fifa_train 데이터

- 연령(age)
- 대륙(continent)
- 계약종료일(contact_until)
- 포지션(position)
- 선호하는 발(prefer_foot): right, left
- 평판(reputation)
- 현재 능력치(stat_overall)
- 잠재 능력치(stat_potential)
- 기술력(stat_skill_moves)
- 이적료(values)

In [94]:
```
import pandas as pd
import numpy as np
```

In [95]:
```
train = pd.read_csv('FIFA_train.csv')
test = pd.read_csv('FIFA_test.csv')
```

In [96]:
```
train
```

Out[96]:

	id	name	age	continent	contract_until	position	prefer_foot	reputation	stat_overall	stat_potential	stat_skill_moves	value
0	0	L. Messi	31	south america	2021	ST	left	5	94	94	4	110500000
1	3	De Gea	27	europe	2020	GK	right	4	91	93	1	72000000
2	7	L. Suárez	31	south america	2021	ST	right	5	91	91	3	80000000
3	8	Sergio Ramos	32	europe	2020	DF	right	4	91	91	3	51000000
4	9	J. Oblak	25	europe	2021	GK	right	3	90	93	1	68000000
...	...	...	...	...	...	...	...	...	...	...	...	...
8878	16925	S. Adewusi	18	africa	2019	MF	right	1	48	63	3	60000
8879	16936	C. Ehlich	19	europe	2020	DF	right	1	47	59	2	40000
8880	16941	N. Fuentes	18	south america	2021	DF	right	1	47	64	2	50000
8881	16942	J. Milli	18	europe	2021	GK	right	1	47	65	1	50000
8882	16948	N. Christoffersson	19	europe	2020	ST	right	1	47	63	2	60000

8883 rows × 12 columns

In [90]:
```
import random
np.random.seed(1000)
random.seed(1000)
```

In [97]:
```
train.dtypes
```

Out[97]:
```
id                 int64
name              object
age                int64
continent         object
contract_until    object
position          object
prefer_foot       object
reputation         int64
stat_overall       int64
stat_potential     int64
stat_skill_moves   int64
value              int64
dtype: object
```

전처리

In [98]:
```
train.isnull().sum()
```

Out[98]:
```
id                 0
name               0
age                0
continent          0
contract_until     0
position           0
prefer_foot        0
reputation         0
stat_overall       0
stat_potential     0
stat_skill_moves   0
value              0
dtype: int64
```

In [99]:
```
train['contract_until'].value_counts()
```

Out[99]:
```
contract_until
2019          2444
2021          2308
2020          2041
2022           761
2023           506
30-Jun-19      388
2018           327
31-Dec-18       60
31-May-19       13
2024            12
31-Jan-19       10
30-Jun-20        6
2025             3
01-Jan-19        2
2026             1
12-Jan-19        1
Name: count, dtype: int64
```

In [101…
```
train['contract_until'] = train['contract_until'].str.replace('30-Jun-19', '2019')
train['contract_until'] = train['contract_until'].str.replace('31-May-19', '2019')
train['contract_until'] = train['contract_until'].str.replace('31-Jan-19', '2019')
train['contract_until'] = train['contract_until'].str.replace('12-Jan-19', '2019')
train['contract_until'] = train['contract_until'].str.replace('01-Jan-19', '2019')
train['contract_until'] = train['contract_until'].str.replace('31-Dec-18', '2018')
train['contract_until'] = train['contract_until'].str.replace('30-Jun-20', '2020')
```

In [102…
```
train['contract_until'].value_counts()
```

Out[102]:
```
contract_until
2019    2858
2021    2308
2020    2047
2022     761
2023     506
2018     387
2024      12
2025       3
2026       1
Name: count, dtype: int64
```

시각화

In [16]:
```
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
```

In [17]:
```
plt.style.use('ggplot')
```

포지션과 선호하는 발 - 바 그래프

In [18]:
```
train[['position', 'prefer_foot','id']]
```

Out[18]:

	position	prefer_foot	id
0	ST	left	0
1	GK	right	3
2	ST	right	7
3	DF	right	8
4	GK	right	9
…	…	…	…
8878	MF	right	16925
8879	DF	right	16936
8880	DF	right	16941
8881	GK	right	16942
8882	ST	right	16948

