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.csvin 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**.

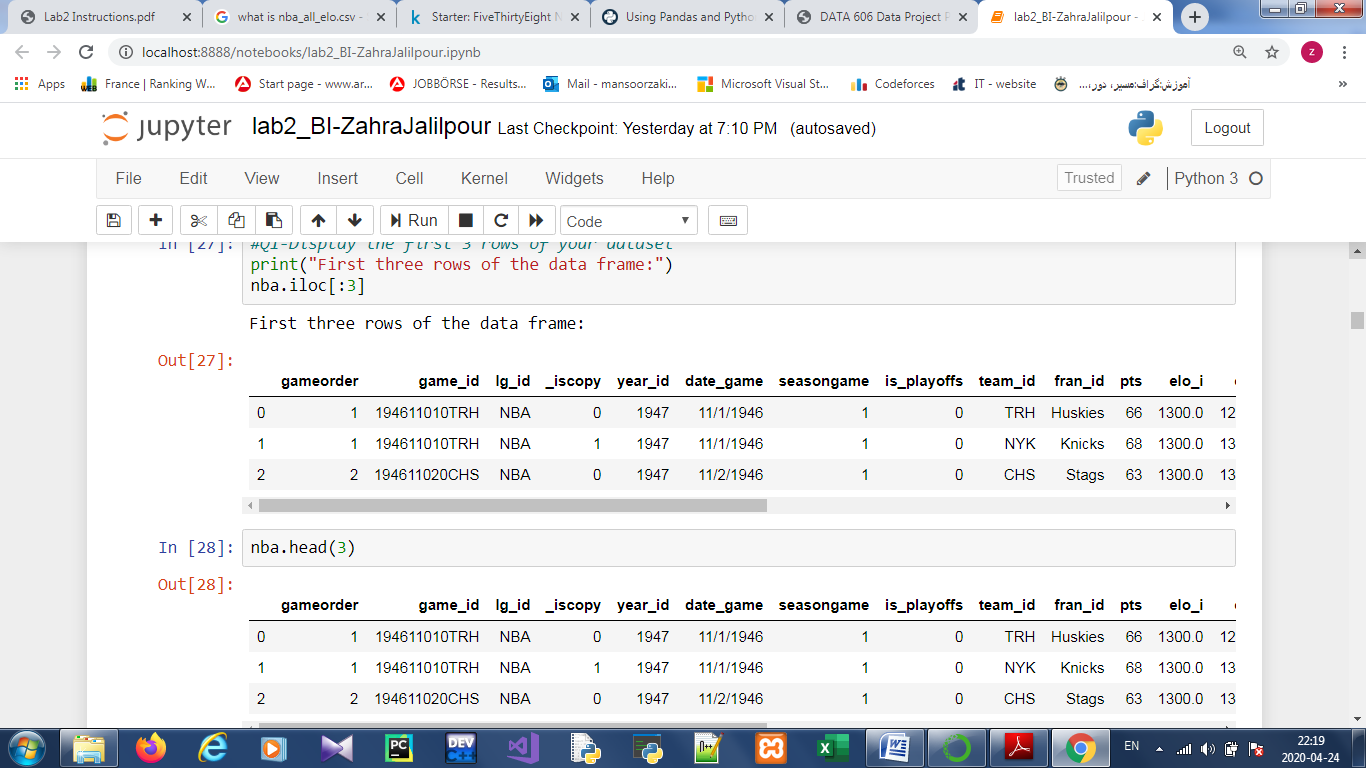
**Q1: Display the first 3 rows of your dataset.**

- It can be done through different commands:

nba.iloc[:3]

