



richardfrancis.info

Web | Data | Process

Digital Design and Management Services



Data Science

Using the Pandas Python Library

<http://realpython.com/pandas-python-explore-dataset> (<http://realpython.com/pandas-python-explore-dataset>).

In this tutorial, you'll learn how to:

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

You'll also learn about the differences between the main data structures that Pandas and Python use. To follow along, you can get all of the example code in this tutorial at the link below:

nba dataset

In [1]:

```
import requests

download_url = "https://raw.githubusercontent.com/fivethirtyeight/data/master/nba-elo/nba_elo.csv"
target_csv_path = "nba_all_elo.csv"

response = requests.get(download_url)
response.raise_for_status() # Check that the request was successful
with open(target_csv_path, "wb") as f:
    f.write(response.content)
print("Download ready.")
```

Download ready.

In [3]:

```
import pandas as pd
nba = pd.read_csv("nba_all_elo.csv")
type(ds)
```

Out[3]:

pandas.core.frame.DataFrame

In [4]:

```
len(nba)
```

Out[4]:

126314

In [5]:

```
nba.shape
```

Out[5]:

(126314, 23)

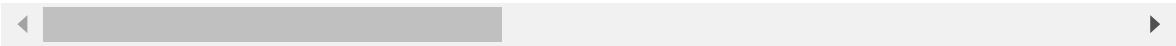
In [6]:

```
nba.head()
```

Out[6]:

	gameorder	game_id	lg_id	_iscopy	year_id	date_game	seasongame	is_playoffs
0	1	194611010TRH	NBA	0	1947	11/1/1946	1	0
1	1	194611010TRH	NBA	1	1947	11/1/1946	1	0
2	2	194611020CHS	NBA	0	1947	11/2/1946	1	0
3	2	194611020CHS	NBA	1	1947	11/2/1946	2	0
4	3	194611020DTF	NBA	0	1947	11/2/1946	1	0

5 rows × 23 columns



In [7]:

```
pd.set_option("display.precision", 2)
```

In [8]:

```
nba.tail()
```

Out[8]:

	gameorder	game_id	lg_id	_iscopy	year_id	date_game	seasongame	is_pla
126309	63155	201506110CLE	NBA	0	2015	6/11/2015	100	
126310	63156	201506140GSW	NBA	0	2015	6/14/2015	102	
126311	63156	201506140GSW	NBA	1	2015	6/14/2015	101	
126312	63157	201506170CLE	NBA	0	2015	6/16/2015	102	
126313	63157	201506170CLE	NBA	1	2015	6/16/2015	103	

5 rows × 23 columns



In [9]:

```
nba.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 126314 entries, 0 to 126313
Data columns (total 23 columns):
gameorder      126314 non-null int64
game_id        126314 non-null object
lg_id          126314 non-null object
_iscopy        126314 non-null int64
year_id        126314 non-null int64
date_game      126314 non-null object
seasongame     126314 non-null int64
is_playoffs    126314 non-null int64
team_id        126314 non-null object
fran_id        126314 non-null object
pts            126314 non-null int64
elo_i          126314 non-null float64
elo_n          126314 non-null float64
win_equiv      126314 non-null float64
opp_id         126314 non-null object
opp_fran       126314 non-null object
opp_pts        126314 non-null int64
opp_elo_i      126314 non-null float64
opp_elo_n      126314 non-null float64
game_location  126314 non-null object
game_result    126314 non-null object
forecast       126314 non-null float64
notes          5424 non-null object
dtypes: float64(6), int64(7), object(10)
memory usage: 22.2+ MB
```

Getting to know the data

In [10]:

```
nba.describe()
```

Out[10]:

	gameorder	_iscopy	year_id	seasongame	is_playoffs	pts	elo_i	elo_n
count	126314.00	126314.0	126314.00	126314.00	126314.00	126314.00	126314.00	126314.00
mean	31579.00	0.5	1988.20	43.53	0.06	102.73	1495.24	14
std	18231.93	0.5	17.58	25.38	0.24	14.81	112.14	1
min	1.00	0.0	1947.00	1.00	0.00	0.00	1091.64	10
25%	15790.00	0.0	1975.00	22.00	0.00	93.00	1417.24	14
50%	31579.00	0.5	1990.00	43.00	0.00	103.00	1500.95	15
75%	47368.00	1.0	2003.00	65.00	0.00	112.00	1576.06	15
max	63157.00	1.0	2015.00	108.00	1.00	186.00	1853.10	18



