MIPdf

May 12, 2021

[24]:
```
import pandas as pd
import numpy as np
```

[25]:
```
#load in normal NBA data (name is different here, but update142 should work)
nba = pd.read_csv('142accentfixed.csv')
```

0.1 2010-2020 MIP

[26]:
```
#select only columns needed
mip = nba.iloc[:,np.r_[1:3, 6:51]]
#set player and season as index
mip = mip.set_index(['Player', 'Season']).sort_index()
```

[27]:
```
#create empty dataframe
mip_df = pd.DataFrame(columns = mip.columns)
mip_df['Player'] = 0
mip_df['Season'] = 0

for i in range(mip.shape[0]-1):
    #if player has played the next year, create column
    if (mip.index[i][0] == mip.index[i+1][0]):
        #create difference row
        mip_df = mip_df.append(mip.iloc[i+1] - mip.iloc[i], ignore_index = True)
        #add corresponding player and season index
        mip_df.loc[mip_df.index[-1], 'Player'] = mip.index[i][0]
        mip_df.loc[mip_df.index[-1], 'Season'] = mip.index[i+1][1]

mip_df = mip_df.set_index(['Player', 'Season'])
```

[28]:
```
nba_copy = nba
nba_copy = nba_copy.set_index(['Player', 'Season']).sort_index()

mip_votes = nba_copy[['MIP']]
```

[29]:
```
#merge with original to get MIP votes
mip_df = mip_df.merge(mip_votes, how='left', left_index = True, right_index =␣
↪True)
```

```
mip_df = mip_df.fillna(0)
mip_df.head()
```

[29]:

Player	Season	G	GS	MP	FGM	FGA	FGP	X3PM	X3PA	X3PP	X2PM	\
A.J. Price	2011.0	-6.0	-2.0	0.5	-0.3	0.1	-0.054	-0.3	-0.1	-0.070	0.0	
	2012.0	-6.0	1.0	-3.0	-1.0	-2.4	-0.017	-0.2	-1.0	0.020	-0.7	
	2013.0	13.0	21.0	9.5	1.5	3.2	0.051	0.6	1.5	0.055	0.8	
	2014.0	-29.0	-22.0	-18.9	-2.1	-5.6	0.023	-1.0	-2.7	-0.077	-1.1	
	2015.0	-2.0	0.0	9.0	1.3	3.7	-0.041	0.4	1.4	-0.010	0.9	

Player	Season	…	USG	OWS	DWS	WS	WS48	OBPM	DBPM	BPM	VORP	MIP
A.J. Price	2011.0	…	0.0	-0.8	-0.1	-0.9	-0.045	-1.8	-0.7	-2.4	-0.5	0.0
	2012.0	…	-5.0	0.6	-0.2	0.4	0.043	1.3	0.6	1.9	0.3	0.0
	2013.0	…	0.3	0.8	0.7	1.5	0.021	0.4	0.1	0.4	0.4	0.0
	2014.0	…	5.2	-1.1	-1.2	-2.2	-0.096	-2.2	-2.0	-4.2	-0.6	0.0
	2015.0	…	-5.0	0.9	0.5	1.4	0.057	1.5	1.3	2.8	0.1	0.0

[5 rows x 46 columns]

[30]:
```
mip_df.to_csv('mip_stats.csv')
```

[31]:
```
mip_df.sort_values(by = ["MIP"], ascending = False).head(10)
```

[31]:

Player	Season	G	GS	MP	FGM	FGA	FGP	X3PM	X3PA	\
Victor Oladipo	2018.0	8.0	8.0	0.8	2.4	4.0	0.035	0.2	0.5	
Pascal Siakam	2019.0	-1.0	74.0	11.2	3.4	5.7	0.041	0.6	1.1	
CJ McCollum	2016.0	18.0	77.0	19.1	5.4	12.0	0.012	1.6	3.7	
Giannis Antetokounmpo	2017.0	0.0	1.0	0.3	1.8	3.0	0.015	0.2	0.9	
Jimmy Butler	2015.0	-2.0	-2.0	0.0	2.4	3.7	0.065	0.1	-0.6	
Kevin Love	2011.0	13.0	51.0	7.2	1.7	3.3	0.020	0.6	1.1	
Brandon Ingram	2020.0	10.0	10.0	0.1	1.2	3.7	-0.034	1.8	4.4	
Goran Dragic	2014.0	-1.0	-2.0	1.6	2.1	2.6	0.062	0.5	0.3	
Bam Adebayo	2020.0	-10.0	44.0	10.3	2.7	5.1	-0.019	0.0	0.0	
Paul George	2013.0	13.0	13.0	7.9	1.9	5.2	-0.021	0.8	2.4	

