M6 database coding session

In [ ]:
```
import pandas as pd
import numpy as np
```

In [4]:
```
# Not stable URL from here: https://stewart-gibson.shinyapps.io/NBA_All_
Data_Downloader/
url = "https://stewart-gibson.shinyapps.io/NBA_All_Data_Downloader/_w_e5
7f2079/session/8211590d61fa5888eafd6dfb23839726/download/downloadData?w=
e57f2079"
```

In [5]:
```
nba = pd.read_csv(url)
nba.head()
```

```
<ipython-input-5-e68c0c818c74>:1: DtypeWarning: Columns (68,69,70) have
mixed types. Specify dtype option on import or set low_memory=False.
  nba = pd.read_csv(url)
```

Out[5]:

	game_id	game_date	OT	H_A	Team_Abbrev	Team_Score	Team_pace	Team_efg_pct	T
0	202203130ATL	2022-03-13	0	A	IND	128	96.3	0.601	
1	202203130ATL	2022-03-13	0	A	IND	128	96.3	0.601	
2	202203130ATL	2022-03-13	0	A	IND	128	96.3	0.601	
3	202203130ATL	2022-03-13	0	A	IND	128	96.3	0.601	
4	202203130ATL	2022-03-13	0	A	IND	128	96.3	0.601	

5 rows × 81 columns

In [12]:
```
# Set pd options display all 7:20 pm
pd.options.display.max_rows = None
nba.T
```

Out[12]:

	0	1	2	3	
game_id	202203130ATL	202203130ATL	202203130ATL	202203130ATL	20
game_date	2022-03-13	2022-03-13	2022-03-13	2022-03-13	
OT	0	0	0	0	
H_A	A	A	A	A	
Team_Abbrev	IND	IND	IND	IND	
Team_Score	128	128	128	128	
Team_pace	96.3	96.3	96.3	96.3	
Team_efg_pct	0.601	0.601	0.601	0.601	
Team_tov_pct	13.0	13.0	13.0	13.0	
Team_orb_pct	35.7	35.7	35.7	35.7	
Team_ft_rate	0.236	0.236	0.236	0.236	
Team_off_rtg	132.9	132.9	132.9	132.9	
Inactives	Malcolm Brogdon, T.J. McConnell, Ricky Rubio, ...	Malcolm Brogdon, T.J. McConnell, Ricky Rubio, ...	Malcolm Brogdon, T.J. McConnell, Ricky Rubio, ...	Malcolm Brogdon, T.J. McConnell, Ricky Rubio, ...	
Opponent_Abbrev	ATL	ATL	ATL	ATL	
Opponent_Score	131	131	131	131	
Opponent_pace	96.3	96.3	96.3	96.3	
Opponent_efg_pct	0.649	0.649	0.649	0.649	
Opponent_tov_pct	10.4	10.4	10.4	10.4	
Opponent_orb_pct	24.3	24.3	24.3	24.3	
Opponent_ft_rate	0.262	0.262	0.262	0.262	
Opponent_off_rtg	136.0	136.0	136.0	136.0	
player	Tyrese Haliburton	Buddy Hield	Oshae Brissett	Isaiah Jackson	
player_id	halibty01	hieldbu01	brissos01	jacksis01	
starter	1	1	1	1	
mp	40:05	39:25	28:58	27:54	
fg	9	9	5	4	
fga	15	20	8	9	
fg_pct	0.6	0.45	0.625	0.444	
fg3	3	2	3	1	
fg3a	5	7	5	1	
fg3_pct	0.6	0.286	0.6	1.0	
ft	4	5	2	3	

	0	1	2	3
fta	4	6	4	4
ft_pct	1.0	0.833	0.5	0.75
orb	1	0	1	6
drb	1	4	3	9
trb	2	4	4	15
ast	10	5	2	0
stl	1	1	1	1
blk	0	0	0	2
tov	5	2	2	2
pf	2	5	4	4
pts	25	25	15	12
plus_minus	2	4	5	0
did_not_play	0	0	0	0
is_inactive	0	0	0	0
ts_pct	0.746	0.552	0.768	0.558
efg_pct	0.7	0.5	0.813	0.5
fg3a_per_fga_pct	0.333	0.35	0.625	0.111
fta_per_fga_pct	0.267	0.3	0.5	0.444
orb_pct	2.9	0.0	3.9	24.6
drb_pct	3.2	13.2	13.4	41.8
trb_pct	3.0	6.2	8.4	32.7
ast_pct	34.0	17.4	8.8	0.0
stl_pct	1.2	1.3	1.7	1.8
blk_pct	0.0	0.0	0.0	7.5
tov_pct	23.0	8.1	17.0	15.7
usg_pct	22.6	26.0	16.9	19.0
off_rtg	135	123	134	122
def_rtg	140	138	137	127
bpm	2.9	-3.9	2.1	-2.9
season	2022	2022	2022	2022
minutes	40.083333	39.416667	28.966667	27.9
double_double	1	0	0	1
triple_double	0	0	0	0
DKP	45.0	39.5	25.5	37.75
FDP	40.4	38.3	23.8	37.0