8883 rows × 3 columns

In [19]:
```
train[['position', 'prefer_foot','id']].groupby(['position','prefer_foot']).count()
```

Out[19]:

		id
position	**prefer_foot**	
DF	**left**	908
	right	1876
GK	**left**	98
	right	907
MF	**left**	757
	right	2645
ST	**left**	320
	right	1372

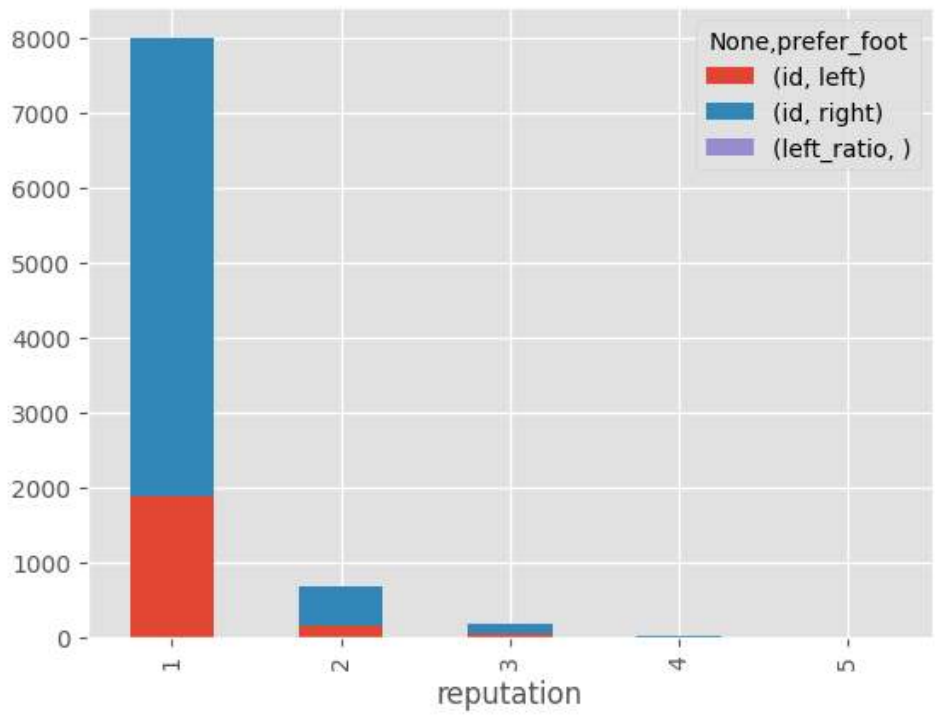
In [23]:
```
embarked_df = train[['position', 'prefer_foot','id']].groupby(['position','prefer_foot']).count().unstack()
embarked_df
```

Out[23]:

		id
prefer_foot	left	right
position		
DF	908	1876
GK	98	907
MF	757	2645
ST	320	1372