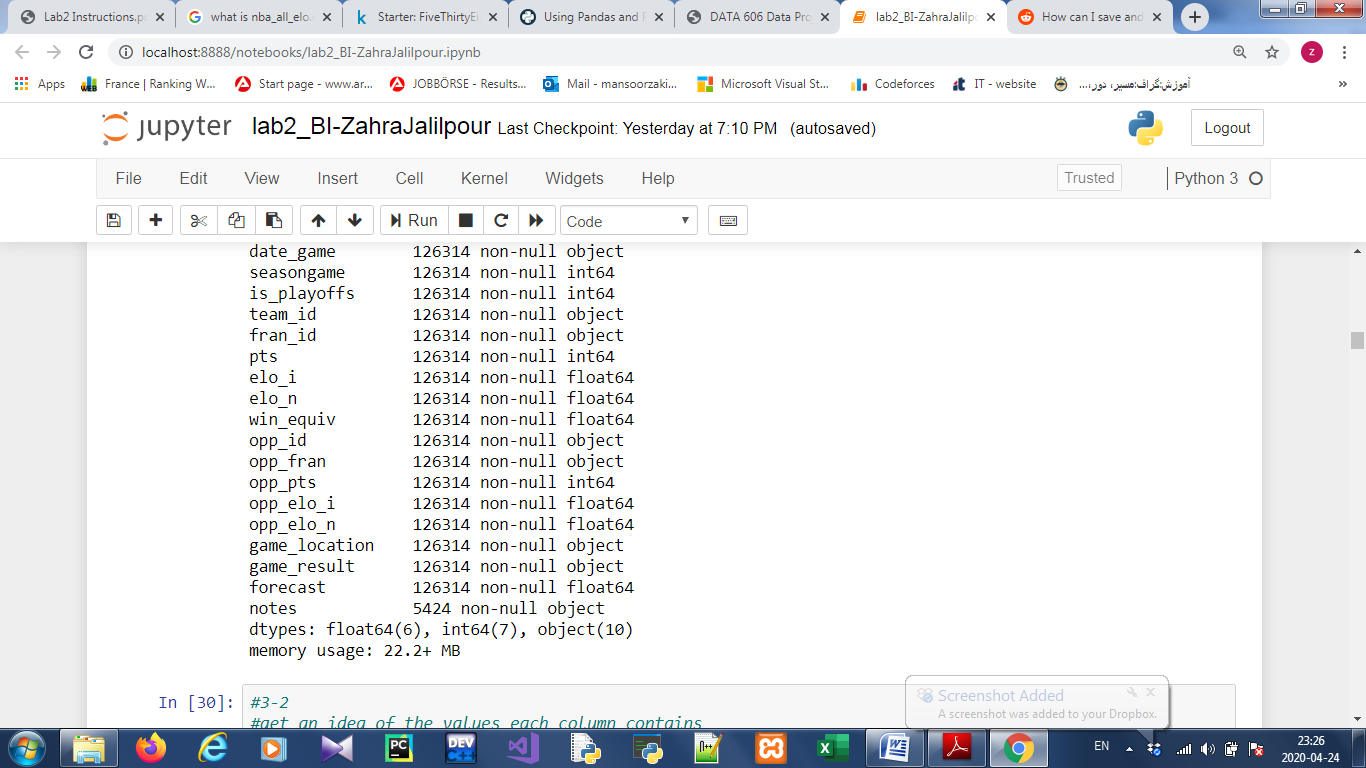
or

nba.head(3)

****

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.

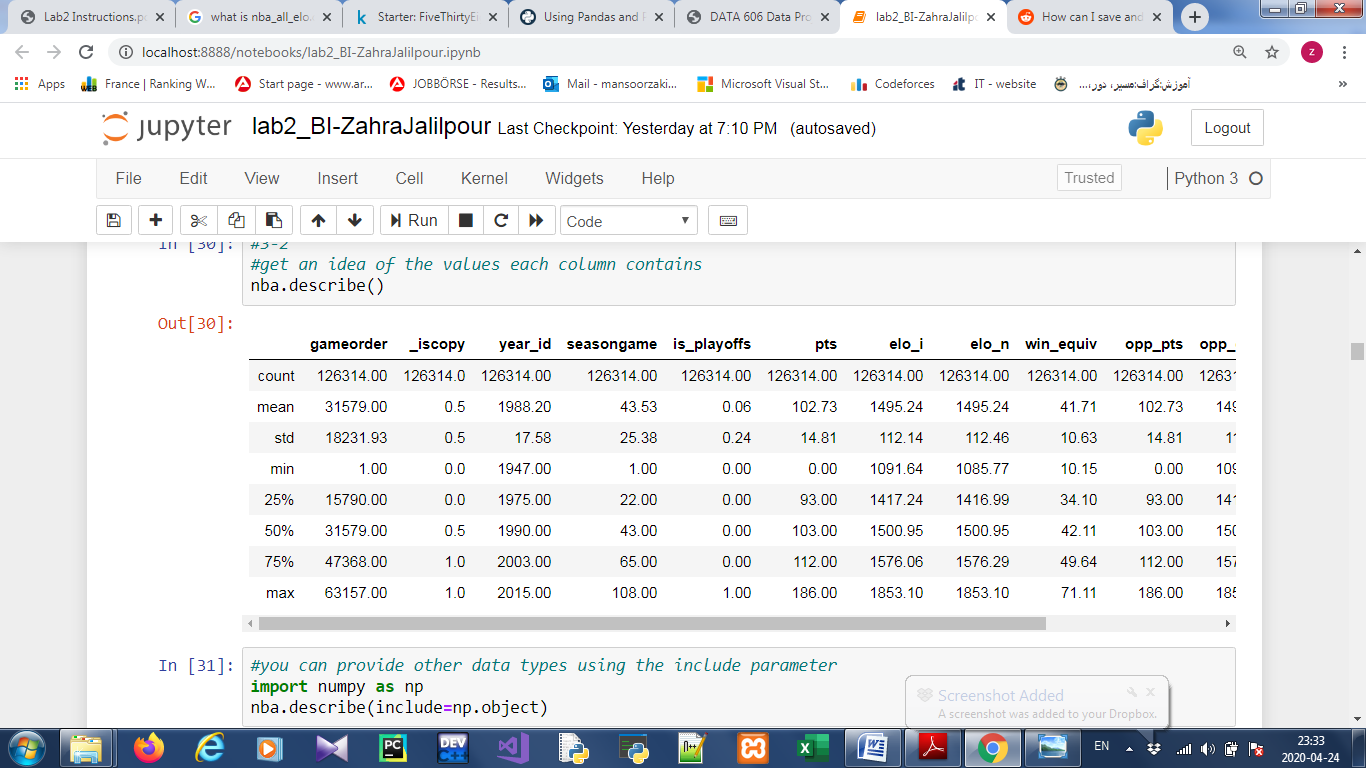
By **nba.info()**, we will see all columns by their data types.



