Zahra Jalilpour

Email: h19zahja@du.se

Report of AMI23B – Business Intelligence Lab2

nba_all_elo.csv is a game information about basketball games in different years.

At first we create a script to download the data. After running the script, we will have nba_all_elo.csv in our current directory.

By using Pandas Python Library , we can analysis our dataset.

Here we can see the definition of the header of dataset:

Header	Definition
gameorder	Play order of game in NBA history
game_id	Unique ID for each game
lg_id	Which league the game was played in
_iscopy	Each row of data is tied to a single team for a single game, so _iscopy flags if this game_id has already occured for the opposing team in the same matchup
year_id	Season id, named based on year in which the season ended
date_game	Game date
is_playoffs	Flag for playoff games
team_id	Three letter code for team name, from Basketball Reference
fran_id	Franchise id. Multiple team_ids can fall under the same fran_id due to name changes or moves. Interactive is grouped by fran_id.
pts	Points scored by team
elo_i	Team elo entering the game
elo_n	Team elo following the game
win_equiv	Equivalent number of wins in a 82-game season for a team of elo_n quality
opp_id	Team id of opponent
opp_fran	Franchise id of opponent
opp_pts	Points scored by opponent
opp_elo_i	Opponent elo entering the game
opp_elo_n	Opponent elo following the game
game_location	Home (H), away (A), or neutral (N)
game_result	Win or loss for team in the team_id column
forecast	Elo-based chances of winning for the team in the team_id column, based on elo ratings and game location
notes	Additional information

The number of rows(Observations) is equal to : **126314**

By **nba.shape** we can find the number of rows and columns: The result is a tuple containing the number of rows and columns. **(126314, 23)**

By using **nba.head(),** we can see the first 5 rows of output.

as I use Jupyter notenook, I cannot see the output completely, but I can scroll it. It is also practical to see all the columns by:

pd.set_option("display.max.columns", None)

By **nba.tail(),** we can see the last 5 rows. Here we can see all columns , because we define the command **pd.set_option**.

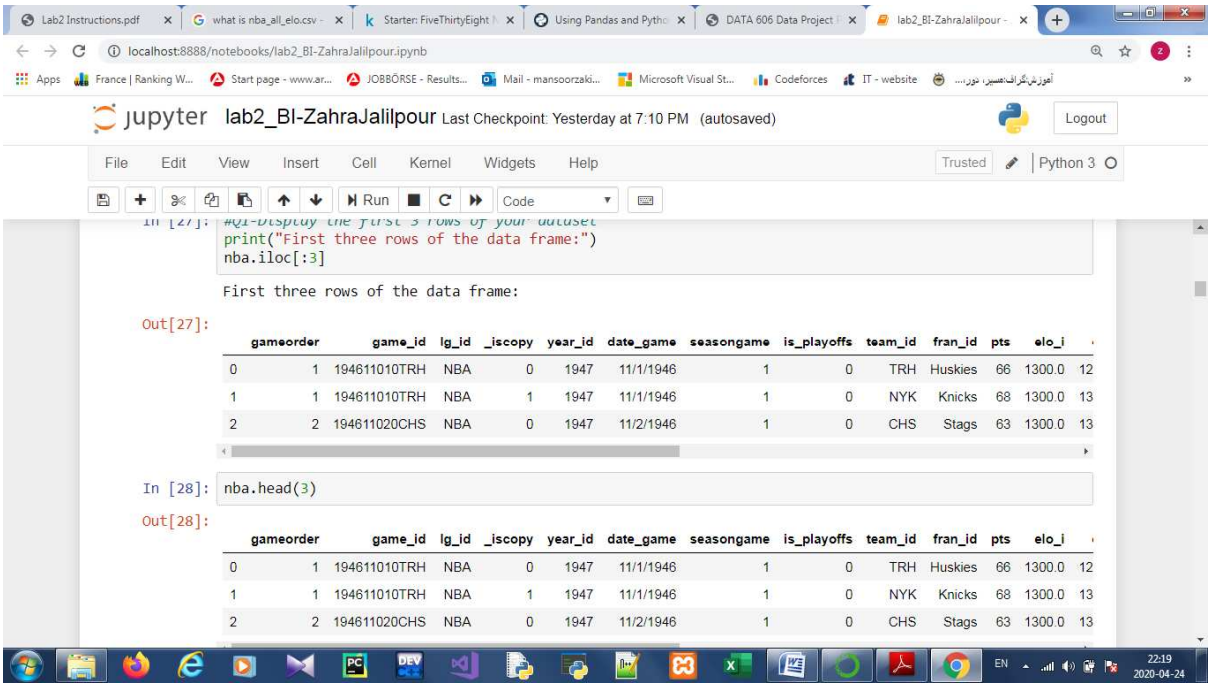
Q1: Display the first 3 rows of your dataset.

