01b_LAB_Reading_Data

May 3, 2022

1 Machine Learning Foundation

1.1 Section 1, Part b: Reading Data Lab

[1]:
```
# Imports
import sqlite3 as sq3
import pandas.io.sql as pds
import pandas as pd
```

1.2 Lab Exercise: Reading in database files

- Create a variable, `path`, containing the path to the `baseball.db` contained in `resources/`
- Create a connection, `con`, that is connected to database at `path`
- Create a variable, `query`, containing a SQL query which reads in all data from the `allstarfull` table
- Create a variable, `observations`, by using pandas' read_sql

1.2.1 Optional

- Create a variable, `tables`, which reads in all data from the table `sqlite_master`
- Pretend that you were interesting in creating a new baseball hall of fame. Join and analyze the tables to evaluate the top 3 all time best baseball players.

[2]:
```
# Download the database
!wget -P data https://cf-courses-data.s3.us.cloud-object-storage.appdomain.
 ↪cloud/IBM-ML0232EN-SkillsNetwork/asset/baseball.db
```

```
--2022-05-03 22:02:19--  https://cf-courses-data.s3.us.cloud-object-
storage.appdomain.cloud/IBM-ML0232EN-SkillsNetwork/asset/baseball.db
Resolving cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-
courses-data.s3.us.cloud-object-storage.appdomain.cloud)… 169.63.118.104
Connecting to cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-
courses-data.s3.us.cloud-object-storage.appdomain.cloud)|169.63.118.104|:443…
connected.
HTTP request sent, awaiting response… 200 OK
Length: 7626752 (7.3M) [binary/octet-stream]
Saving to: 'data/baseball.db'

baseball.db         100%[===================>]   7.27M  16.5MB/s    in 0.4s
```

```
2022-05-03 22:02:19 (16.5 MB/s) - ‘data/baseball.db’ saved [7626752/7626752]
```

[3]:

```
### BEGIN SOLUTION
# Create a variable, `path`, containing the path to the `baseball.db` contained␣
 ↪in `resources/`
path = 'data/baseball.db'

# Create a connection, `con`, that is connected to database at `path`
con = sq3.Connection(path)

# Create a variable, `query`, containing a SQL query which reads in all data␣
 ↪from the `` table

query = """
SELECT *
    FROM allstarfull
    ;
"""

allstar_observations = pd.read_sql(query, con)

# print(allstar_observations)

# Create a variable, tables, which reads in all data from the table␣
 ↪sqlite_master
all_tables = pd.read_sql('SELECT * FROM sqlite_master', con)
print(all_tables)

# Pretend that you were interesting in creating a new baseball hall of fame.␣
 ↪Join and analyze the tables to evaluate the top 3 all time best baseball␣
 ↪players
best_query = """
SELECT playerID, sum(GP) AS num_games_played, AVG(startingPos) AS␣
 ↪avg_starting_position
    FROM allstarfull
    GROUP BY playerID
    ORDER BY num_games_played DESC, avg_starting_position ASC
    LIMIT 3
"""
best = pd.read_sql(best_query, con)
print(best.head())
### END SOLUTION
```

```
    type                name     tbl_name  rootpage  \
0  table         allstarfull  allstarfull         2
```

```
1  index  ix_allstarfull_index  allstarfull         3
2  table               schools      schools        26
3  index      ix_schools_index      schools        31
4  table               batting      batting        99
5  index      ix_batting_index      batting       100

                                                  sql
0  CREATE TABLE "allstarfull" (\n"index" INTEGER,…
1  CREATE INDEX "ix_allstarfull_index"ON "allstar…
2  CREATE TABLE "schools" (\n"index" INTEGER,\n  …
3  CREATE INDEX "ix_schools_index"ON "schools" ("…
4  CREATE TABLE "batting" (\n"index" INTEGER,\n  …
5  CREATE INDEX "ix_batting_index"ON "batting" ("…
    playerID  num_games_played  avg_starting_position
0  musiast01              24.0               6.357143
1   mayswi01              24.0               8.000000
2  aaronha01              24.0               8.470588
```

1.2.2 Machine Learning Foundation (C) 2020 IBM Corporation