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

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

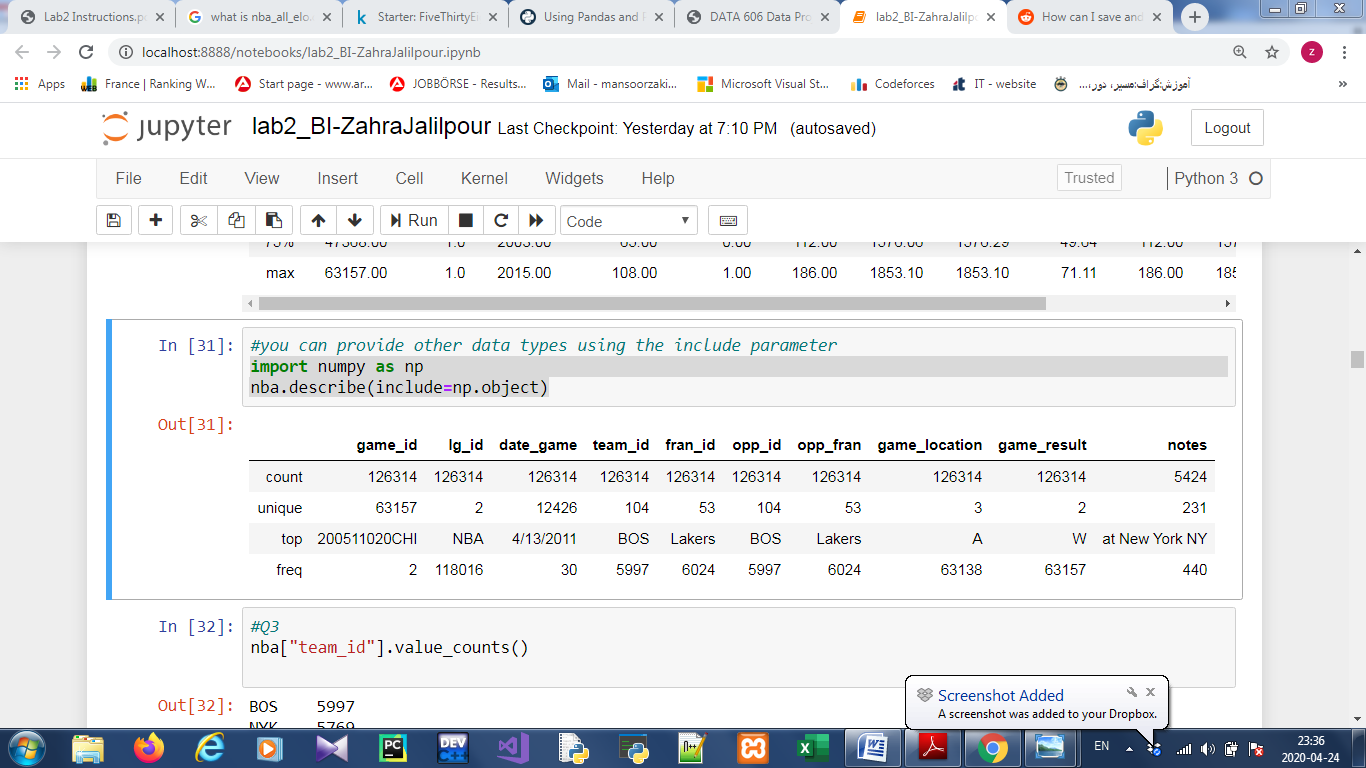
Now we can see an overview of the values each column contains. By this command **nba.describe():**



