Business Intelligence
Lab Report
On
Exploring a Dataset using Pandas

HÖGSKOLAN
DALARNA

By
Manthri Bala Kiran
H19baman@du.se

Introduction:

In this Lab we would be analysing NBA results provided by FiveThrityEight which is provided in a CSV file. We can download the same CSV file from the web.

Using the dataset we would be analysing the following things:

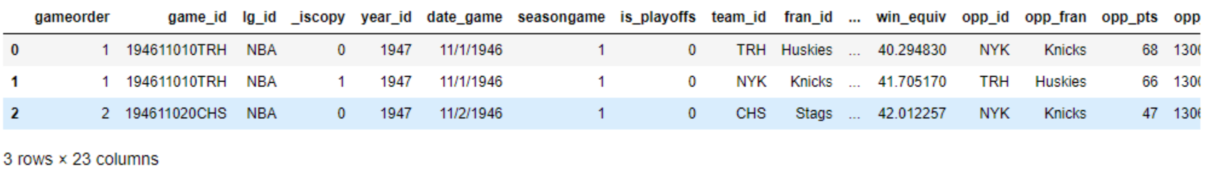
- Calculate metrics about your data
- Perform basic queries and aggregations
- Discover and handle incorrect data, inconsistencies, and missing values.
- Visualize your data with plots

Results:

Question.1 (report your answer): Display the first 3 rows of your dataset. Remember that the default of nba.head() shows the first 5 rows.

ANS) nba.head(3)

Normally the head() function would be giving the first 5 rows of the dataset. If we want the first three rows we need to mention as 3.



	gameorder	game_id	lg_id	_iscopy	year_id	date_game	seasongame	is_playoffs	team_id	fran_id	...	win_equiv	opp_id	opp_fran	opp_pts	opp
0	1	194611010TRH	NBA	0	1947	11/1/1946	1	0	TRH	Huskies	...	40.294830	NYK	Knicks	68	130
1	1	194611010TRH	NBA	1	1947	11/1/1946	1	0	NYK	Knicks	...	41.705170	TRH	Huskies	66	130
2	2	194611020CHS	NBA	0	1947	11/2/1946	1	0	CHS	Stags	...	42.012257	NYK	Knicks	47	130

3 rows × 23 columns

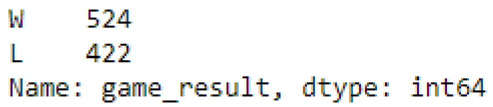
Question.2 (report your answer): Take a look at the team_id and fran_id (franchise) columns, what observations can you make at this point (i.e. do you see anything strange here)? Write your initial observation then carry on with section 3.3 to be able to answer it by exploring your dataset.

ANS) It looks like the data set contains 104 unique team ids but only 53 unique franchise ids. And the most played team is BOS and franchise is Lakers. This is may be because there may be any another team under the franchise. Usually a franchise could have multiple teams under them, or they may be any change of the name for the team. Let us see the data and get conclude on this.

Question.3 (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.

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

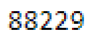
We have used loc() function from the pandas library which is used to access multiple columns and rows by their label or Boolean array. In the above query we have used loc() function which is accessing the team_id for the MNL and the game results. And we are using the value_count() function to get the count for the values in the game_results column for the team id “MNL”



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

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

Now we are using loc() function for the columns team_id and the points column for the team “MNL” and then we are using the sum function to add all the points for the team “MNL”.

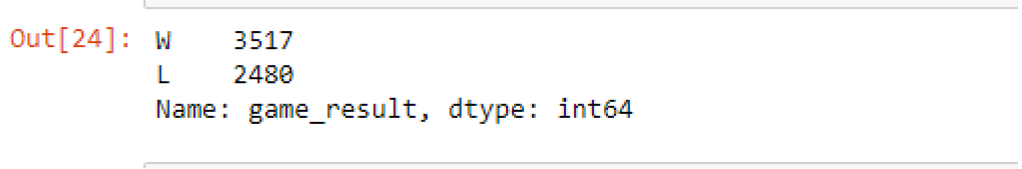


```
88229
```

Question.4 (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.

