Football player's market value

https://arxiv.org/ftp/arxiv/papers/1807/1807.01104.pdf

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df = pd.read_csv("epldata_final.csv")
```

```
df.head()
```

	name	club	age	position	position_cat	market_value	page_views	fpl_value
0	Alexis Sanchez	Arsenal	28	LW	1	65.0	4329	12.(
1	Mesut Ozil	Arsenal	28	AM	1	50.0	4395	9.5
2	Petr Cech	Arsenal	35	GK	4	7.0	1529	5.5
3	Theo Walcott	Arsenal	28	RW	1	20.0	2393	7.5

```
df.shape
```

```
(461, 17)
```

EDA

```
df.dtypes
```

```
name             object
club             object
age               int64
position         object
position_cat      int64
market_value    float64
page_views        int64
fpl_value       float64
fpl_sel          object
fpl_points        int64
region          float64
nationality      object
new_foreign       int64
age_cat           int64
club_id           int64
big_club          int64
new_signing       int64
dtype: object
```

```
df.describe()
```

	age	position_cat	market_value	page_views	fpl_value	fpl_points
count	461.000000	461.000000	461.000000	461.000000	461.000000	461.000000
mean	26.804772	2.180043	11.012039	763.776573	5.447939	57.314534
std	3.961892	1.000061	12.257403	931.805757	1.346695	53.113811
min	17.000000	1.000000	0.050000	3.000000	4.000000	0.000000
25%	24.000000	1.000000	3.000000	220.000000	4.500000	5.000000
50%	27.000000	2.000000	7.000000	460.000000	5.000000	51.000000
75%	30.000000	3.000000	15.000000	896.000000	5.500000	94.000000
max	38.000000	4.000000	75.000000	7664.000000	12.500000	264.000000

```
df[df.isna()==False]
df.isna().sum()
```

```
name            0
club            0
```

```
age             0
position        0
position_cat    0
market_value    0
page_views      0
fpl_value       0
fpl_sel         0
fpl_points      0
region          1
nationality     0
new_foreign     0
age_cat         0
club_id         0
big_club        0
new_signing     0
dtype: int64
```

```
df.dropna(inplace=True)
```

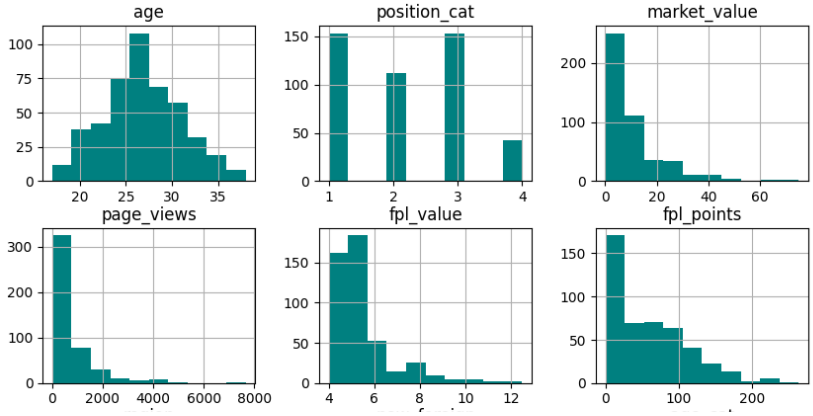
```
df[df.isna()==False]
df.isna().sum()
```

```
name            0
club            0
age             0
position        0
position_cat    0
market_value    0
page_views      0
fpl_value       0
fpl_sel         0
fpl_points      0
region          0
nationality     0
new_foreign     0
age_cat         0
club_id         0
big_club        0
new_signing     0
dtype: int64
```