In [11]:

```
import numpy as np
nba.describe(include=np.object)
```

Out[11]:

	game_id	lg_id	date_game	team_id	fran_id	opp_id	opp_fran	game_locatio
count	126314	126314	126314	126314	126314	126314	126314	126314
unique	63157	2	12426	104	53	104	53	53
top	199611290DAL	NBA	4/16/2014	BOS	Lakers	BOS	Lakers	Lakers
freq	2	118016	30	5997	6024	5997	6024	6313

In [12]:

```
nba["team_id"].value_counts()
```

Out[12]:

```
BOS    5997
NYK    5769
LAL    5078
DET    4985
PHI    4533
...
PIT      60
INJ      60
DTF      60
TRH      60
SDS      11
Name: team_id, Length: 104, dtype: int64
```

In [13]:

```
nba["fran_id"].value_counts()
```

Out[13]:

Lakers	6024
Celtics	5997
Knicks	5769
Warriors	5657
Pistons	5650
Sixers	5644
Hawks	5572
Kings	5475
Wizards	4582
Spurs	4309
Bulls	4307
Pacers	4227
Thunder	4178
Rockets	4154
Nuggets	4120
Nets	4106
Suns	4080
Bucks	4034
Trailblazers	3870
Cavaliers	3810
Clippers	3733
Jazz	3555
Mavericks	3013
Heat	2371
Pelicans	2254
Magic	2207
Timberwolves	2131
Grizzlies	1657
Raptors	1634
Hornets	894
Colonels	846
Squires	799
Spirits	777
Stars	756
Sounds	697
Baltimore	467
Floridians	440
Condors	430
Capitols	291
Olympians	282
Sails	274
Stags	260
Bombers	249
Steamrollers	168
Packers	72
Redskins	65
Rebels	63
Denver	62
Waterloo	62
Falcons	60
Ironmen	60
Huskies	60
Jets	60

Name: fran_id, dtype: int64

In [14]:

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

Out[14]:

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

In [17]:

```
nba.loc[nba["team_id"] == "MNL", "date_game"].agg(("min", "max"))
```

Out[17]:

```
min    1/1/1949
max    4/9/1959
Name: date_game, dtype: object
```

In [19]:

```
nba.loc[nba["team_id"] == "MNL", "date_game"].max()
```

Out[19]:

```
'4/9/1959'
```

In [20]:

```
nba.loc[nba["team_id"] == "MNL", "date_game"].min()
```

Out[20]:

```
'1/1/1949'
```

Getting to know Pandas Data Structures

In [24]:

```
revenues = pd.Series([5555, 7000, 1980])
revenues
```

Out[24]:

```
0    5555
1    7000
2    1980
dtype: int64
```

In [25]:

```
revenues.values
```

Out[25]:

```
array([5555, 7000, 1980], dtype=int64)
```

In [26]:

```
revenues.index
```

Out[26]:

```
RangeIndex(start=0, stop=3, step=1)
```

In [27]:

```
type(revenues.values)
```

Out[27]:

```
numpy.ndarray
```

In [28]:

```
city_revenues = pd.Series([4200, 8000, 6500], index=["Amsterdam", "London", "New York"])
```

In [29]:

```
city_revenues
```

Out[29]:

```
Amsterdam    4200  
London       8000  
New York     6500  
dtype: int64
```

In [31]:

```
city_employee_count = pd.Series({"Amsterdam": 5, "London": 8})  
city_employee_count
```

Out[31]:

```
Amsterdam    5  
London       8  
dtype: int64
```

In [32]:

```
city_employee_count.keys()
```

Out[32]:

```
Index(['Amsterdam', 'London'], dtype='object')
```

In [33]:

```
"Tokyo" in city_employee_count
```

Out[33]:

```
False
```

In [34]:

```
"London" in city_employee_count
```

Out[34]:

True

In [35]:

```
city_data = pd.DataFrame({"revenue": city_revenues, "employee_count": city_employee_count})
```