- It can be done through different commands:

nba.iloc[:3]

or

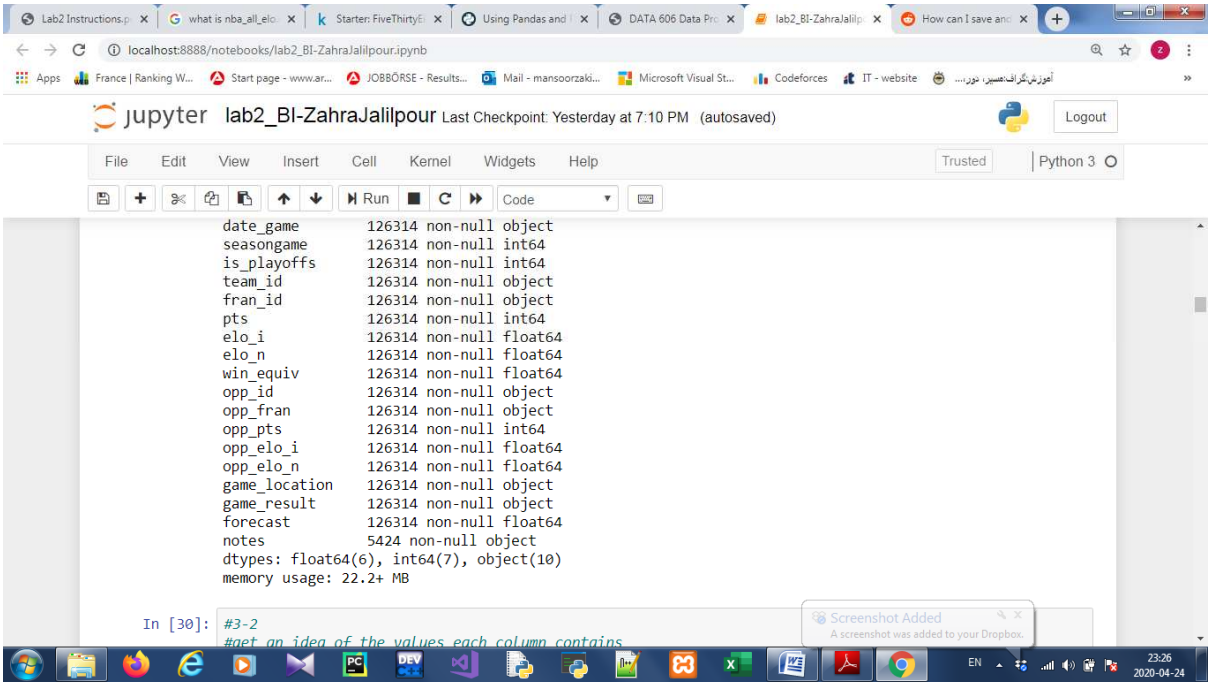
nba.head(3)



	gameorder	game_id	lg_id	_iscopy	year_id	date_game	seasongame	is_playoffs	team_id	fran_id	pts	elo_i
0	1	194611010TRH	NBA	0	1947	11/1/1946	1	0	TRH	Huskies	66	1300.0
1	1	194611010TRH	NBA	1	1947	11/1/1946	1	0	NYK	Knicks	68	1300.0
2	2	194611020CHS	NBA	0	1947	11/2/1946	1	0	CHS	Stags	63	1300.0

In previous task we learn how to display first and last rows , size of dataset, how to import csv.file, now we learn how to examine our data. At first we see the different data type that exist in our dataset.

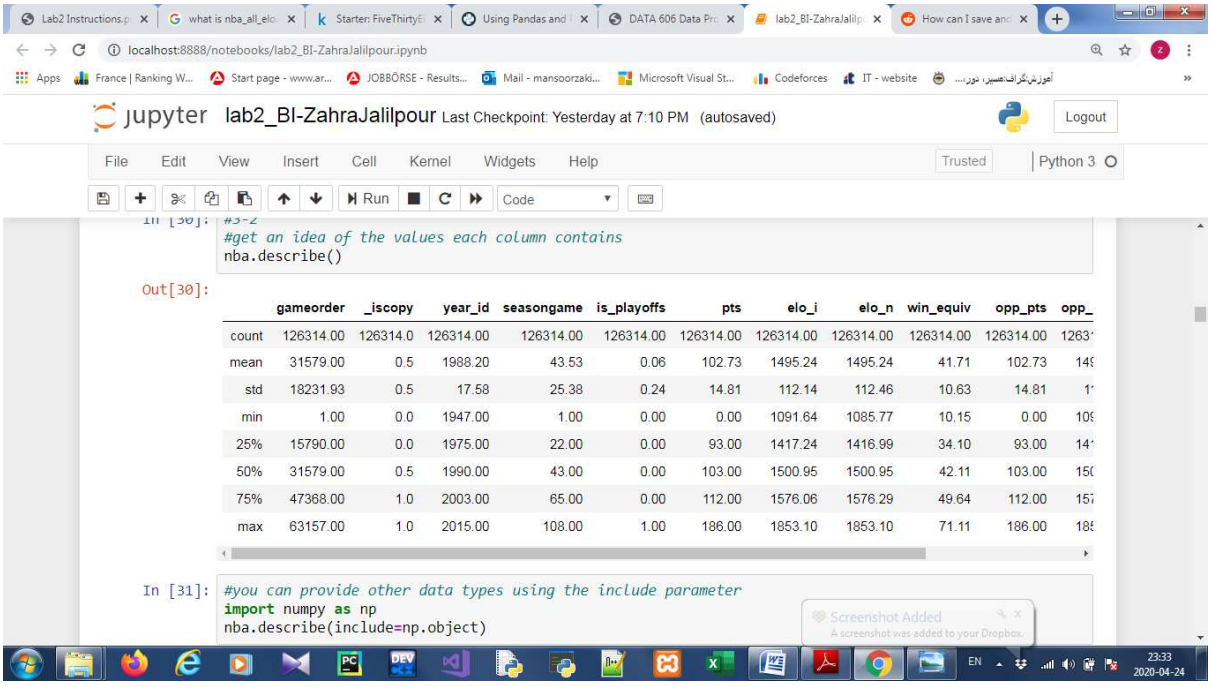
By **nba.info()**, we will see all columns by their data types.



