Mini-Project - Analysis of FIFA Player's Attributes

July 10, 2017

1 Dataset selection

The selected dataset is the *European Soccer Database* which can be found on Kaggle.

2 Dataset exploration

2.1 Required libraries

Since the dataset comes in an SQL format, the required libraries need to be imported:

- ***sqllite3*** for interacting with a local relational database.
- ***pandas*** and ***numpy*** for data ingestion and manipulation.
- ***matplotlib*** for data visualization.

```
In [1]: import sqlite3 #To read SQL files
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
```

2.2 Data ingestion

A connection to the data base is created followed by the creation of the *pandas* data frames.

```
In [2]: # Connection to SQL data base
        connection = sqlite3.connect('database.sqlite')

        # Data frame creation
        df_attr = pd.read_sql_query("SELECT * FROM Player_Attributes", connection)
        df_players = pd.read_sql_query("SELECT * FROM Player", connection)
```

2.3 Data exploration

```
In [3]: print("Dimensions of df_attr: ", df_attr.shape)
        df_attr.head(3)
```

```
Dimensions of df_attr:  (183978, 42)
```

```
Out[3]:    id  player_fifa_api_id  player_api_id                 date  overall_rating  \
        0   1              218353         505942  2016-02-18 00:00:00            67.0
        1   2              218353         505942  2015-11-19 00:00:00            67.0
        2   3              218353         505942  2015-09-21 00:00:00            62.0

           potential preferred_foot attacking_work_rate defensive_work_rate  crossing  \
        0       71.0          right              medium              medium      49.0
        1       71.0          right              medium              medium      49.0
        2       66.0          right              medium              medium      49.0

              ...        vision  penalties  marking  standing_tackle  sliding_tackle  \
        0     ...          54.0       48.0     65.0             69.0            69.0
        1     ...          54.0       48.0     65.0             69.0            69.0
        2     ...          54.0       48.0     65.0             66.0            69.0

           gk_diving  gk_handling  gk_kicking  gk_positioning  gk_reflexes
        0        6.0         11.0        10.0             8.0          8.0
        1        6.0         11.0        10.0             8.0          8.0
        2        6.0         11.0        10.0             8.0          8.0

        [3 rows x 42 columns]
```

```
In [4]: print("Dimensions of df_players: ", df_players.shape)
        df_players.head(3)
```

```
Dimensions of df_players:  (11060, 7)
```

```
Out[4]:    id  player_api_id         player_name  player_fifa_api_id  \
        0   1         505942  Aaron Appindangoye              218353
        1   2         155782     Aaron Cresswell              189615
        2   3         162549         Aaron Doran              186170

                      birthday  height  weight
        0  1992-02-29 00:00:00  182.88     187
        1  1989-12-15 00:00:00  170.18     146
        2  1991-05-13 00:00:00  170.18     163
```

3 Research question

Since the information about the player's overall rating is a parameter that is really important not only in real life (e.g. to plan and execute transfers) but also is a valuable evaluation parameter for *FIFA (video game)* players. Therefore, the chosen research question for this Mini-Project is:

How are the skills related specially with the player's overall rating?

4 Data deep exploration and analysis

Which information is contained in the attribute dataset columns?

```
In [5]: df_attr.columns
```

```
Out[5]: Index(['id', 'player_fifa_api_id', 'player_api_id', 'date', 'overall_rating',
               'potential', 'preferred_foot', 'attacking_work_rate',
               'defensive_work_rate', 'crossing', 'finishing', 'heading_accuracy',
               'short_passing', 'volleys', 'dribbling', 'curve', 'free_kick_accuracy',
               'long_passing', 'ball_control', 'acceleration', 'sprint_speed',
               'agility', 'reactions', 'balance', 'shot_power', 'jumping', 'stamina',
               'strength', 'long_shots', 'aggression', 'interceptions', 'positioning',
               'vision', 'penalties', 'marking', 'standing_tackle', 'sliding_tackle',
               'gk_diving', 'gk_handling', 'gk_kicking', 'gk_positioning',
               'gk_reflexes'],
              dtype='object')
```