In [36]:

```
city_data
```

Out[36]:

	revenue	employee_count
Amsterdam	4200	5.0
London	8000	8.0
New York	6500	NaN

In [37]:

```
city_data.index
```

Out[37]:

```
Index(['Amsterdam', 'London', 'New York'], dtype='object')
```

In [39]:

```
city_data.values
```

Out[39]:

```
array([[4.2e+03, 5.0e+00],  
       [8.0e+03, 8.0e+00],  
       [6.5e+03,    nan]])
```

In [40]:

```
city_data.axes
```

Out[40]:

```
[Index(['Amsterdam', 'London', 'New York'], dtype='object'),  
 Index(['revenue', 'employee_count'], dtype='object')]
```

In [41]:

```
city_data.axes[0]
```

Out[41]:

```
Index(['Amsterdam', 'London', 'New York'], dtype='object')
```

In [42]:

```
city_data.axes[1]
```

Out[42]:

```
Index(['revenue', 'employee_count'], dtype='object')
```

In [43]:

```
city_data.keys()
```

Out[43]:

```
Index(['revenue', 'employee_count'], dtype='object')
```

In [44]:

```
"revenue" in city_data
```

Out[44]:

```
True
```

In [45]:

```
"Tokyo" in city_data
```

Out[45]:

```
False
```

In [46]:

```
nba.index
```

Out[46]:

```
RangeIndex(start=0, stop=126314, step=1)
```

In [47]:

```
nba.axes
```

Out[47]:

```
[RangeIndex(start=0, stop=126314, step=1),  
 Index(['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', 'opp_fran', 'opp_pts', 'opp_elo_i',  
        'opp_elo_n', 'game_location', 'game_result', 'forecast', 'notes'],  
       dtype='object')]
```

In [50]:

```
"pts" in nba.keys()
```

Out[50]:

```
True
```

Accessing series elements

In [51]:

```
city_revenues["London"]
```

Out[51]:

8000

In [52]:

```
city_revenues[1]
```

Out[52]:

8000

In [53]:

```
city_revenues[-1]
```

Out[53]:

6500

In [54]:

```
city_revenues[1:]
```

Out[54]:

```
London      8000
New York    6500
dtype: int64
```

In [56]:

```
city_revenues["New York"]
```

Out[56]:

6500

In [58]:

```
colors = pd.Series(["red", "purple", "blue", "green", "yellow"], index=[1, 2, 3, 5, 8])
```

In [60]:

```
colors
```

Out[60]:

```
1      red
2    purple
3      blue
5     green
8     yellow
dtype: object
```

In [61]:

```
colors.loc[1]
```

Out[61]:

```
'red'
```

In [62]:

```
colors.iloc[1]
```

Out[62]:

```
'purple'
```

In [63]:

```
colors.iloc[1:3]
```

Out[63]:

```
2    purple
3     blue
dtype: object
```

In [65]:

```
colors.loc[3:8]
```

Out[65]:

```
3     blue
5    green
8    yellow
dtype: object
```

In [66]:

```
colors.iloc[-2]
```

Out[66]:

```
'green'
```

In [67]:

```
city_data["revenue"]
```

Out[67]:

```
Amsterdam    4200
London       8000
New York     6500
Name: revenue, dtype: int64
```

In [68]:

```
type(city_data["revenue"])
```

Out[68]:

pandas.core.series.Series

In [69]:

```
city_data.revenue
```

Out[69]:

```
Amsterdam    4200
London        8000
New York      6500
Name: revenue, dtype: int64
```

In [70]:

```
city_data
```

Out[70]:

	revenue	employee_count
Amsterdam	4200	5.0
London	8000	8.0
New York	6500	NaN

In [71]:

```
toys = pd.DataFrame([
    {"name": "ball", "shape": "sphere"},
    {"name": "rubik's cube", "shape": "cube"}
])
```

In [72]:

```
toys["shape"]
```

Out[72]:

```
0    sphere
1     cube
Name: shape, dtype: object
```

In [73]:

```
city_data.loc["Amsterdam"]
```

Out[73]:

```
revenue    4200.0
employee_count    5.0
Name: Amsterdam, dtype: float64
```

In [74]:

```
city_data.iloc[1]
```

Out[74]:

```
revenue          8000.0
employee_count    8.0
Name: London, dtype: float64
```