.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)**

****

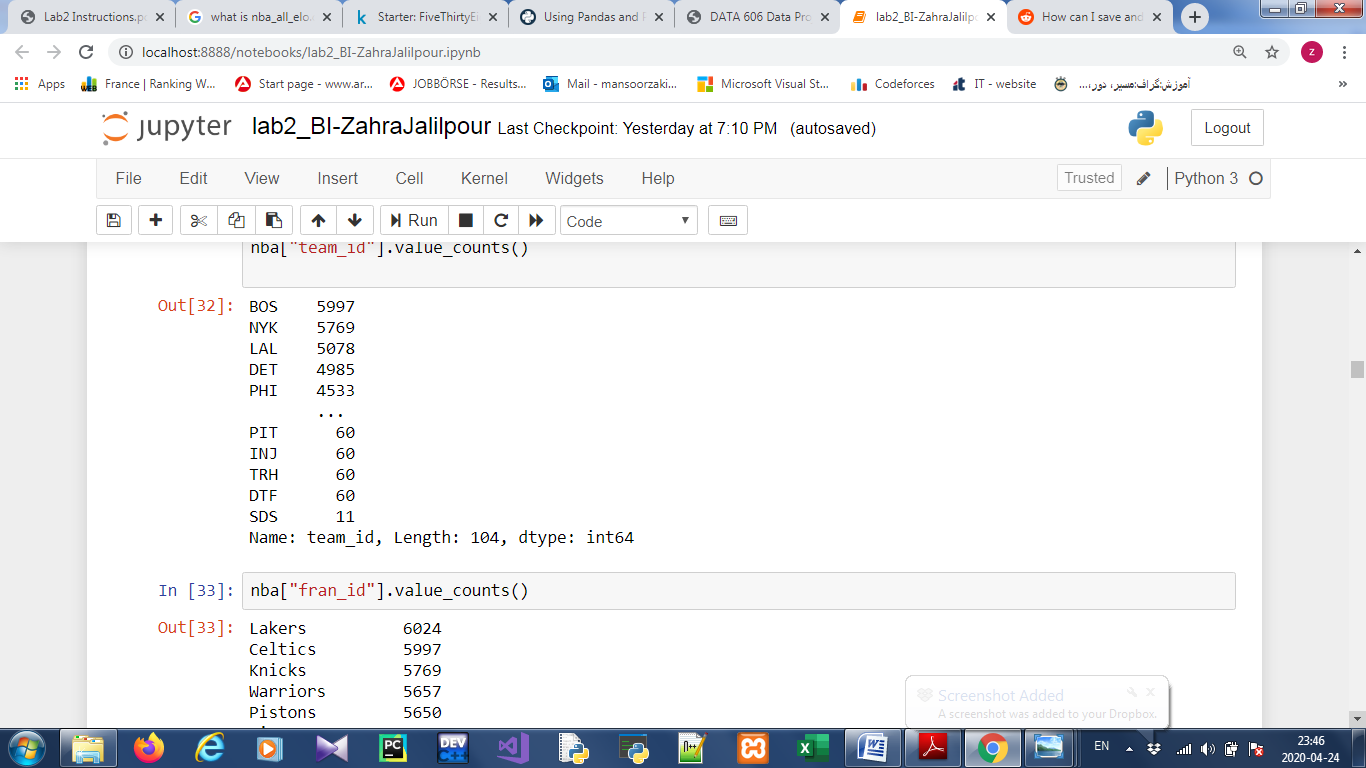
**Q2: Take a look at the team\_id and fran\_id (franchise) columns, what observations can you make at this point (i.e. do you see anything strange here)?**