4.1 Preparation of working dataset

In order to prepare the final dataset to work with, a merge of the dataset is performed

```
In [6]: df = df_players.merge(df_attr, on='player_fifa_api_id', how='inner')
        print(df.shape)
        df.iloc[0:4,0:6]
```

```
(183929, 48)
```

```
Out[6]:    id_x  player_api_id_x         player_name  player_fifa_api_id  \
        0     1           505942  Aaron Appindangoye              218353
        1     1           505942  Aaron Appindangoye              218353
        2     1           505942  Aaron Appindangoye              218353
        3     1           505942  Aaron Appindangoye              218353

                      birthday  height
        0  1992-02-29 00:00:00  182.88
        1  1992-02-29 00:00:00  182.88
        2  1992-02-29 00:00:00  182.88
        3  1992-02-29 00:00:00  182.88
```

Since the dataset is constantly updated, it is expected to have historical statistics of the same player.

```
In [7]: # Example: filter sigle player stats
        df[df['player_name']=='James Rodriguez']     \
             .head(10)[['player_name', 'overall_rating', 'date']]
```

```
Out[7]:            player_name  overall_rating                 date
        76411  James Rodriguez            87.0  2016-02-18 00:00:00
        76412  James Rodriguez            87.0  2015-10-16 00:00:00
        76413  James Rodriguez            87.0  2015-09-21 00:00:00
        76414  James Rodriguez            86.0  2015-05-01 00:00:00
```

```
76415   James Rodriguez            86.0  2014-12-05 00:00:00
76416   James Rodriguez            86.0  2014-11-28 00:00:00
76417   James Rodriguez            86.0  2014-10-31 00:00:00
76418   James Rodriguez            86.0  2014-10-10 00:00:00
76419   James Rodriguez            86.0  2014-09-18 00:00:00
76420   James Rodriguez            83.0  2014-02-07 00:00:00
```

It is also needed to remove the null/missing values. First, we check if there are missing values:

In [8]:
```
init_rows = df.shape[0]
df.isnull().any().any(), df.shape
```

Out[8]:
```
(True, (183929, 48))
```

In [9]:
```
df.isnull().sum(axis=0)
```

Out[9]:
```
id_x                      0
player_api_id_x           0
player_name               0
player_fifa_api_id        0
birthday                  0
height                    0
weight                    0
id_y                      0
player_api_id_y           0
date                      0
overall_rating          787
potential               787
preferred_foot          787
attacking_work_rate    3181
defensive_work_rate     787
crossing                787
finishing               787
heading_accuracy        787
short_passing           787
volleys                2664
dribbling               787
curve                  2664
free_kick_accuracy      787
long_passing            787
ball_control            787
acceleration            787
sprint_speed            787
agility                2664
reactions               787
balance                2664
shot_power              787
jumping                2664
stamina                 787
```

```
strength                787
long_shots              787
aggression              787
interceptions           787
positioning             787
vision                 2664
penalties               787
marking                 787
standing_tackle         787
sliding_tackle         2664
gk_diving               787
gk_handling             787
gk_kicking              787
gk_positioning          787
gk_reflexes             787
dtype: int64
```

It can be observed that the `attacking_work_rate` attribute is the column which has the most number of missing values, however, to be sure the number of dropped rows is calculated.

```
In [10]: df = df.dropna()
         final_rows = df.shape[0]
         drop_rows = init_rows - final_rows

         print("# of original rows: ", init_rows)
         print("# of rows after cleaning: ", final_rows)
         print("# of deleted rows: ", drop_rows)
```

```
# of original rows:  183929
# of rows after cleaning:  180354
# of deleted rows:  3575
```

Assuring the dataset doesn't have null values anymore:

```
In [11]: df.isnull().any().any(), df.shape
```

```
Out[11]: (False, (180354, 48))
```

Exploring the clean version of the dataset (by checking a slice):

```
In [12]: df.iloc[0:3,0:9]
```

```
Out[12]:    id_x  player_api_id_x         player_name  player_fifa_api_id  \
         0     1           505942  Aaron Appindangoye              218353
         1     1           505942  Aaron Appindangoye              218353
         2     1           505942  Aaron Appindangoye              218353

                       birthday  height  weight  id_y  player_api_id_y
         0  1992-02-29 00:00:00  182.88     187     1           505942
         1  1992-02-29 00:00:00  182.88     187     2           505942
         2  1992-02-29 00:00:00  182.88     187     3           505942
```