In [128…
```
embarked_df.plot.bar(stacked = True)
```

Out[128]:
```
<Axes: xlabel='reputation'>
<Figure size 500x300 with 0 Axes>
```



평판과 발 선호도

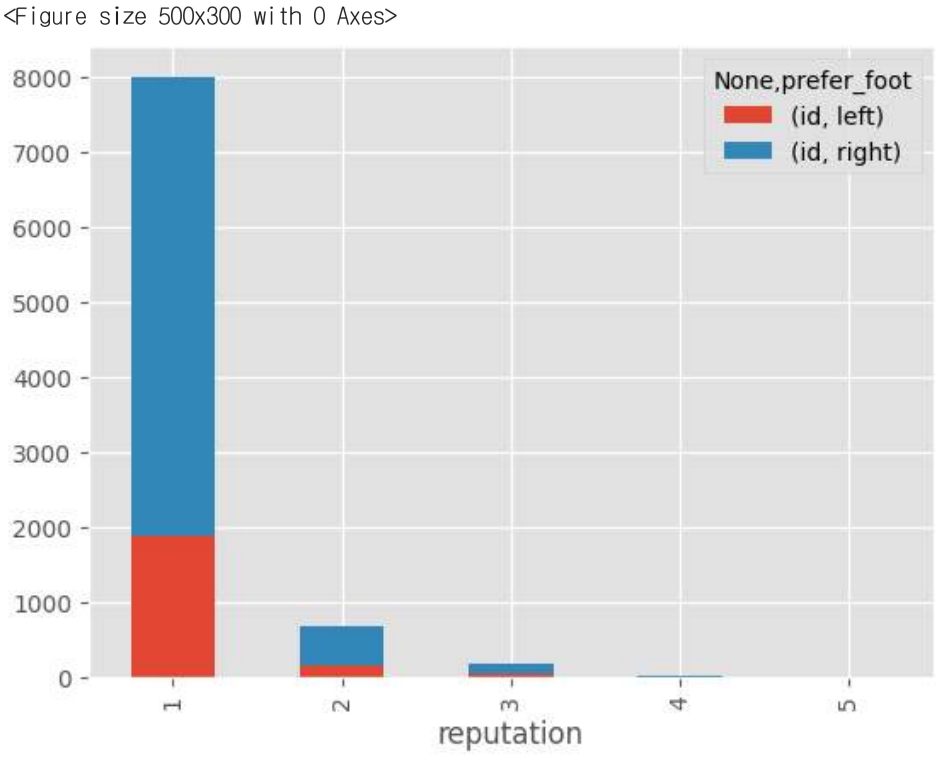
In [106…
```
embarked_df = train[['reputation', 'prefer_foot','id']].groupby(['reputation','prefer_foot']).count().unstack()
embarked_df
```

Out[106]:

		id
prefer_foot	left	right
reputation		
1	1877	6127
2	162	511
3	36	136
4	7	23
5	1	3

In [127…
```
ticket_df = train[['reputation','prefer_foot','id']].dropna().groupby(['reputation','prefer_foot']).count().unstack()
ticket_df.plot.bar(stacked=True)
```

Out[127]:
```
<Axes: xlabel='reputation'>
<Figure size 500x300 with 0 Axes>
```



In [107…
```
embarked_df['left_ratio'] = embarked_df['id']['left'] / embarked_df['id']['right']
embarked_df
```

Out[107]:

prefer_foot	left	right	id left_ratio
reputation			
1	1877	6127	0.306349
2	162	511	0.317025
3	36	136	0.264706
4	7	23	0.304348
5	1	3	0.333333

연령 별 선호하는 발을 히스토그램으로 표현

In [33]:
```
train['age'].describe()
```

Out[33]:
```
count    8883.000000
mean       25.208713
std         4.643264
min        16.000000
25%        21.000000
50%        25.000000
75%        29.000000
max        40.000000
Name: age, dtype: float64
```

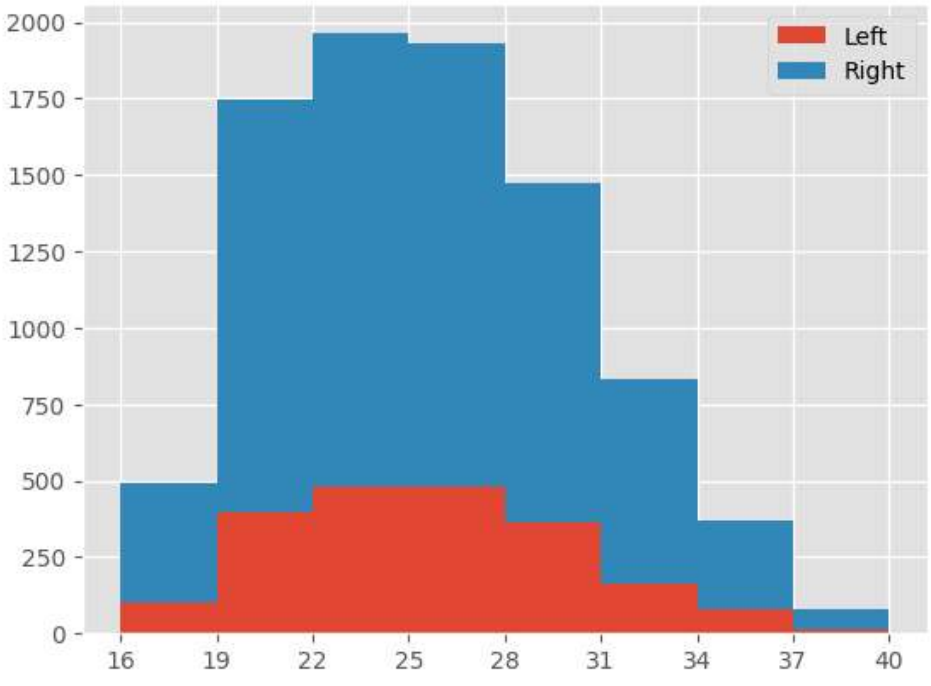
In [50]:
```
embarked_df = train[['age', 'prefer_foot','id']].groupby(['age','prefer_foot']).count().unstack()
embarked_df
```