we can see the data types int64, float64, and object.

Object means that all of the values in the column are strings.

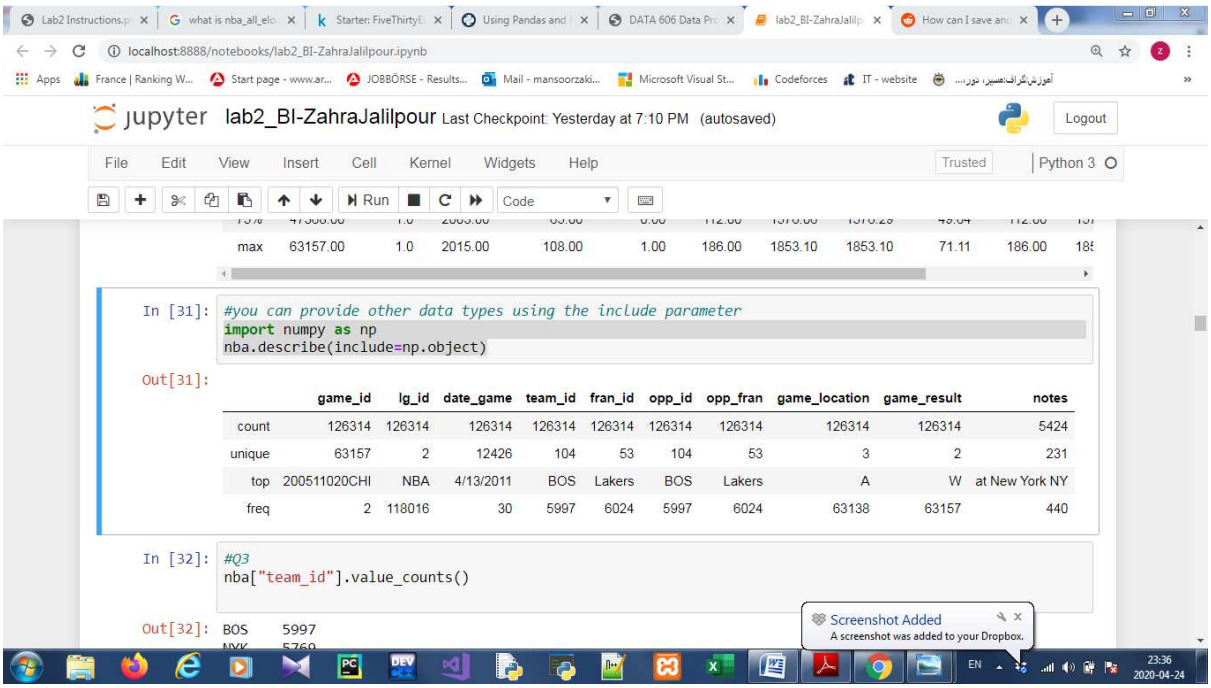
Now we can see an overview of the values each column contains. By this command **nba.describe():**



.describe() only analyzes numeric columns but we canprovide other data types if we use the include parameter:

import numpy as np

nba.describe(include=np.object)