As the result of merging the *attribute* and *players* datasets, the *player_api_id* and the *player_fifa_api_id* are repeated columns, therefore to get rid of those columns:

```
In [13]: df.drop(['id_x', 'id_y', 'player_api_id_y'],  \
                  inplace=True, axis=1, errors='ignore')
         df.iloc[-4:-1,0:9]
```

```
Out[13]:         player_api_id_x         player_name  player_fifa_api_id  \
         183925            39902  Zvjezdan Misimovic              102359
         183926            39902  Zvjezdan Misimovic              102359
         183927            39902  Zvjezdan Misimovic              102359

                            birthday  height  weight                 date  \
         183925  1982-06-05 00:00:00  180.34     176  2009-02-22 00:00:00
         183926  1982-06-05 00:00:00  180.34     176  2008-08-30 00:00:00
         183927  1982-06-05 00:00:00  180.34     176  2007-08-30 00:00:00

                 overall_rating  potential
         183925            78.0       80.0
         183926            77.0       80.0
         183927            78.0       81.0
```

4.2 Analysis of data

Getting an overview of the player's attributes by looking at the data columns:

```
In [14]: df.columns
```

```
Out[14]: Index(['player_api_id_x', 'player_name', 'player_fifa_api_id', 'birthday',
                'height', 'weight', 'date', 'overall_rating', 'potential',
                'preferred_foot', 'attacking_work_rate', 'defensive_work_rate',
                'crossing', 'finishing', 'heading_accuracy', 'short_passing', 'volleys',
                'dribbling', 'curve', 'free_kick_accuracy', 'long_passing',
                'ball_control', 'acceleration', 'sprint_speed', 'agility', 'reactions',
                'balance', 'shot_power', 'jumping', 'stamina', 'strength', 'long_shots',
                'aggression', 'interceptions', 'positioning', 'vision', 'penalties',
                'marking', 'standing_tackle', 'sliding_tackle', 'gk_diving',
                'gk_handling', 'gk_kicking', 'gk_positioning', 'gk_reflexes'],
               dtype='object')
```

The initial merging of the data was useful to get an insight of the data and understand, for instance the reason of the multiple values for the same player, nevertheless, for the analysis only the attribute columns are needed.

```
In [15]: df_values = df.iloc[:,7:] #Slicing the attribute columns
         df_values.iloc[:3,:6] #Slice of the dataset
```

```
Out[15]:    overall_rating  potential preferred_foot attacking_work_rate  \
         0            67.0       71.0          right              medium
```

```
1            67.0       71.0          right                  medium
2            62.0       66.0          right                  medium

  defensive_work_rate  crossing
0              medium      49.0
1              medium      49.0
2              medium      49.0
```