Out[50]:

prefer_foot	id left	right
age		
16	3.0	15.0
17	31.0	100.0
18	69.0	275.0
19	117.0	373.0
20	127.0	454.0
21	154.0	522.0
22	137.0	491.0
23	156.0	497.0
24	187.0	495.0
25	160.0	493.0
26	179.0	523.0
27	144.0	430.0
28	131.0	397.0
29	112.0	359.0
30	121.0	353.0
31	65.0	280.0
32	58.0	227.0
33	40.0	161.0
34	56.0	160.0
35	17.0	72.0
36	8.0	56.0
37	6.0	36.0
38	2.0	15.0
39	3.0	13.0
40	NaN	3.0

In [131…
```
plt.hist(x=[train.age[train.prefer_foot =="left"], train.age[train.prefer_foot == "right"]],
           bins = 8, histtype='barstacked', label = ['Left','Right'])
plt.xticks(range(16, 41, 3), [str(i) for i in range(16, 41, 3)])
plt.legend()
```

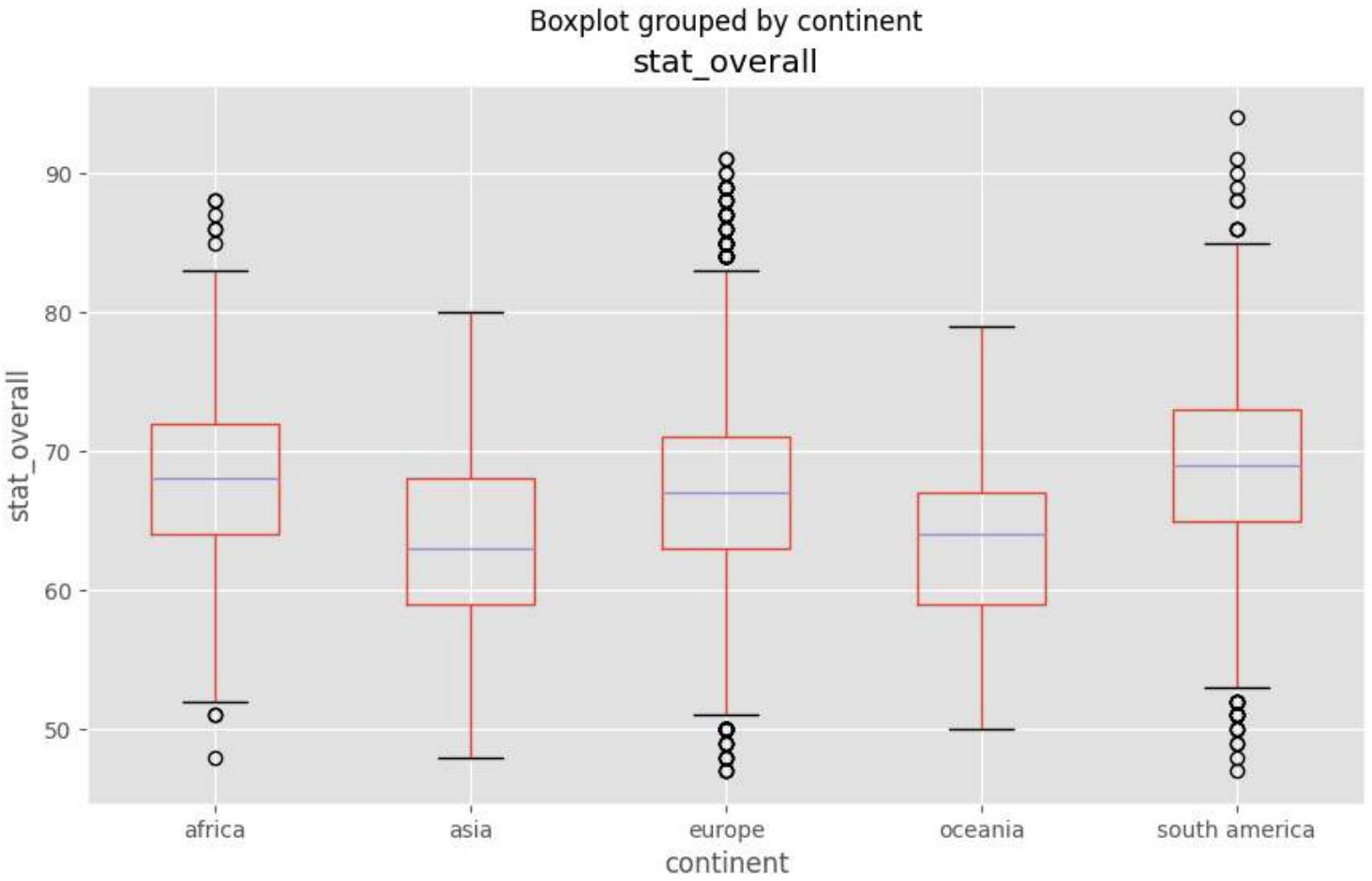
Out[131]:
```
<matplotlib.legend.Legend at 0x28aff8cf670>
```