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

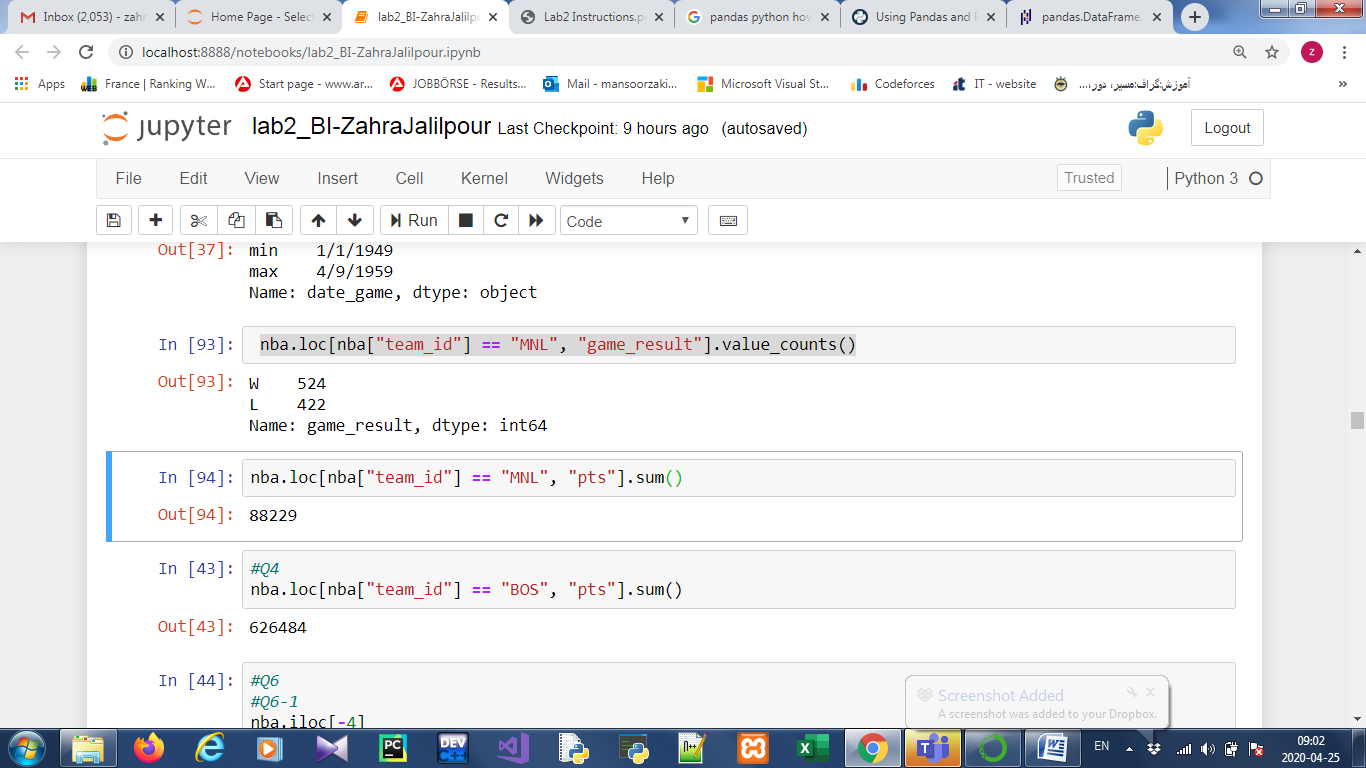
Name: game\_result, dtype: int64

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