Now the attribute data is isolated, the numerical analysis is done by calculating the correlation matrix of the data:

```
In [16]: corr = df_values.corr()
         corr
```

```
Out[16]:                     overall_rating  potential  crossing  finishing  \
         overall_rating            1.000000   0.765435  0.357320   0.330079
         potential                 0.765435   1.000000  0.277284   0.287838
         crossing                  0.357320   0.277284  1.000000   0.576896
         finishing                 0.330079   0.287838  0.576896   1.000000
         heading_accuracy          0.313324   0.206063  0.368956   0.373459
         short_passing             0.458243   0.382538  0.790323   0.580245
         volleys                   0.361739   0.301678  0.637527   0.851482
         dribbling                 0.354191   0.339978  0.809747   0.784988
         curve                     0.357566   0.296050  0.788924   0.691082
         free_kick_accuracy        0.349800   0.262842  0.708763   0.633274
         long_passing              0.434525   0.343133  0.685649   0.341121
         ball_control              0.443991   0.401803  0.807721   0.720694
         acceleration              0.243998   0.338820  0.599439   0.529355
         sprint_speed              0.253048   0.340698  0.579506   0.509647
         agility                   0.239963   0.293714  0.599561   0.554396
         reactions                 0.771856   0.580991  0.384999   0.354769
         balance                   0.160211   0.202232  0.519778   0.394978
         shot_power                0.428053   0.325459  0.656740   0.727835
         jumping                   0.258978   0.174532  0.021270   0.008948
         stamina                   0.325606   0.259432  0.565935   0.347853
         strength                  0.315684   0.122392 -0.072915  -0.054596
         long_shots                0.392668   0.313059  0.716515   0.806895
         aggression                0.322782   0.162137  0.324625   0.044465
         interceptions             0.249094   0.163292  0.306446  -0.152560
         positioning               0.368978   0.326898  0.684803   0.803687
         vision                    0.431493   0.379278  0.693978   0.652376
         penalties                 0.392715   0.315207  0.574208   0.726234
         marking                   0.132185   0.054094  0.234886  -0.285416
         standing_tackle           0.163986   0.082073  0.285018  -0.230453
         sliding_tackle            0.128054   0.063284  0.274673  -0.262144
         gk_diving                 0.027675  -0.012283 -0.604567  -0.479370
         gk_handling               0.006717   0.005865 -0.595646  -0.465135
         gk_kicking                0.028799   0.092299 -0.356728  -0.292349
```

```
gk_positioning           0.008029   0.004472 -0.597742  -0.470758
gk_reflexes              0.007804   0.004936 -0.601696  -0.473302

                    heading_accuracy  short_passing   volleys  dribbling  \
overall_rating              0.313324       0.458243  0.361739   0.354191
potential                   0.206063       0.382538  0.301678   0.339978
crossing                    0.368956       0.790323  0.637527   0.809747
finishing                   0.373459       0.580245  0.851482   0.784988
heading_accuracy            1.000000       0.548435  0.391129   0.400803
short_passing               0.548435       1.000000  0.639995   0.788935
volleys                     0.391129       0.639995  1.000000   0.784247
dribbling                   0.400803       0.788935  0.784247   1.000000
curve                       0.320384       0.731948  0.752410   0.810353
free_kick_accuracy          0.306013       0.693490  0.682909   0.707322
long_passing                0.362741       0.803073  0.414520   0.579201
ball_control                0.550956       0.890622  0.749459   0.901730
acceleration                0.198164       0.502893  0.512931   0.698906
sprint_speed                0.265430       0.490562  0.493721   0.669779
agility                     0.068570       0.510650  0.560021   0.703528
reactions                   0.295601       0.460469  0.397448   0.377852
balance                     0.077255       0.462617  0.416578   0.547666
shot_power                  0.541365       0.722320  0.746622   0.744960
jumping                     0.286305       0.060067  0.023143   0.008645
stamina                     0.477830       0.611422  0.382636   0.527134
strength                    0.493543       0.089782 -0.037103  -0.114107
long_shots                  0.406003       0.729741  0.814894   0.807175
aggression                  0.577304       0.455426  0.127425   0.204592
interceptions               0.454187       0.425764 -0.038534   0.106897
positioning                 0.408972       0.679014  0.779166   0.798720
vision                      0.336472       0.766401  0.690716   0.734119
penalties                   0.431291       0.612511  0.713116   0.663420
marking                     0.460831       0.349578 -0.170094   0.004345
standing_tackle             0.480054       0.415427 -0.108062   0.067306
sliding_tackle              0.441134       0.380148 -0.127810   0.044988
gk_diving                  -0.665600      -0.694111 -0.508029  -0.654097
gk_handling                -0.649145      -0.689874 -0.486178  -0.650645
gk_kicking                 -0.402865      -0.422659 -0.279492  -0.432452
gk_positioning             -0.648981      -0.691030 -0.490148  -0.653560
gk_reflexes                -0.652494      -0.693260 -0.492267  -0.656195

                       curve  free_kick_accuracy     ...        vision  \
overall_rating      0.357566            0.349800     ...      0.431493
potential           0.296050            0.262842     ...      0.379278
crossing            0.788924            0.708763     ...      0.693978
finishing           0.691082            0.633274     ...      0.652376
heading_accuracy    0.320384            0.306013     ...      0.336472
short_passing       0.731948            0.693490     ...      0.766401
volleys             0.752410            0.682909     ...      0.690716
```