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

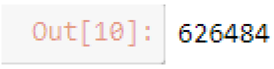
We are using loc() function to combine the team_id and game_results column and using the value_count() function to see how many games did “BOS” team won and lost.



```
Out[24]: W    3517
         L    2480
         Name: game_result, dtype: int64
```

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

Now we are trying to find the total sum of the points for the “BOS” team by using the sum() function.



```
Out[10]: 626484
```

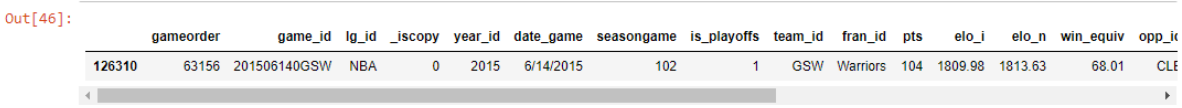
Question.5 (report your answer): After having explored your dataset, explain your observations from Question.2 in a structured way.

ANS) We see that the Lakers Franchise has the highest played matches but under Lakers we had two different teams one was MNL which was playing during the years 1949-1959 they later they have new team playing under the franchise which is LAL. So according to the most player matches count BOS stands at the first.

Question.6 (report your answer):
6.1) Use a data access method to display the 4th row from the bottom of the nba dataset.

ANS) nba.iloc[[-4]]

iloc() function in the pandas library is used access rows and columns by indexing. To display the last 4th row of the dataset we need to mention as -4 in the parameters for the iloc() function.

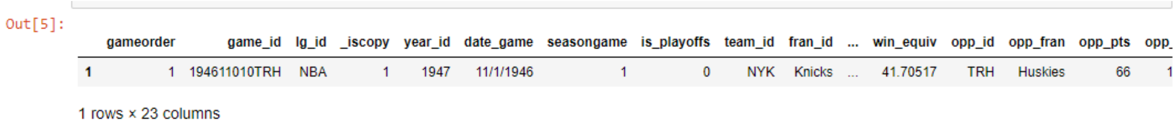


Out[46]:

	gameorder	game_id	lg_id	_iscopy	year_id	date_game	seasongame	is_playoffs	team_id	fran_id	pts	elo_i	elo_n	win_equiv	opp_id
126310	63156	201506140GSW	NBA	0	2015	6/14/2015	102	1	GSW	Warriors	104	1809.98	1813.63	68.01	CLE

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

ANS) nba.iloc[[1]]



Out[5]:

	gameorder	game_id	lg_id	_iscopy	year_id	date_game	seasongame	is_playoffs	team_id	fran_id	...	win_equiv	opp_id	opp_fran	opp_pts	opp_
1	1	194611010TRH	NBA	1	1947	11/1/1946	1	0	NYK	Knicks	...	41.70517	TRH	Huskies	66	1

1 rows × 23 columns

The second row from the top would be the row which has the index value 1.

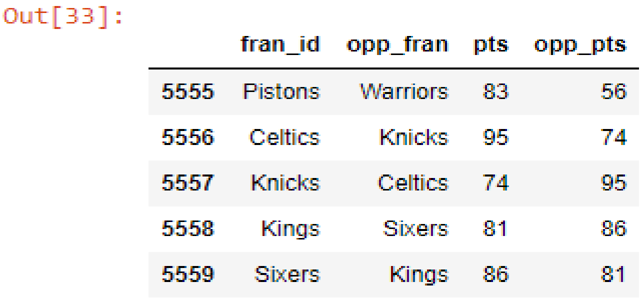
6.3) Access all games between the labels 5555 and 5559, you only want to see the names of teams and the scores.

ANS) nba.iloc[5555:5559]

In the above query we have requesting for all the columns information from the specific rows in the dataset.

2 - nba.loc[5555:5559, ["fran_id", "opp_fran", "pts", "opp_pts"]]

In the above query we are using loc() function to access from the specific rows of the dataset and we are mentioning the required columns so to print on the columns which are required.



Out[33]:

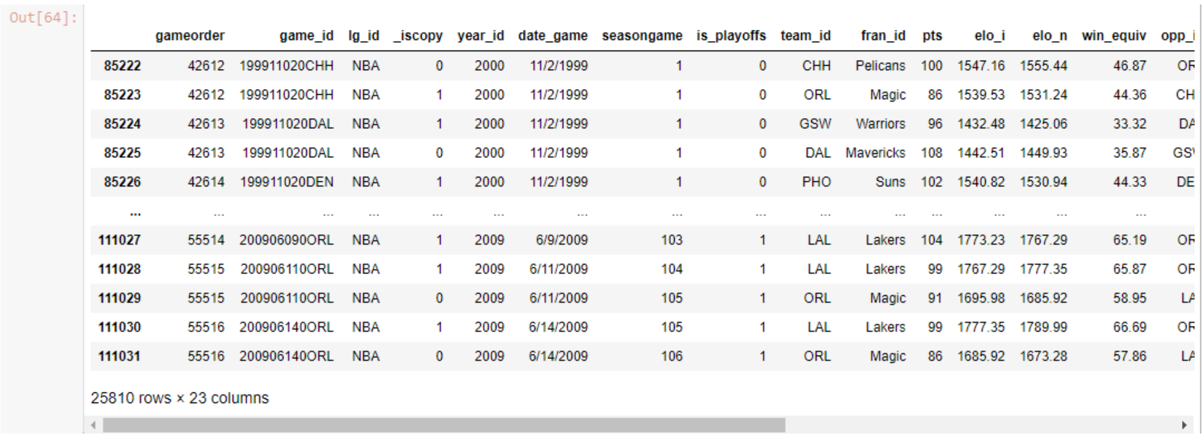
	fran_id	opp_fran	pts	opp_pts
5555	Pistons	Warriors	83	56
5556	Celtics	Knicks	95	74
5557	Knicks	Celtics	74	95
5558	Kings	Sixers	81	86
5559	Sixers	Kings	86	81

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

ANS) games_00_09 = nba[(nba["year_id"]>=2000) & (nba["year_id"]<=2009)]

games_00_09

Now we are creating a subset for the dataset by giving some conditions with the year_id. From the above query a new dataset would be created and we are calling the subset of the data as games_00_09.



Out[64]:

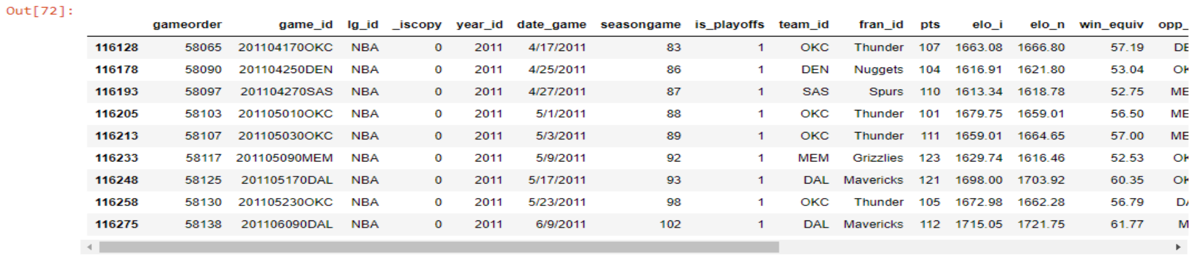
	gameorder	game_id	lg_id	_iscopy	year_id	date_game	seasongame	is_playoffs	team_id	fran_id	pts	elo_i	elo_n	win_equiv	opp_i
85222	42612	199911020CHH	NBA	0	2000	11/2/1999	1	0	CHH	Pelicans	100	1547.16	1555.44	46.87	OF
85223	42612	199911020CHH	NBA	1	2000	11/2/1999	1	0	ORL	Magic	86	1539.53	1531.24	44.36	CH
85224	42613	199911020DAL	NBA	1	2000	11/2/1999	1	0	GSW	Warriors	96	1432.48	1425.06	33.32	DA
85225	42613	199911020DAL	NBA	0	2000	11/2/1999	1	0	DAL	Mavericks	108	1442.51	1449.93	35.87	GS
85226	42614	199911020DEN	NBA	1	2000	11/2/1999	1	0	PHO	Suns	102	1540.82	1530.94	44.33	DE
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
111027	55514	200906090ORL	NBA	1	2009	6/9/2009	103	1	LAL	Lakers	104	1773.23	1767.29	65.19	OF
111028	55515	200906110ORL	NBA	1	2009	6/11/2009	104	1	LAL	Lakers	99	1767.29	1777.35	65.87	OF
111029	55515	200906110ORL	NBA	0	2009	6/11/2009	105	1	ORL	Magic	91	1695.98	1685.92	58.95	LA
111030	55516	200906140ORL	NBA	1	2009	6/14/2009	105	1	LAL	Lakers	99	1777.35	1789.99	66.69	OF
111031	55516	200906140ORL	NBA	0	2009	6/14/2009	106	1	ORL	Magic	86	1685.92	1673.28	57.86	LA

25810 rows × 23 columns

Question.8 (report your answer): Filter your dataset and find all the playoffs games where the number of points scored by both home and aways is more than 100, in the year 2011 and make sure you don't include duplicates (don't forget the parentheses).