	0	1	2	3
SDP	43.5	39.5	25.5	39.75
DKP_per_minute	1.122661	1.002114	0.880322	1.353047
FDP_per_minute	1.0079	0.97167	0.821634	1.326165
SDP_per_minute	1.085239	1.002114	0.880322	1.424731
pf_per_minute	0.049896	0.12685	0.13809	0.143369
ts	16.76	22.64	9.76	10.76
last_60_minutes_per_game_starting	37.403623	39.205128	31.130769	19.821212
last_60_minutes_per_game_bench	20.17601	26.506944	20.932051	10.313636
PG%	69.0	1.0	0.0	0.0
SG%	28.0	50.0	0.0	0.0
SF%	4.0	48.0	18.0	0.0
PF%	0.0	1.0	79.0	27.0
C%	0.0	0.0	3.0	73.0
active_position_minutes	49.812593	48.305206	42.877343	47.1779

81 rows × 115719 columns

In [13]:
```
nba = nba.drop('Inactives', axis=1)
```

TODOS: trategy for normalization down to the 3d NF:**

- T1: info about just the team
- T2: info about the team in each game
- T3: info about the player in each game
- T4: inf about players, not dependent on games (season totals)
- T5: info about the teams, not dependent on games (season totals)

Information overall about the game

In [30]:
```
!pip install psycopg2
```

```
Collecting psycopg2
  Downloading psycopg2-2.9.3.tar.gz (380 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 380.6/380.6 KB 8.3 MB/s et
a 0:00:0000:01
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: psycopg2
  Building wheel for psycopg2 (setup.py) ... done
  Created wheel for psycopg2: filename=psycopg2-2.9.3-cp38-cp38-macosx_
10_9_x86_64.whl size=142673 sha256=1524b13def4da86bd28e3d42306c238d7b2a
91550d443c5218a7f8e0c47a2d5c
  Stored in directory: /Users/dmitrymikhaylov/Library/Caches/pip/wheel
s/f3/dc/e2/b8e0e2142eff7fd680295ecd2d92e3bfbb90195523e43da161
Successfully built psycopg2
Installing collected packages: psycopg2
Successfully installed psycopg2-2.9.3
WARNING: You are using pip version 22.0.3; however, version 22.0.4 is a
vailable.
You should consider upgrading via the '/Users/dmitrymikhaylov/opt/anaco
nda3/bin/python -m pip install --upgrade pip' command.
```

In [31]:
```
import dotenv
import sqlite3
import os
import psycopg2
from sqlalchemy import create_engine
```

In [14]:
```
nba.head(5)
```

Out[14]:

	game_id	game_date	OT	H_A	Team_Abbrev	Team_Score	Team_pace	Team_efg_pct	T
0	202203130ATL	2022-03-13	0	A	IND	128	96.3	0.601	
1	202203130ATL	2022-03-13	0	A	IND	128	96.3	0.601	
2	202203130ATL	2022-03-13	0	A	IND	128	96.3	0.601	
3	202203130ATL	2022-03-13	0	A	IND	128	96.3	0.601	
4	202203130ATL	2022-03-13	0	A	IND	128	96.3	0.601	

5 rows × 80 columns

Information overall about the game: OT, date, location, etc

In [17]:
```
nba2022 = nba.query("season==2022")
```

Information overall about the game: OT, date, location, etc

In [19]:
```
game_info = nba2022[['game_id', 'game_date', 'OT']].drop_duplicates()
game_info.head()
```

Out[19]:

	game_id	game_date	OT
0	202203130ATL	2022-03-13	0
24	202203130BOS	2022-03-13	0
50	202203130BRK	2022-03-13	0
73	202203130DET	2022-03-13	0
96	202203130NOP	2022-03-13	0