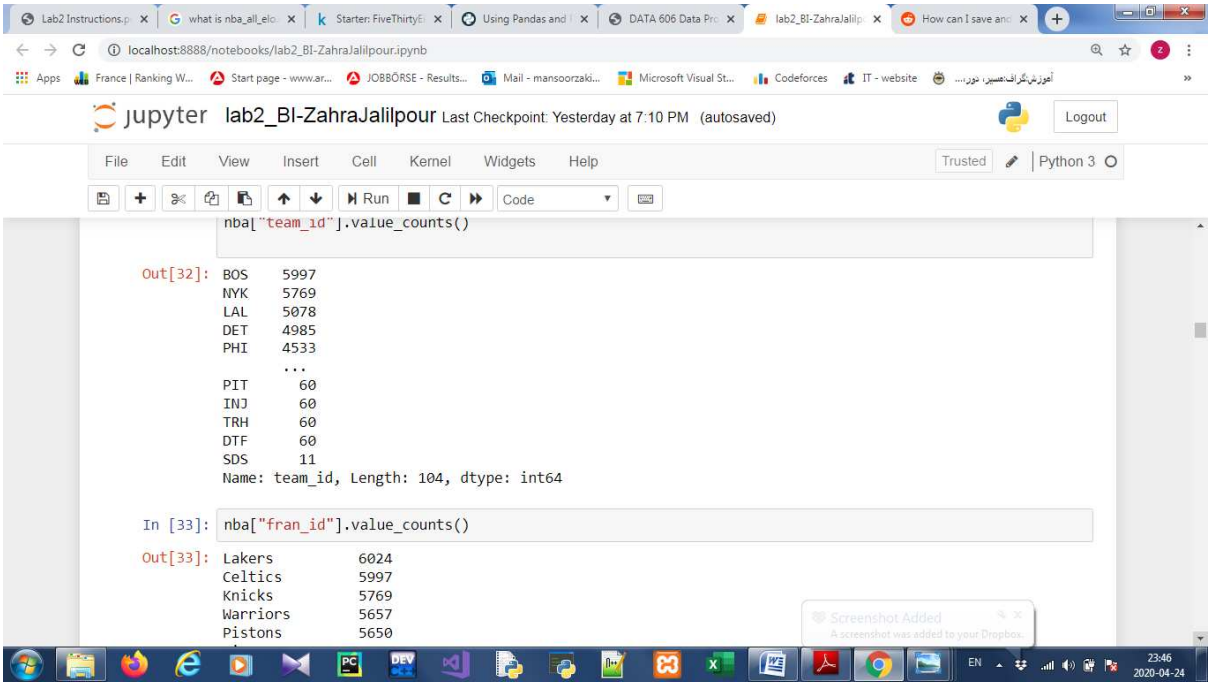
Q2: Take a look at the team_id and fran_id (franchise) columns, what observations can you make at this point (i.e. do you see anything strange here)?

By looking at team_id and fran_id, we see that our dataset contains 104 different team id, but 53 different fran id, and the most frequent team id is BOS, but the most frequent fran id is Lakers.

3.3-Exploring the dataset

nba["team_id].value_counts()

nba["fran_id"].value.counts()



nba.loc[nba["fran_id"] == "Lakers", "team_id"].value_counts()

```
LAL     5078
MNL      946
Name: team_id, dtype: int64
```

The Minneapolis Lakers ("MNL") played 946 games.

Q3: (report your answer): Find out how many wins and losses the Minneapolis Lakers had, also find how many points they scored during the matches contained in the dataset.

nba.loc[nba["team_id"] == "MNL", "game_result"].value_counts()

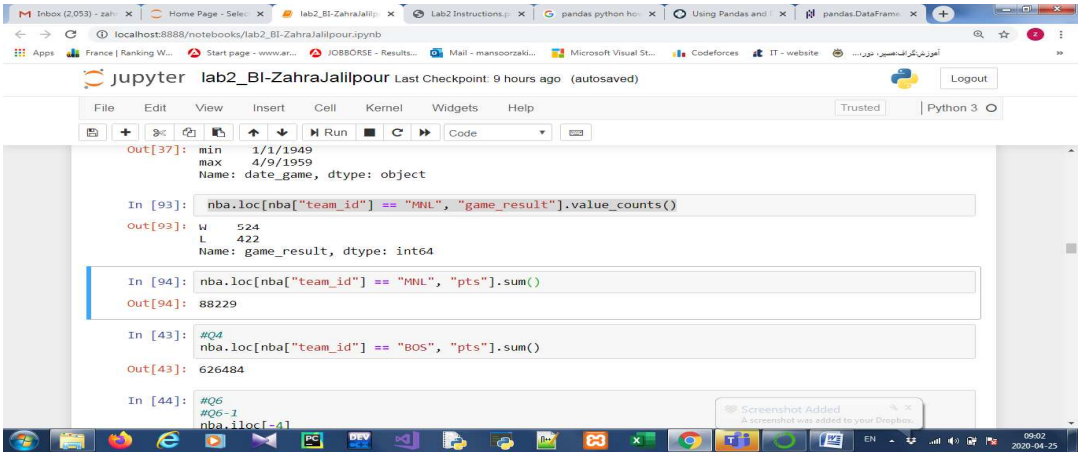
```
W    524
L    422
Name: game_result, dtype: int64
```

The Minneapolis Lakers ("MNL") played 946 games and during these games , the number of win is 524 and the number of loss is 422.

nba.loc[nba["team_id"] == "MNL", "pts"].sum()

```
88229
```

The number of points of "MNL" is equal to 88229.