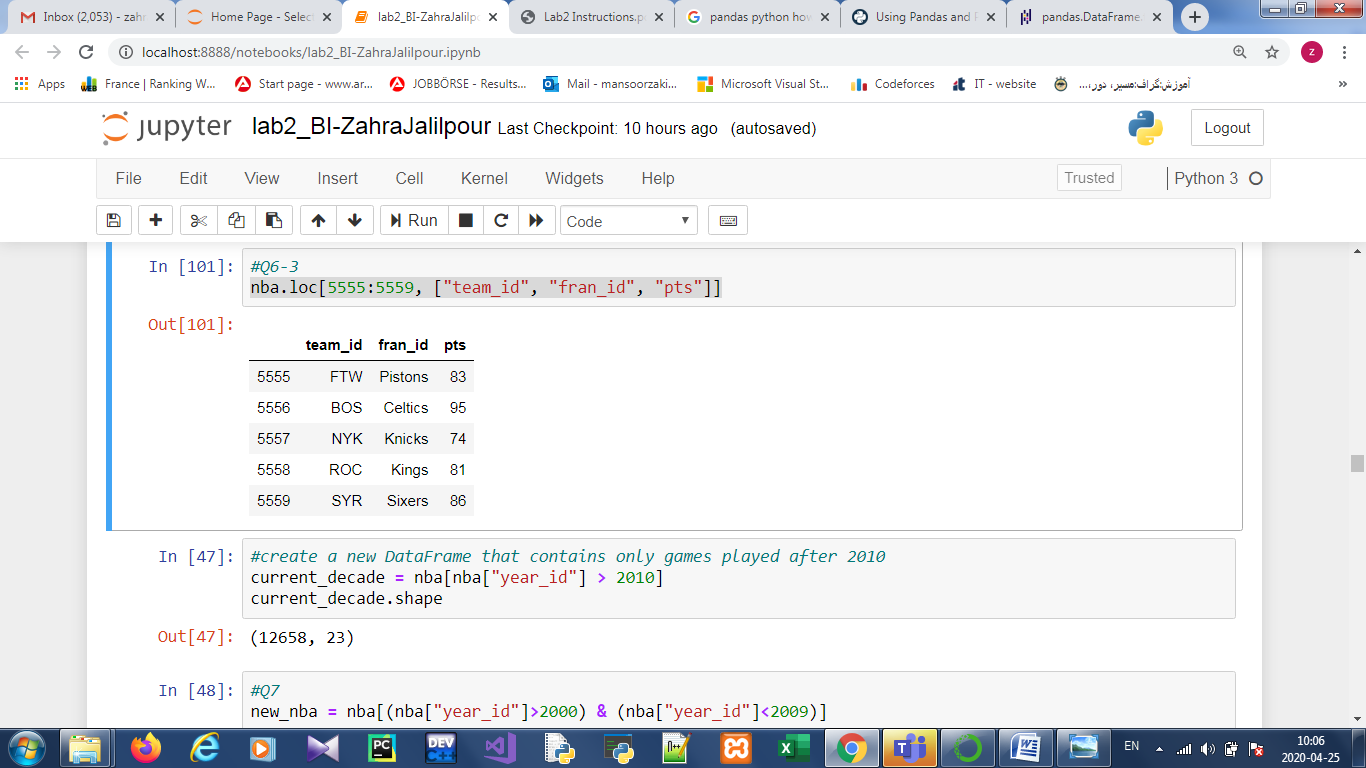
forecast 0.36

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.

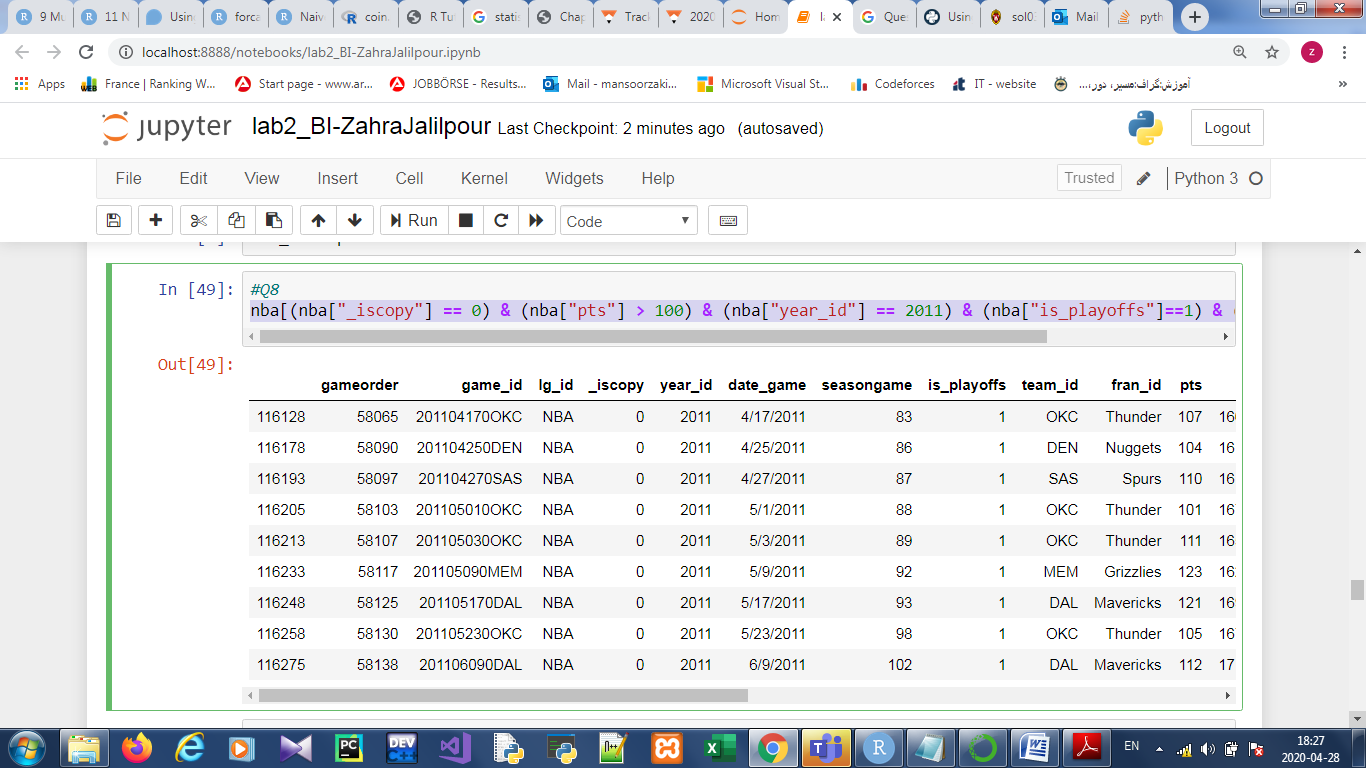
new\_nba = nba[(nba["year\_id"]>=2000) & (nba["year\_id"]<=2009)]

