
Test DEP – Oct 4th, 2023

This test is for the Topic DEP (with Python). You may use your (not scientific) calculator. You have from 8:45 – 10:45 to finish both topic tests.
Number of questions: 12
You can score a total of 100 points for this exam, you need 55 points to pass the exam.

Section "Concepts"

Question 1. [4 pt]

What is a "foreign key"?

a) attribute(s) in a table that determine a logical group of records in a table
b) attribute(s) in a table that uniquely determine a record in a table
c) attribute(s) in a table that form a reference to the primary key of another table
d) artificially introduced code or number to function as a key

Question 2. [4 pt]

You obtain a source with data on buildings. You suspect that the attribute with the height of the buildings might contain typing errors. Which of the mentioned statistics would be useful to calculate for this attribute to explore this data quality problem? Usefulness of the statistic can be on any level: for detecting the presence of such errors, estimating how many such errors exist, detecting the doubtful values, etc.
(more than one possible answer; you need to give all)

a) Correlation
b) Outliers
c) Histogram
d) Missings
e) Mean

Question 3. [6 pt]

Select all the reasons why domain understanding is important in the field of data analysis.

a) It allows for the use of of statistical techniques without the risk of bias
b) It helps in identifying relevant variables and features for analysis
c) It allows for better data cleaning
d) It enables the creation of visually appealing data visualizations
e) It guarantees accurate predictive modeling results without the need for feature engineering
f) It assists in interpreting data patterns in context
g) It automates data cleaning and preprocessing tasks
h) It aids in formulating meaningful research questions

Question 4. [4 pt]

What is the technical term for the summarization of the characteristics of individual variables?

a) Data Imputation

b) Data Transformation
c) Data Segmentation
d) Data Profiling
e) Data Cleaning
f) Data Normalization

Question 5. [4 pt]

What is the primary purpose of data imputation?

a) To remove outliers from the dataset
b) To create new variables from existing ones
c) To standardize the data for modeling purposes
d) To fill in missing values in the dataset
e) To reduce the dimensionality of the dataset

Question 6. [a: 6 pt; b:2 pt]

For analysing traffic patterns, an inlined logical design for a cube has been made in the form of a table. The table holds information on how many bikes, cars, and trucks pass by various points of particular roads. It also registers the speed of these vehicles.

a) For each attribute, determine whether it is a fact or dimension attribute.

	Fact attribute	Dimension attribute
RoadID (identfier that determines the specific road)		**X**
Speed (of the vehicles passing by)	**X**	
Vehicle kind (bike, car or truck)		**X**
Day (day of the traffic measurement)		**X**
Count (how many vehicles passed by)	**X**	
PointID (identifier that determines a specific point on a road)		**X**

b) How many dimensions are there?
(NB: two or more dimension attributes can together form one dimension!)

ANSWER: 3
Although there are 4 Xs in the dimension column, the RoadID is a grouping over the PointID, because every road is composed of several measurement points.

Question 7. [6 pt]

Match the technical terms to which phase in the Data Science process they belong.

	Data preparation	Analysis	Use
Data transformation	**X**		
Machine learning		**X**	
Data cleaning	**X**		
Decision making			**X**
Data mining		**X**	
Deployment			**X**

Data exploration	**X**		
Data visualization		**X**	

Question 8. [4 pt]

What is the "DBMS"?

a) The computer where the data in the database is stored
b) The programming language for specifying queries, inserts, deletes and updates to the database
c) The tool accessing the database server
d) The software that runs on the database server computer

Section "Code understanding" – Python version

The test had two versions that were the same except for this "code understanding" section. The questions below are from the Python-version of the test.