Q4: (report your answer): Now you understand why the Boston Celtics team "BOS" played the most games in the dataset, find out how many points the Boston Celtics have scored during all matches contained in this dataset.

By this command we can find the total points that Boston have scored during all matches:

nba.loc[nba["team_id"] == "BOS", "pts"].sum()

The Boston Celtics scored a total of 626,484 points.

Q5: (report your answer): After having explored your dataset, explain your observations from Question.2 in a structured way.

It looks like the Minneapolis Lakers played between the years of 1949 and 1959 and "LAL" played between the years of 1961 and 2009 and Boston played between 1948 and 2012. That explains why we might not recognize Minneapolis Lakers team!

We've also found out why the Boston Celtics team "BOS" played the most games in the dataset.

Q6-

6.1) Use a data access method to display the 4th row from the bottom of the nba dataset:

nba.iloc[-4]

```
gameorder                63156
game_id           201506140GSW
lg_id                      NBA
 iscopy                      0
year_id                  2015
date_game            6/14/2015
seasongame                 102
is_playoffs                  1
team_id                    GSW
fran_id               Warriors
pts                        104
elo_i                   1.8e+03
```

```
elo_n                 1.8e+03
win_equiv                  68
opp_id                    CLE
opp_fran            Cavaliers
opp_pts                    91
opp_elo_i             1.7e+03
opp_elo_n             1.7e+03
game_location               H
game_result                 W
forecast                 0.77
notes                     NaN
Name: 126310, dtype: object
```

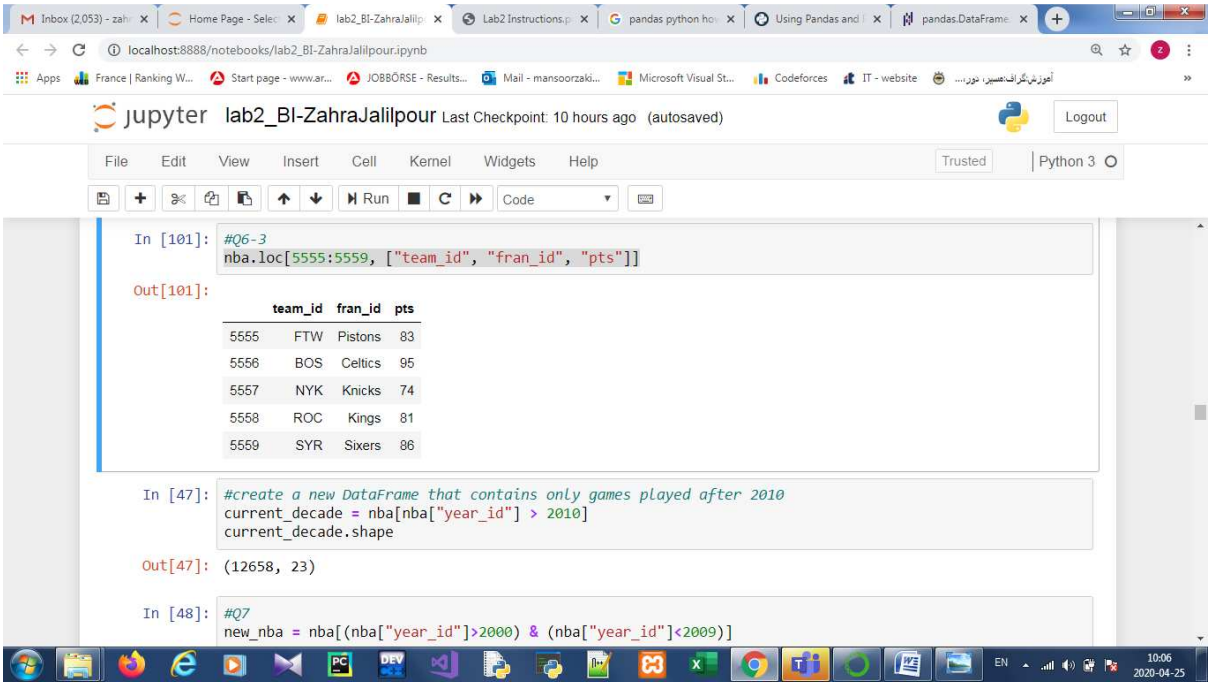
6.2) Use a data access method to display the 2nd row from the top of the nba dataset.