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) ]



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

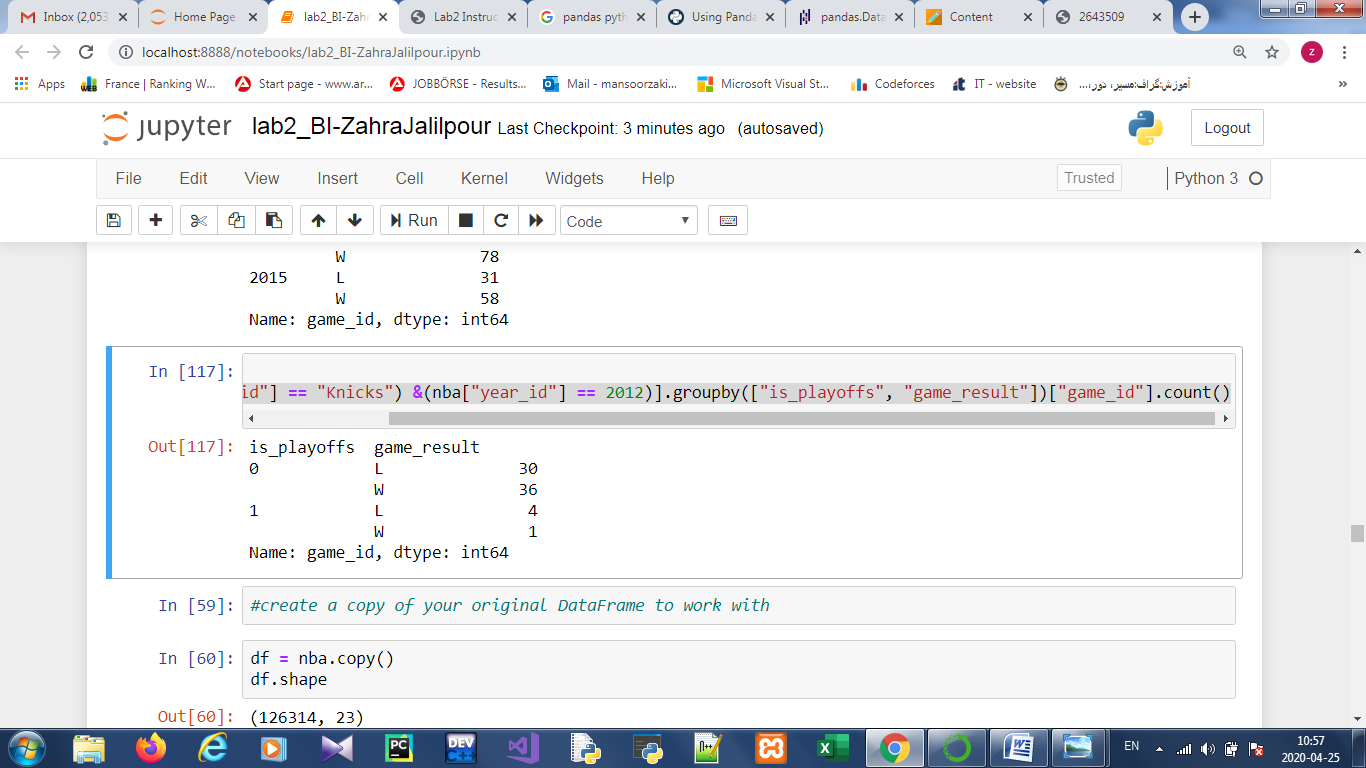
0 L 30

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.