Info about how the team overall did in the game

In [40]:
```
nba2022['win'] = nba2022['Team_Score'] > nba2022['Opponent_Score']
team_game = nba2022[['Team_Abbrev', 'H_A', 'win', 'game_id', 'fg', 'fga'
, 'fg3', 'fg3a',
                     'ft', 'fta', 'orb', 'drb', 'ast', 'stl', 'blk', 'tov', 'pf'
, 'pts']]
team_game = team_game.groupby(['game_id','Team_Abbrev']).agg({'H_A': pd.
Series.mode,
                                                              'win': 'me
an',
                                                              'fg': sum,
                                                              'fga': sum
,
                                                              'fg3': sum
,
                                                              'fg3a': su
m,
                                                              'ft': sum,
                                                              'fta': sum
,
                                                              'orb': sum
,
                                                              'drb': sum
,
                                                              'ast': sum
,
                                                              'stl': sum
,
                                                              'blk': sum
,
                                                              'tov': sum
,
                                                              'pf': sum,
                                                              'pts': sum
})
team_game = team_game.reset_index()
```

```
<ipython-input-40-64475f1533d7>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  nba2022['win'] = nba2022['Team_Score'] > nba2022['Opponent_Score']
```

Info about how the player did personally in the game

In [33]:
```
player_game = nba2022[['game_id', 'player_id', 'starter', 'minutes', 'fg', 'fga', 'fg3', 'fg3a',
                       'ft', 'fta', 'orb', 'drb', 'ast', 'stl', 'blk', 'tov', 'pf', 'pts', 'usg_pct',
                       'is_inactive', 'PG%', 'SG%', 'SF%', 'PF%', 'C%']]

player_game.head()
```

Out[33]:

	game_id	player_id	starter	minutes	fg	fga	fg3	fg3a	ft	fta	...	tov	pf	pts	usg_
0	202203130ATL	halibty01	1	40.083333	9	15	3	5	4	4	...	5	2	25	2
1	202203130ATL	hieldbu01	1	39.416667	9	20	2	7	5	6	...	2	5	25	2
2	202203130ATL	brissos01	1	28.966667	5	8	3	5	2	4	...	2	4	15	1
3	202203130ATL	jacksis01	1	27.900000	4	9	1	1	3	4	...	2	4	12	1
4	202203130ATL	taylote01	1	24.766667	6	8	1	2	3	4	...	0	0	16	1

5 rows × 25 columns

Info about the team's total season stats so far

In [28]:
```
nba2022.columns
```

Out[28]:
```
Index(['game_id', 'game_date', 'OT', 'H_A', 'Team_Abbrev', 'Team_Score',
       'Team_pace', 'Team_efg_pct', 'Team_tov_pct', 'Team_orb_pct',
       'Team_ft_rate', 'Team_off_rtg', 'Opponent_Abbrev', 'Opponent_Score',
       'Opponent_pace', 'Opponent_efg_pct', 'Opponent_tov_pct',
       'Opponent_orb_pct', 'Opponent_ft_rate', 'Opponent_off_rtg', 'player',
       'player_id', 'starter', 'mp', 'fg', 'fga', 'fg_pct', 'fg3', 'fg3a',
       'fg3_pct', 'ft', 'fta', 'ft_pct', 'orb', 'drb', 'trb', 'ast', 'stl',
       'blk', 'tov', 'pf', 'pts', 'plus_minus', 'did_not_play', 'is_inactive',
       'ts_pct', 'efg_pct', 'fg3a_per_fga_pct', 'fta_per_fga_pct', 'orb_pct',
       'drb_pct', 'trb_pct', 'ast_pct', 'stl_pct', 'blk_pct', 'tov_pct',
       'usg_pct', 'off_rtg', 'def_rtg', 'bpm', 'season', 'minutes',
       'double_double', 'triple_double', 'DKP', 'FDP', 'SDP', 'DKP_per_minute',
       'FDP_per_minute', 'SDP_per_minute', 'pf_per_minute', 'ts',
       'last_60_minutes_per_game_starting', 'last_60_minutes_per_game_bench',
       'PG%', 'SG%', 'SF%', 'PF%', 'C%', 'active_position_minutes', 'win'],
      dtype='object')
```