nba.iloc[1]

```
gameorder                   1
game_id           194611010TRH
lg_id                      NBA
_iscopy                      1
year_id                   1947
date_game           11/1/1946
seasongame                   1
is_playoffs                  0
team_id                    NYK
fran_id                Knicks
pts                         68
elo_i                 1.3e+03
elo_n                 1.3e+03
win_equiv                   42
opp_id                     TRH
opp_fran               Huskies
opp_pts                     66
opp_elo_i             1.3e+03
opp_elo_n             1.3e+03
game_location                A
game_result                  W
forecast                  0.36
notes                      NaN
Name: 1, dtype: object
```

6.3) Access all games between the labels 5555 and 5559, you only want to see the names of teams and the scores.

nba.loc[5555:5559, ["team_id", "fran_id", "pts"]]



Question.7 (report your answer): Create a new DataFrame which consists of the games played between 2000 and 2009.

new_nba = nba[(nba["year_id"]>=2000) & (nba["year_id"]<=2009)]

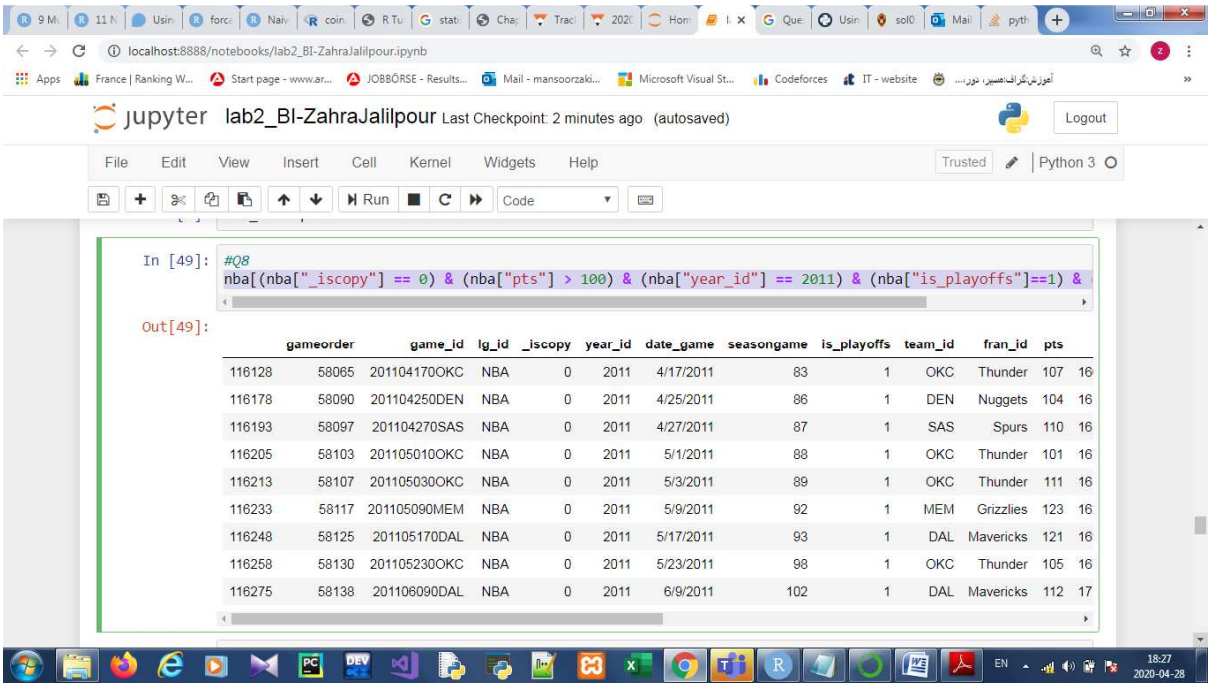
new_nba.shape

```
(25810, 23)
```

Question 8:

Filter your dataset and find all the playoffs games where the number of points scored by both home and away is more than 100, in the year 2011 and make sure you don't include duplicates (don't forget the parentheses).

nba[(nba["_iscopy"] == 0) & (nba["pts"] > 100) & (nba["year_id"] == 2011) & (nba["is_playoffs"]==1) & (nba["opp_pts"]>100)]