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

We can use the queries to filter the data based on the requirements we can display the necessary data. Now we are looking at only the playoff games which have been played in the year 2011 and looking for those home and away teams who have scored more than 100 points during the game and we are also eliminating the duplicates.



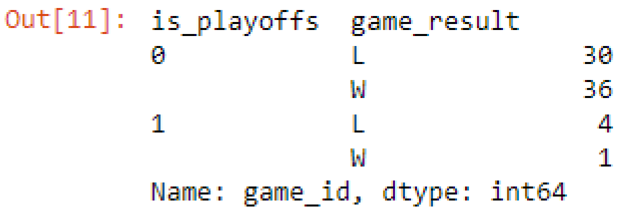
Out[72]:

	gameorder	game_id	lg_id	_iscopy	year_id	date_game	seasongame	is_playoffs	team_id	fran_id	pts	elo_i	elo_n	win_equiv	opp_
116128	58065	201104170OKC	NBA	0	2011	4/17/2011	83	1	OKC	Thunder	107	1663.08	1666.80	57.19	DE
116178	58090	201104250DEN	NBA	0	2011	4/25/2011	86	1	DEN	Nuggets	104	1616.91	1621.80	53.04	OK
116193	58097	201104270SAS	NBA	0	2011	4/27/2011	87	1	SAS	Spurs	110	1613.34	1618.78	52.75	ME
116205	58103	201105010OKC	NBA	0	2011	5/1/2011	88	1	OKC	Thunder	101	1679.75	1659.01	56.50	ME
116213	58107	201105030OKC	NBA	0	2011	5/3/2011	89	1	OKC	Thunder	111	1659.01	1664.65	57.00	ME
116233	58117	201105090MEM	NBA	0	2011	5/9/2011	92	1	MEM	Grizzlies	123	1629.74	1616.46	52.53	OK
116248	58125	201105170DAL	NBA	0	2011	5/17/2011	93	1	DAL	Mavericks	121	1698.00	1703.92	60.35	OK
116258	58130	201105230OKC	NBA	0	2011	5/23/2011	98	1	OKC	Thunder	105	1672.98	1662.28	56.79	DA
116275	58138	201106090DAL	NBA	0	2011	6/9/2011	102	1	DAL	Mavericks	112	1715.05	1721.75	61.77	M

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?

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

In the above query we are looking for the team “NYK” for the year 2012 and trying to see how many games they have won and lost during the season and playoffs. In the dataset we have the column saying is_playoffs it means if the game is playoff it would be having 1 and 0 if its not. And we can also say that if its not a playoff game it would be a season game. So we are grouping the team and year with the playoff and the game_results and we are using the count() function to count the wins and losses for the games.

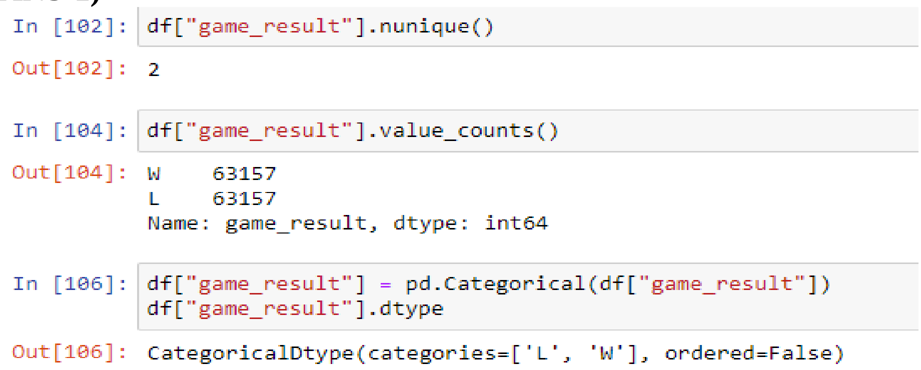


```
Out[11]: is_playoffs  game_result
         0            L              30
                      W              36
         1            L               4
                      W               1
         Name: game_id, dtype: int64
```

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.