Question 9. [a: 5pt; b: 5pt; c: 5pt]

Given the following example Python code from the DEP course guide that constructs the 'product' dimension.

```
product=data0[['Product_Name', 'Product_Category']]
product=product.rename(columns={'Product_Name':'name','Pr
oduct_Category':'c
ategory'})
product=product.drop_duplicates(ignore_index=True,keep='l
ast')
product['productid'] = product.reset_index().index
product=product[['productid','name','category']]
```

NB: In the questions below, we use the terminology "on the left" and "on the right" for whatever is coming before the operator and whatever is coming after the operator, respectively. For example, regarding the operator "=" in the first line, its left is "`product`" and its right is everything from "`data0[['Product_Name','Product_Category']]`".

a) What do the "[[" and "]]" brackets do in the first and last line?
***NB**: I approved both answers (c) and (e).*
    a. They omit the columns between the brackets from the variable on the left and leave the remaining columns
    b. They specify that a join should be performed between the two variables on the left using the columns between brackets for matching
    c. They convert the columns between brackets of the variable to the left into a data frame
    d. They check that the columns between brackets are present in the variable on the left; an error is produced if one or more are not present
    e. **They select only the columns between the brackets from the variable on the left and omit the other columns**

b) What would happen differently if we would say "keep=first" instead of "keep=last"
    a. Although a duplicate means that two rows have the same attribute values suggesting that it doesn't matter which one is kept, the first one or the last one, the index numbering can be different after all five lines

b. Nothing: a duplicate means two rows have the same attribute values, hence it doesn't matter which one is kept, the first one or the last one; and also the index number is the same, because we reset the index in the fourth line
c. A duplicate doesn't mean that the values of all attributes need to be the same, so keeping a first or last makes a difference for the attribute values

c) Which of the following pieces of code result in a data frame?
   a. product
   b. product['productid']
   c. product.reset_index().index
   d. data0
   e. 'Product_Name'
   f. product[['productid','name','category']]

Question 10. [a: 5pt; b: 5pt; c: 5 pt]

Given the following example Python code from the DEP course guide that joins the product dimension with the sales table.

```
sales = pd.merge(sales,product,how='outer',
                 left_on =['Product_Name','Product_Category'],
                 right_on = ['name','category'])
```

NB: In the questions below, we use the terminology "on the left" and "on the right" for whatever is coming before the operator and whatever is coming after the operator, respectively. For example, regarding the operator "=", its left is "`sales`" and its right is everything from "`pd.merge`" until "`['name','category'])`".

a) What is the purpose of "`left_on=['Product_Name','Product_Category']`" and "`right_on = ['name','category']`"?
   a. It appends the 'name' values of rows in the product table to the 'Product_Name' values in the sales table. 'category' and 'Product_Category' analogously
   b. It renames 'Product_Name' and 'Product_Category' in the sales table to 'name' and 'category'
   c. It specifies which attributes to use for the join: 'Product_Name' and 'Product_Category' from the 'sales' table and 'name' and 'category' from the 'product' table
   d. It specifies that 'Product_Name' is a naming attribute and that 'Product_Category' is a categorical attribute
   e. It specifies that lower/uppercase and anything before the '_' is irrelevant in comparisons, i.e., that 'Product_Name' is the same as 'name' and that 'Product_Category' is the same as 'category'

b) What does "how=outer" specify?
   a. It specifies that product names and categories are also a match if they match partially at the beginning or the end of the product names and categories
   b. It specifies that the join should be executed in full and not stop when an error occurs in the join process
   c. It specifies the join type to be 'outer join' as opposed to one of the other types of join like inner join, left join, right join, etc.
   d. It specifies that the attributes on the outside are used for matching, i.e., the first attribute of sales and the last attribute of product

c) What does"pd.merge" specify?

a. It specifies that we select the "merge" attribute from the "pd" data frame
b. It specifies that the "pd" and "merge" attributes of the "sales" data frame should not be affected by the join
c. It specifies that we mean to execute the "merge" function from the "pd" package
d. It specifies that the "pd" parameter of the join operation should be set to the value "merge"

Section "Code understanding" – R version

The test had two versions that were the same except for this "code understanding" section. The questions below are from the R-version of the test.