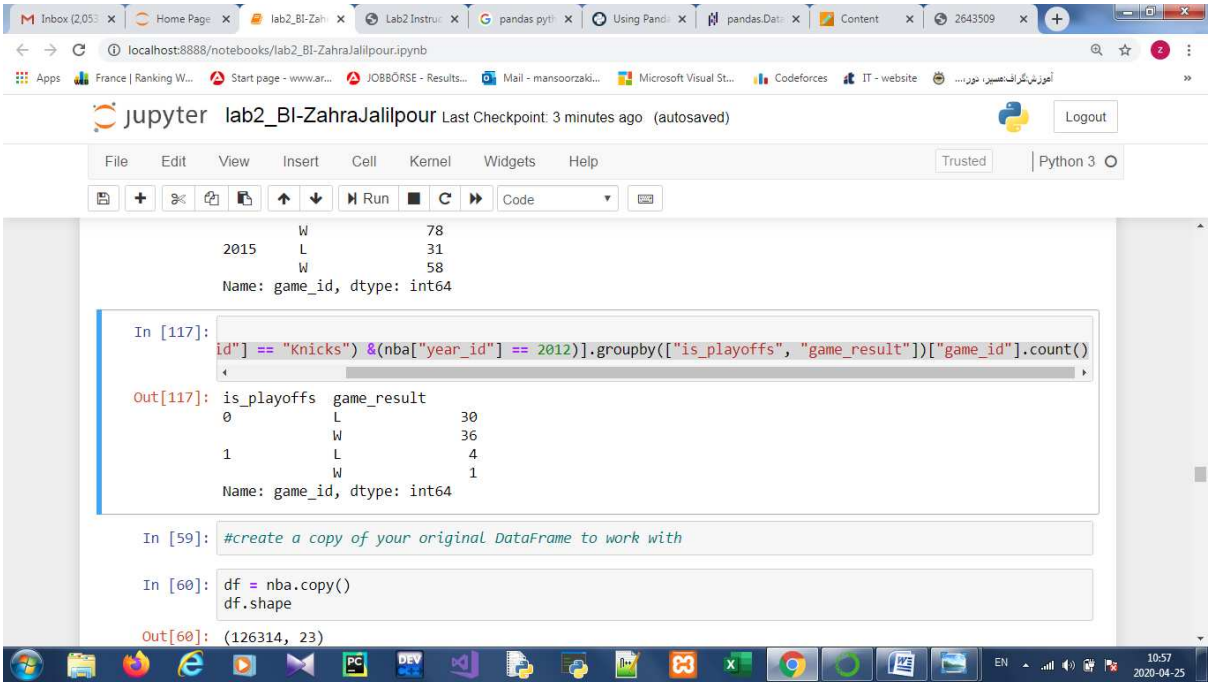


	gameorder	game_id	lg_id	_iscopy	year_id	date_game	seasongame	is_playoffs	team_id	fran_id	pts	
116128	58065	201104170OKC	NBA	0	2011	4/17/2011	83	1	OKC	Thunder	107	16
116178	58090	201104250DEN	NBA	0	2011	4/25/2011	86	1	DEN	Nuggets	104	16
116193	58097	201104270SAS	NBA	0	2011	4/27/2011	87	1	SAS	Spurs	110	16
116205	58103	201105010OKC	NBA	0	2011	5/1/2011	88	1	OKC	Thunder	101	16
116213	58107	201105030OKC	NBA	0	2011	5/3/2011	89	1	OKC	Thunder	111	16
116233	58117	201105090MEM	NBA	0	2011	5/9/2011	92	1	MEM	Grizzlies	123	16
116248	58125	201105170DAL	NBA	0	2011	5/17/2011	93	1	DAL	Mavericks	121	16
116258	58130	201105230OKC	NBA	0	2011	5/23/2011	98	1	OKC	Thunder	105	16
116275	58138	201106090DAL	NBA	0	2011	6/9/2011	102	1	DAL	Mavericks	112	17

Question.9 (report your answer): Take a look at the New York Knicks 2011-12 season (year_id: 2012). How many wins and losses did they score during the regular season and the playoffs?

nba[(nba["fran_id"] == "Knicks") &(nba["year_id"] == 2012)].groupby(["is_playoffs", "game_result"])["game_id"].count()

```
is_playoffs  game_result
0            L              30
             W              36
1            L               4
             W               1
Name: game_id, dtype: int64
```



Question.10 (report your answer): Find another column in the nba dataset that has a generic data type and convert it to a more specific one.

