Data Science Workshop

Safouene Kallel

2023-12-01

SQL Quizz

SQL is a ... language.

... clause is used to choose the columns

... clause is used to filter the rows

... gets rid of the duplicates

... is used to create aggregated summaries

A Primary key is ...

SQL Quizz

Please complete the Query:

```
SELECT Sales.*, Customers.*
FROM Sales
LEFT JOIN Customers
... Customers.CustomerID = Sales.CustomerID
```

R Quizz

{...} helps us connect to SQL
{...} helps us handle data
{...} helps us visualize data

```
pulsedIV <- pulsedIV ...
  filter(Example==1)
```

R Quizz

```r
ggplot(pulsedIV,aes(x=V1,
                    y=I,
                    ... = V2)) +
  geom_point() +
  geom_line()
```

R Quizz

```r
pulsedIV_summary <- pulsedIV %>%
  ...(V1) %>%
  ...(maxI=max(I, na.rm = TRUE))
```

Python

Please open Anaconda:

We have two choices:

- Jupyter (**Ju**lia, **Py**thon and **R**) Web-based, Interactive
- Spyder Integrated Development Environment (close to Matlab and RStudio)

Data Science Libraries

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

numpy

```
python_list = [2, 5, "1", [1 ,3], False]

list = [2,15,30,60]

one_d = np.array(list)
two_d = np.array([list,list]) #list of lists
```

numpy, dicing and slicing

```
two_d[1,1]


two_d[0:1,2:5]
```

numpy can only work on numerical data!

Pandas

Figure 1: not these pandas

pandas dicing and slicing

```
df = pd.read_csv("./Data/gapminder.csv")
display(head(df,5))
display(df.info())
display(df.describe())
```

pandas dicing and slicing

```
display(df.index)
display(df.index.values)
display(df.columns)
```

pandas, dicing and slicing

```
df.iloc[0]
df.iloc[0:2]
df.iloc[:,0]
df.iloc[1:3,0]
countries = df.iloc[:,0]
countries = df['country']
```

pandas

```python
df['country'].nunique()
df['country'].unique()

mean_lifeExp = df['lifeExp'].mean()
```

pandas, dicing and slicing

```
df1952 = df[df['year']==1952]
df1952.drop(columns='year', inplace=True)

maxpop = df1952['pop'].max()
country_1952_maxpop = df1952[df1952['pop'] == maxpop]
display(country_1952_maxpop)
```

pandas, sorting

```
df1952= df1952.sort_values(by=["gdpPercap"])
df1952.sort_values(by=["gdpPercap"], inplace=True)

df1952.sort_values(by=["gdpPercap"], inplace=True, ascending=F
```

pandas plotting

```
df1952.plot.scatter(x='gdpPercap', y='lifeExp')
```

SQL

```python
import pandas as pd
import numpy as np
import MySQLdb
```

SQL

```
conn = MySQLdb.connect(host='stgux-db.vih.infineon.com',
                                     database='DS_Training',
                                     user='DS_Training_u01',
                                     password='')
```

SQL

```
Employees =
  pd.read_sql_query(sql= "SELECT * FROM employees", con=conn)
```

SQL

```
Employee2 =
  Employees[Employees["EmployeeID"]==2]
```

```
Employee2 =
  pd.read_sql_query(
    sql= "SELECT * FROM employees WHERE EmployeeID = 2", con=c
```

Exercice

Do the same thing we did last time in R (PulsedIV data) but this time in Python.

Solution

We load the usual suspects

```python
import pandas as pd
import numpy as np
import seaborn as sns
```

Solution

Let's do some data exploration

```
pulsedIV = pd.read_csv("./Data/TestID2.csv")

pulsedIV.dtypes

pulsedIV.describe()
```

Solution

filter data to have only Example = 1

```
pulsedIV = pulsedIV[pulsedIV['Example']==1]
```

Solution

create a plot to show the raw data (x axis : Vd_Target, y axis: Id)

```
sns.scatterplot(data=pulsedIV,x='V2',\
                y='I', hue='V1')

sns.scatterplot(data=pulsedIV,x='V2',\
                y='I', hue='V1',legend='full')
```

create a line between the points

```
sns.lineplot(data=pulsedIV,x='V2',y='I', \
            hue='V1', legend='full')
```

Solution

modify labels (to add the units)

```
p=sns.lineplot(data=pulsedIV,x='V2',\
              y='I', hue='V1', legend='full')

p.legend(title='V1 [V]',\
                loc='center left', bbox_to_anchor=(1, 0.5))
p.set_xlabel('V1 [V]')
p.set_ylabel('I [A]')
```

Solution

get the maximum value of current per line the data

```
pulsed_summary = pulsedIV.groupby('V1').max('I')
```

```
sns.lineplot(data=pulsed_summary,x='V1',y='I')
```