```
clubs=df['club'].unique()
print(clubs)
print("Total clubs", len(clubs))
```

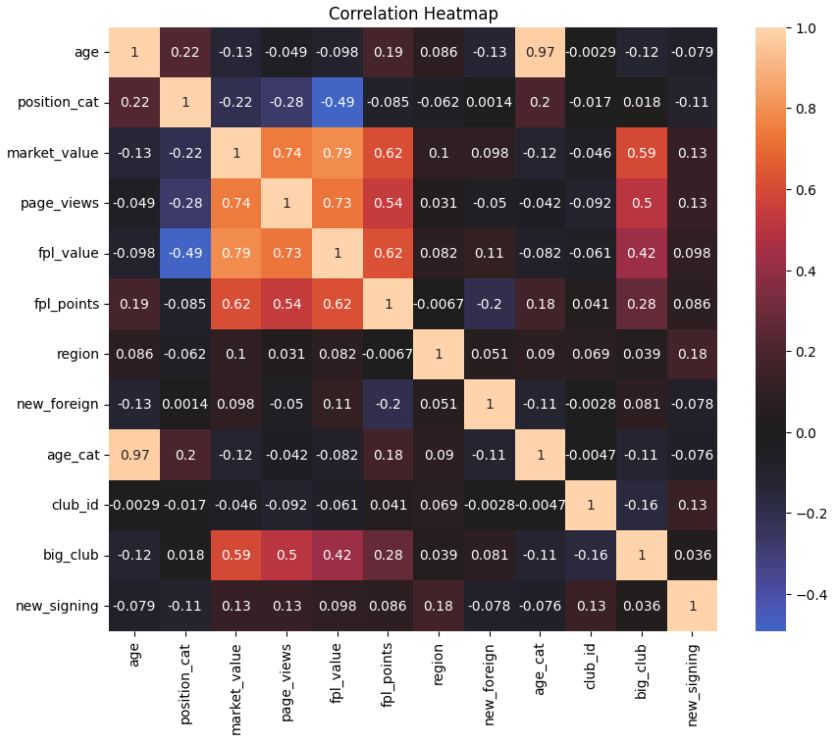
```
['Arsenal' 'Bournemouth' 'Brighton+and+Hove' 'Burnley' 'Chelsea'
 'Crystal+Palace' 'Everton' 'Huddersfield' 'Leicester+City' 'Liverpool'
 'Manchester+City' 'Manchester+United' 'Newcastle+United' 'Southampton'
 'Stoke+City' 'Swansea' 'Tottenham' 'Watford' 'West+Brom' 'West+Ham']
Total clubs 20
```

```
df.hist(figsize=(10,10), color='teal')
plt.title("Histogram Plot of the Features")
plt.show()
```




```
plt.figure(figsize=(10, 8))  # Adjust the figure size as needed
sns.heatmap(df.corr(), annot=True, center=0)
plt.title("Correlation Heatmap")
plt.show()
```

```
<ipython-input-20-5d3363fde466>:2: FutureWarning: The default value of numeric_onl
  sns.heatmap(df.corr(), annot=True, center=0)
```



Train Test Split

```
df.columns
```

```
Index(['name', 'club', 'age', 'position', 'position_cat', 'market_value',
       'page_views', 'fpl_value', 'fpl_sel', 'fpl_points', 'region',
       'nationality', 'new_foreign', 'age_cat', 'club_id', 'big_club',
       'new_signing'],
      dtype='object')
```

```
X = df.drop(["name", "club", "age", "position", "market_value", "page_views", "fpl_sel", "nationality"], axis=1)
Y = df["market_value"]
```

```
print("X-shape", X.shape)
print("Y-shape", Y.shape)
```

```
X-shape (460, 9)
Y-shape (460,)
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)
```

```
print("Training: ", X_train.shape, "   ", y_train.shape)
print("Testing: ",X_test.shape, "   ", y_test.shape)
```

```
Training:  (368, 9)     (368,)
Testing:  (92, 9)     (92,)
```

Fitting Model

```
from sklearn.linear_model import LinearRegression
```

```
lr_model = LinearRegression()
```

```
lr_model.fit(X_train, y_train)
```

```
LinearRegression
LinearRegression()
```

```
y_pred = lr_model.predict(X_test)
```

Accuracy of Model

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
```

```
mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error (MSE):", mse)

mae = mean_absolute_error(y_test, y_pred)
print("Mean Absolute Error (MAE):", mae)

r_squared = r2_score(y_test, y_pred)
print("R-squared:", r_squared)
```

```
Mean Squared Error (MSE): 57.55466975972589
Mean Absolute Error (MAE): 4.547691897399721
R-squared: 0.7047079603681493
```