In [35]:

```
teams = nba2022[['Team_Abbrev', 'game_id', 'fg', 'fga', 'fg3', 'fg3a',
                 'ft', 'fta', 'orb', 'drb', 'ast', 'stl', 'blk', 'tov', 'pf'
, 'pts']]
teams = teams.groupby('Team_Abbrev').agg({'game_id': lambda x: x.nunique
(),
                                          'fg': sum,
                                          'fga': sum,
                                          'fg3': sum,
                                          'fg3a': sum,
                                          'ft': sum,
                                          'fta': sum,
                                          'orb': sum,
                                          'drb': sum,
                                          'ast': sum,
                                          'stl': sum,
                                          'blk': sum,
                                          'tov': sum,
                                          'pf': sum,
                                          'pts': sum})
WL = nba2022.groupby(['Team_Abbrev', 'game_id']).agg({'win': 'mean'})
WL = WL.groupby('Team_Abbrev').agg({'win':sum})
teams = pd.merge(teams, WL, on=['Team_Abbrev'], validate='one_to_one')

teams = teams.rename({'game_id':'total_games'}, axis=1)

teams = teams.reset_index()
teams
```

Out[35]:

	Team_Abbrev	total_games	fg	fga	fg3	fg3a	ft	fta	orb	drb	ast	stl	blk
0	ATL	67	2762	5884	841	2257	1193	1479	665	2282	1629	456	293
1	BOS	69	2752	6037	864	2520	1186	1455	748	2448	1654	498	416
2	BRK	68	2825	6021	768	2168	1153	1445	694	2308	1697	479	363
3	CHI	67	2820	5833	730	1948	1153	1414	577	2302	1616	481	293
4	CHO	68	2931	6349	954	2656	1101	1497	762	2357	1891	604	334
5	CLE	67	2640	5665	768	2201	1099	1441	684	2303	1680	477	288
6	DAL	68	2693	5909	885	2564	1087	1422	646	2319	1603	484	286
7	DEN	68	2794	5867	866	2470	1101	1394	616	2387	1859	496	257
8	DET	68	2566	6026	753	2328	1158	1485	760	2170	1557	509	323
9	GSW	68	2768	5901	987	2717	1050	1377	663	2430	1863	627	322
10	HOU	68	2699	5956	919	2679	1196	1696	663	2242	1601	503	311
11	IND	69	2888	6256	843	2464	1142	1504	781	2361	1737	480	404
12	LAC	70	2822	6202	879	2399	1066	1360	644	2499	1672	532	341
13	LAL	67	2811	6043	819	2345	1103	1517	653	2361	1622	539	380
14	MEM	69	2979	6488	761	2232	1175	1604	977	2431	1751	679	446
15	MIA	69	2767	5980	929	2484	1203	1492	718	2389	1796	533	238
16	MIL	68	2810	6064	971	2654	1197	1547	706	2478	1610	515	272
17	MIN	69	2883	6375	1021	2917	1221	1581	798	2324	1783	614	404
18	NOP	68	2745	6052	741	2251	1242	1583	805	2330	1724	549	283
19	NYK	68	2594	5953	891	2515	1225	1641	785	2412	1480	469	335
20	OKC	67	2590	6102	797	2532	1023	1341	707	2412	1479	529	314
21	ORL	69	2675	6144	841	2542	1117	1413	632	2464	1635	481	321
22	PHI	66	2618	5680	758	2106	1300	1584	569	2260	1552	508	357
23	PHO	68	2978	6178	796	2202	1104	1388	670	2466	1876	591	310
24	POR	66	2541	5736	867	2476	1124	1468	669	2204	1488	505	293
25	SAC	69	2808	6120	789	2305	1201	1572	687	2302	1623	493	310
26	SAS	68	3018	6443	756	2166	1014	1366	772	2366	1937	526	342
27	TOR	67	2741	6189	812	2303	1090	1449	900	2150	1485	615	316
28	UTA	67	2768	5844	999	2744	1208	1557	713	2434	1530	484	332
29	WAS	66	2720	5794	709	2079	1151	1467	620	2281	1656	432	336

Info about the player's total season so far

In [24]:
```
players = nba2022[['player', 'player_id', 'starter', 'minutes', 'fg', 'fga', 'fg3', 'fg3a',
                   'ft', 'fta', 'orb', 'drb', 'ast', 'stl', 'blk', 'tov', 'pf', 'pts', 'usg_pct',
                   'did_not_play', 'is_inactive', 'ts_pct', 'PG%', 'SG%', 'SF%', 'PF%', 'C%']]
players = players.groupby(['player','player_id']).agg({'starter': sum,
                                                       'minutes': 'mean',
                                                       'fg': sum,
                                                       'fga': sum,
                                                       'fg3': sum,
                                                       'fg3a': sum,
                                                       'ft': sum,
                                                       'fta': sum,
                                                       'orb': sum,
                                                       'drb': sum,
                                                       'ast': sum,
                                                       'stl': sum,
                                                       'blk': sum,
                                                       'tov': sum,
                                                       'pf': sum,
                                                       'pts': sum,
                                                      'usg_pct': 'mean',
                                                       'ts_pct': 'mean',
                                                      'did_not_play': sum,
                                                      'is_inactive': sum,
                                                      'PG%': 'mean',
                                                       'SG%': 'mean',
                                                       'SF%': 'mean',
                                                       'PF%': 'mean',
                                                       'C%': 'mean'}).add_prefix('season_')
```