Question 11. [a:5 pt; b:5 pt; c:5 pt]

Given the following example R code from the DEP course guide that constructs the 'product' dimension.

```
product <- data0 %>%
   select(Product_Name, Product_Category) %>%
   rename(name = Product_Name, category = Product_Category) %>%
   arrange(name, category) %>%
   group_by(name, category) %>%
   distinct() %>%
   ungroup() %>%
   mutate(productid = row_number())
```

NB: In the questions below, we use the terminology "on the left" and "on the right" for whatever is coming before the operator and whatever is coming after the operator, respectively. For example, regarding the operator "<-", its left is "product" and its right is everything from "data0" until "row_number())".

a) What does the "%>%" operator do?
   a. It is used for creating comments (we only didn't write comments, but whatever is written to the right of it is interpreted as a comment)
   b. It makes sure that the result of the operation on the left is assigned to the given variable to the left of '<-' (here the variable "product")
   c. It is used to merge the output of the operation on the left with the output of the operation on the right
   d. It doesn't do anything, but is only a delimiter for separating the operations in the list
   e. It takes the output of the variable or operation on the left and gives it as input to the operator on the right

b) What does the '<-' operator do?
   a. It checks whether or not the result of the operations on the right is equal to the contents of the variable on the left
   b. It makes sure that the result of the operation(s) on the right is assigned to the given variable to the left
   c. It is used to rename the variable on the right to the name on the left before the operations on the right are applied
   d. It indicates that all operations on the right are applied to the variable on the left

c) What is the function "row_number()" used for here?
   a. It creates in the resulting table a "row_number" attribute with a unique number for each row
   b. It makes sure that the operations before the mutate are applied in the order of the row number
   c. It provides for a unique number to be used as product identifier
   d. It generates unique numbers for the product categories so that they can be used for grouping productids

Question 12. [a:5pt; b:5pt; c:5pt]

Given the following example R code from the DEP course guide that joins the product dimension with the sales table.

```
sales <- sales %>%
   full_join(product, by = c("Product_Name" = "name",
          "Product_Category" = "category")) %>%
   select( -Product_Name, -Product_Category)
```

NB: In the questions below, we use the terminology "on the left" and "on the right" for whatever is coming before the operator and whatever is coming after the operator, respectively. For example, regarding the operator "<-", its left is "product" and its right is everything from "data0" until "row_number())".

a) What is the purpose of "by = c("Product_Name" = "name", "Product_Category" = "category")"?
   a. It specifies that lower/uppercase and anything before the '_' is irrelevant in comparisons, i.e., that 'Product_Name' is the same as 'name' and that 'Product_Category' is the same as 'category'
   b. It specifies which attributes to use for the join: 'Product_Name' and 'Product_Category' from the 'sales' table and 'name' and 'category' from the 'product' table
   c. It renames 'Product_Name' and 'Product_Category' in the sales table to 'name' and 'category'
   d. It appends the 'name' values of rows in the product table to the 'Product_Name' values in the sales table. 'category' and 'Product_Category' analogously
   e. It specifies that 'Product_Name' is a naming attribute and that 'Product_Category' is a categorical attribute

b) Why does it say 'full_join' and not just 'join'?
   a. It specifies the join type to be 'full_join' as opposed to one of the other types of join like inner_join, left_join, right_join, etc.
   b. It specifies that the join should be executed in 'full' and not stop when an error occurs in the join process
   c. It specifies that the full set of product names and categories are joined
   d. It specifies that product names and categories only a match if they match in 'full' and not partially

c) What does the line "select(-Product_Name, -Product_Category)" do?

a. It selects the "Product_Name" and "Product_Category" attributes as target attributes for subsequent operations
b. It keeps the "Product_Name" and "Product_Category" attributes and removes the other attributes
c. It removes the "Product_Name" and "Product_Category" attributes and keeps the others
d. It specifies that join is performed by matching the "Product_Name" and "Product_Category" attributes