Player	Season	X3PP	X2PM	…	USG	OWS	DWS	WS	WS48	\
Victor Oladipo	2018.0	0.010	2.1	…	8.7	2.6	1.7	4.2	0.070	
Pascal Siakam	2019.0	0.149	2.7	…	5.1	3.6	1.1	4.6	0.042	
CJ McCollum	2016.0	0.021	3.9	…	16.2	4.3	1.6	5.7	0.002	
Giannis Antetokounmpo	2017.0	0.015	1.5	…	6.0	3.7	1.6	5.3	0.089	
Jimmy Butler	2015.0	0.095	2.3	…	5.6	-1.4	-3.0	-4.4	-0.040	
Kevin Love	2011.0	0.087	1.1	…	0.5	5.5	1.0	6.5	0.072	
Brandon Ingram	2020.0	0.061	-0.6	…	4.9	2.7	0.4	3.1	0.060	

```
Goran Dragic          2014.0  0.089   1.6  …   2.8  4.3  0.4  4.6  0.080
Bam Adebayo           2020.0 -0.057   2.7  …   5.4  1.2  0.5  1.7 -0.003
Paul George           2013.0 -0.023   1.2  …   4.2 -0.1  3.1  3.0 -0.003
```

Player	Season	OBPM	DBPM	BPM	VORP	MIP
Victor Oladipo	2018.0	4.4	1.8	6.3	4.2	0.988
Pascal Siakam	2019.0	1.9	-0.8	1.1	1.4	0.938
CJ McCollum	2016.0	3.3	-1.9	1.4	2.2	0.860
Giannis Antetokounmpo	2017.0	3.6	1.7	5.2	3.8	0.856
Jimmy Butler	2015.0	0.1	-2.3	-2.2	-1.7	0.829
Kevin Love	2011.0	3.0	-0.1	2.9	2.7	0.690
Brandon Ingram	2020.0	3.8	0.3	4.1	2.2	0.652
Goran Dragic	2014.0	2.4	0.1	2.5	1.8	0.648
Bam Adebayo	2020.0	1.8	-0.2	1.6	1.5	0.590
Paul George	2013.0	0.1	0.2	0.4	1.7	0.518

```
[10 rows x 46 columns]
```

0.2 2021 MIP

[32]:
```
data2021 = pd.read_csv("nba2021.csv")
```

[33]:
```
nba_2020 = nba[nba['Season'] == 2020]
mip2021 = nba_2020.append(data2021).set_index(['Player', 'Season']).sort_index()
mip2021 = mip2021.iloc[:,np.r_[4:49]]
```

[34]:
```
mip2021_df = pd.DataFrame(columns = mip2021.columns)
mip2021_df['Player'] = 0
mip2021_df['Season'] = 0

for i in range(mip2021.shape[0]-1):
    #if player has played the next year, create column
    if (mip2021.index[i][0] == mip2021.index[i+1][0]):
        mip2021_df = mip2021_df.append(mip2021.iloc[i+1] - mip2021.iloc[i],␣
 ↪ignore_index = True)
        mip2021_df.loc[mip2021_df.index[-1], 'Player'] = mip2021.index[i][0]
        mip2021_df.loc[mip2021_df.index[-1], 'Season'] = mip2021.index[i+1][1]

mip2021_df = mip2021_df.set_index(['Player', 'Season'])
```

[35]:
```
mip2021_df.to_csv('mip2021_stats.csv')
```