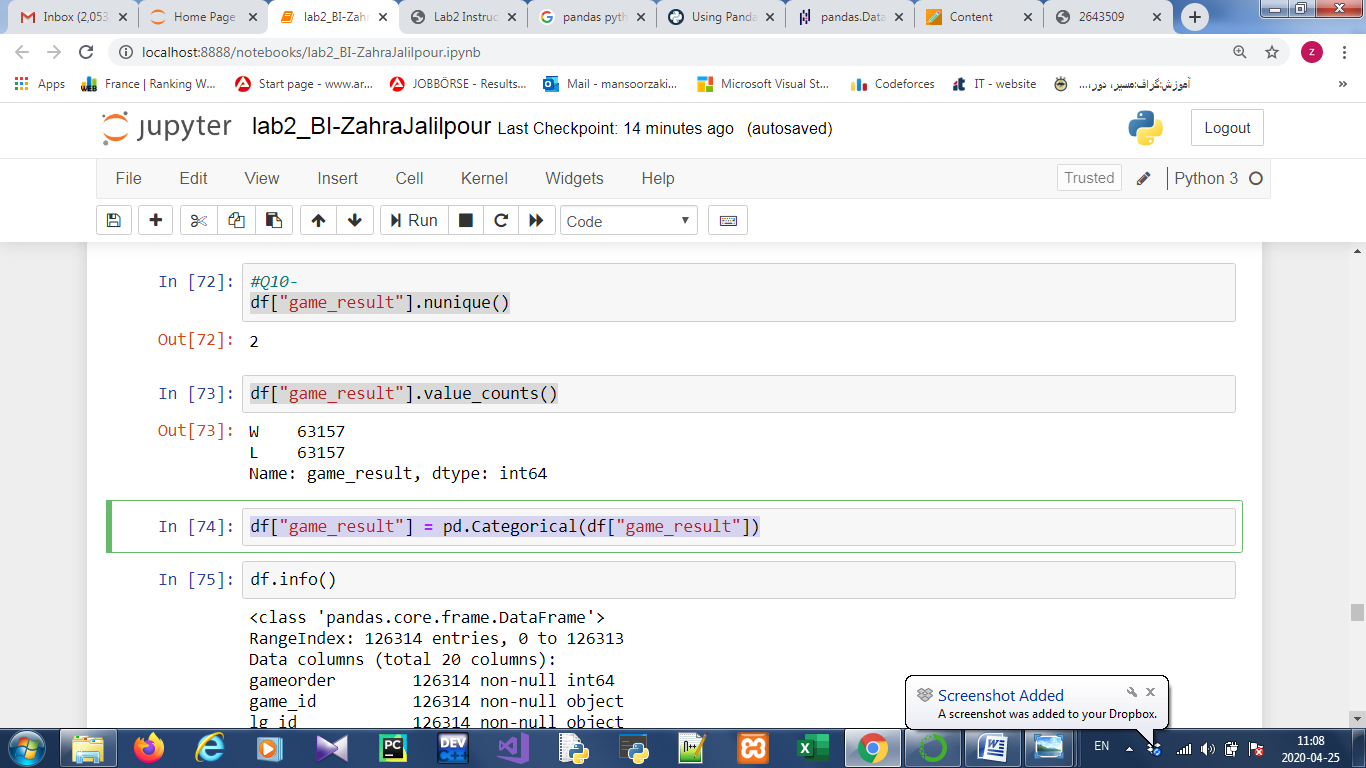
By using df.info() we see that ten of your columns have the data type object. Some of these objects can be converted to specific data type, like game\_date, game\_location and game\_result.

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"])

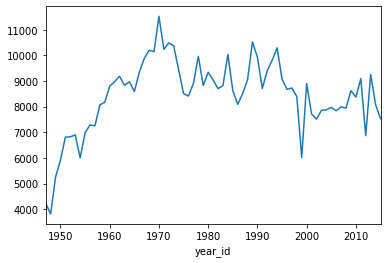


Question.10 (report your answer):

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

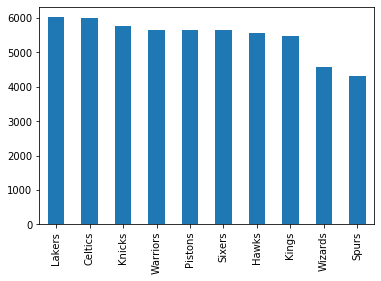
nba[nba["fran\_id"] == "Knicks"].groupby("year\_id")["pts"].sum().plot()



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.