Section "Case"

Question 13. [15 pt]

The management of a chain of car garages would like to optimize their processes regarding service and repairs. Their process is as follows: customers reserve a timeslot (day) themselves with a garage in their vicinity. The reservation can be for a number of different kinds of requests like changing tires, repair of a dent, thorough periodic check and service, casual periodic check and service, and many more. To keep an overview, these request kinds are divided into three categories: periodic service, one-off service, damage repair. They garage registers the moment of reservation, the moment when the car is delivered to the garage, the moment the request started, the moment the request is finished, and the moment the customer picks up the car again. For the optimization of this process, management would like to analyse the waiting time for the customer (time between reservation and delivery of car), request duration (time between start en finish of request), and parking time (time between delivery of car and start of request plus time between finish of request and pick-up). Obviously the car brand and type (e.g. "Renault" is a car brand and "Renauilt Clio" is the car type) influences these times, so management would also like to analyze which car brands and types have higher waiting time and request duration. They are also interested in trends over time, for example, to see that the request duration becomes lower in the period after a certain change in procedures. To determine the exact 'day of the request', we take the moment of the start of the request as a basis.

For each attribute, determine whether it is a fact or lowest-level dimension attribute or a higher-level dimension attribute. You can also decide to not include the attribute in the cube / star schema.

To illustrate the dimension levels, when you have a dimension "location" which is analyzed with a granularity of "city", then the lowest-level dimension attribute is "city" and higher-level dimension attributes are "region", "province", and "country". Together they form one dimension. The higher-level dimension attributes are used for aggregation.

	Fact attribute	Lowest-level dimension attribute	Higher-level dimension attribute	Not in cube / star schema
Car type		**X**		
Datetime of car pick-up				**X**
Datetime of delivery of car				**X**

Request category (periodice service, one-off service, or damage repair)			**X**[1]	
Datetime of reservation				**X**
Customer		**X**		
Datetime of request finish				**X**
Request duration	**X**			
Datetime of request start		**X**[2]		
Request kind (i.e., tire change, dent repair, etc.)		**X**		
Car brand			**X**[3]	
Waiting time for customer	**X**			
Parking time	**X**			

Question 14. [15 pt]

We collected data from wide range of musea in The Netherlands containing information on paintings. In a sense, each museum sent a list with one row per painting in their collection. For ech painting, they give the name of the painter, the painting style, the year it was painted, in which city it was painted, and the number of people on the painting.
The organization who collected this data asked you to analyse it. They are interested in trends in time concerning how many people are on the paintings (going up or down in certain time periods). They are also interested in comparing these trends between painting styles and between the provinces where the paintings were painted.
Give a conceptual design for the cube for this data and questions in terms of a star schema.

Answer

Star schema means: what is the fact and which are the dimensions?
I subtracted 1 point for each dimension when it is not at the right granularity.

Fact: number of people on the paintings (3 points)
Dimension: year (4 points)

- The source data only contains ‘year it was painted’, so any granularity finer than this cannot be derived from this source data.

Dimension: Painting style (4 points)

- The painter itself is not relevant for the questions; I regarded ‘painting style’ as a grouping of ‘painter’, so in that sense, an answer ‘painter’ is not of the right granularity hence still be worth 3 points.

Dimension: Province (4 points)

- Although ‘city’ is recorded in the source data, only data on ‘province’ level is needed for answering the questions.

---

[1] Grouping of request kind.
[2] To answer the questions, only the durations are relevant as facts and not the actual moments in time (as in the ‘datetime’ attributes). But they are also interested in trends in time, and because of the comment “To determine the exact 'day of the request', we take the moment of the start of the request as a basis”, the attribute holding the ‘datetime of request start’ is relevant as a dimension attribute.
[3] Grouping of car type