In [75]:

```
nba.iloc[-2]
```

Out[75]:

```
gameorder          63157
game_id            201506170CLE
lg_id              NBA
_iscopy            0
year_id            2015
date_game          6/16/2015
seasongame          102
is_playoffs        1
team_id            CLE
fran_id            Cavaliers
pts                97
elo_i              1.7e+03
elo_n              1.7e+03
win_equiv          59
opp_id             GSW
opp_fran           Warriors
opp_pts            105
opp_elo_i          1.8e+03
opp_elo_n          1.8e+03
game_location      H
game_result        L
forecast           0.48
notes              NaN
Name: 126312, dtype: object
```

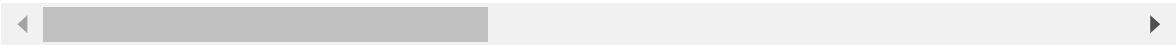
In [78]:

```
nba[
    (nba["_iscopy"] == 0) &
    (nba["pts"] > 100) &
    (nba["opp_pts"] > 100) &
    (nba["team_id"] == "BLB")
]
```

Out[78]:

	gameorder	game_id	lg_id	_iscopy	year_id	date_game	seasongame	is_playoff
1726	864	194902260BLB	NBA	0	1949	2/26/1949	53	
4890	2446	195301100BLB	NBA	0	1953	1/10/1953	32	
4909	2455	195301140BLB	NBA	0	1953	1/14/1953	34	
5208	2605	195303110BLB	NBA	0	1953	3/11/1953	66	
5825	2913	195402220BLB	NBA	0	1954	2/22/1954	60	

5 rows × 23 columns



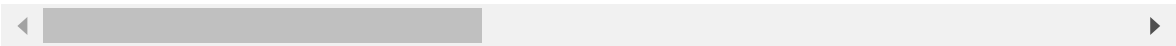
In [79]:

```
nba[
    (nba["_iscopy"] == 0) &
    (nba["team_id"].str.startswith("LA")) &
    (nba["year_id"] == 1992) &
    (nba["notes"].notnull())
]
```

Out[79]:

	gameorder	game_id	lg_id	_iscopy	year_id	date_game	seasongame	is_playoff
68901	34451	199205030LAC	NBA	0	1992	5/3/1992	86	
68903	34452	199205030LAL	NBA	0	1992	5/3/1992	86	

2 rows × 23 columns



Grouping and aggregating the data

In [80]:

```
city_revenues.sum()
```

Out[80]:

18700

In [81]:

```
city_revenues
```

Out[81]:

```
Amsterdam    4200  
London       8000  
New York     6500  
dtype: int64
```

In [83]:

```
city_revenues.max()
```

Out[83]:

8000

In [84]:

```
points = nba["pts"]
```

In [85]:

```
type(points)
```

Out[85]:

pandas.core.series.Series

In [86]:

```
points.sum()
```

Out[86]:

12976235

In [87]:

```
nba.groupby("fran_id", sort=False)["pts"].sum()
```

Out[87]:

fran_id	
Huskies	3995
Knicks	582497
Stags	20398
Falcons	3797
Capitols	22387
Celtics	626484
Steamrollers	12372
Ironmen	3674
Bombers	17793
Rebels	4474
Warriors	591224
Baltimore	37219
Jets	4482
Pistons	572758
Lakers	637444
Kings	569245
Hawks	567261
Denver	4818
Olympians	22864
Redskins	5372
Waterloo	4921
Packers	6193
Sixers	585891
Wizards	474809
Bulls	437269
Thunder	437735
Squires	91127
Stars	84940
Rockets	432504
Colonels	94435
Pacers	438288
Nuggets	445780
Spurs	453822
Spirits	85874
Sounds	75582
Floridians	49568
Nets	417809
Condors	49642
Bucks	418326
Suns	437486
Clippers	380523
Cavaliers	380416
Trailblazers	402695
Sails	30080
Jazz	363155
Mavericks	309239
Pelicans	220794
Heat	229103
Timberwolves	207693
Magic	219436
Grizzlies	157683
Raptors	158370
Hornets	84489
Name: pts, dtype: int64	