dribbling	0.810353	0.707322	...	0.734119
curve	1.000000	0.797842	...	0.728198
free_kick_accuracy	0.797842	1.000000	...	0.697943
long_passing	0.586313	0.603286	...	0.670151
ball_control	0.798598	0.720674	...	0.773185
acceleration	0.549135	0.430657	...	0.470370
sprint_speed	0.516366	0.394006	...	0.435667
agility	0.619243	0.505257	...	0.559152
reactions	0.392756	0.369191	...	0.452559
balance	0.494479	0.431480	...	0.507510
shot_power	0.694945	0.684191	...	0.647262
jumping	-0.017059	-0.033555	...	0.017372
stamina	0.454458	0.416764	...	0.506252
strength	-0.115739	-0.059102	...	-0.039883
long_shots	0.783732	0.773887	...	0.730112
aggression	0.203332	0.232394	...	0.276333
interceptions	0.136119	0.176245	...	0.233281
positioning	0.721106	0.656253	...	0.740857
vision	0.728198	0.697943	...	1.000000
penalties	0.649737	0.669018	...	0.665802
marking	0.032956	0.072918	...	0.080042
standing_tackle	0.094466	0.133147	...	0.144749
sliding_tackle	0.080110	0.105894	...	0.118656
gk_diving	-0.556625	-0.498347	...	-0.502582
gk_handling	-0.544940	-0.491631	...	-0.461778
gk_kicking	-0.333784	-0.279713	...	-0.201738
gk_positioning	-0.549870	-0.494253	...	-0.465109
gk_reflexes	-0.551574	-0.495868	...	-0.470499

	penalties	marking	standing_tackle	sliding_tackle	\
overall_rating	0.392715	0.132185	0.163986	0.128054	
potential	0.315207	0.054094	0.082073	0.063284	
crossing	0.574208	0.234886	0.285018	0.274673	
finishing	0.726234	-0.285416	-0.230453	-0.262144	
heading_accuracy	0.431291	0.460831	0.480054	0.441134	
short_passing	0.612511	0.349578	0.415427	0.380148	
volleys	0.713116	-0.170094	-0.108062	-0.127810	
dribbling	0.663420	0.004345	0.067306	0.044988	
curve	0.649737	0.032956	0.094466	0.080110	
free_kick_accuracy	0.669018	0.072918	0.133147	0.105894	
long_passing	0.476750	0.441837	0.496679	0.462544	
ball_control	0.684410	0.188479	0.252325	0.220588	
acceleration	0.428884	-0.034449	-0.006462	0.000558	
sprint_speed	0.411872	0.005501	0.033576	0.038742	
agility	0.442737	-0.126362	-0.091032	-0.080848	
reactions	0.390045	0.123173	0.159255	0.137098	
balance	0.391050	0.036002	0.063704	0.075284	
shot_power	0.680887	0.094068	0.155941	0.117569	

jumping	0.058097	0.194289	0.188607	0.199657
stamina	0.399233	0.416995	0.459434	0.437001
strength	0.056923	0.356111	0.365857	0.324646
long_shots	0.714596	-0.011926	0.054655	0.022683
aggression	0.218452	0.652986	0.682715	0.654777
interceptions	0.077169	0.835412	0.848326	0.825515
positioning	0.753908	-0.075977	-0.015256	-0.042476
vision	0.665802	0.080042	0.144749	0.118656
penalties	1.000000	-0.043649	0.005405	-0.030404
marking	-0.043649	1.000000	0.950370	0.937716
standing_tackle	0.005405	0.950370	1.000000	0.953264
sliding_tackle	-0.030404	0.937716	0.953264	1.000000
gk_diving	-0.470286	-0.382682	-0.418985	-0.399978
gk_handling	-0.435311	-0.376709	-0.418084	-0.392203
gk_kicking	-0.189194	-0.202480	-0.244711	-0.212487
gk_positioning	-0.439610	-0.370921	-0.412529	-0.386481
gk_reflexes	-0.443635	-0.373047	-0.415528	-0.389127