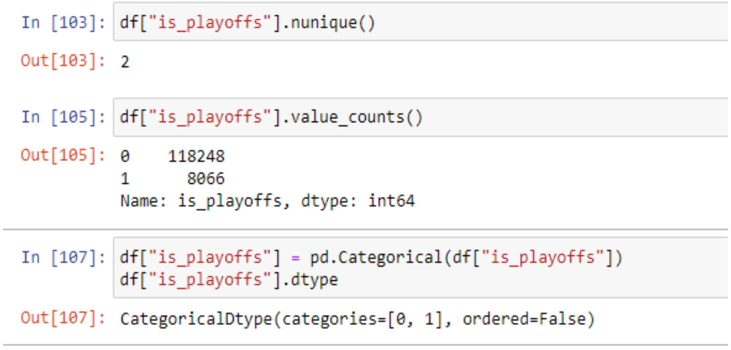
For this, I have found two different columns in the dataset which can we converted to generic data type to Categorical data values. For this, firstly I have checking the columns with the number of unique values by using the nunique() function. Then I was trying with the count of the values by using the value_count() function. Next using the pandas library saying pd.Categorical I am changing the data type for the columns game_results and is_playoffs from generic to Categorical type.

ANS-1)



```
In [102]: df["game_result"].nunique()

Out[102]: 2

In [104]: df["game_result"].value_counts()

Out[104]: W    63157
          L    63157
          Name: game_result, dtype: int64

In [106]: df["game_result"] = pd.Categorical(df["game_result"])
          df["game_result"].dtype

Out[106]: CategoricalDtype(categories=['L', 'W'], ordered=False)
```

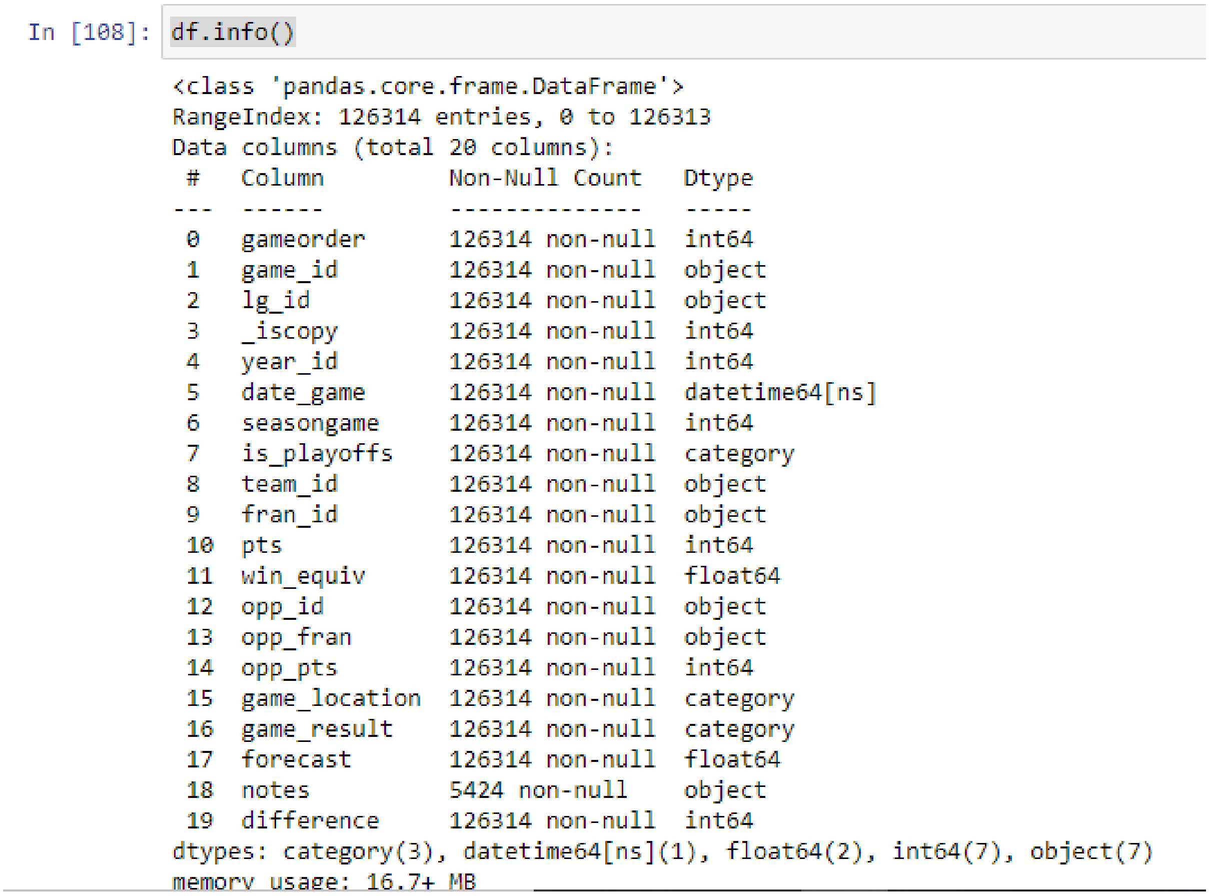
ANS-2)



```
In [103]: df["is_playoffs"].nunique()

Out[103]: 2

In [105]: df["is_playoffs"].value_counts()

Out[105]: 0    118248
          1      8066
          Name: is_playoffs, dtype: int64

In [107]: df["is_playoffs"] = pd.Categorical(df["is_playoffs"])
          df["is_playoffs"].dtype

Out[107]: CategoricalDtype(categories=[0, 1], ordered=False)
```