In [89]:

```
maxpoints = nba.groupby("fran_id", sort=False)["pts"].max()
```

In [90]:

```
maxpoints.max()
```

Out[90]:

186

In [91]:

```
nba[
    (nba["fran_id"] == "Spurs") &
    (nba["year_id"] > 2010)
].groupby(["year_id", "game_result"])["game_id"].count()
```

Out[91]:

year_id	game_result	
2011	L	25
	W	63
2012	L	20
	W	60
2013	L	30
	W	73
2014	L	27
	W	78
2015	L	31
	W	58

Name: game_id, dtype: int64

In [93]:

```
nba[
    (nba["fran_id"] == "Warriors") &
    (nba["year_id"] == 2015)
].groupby(["is_playoffs", "game_result"])["game_id"].count()
```

Out[93]:

is_playoffs	game_result	
0	L	15
	W	67
1	L	5
	W	16

Name: game_id, dtype: int64

Manipulating Columns

In [94]:

```
df = nba.copy()
```

In [95]:

```
df.shape
```

Out[95]:

```
(126314, 23)
```

In [96]:

```
df["difference"] = df.pts - df.opp_pts  
df.shape
```

Out[96]:

```
(126314, 24)
```

In [97]:

```
df["difference"].max()
```

Out[97]:

```
68
```

In [98]:

```
renamed_df = df.rename(  
    columns={"game_result": "result", "game_location": "location"}  
)
```

In [99]:

```
renamed_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 126314 entries, 0 to 126313
Data columns (total 24 columns):
gameorder      126314 non-null int64
game_id        126314 non-null object
lg_id          126314 non-null object
_iscopy        126314 non-null int64
year_id        126314 non-null int64
date_game      126314 non-null object
seasongame     126314 non-null int64
is_playoffs    126314 non-null int64
team_id        126314 non-null object
fran_id        126314 non-null object
pts            126314 non-null int64
elo_i          126314 non-null float64
elo_n          126314 non-null float64
win_equiv      126314 non-null float64
opp_id         126314 non-null object
opp_fran       126314 non-null object
opp_pts        126314 non-null int64
opp_elo_i      126314 non-null float64
opp_elo_n      126314 non-null float64
location       126314 non-null object
result         126314 non-null object
forecast       126314 non-null float64
notes          5424 non-null object
difference     126314 non-null int64
dtypes: float64(6), int64(8), object(10)
memory usage: 23.1+ MB
```

In [100]:

```
df.shape
```

Out[100]:

```
(126314, 24)
```

In [101]:

```
elo_columns = ["elo_i", "elo_n", "opp_elo_i", "opp_elo_n"]
```

In [103]:

```
df.drop(elo_columns, inplace=True, axis=1)
```

In [104]:

```
df.shape
```

Out[104]:

```
(126314, 20)
```

Specifying data types

In [106]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 126314 entries, 0 to 126313
Data columns (total 20 columns):
gameorder      126314 non-null int64
game_id        126314 non-null object
lg_id          126314 non-null object
_iscopy        126314 non-null int64
year_id        126314 non-null int64
date_game      126314 non-null object
seasongame     126314 non-null int64
is_playoffs    126314 non-null int64
team_id        126314 non-null object
fran_id        126314 non-null object
pts            126314 non-null int64
win_equiv      126314 non-null float64
opp_id         126314 non-null object
opp_fran       126314 non-null object
opp_pts        126314 non-null int64
game_location  126314 non-null object
game_result    126314 non-null object
forecast       126314 non-null float64
notes          5424 non-null object
difference     126314 non-null int64
dtypes: float64(2), int64(8), object(10)
memory usage: 19.3+ MB
```

In [107]:

```
df["date_game"] = pd.to_datetime(df["date_game"])
```

In [108]:

```
df["game_location"].nunique()
```

Out[108]:

3

In [109]:

```
df["game_location"].value_counts()
```

Out[109]:

```
H    63138
A    63138
N      38
Name: game_location, dtype: int64
```

In [111]:

```
df["game_location"] = pd.Categorical(df["game_location"])
```

In [112]:

```
df["game_location"].dtype
```

Out[112]:

```
CategoricalDtype(categories=['A', 'H', 'N'], ordered=False)
```

In [113]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 126314 entries, 0 to 126313
Data columns (total 20 columns):
gameorder      126314 non-null int64
game_id        126314 non-null object
lg_id          126314 non-null object
_iscopy        126314 non-null int64
year_id        126314 non-null int64
date_game      126314 non-null datetime64[ns]
seasongame     126314 non-null int64
is_playoffs    126314 non-null int64
team_id        126314 non-null object
fran_id        126314 non-null object
pts            126314 non-null int64
win_equiv      126314 non-null float64
opp_id         126314 non-null object
opp_fran       126314 non-null object
opp_pts        126314 non-null int64
game_location  126314 non-null category
game_result    126314 non-null object
forecast       126314 non-null float64
notes          5424 non-null object
difference     126314 non-null int64
dtypes: category(1), datetime64[ns](1), float64(2), int64(8), object(8)
memory usage: 18.4+ MB
```

In [114]:

```
df["game_result"].nunique()
```

Out[114]:

```
2
```

In [115]:

```
df["game_result"].value_counts()
```

Out[115]:

```
L    63157
W    63157
Name: game_result, dtype: int64
```

In [116]:

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

In [117]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 126314 entries, 0 to 126313
Data columns (total 20 columns):
gameorder      126314 non-null int64
game_id        126314 non-null object
lg_id          126314 non-null object
_iscopy        126314 non-null int64
year_id        126314 non-null int64
date_game      126314 non-null datetime64[ns]
seasongame     126314 non-null int64
is_playoffs    126314 non-null int64
team_id        126314 non-null object
fran_id        126314 non-null object
pts            126314 non-null int64
win_equiv      126314 non-null float64
opp_id         126314 non-null object
opp_fran       126314 non-null object
opp_pts        126314 non-null int64
game_location  126314 non-null category
game_result    126314 non-null category
forecast       126314 non-null float64
notes          5424 non-null object
difference     126314 non-null int64
dtypes: category(2), datetime64[ns](1), float64(2), int64(8), object(7)
memory usage: 17.6+ MB
```

Cleaning data

In [118]:

```
rows_without_missing_data = nba.dropna()
```

In [119]:

```
rows_without_missing_data.shape
```

Out[119]:

```
(5424, 23)
```

In [120]:

```
data_without_missing_columns = nba.dropna(axis=1)
```

In [121]:

```
data_without_missing_columns.shape
```

Out[121]:

```
(126314, 22)
```

In [122]:

```
data_with_default_notes = nba.copy()
```

In [123]:

```
data_with_default_notes["notes"].fillna(value="no notes at all", inplace=True)
```

In [124]:

```
data_with_default_notes["notes"].describe()
```

Out[124]:

```
count          126314
unique           232
top      no notes at all
freq          120890
Name: notes, dtype: object
```

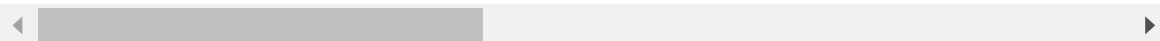
In [125]:

```
nba[nba["pts"] == 0]
```

Out[125]:

	gameorder	game_id	lg_id	_iscopy	year_id	date_game	seasongame	is_playo
26684	13343	197210260VIR	ABA	1	1973	10/26/1972	7	

1 rows × 23 columns



In [126]:

```
nba[(nba["pts"] > nba["opp_pts"]) & (nba["game_result"] != 'W')].empty
```

Out[126]:

True

In [127]:

```
nba[(nba["pts"] < nba["opp_pts"]) & (nba["game_result"] != 'L')].empty
```

Out[127]:

True

Combining multiple datasets

In [129]:

```
further_city_data = pd.DataFrame(  
    {"revenue": [7000, 3400], "employee_count": [2, 2]},  
    index=["New York", "Barcelona"]  
)
```

In [130]:

```
all_city_data = pd.concat([city_data, further_city_data], sort=False)
```

In [131]:

```
all_city_data
```

Out[131]:

	revenue	employee_count
Amsterdam	4200	5.0
London	8000	8.0
New York	6500	NaN
New York	7000	2.0
Barcelona	3400	2.0

Querying the dataset

In [132]:

```
current_decade = nba[nba["year_id"] > 2010]
```

In [133]:

```
current_decade.shape
```

Out[133]:

```
(12658, 23)
```

In [134]:

```
games_with_notes = nba[nba["notes"].notnull()]
```

In [135]:

```
games_with_notes.shape
```

Out[135]:

```
(5424, 23)
```

In [137]:

```
ers = nba[nba["fran_id"].str.endswith("ers")]
```

In [138]:

```
ers.shape
```

Out[138]:

(27797, 23)

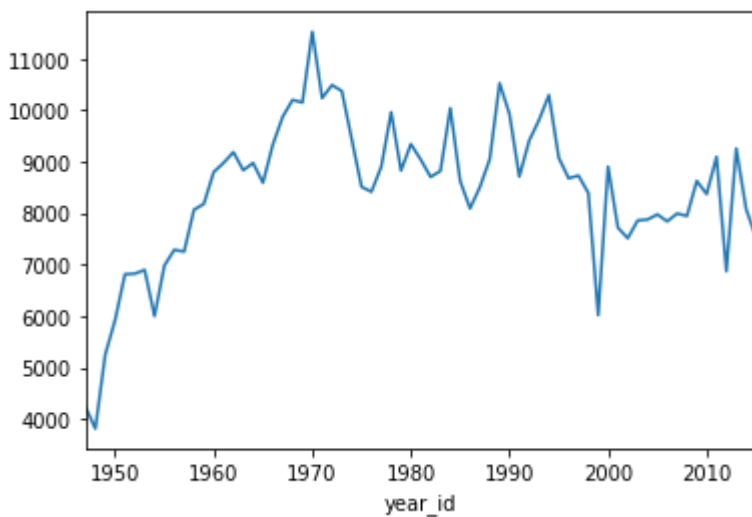
Visualising the dataframe

In [140]:

```
%matplotlib inline  
nba[nba["fran_id"] == "Knicks"].groupby("year_id")["pts"].sum().plot()
```

Out[140]:

<matplotlib.axes._subplots.AxesSubplot at 0x294a51a4e48>

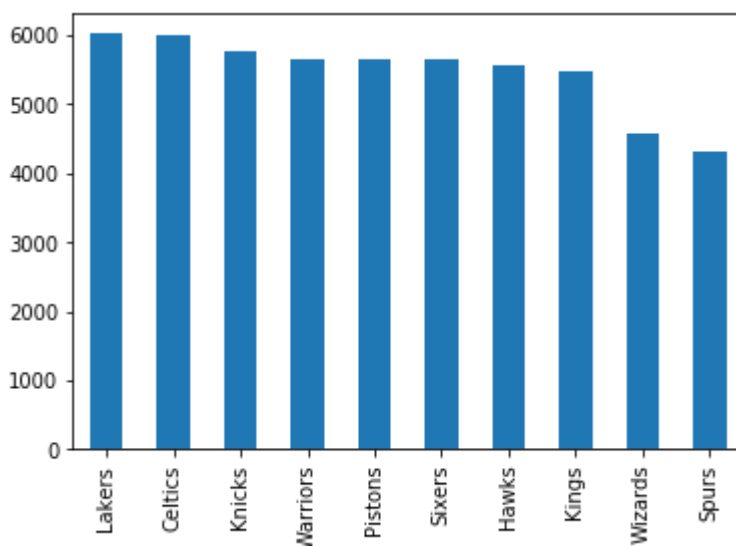


In [142]:

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

Out[142]:

<matplotlib.axes._subplots.AxesSubplot at 0x294a5279e48>

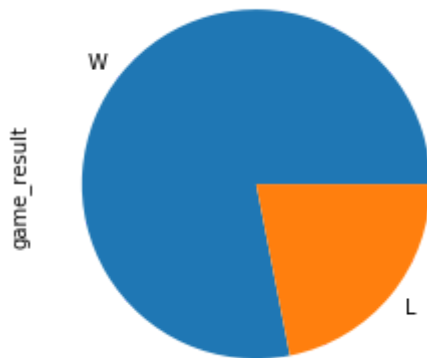


In [143]:

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

Out[143]:

<matplotlib.axes._subplots.AxesSubplot at 0x294a0396488>



In []:

Copyright © richardfrancis.info 2022

[Privacy](#) · [Terms](#) · [Contact](#)