	gk_diving	gk_handling	gk_kicking	gk_positioning \
overall_rating	0.027675	0.006717	0.028799	0.008029
potential	-0.012283	0.005865	0.092299	0.004472
crossing	-0.604567	-0.595646	-0.356728	-0.597742
finishing	-0.479370	-0.465135	-0.292349	-0.470758
heading_accuracy	-0.665600	-0.649145	-0.402865	-0.648981
short_passing	-0.694111	-0.689874	-0.422659	-0.691030
volleys	-0.508029	-0.486178	-0.279492	-0.490148
dribbling	-0.654097	-0.650645	-0.432452	-0.653560
curve	-0.556625	-0.544940	-0.333784	-0.549870
free_kick_accuracy	-0.498347	-0.491631	-0.279713	-0.494253
long_passing	-0.464221	-0.466906	-0.261361	-0.468453
ball_control	-0.741678	-0.732701	-0.465803	-0.735166
acceleration	-0.481988	-0.465823	-0.279128	-0.468226
sprint_speed	-0.497846	-0.479999	-0.285362	-0.484227
agility	-0.388833	-0.380188	-0.243554	-0.382839
reactions	-0.073940	-0.079753	-0.033067	-0.078091
balance	-0.386654	-0.362838	-0.184204	-0.364395
shot_power	-0.584551	-0.589048	-0.398712	-0.591626
jumping	-0.038265	-0.037035	-0.014910	-0.035719
stamina	-0.554568	-0.543677	-0.320408	-0.542760
strength	-0.070762	-0.083506	-0.061410	-0.084202
long_shots	-0.545446	-0.538702	-0.337826	-0.542884
aggression	-0.433406	-0.431515	-0.270683	-0.430153
interceptions	-0.375410	-0.336190	-0.090819	-0.330958
positioning	-0.546791	-0.505852	-0.233736	-0.511490
vision	-0.502582	-0.461778	-0.201738	-0.465109
penalties	-0.470286	-0.435311	-0.189194	-0.439610
marking	-0.382682	-0.376709	-0.202480	-0.370921
standing_tackle	-0.418985	-0.418084	-0.244711	-0.412529

```
sliding_tackle      -0.399978    -0.392203   -0.212487        -0.386481
gk_diving            1.000000     0.926869    0.583455         0.928024
gk_handling          0.926869     1.000000    0.746524         0.965876
gk_kicking           0.583455     0.746524    1.000000         0.745670
gk_positioning       0.928024     0.965876    0.745670         1.000000
gk_reflexes          0.936807     0.966029    0.739146         0.966407

                    gk_reflexes
overall_rating         0.007804
potential              0.004936
crossing              -0.601696
finishing             -0.473302
heading_accuracy      -0.652494
short_passing         -0.693260
volleys               -0.492267
dribbling             -0.656195
curve                 -0.551574
free_kick_accuracy    -0.495868
long_passing          -0.469598
ball_control          -0.738085
acceleration          -0.469669
sprint_speed          -0.485750
agility               -0.383367
reactions             -0.078300
balance               -0.366849
shot_power            -0.593827
jumping               -0.034010
stamina               -0.547975
strength              -0.083624
long_shots            -0.544979
aggression            -0.432776
interceptions         -0.335409
positioning           -0.516008
vision                -0.470499
penalties             -0.443635
marking               -0.373047
standing_tackle       -0.415528
sliding_tackle        -0.389127
gk_diving              0.936807
gk_handling            0.966029
gk_kicking             0.739146
gk_positioning         0.966407
gk_reflexes            1.000000

[35 rows x 35 columns]
```

A glance at the correlation matrix allows to see that in general the data do not show a strong dependence (direct or inverse) between the attributes. However, since it is hard to see the relationship by reading the numbers, a visualization of the correlation matrix is used:

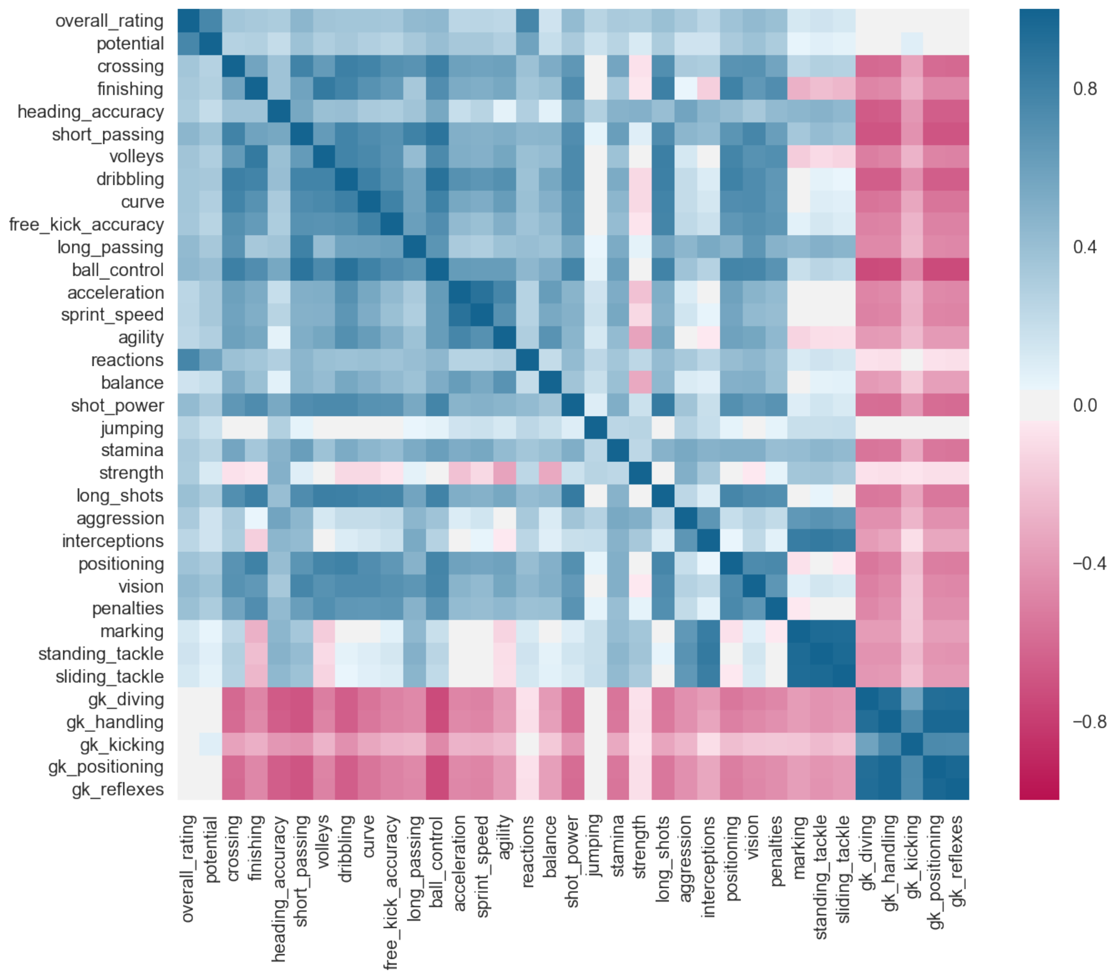
```
In [28]: %matplotlib inline
         %config InlineBackend.figure_format = 'retina'

         import seaborn as sns

         f, ax = plt.subplots(figsize=(10,8))
         sns.heatmap(corr, cmap=sns.diverging_palette(0, 240, s=95, l=40, as_cmap=True),
                     square=True, ax=ax)

         #Optional: rotate & align tick labels
         #for tick in ax.get_xticklabels():
         #    print(tick)
         #    tick.set_rotation(80)
         #    tick.set_horizontalalignment('right')
```

```
Out[28]: <matplotlib.axes._subplots.AxesSubplot at 0x225d3ffada0>
```



Now it is easier to see the dependence between the skills.

It can be confirmed the initial statement about the weak dependence of the data, but some exceptions, e.g. GK related skills and Defensive skills, which make sense, can be also noticed.

Note that the main diagonal of the matrix has a correlation factor of 1.0 because is shows the correlation of the attribute itself.

It is really interesting that the correlation values of the *overall_rating* are surprisingly low which can indicate the use of a complex formula to calculate that parameter.

A bit of searching on the web can reinforce this hypothesis; this article an EA representative explains that the formula used for the overall rating is more complex in the sense that are many other parameters that should be considered, for instance real-life performance and the specific league in which a player plays, among some others. The following is an excerpt of the this article:

ńAll that data is then put into a formula, which spits out the rating we see in game. However, some players just don't work well with this formula, meaning they end up getting rated much lower than their real-world performance would indicate.

Mueller-Moehring gives the example of Thomas Muller, who isn't particularly good at any one thing, according to him. "He always finds the right spot on the pitch, it's amazing. But he's not a great dribbler and he can't really strike the ball properly — his finishing is sometimes really, really off. Shot power is not his strength as well," he went on.

"So if you rate Thomas Muller properly, he ends up with a rating that we say doesn't make sense. It's too low."ż

4.3 Conclusions

- The dataset is extended and it includes historical updates on the skills of players, this allows a better analysis since much more data can be included.
- A direct relationship between the related parameters, e.g. jumping with heading, GK skills, defensive skills and so forth, however the correlation values show this relationship is not as strong as one would expect.
- This analysis could indicate the use of a complex formula in the calculation of the *overall_rating* of the player.
- The correlation matrix analysis therefore is not conclusive in this specific case.