In [36]:
```
players = players.reset_index()
```

In [37]:
```
players.head()
```

Out[37]:

	player	player_id	season_starter	season_minutes	season_fg	season_fga	season_fg3	seas
0	Aaron Gordon	gordoaa01	62	31.671237	344	674	68	
1	Aaron Henry	henryaa01	0	0.999020	1	5	0	
2	Aaron Holiday	holidaa01	14	14.848305	136	289	34	
3	Aaron Nesmith	nesmiaa01	2	8.159770	56	153	23	
4	Aaron Wiggins	wiggiaa01	27	19.967708	119	249	35	

5 rows × 27 columns

Create SQLite database

In [38]:
```
nbadb = sqlite3.connect("nba.dn")
```

In [41]:
```
game_info.to_sql('game_info', nbadb, index=False, chunksize=1000, if_exi
sts = 'replace') # to be run
```

Out[41]: 1018

In [42]:
```
team_game.to_sql('team_game', nbadb, index=False, chunksize=1000, if_exi
sts = 'replace')
```

Out[42]: 2036

In [43]:
```
player_game.to_sql('player_game', nbadb, index=False, chunksize=1000, if
_exists = 'replace')
```

Out[43]: 26220

In [44]:
```
teams.to_sql('teams_season', nbadb, index=False, chunksize=1000, if_exis
ts = 'replace')
```

Out[44]: 30

In [45]:
```
players.to_sql('players_season', nbadb, index=False, chunksize=1000, if_
exists = 'replace')
```

Out[45]: 612

In [48]:
```
pd.read_sql_query("SELECT * FROM players_season", nbadb).head(10)
```

Out[48]:

	player	player_id	season_starter	season_minutes	season_fg	season_fga	season_fg3	sea
0	Aaron Gordon	gordoaa01	62	31.671237	344	674	68	
1	Aaron Henry	henryaa01	0	0.999020	1	5	0	
2	Aaron Holiday	holidaa01	14	14.848305	136	289	34	
3	Aaron Nesmith	nesmiaa01	2	8.159770	56	153	23	
4	Aaron Wiggins	wiggiaa01	27	19.967708	119	249	35	
5	Abdel Nader	naderab01	0	6.050000	12	35	4	
6	Ade Murkey	murkead01	0	1.466667	0	0	0	
7	Admiral Schofield	schofad01	1	7.846491	36	79	14	
8	Ahmad Caver	caverah01	0	0.208333	1	1	0	
9	Al Horford	horfoal01	59	28.522500	230	506	75	

10 rows × 27 columns

In [49]:
```
pd.read_sql_query("SELECT * FROM team_game", nbadb).head(10)
```

Out[49]:

	game_id	Team_Abbrev	H_A	win	fg	fga	fg3	fg3a	ft	fta	orb	drb	ast	stl	blk
0	202110190LAL	GSW	A	1.0	41	93	14	39	25	30	9	41	30	9	2
1	202110190LAL	LAL	H	0.0	45	95	15	42	9	19	5	40	21	7	4
2	202110190MIL	BRK	A	0.0	37	84	17	32	13	23	5	39	19	3	9
3	202110190MIL	MIL	H	1.0	48	105	17	45	14	18	13	41	25	8	9
4	202110200CHO	CHO	H	1.0	46	107	13	31	18	27	12	34	29	9	5
5	202110200CHO	IND	A	0.0	42	90	17	47	21	24	8	43	29	2	10
6	202110200DET	CHI	A	1.0	37	86	7	23	13	15	9	39	18	8	5
7	202110200DET	DET	H	0.0	36	90	6	28	10	13	11	36	17	7	5
8	202110200MEM	CLE	A	0.0	47	93	14	38	13	18	7	29	38	6	5
9	202110200MEM	MEM	H	1.0	53	100	14	33	12	12	13	40	28	8	8