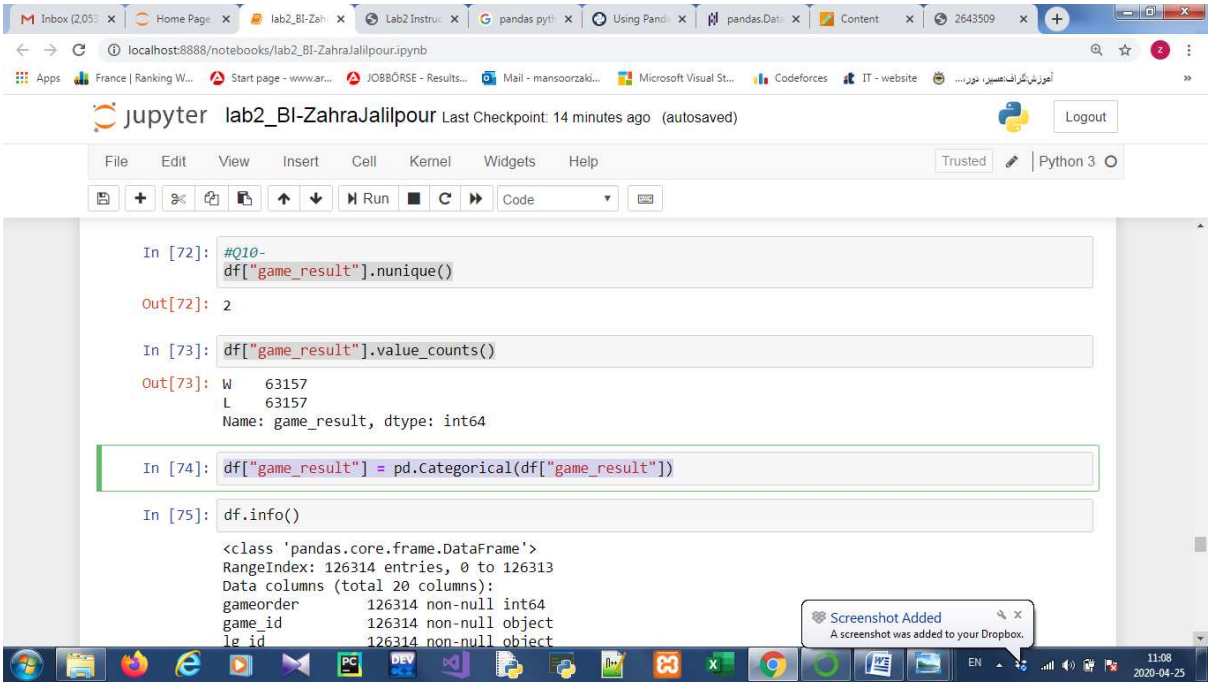
By using df.info() we see that ten of your columns have the data type object. Some of these objects can be converted to specific data type, like game_date, game_location and game_result.

When we specify the categorical data type, we make validation easier and save a ton of memory.

df["game_result"].nunique()

df["game_result"].value_counts()

df["game_result"] = pd.Categorical(df["game_result"])

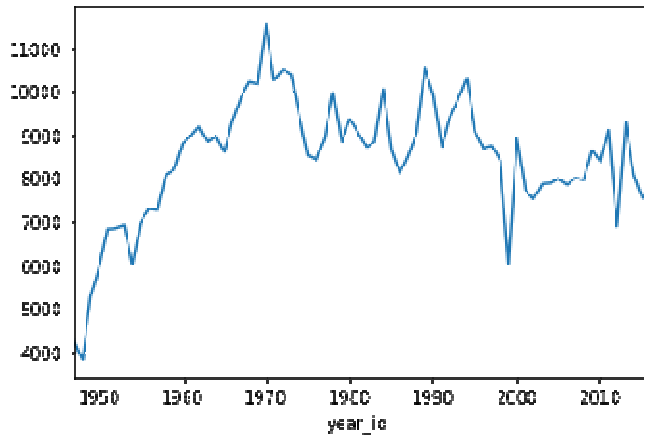


Question.10 (report your answer):

10.1) Explain what the above line plot, showing how many points the Knicks scored throughout the seasons, reveals to you (i.e. describe what you find out).

%matplotlib inline

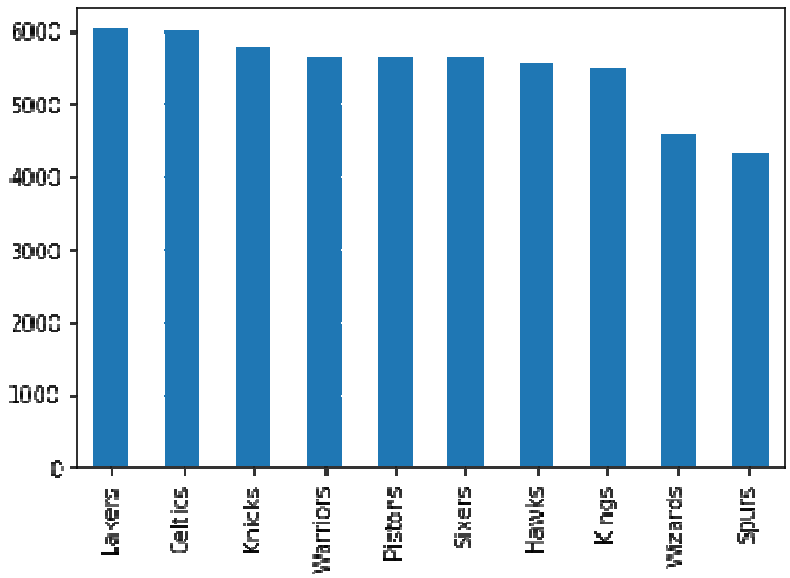
nba[nba["fran_id"] == "Knicks"].groupby("year_id")["pts"].sum().plot()



This shows a line plot with several peaks and two notable valleys around the years 2000 and 2010.

10.2) Describe what the above bar plot reveals to you about the franchises with the most games played.

nba["fran_id"].value_counts().head(10).plot(kind="bar")



The Lakers and Celtics have the most games, and there are six further teams with a game count above 5000.