박스플롯

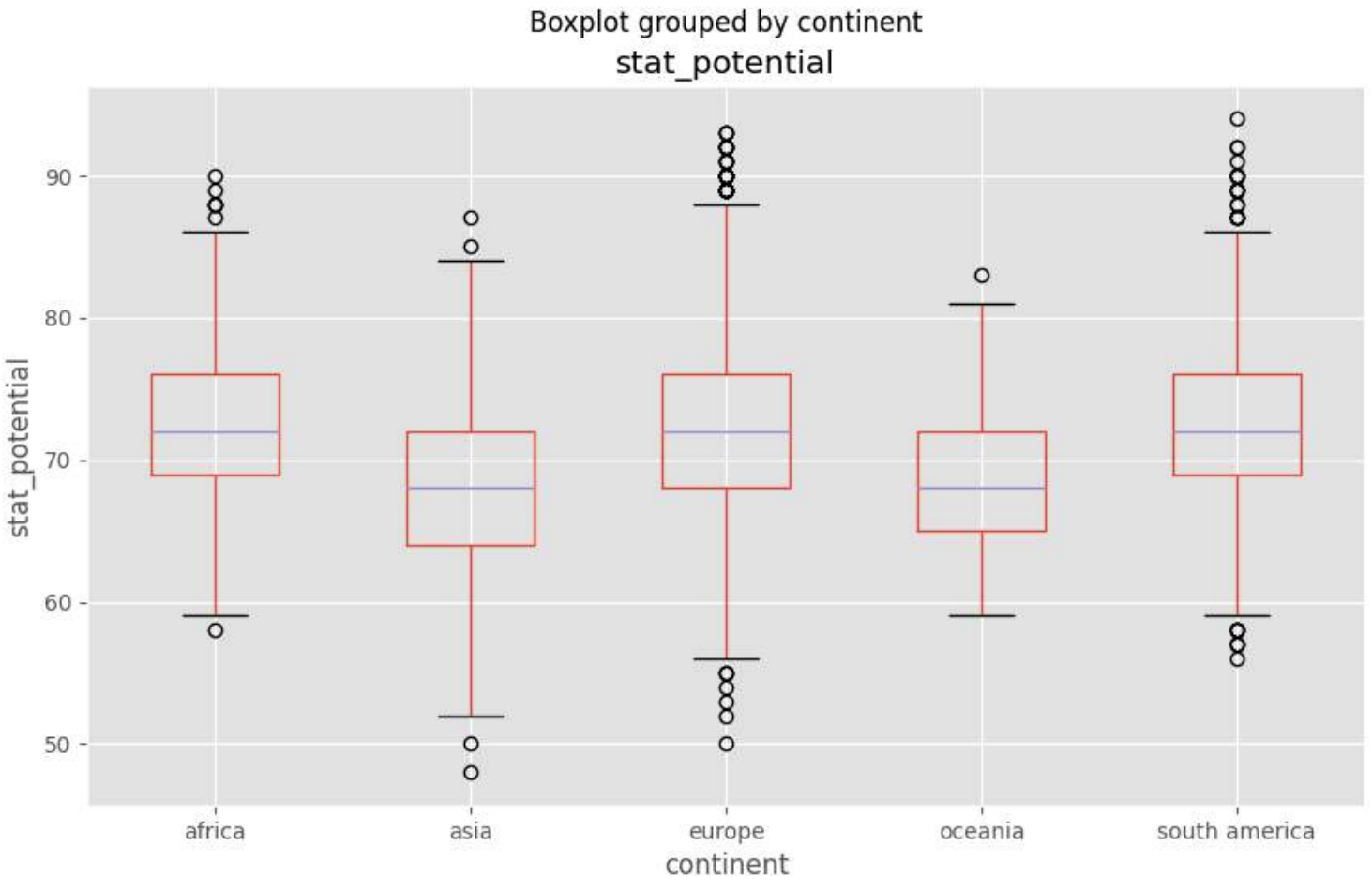
In [129…
```
#대륙별 현재 능력치
train.boxplot(column='stat_overall', by='continent', figsize=(10, 6))
plt.ylabel('stat_overall')
```