FINAL)

Now we can check the memory usage has been reduced by executing df.info(). By this we can say that its improving the performance.



```
In [108]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 126314 entries, 0 to 126313
Data columns (total 20 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   gameorder      126314 non-null  int64
 1   game_id        126314 non-null  object
 2   lg_id          126314 non-null  object
 3   _iscopy        126314 non-null  int64
 4   year_id        126314 non-null  int64
 5   date_game      126314 non-null  datetime64[ns]
 6   seasongame     126314 non-null  int64
 7   is_playoffs    126314 non-null  category
 8   team_id        126314 non-null  object
 9   fran_id        126314 non-null  object
 10  pts            126314 non-null  int64
 11  win_equiv      126314 non-null  float64
 12  opp_id         126314 non-null  object
 13  opp_fran       126314 non-null  object
 14  opp_pts        126314 non-null  int64
 15  game_location  126314 non-null  category
 16  game_result    126314 non-null  category
 17  forecast       126314 non-null  float64
 18  notes          5424 non-null    object
 19  difference     126314 non-null  int64
dtypes: category(3), datetime64[ns](1), float64(2), int64(7), object(7)
memory usage: 16.7+ MB
```

Question.11 (report your answer):
11.1) 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).

ANS) The line plot shows the sum of the points where the Knicks scored throughout the season. The first season the sum of total points is above 4000. And the next season it got decreased and then it was increasing. In the year 1970 the total above of all points crossed 11000 points which is the highest compared to all the season points. And then on an average they have scored nearly 8000 points for all the seasons. And from after the peak high, they were at their lowest points in the year 2000 and 2010.

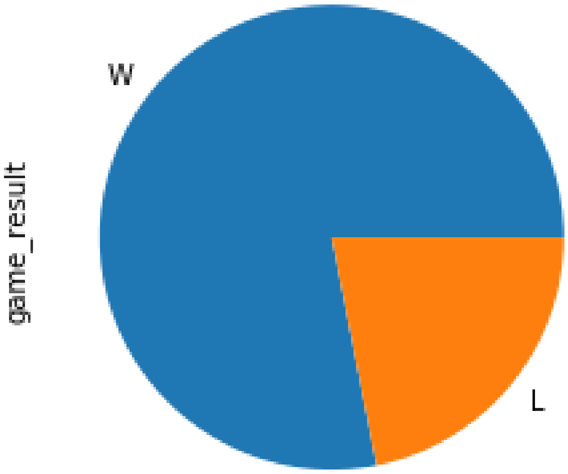
11.2) Describe what the above bar plot reveals to you about the franchises with the most games played.

ANS) The size of the Bar plot shows value from the bar plot we can say that the Lakers are slightly leading the Celtics with nearly 6000 games. And there are six other teams which has more than 5000 games played.

11.3) In 2013, the Miami Heat won the championship. Create a pie plot showing the count of their wins and losses during that season. (First, define a criteria to include only the Heat’s games from 2013. Then, create a plot in the same way as you’ve seen above).

ANS) nba[

```
nba[
    (nba["fran_id"] == "Heat") &
    (nba["year_id"] == 2013)]["game_result"].value_counts().plot(kind="pie")
```



Now we are trying to see plot for the franchise Heat for the year 2013 and counting the values for the wins and losses for the year. The above graph shows that percentage for the wins and losses for the season.

Conclusion:

Its and great opportunity to learn about the pandas library in python. All the instruction which are proved were very useful. Thanks for this opportunity.

References:

1. https://realpython.com/pandas-python-explore-dataset/
2. Lab 2 Instructions by Professor