Out[129]:
```
Text(0, 0.5, 'stat_overall')
<Figure size 500x300 with 0 Axes>
```



In [130…
```
#대륙별 잠재 능력치
train.boxplot(column='stat_potential', by='continent', figsize=(10, 6))
plt.ylabel('stat_potential')
```

Out[130]:
```
Text(0, 0.5, 'stat_potential')
<Figure size 500x300 with 0 Axes>
```



상관관계분석

In [62]:
```
train.dtypes
```

Out[62]:
```
id                 int64
name              object
age                int64
continent         object
contract_until    object
position          object
prefer_foot       object
reputation         int64
stat_overall       int64
stat_potential     int64
stat_skill_moves   int64
value              int64
dtype: object
```

In [66]:
```
df_corr = train[['age','continent','position','reputation','stat_overall','stat_potential','stat_skill_moves','value']]
```

In [67]:
```
df_corr
```

Out[67]:

	age	continent	position	reputation	stat_overall	stat_potential	stat_skill_moves	value
0	31	south america	ST	5	94	94	4	110500000
1	27	europe	GK	4	91	93	1	72000000
2	31	south america	ST	5	91	91	3	80000000
3	32	europe	DF	4	91	91	3	51000000
4	25	europe	GK	3	90	93	1	68000000
...	...	...	...	...	...	...	...	...
8878	18	africa	MF	1	48	63	3	60000
8879	19	europe	DF	1	47	59	2	40000
8880	18	south america	DF	1	47	64	2	50000
8881	18	europe	GK	1	47	65	1	50000
8882	19	europe	ST	1	47	63	2	60000

8883 rows × 8 columns

In [69]:
```
df_corr = pd.get_dummies(df_corr, columns=['continent','position'])
```

In [70]:
```
df_corr.head()
```

Out[70]:

	age	reputation	stat_overall	stat_potential	stat_skill_moves	value	continent_africa	continent_asia	continent_europe	continent_oceania	continent_south america	position_DF	positio
0	31	5	94	94	4	110500000	False	False	False	False	True	False	
1	27	4	91	93	1	72000000	False	False	True	False	False	False	
2	31	5	91	91	3	80000000	False	False	False	False	True	False	
3	32	4	91	91	3	51000000	False	False	True	False	False	True	
4	25	3	90	93	1	68000000	False	False	True	False	False	False	

In [72]:
```
df_corr_res = df_corr.corr()
```

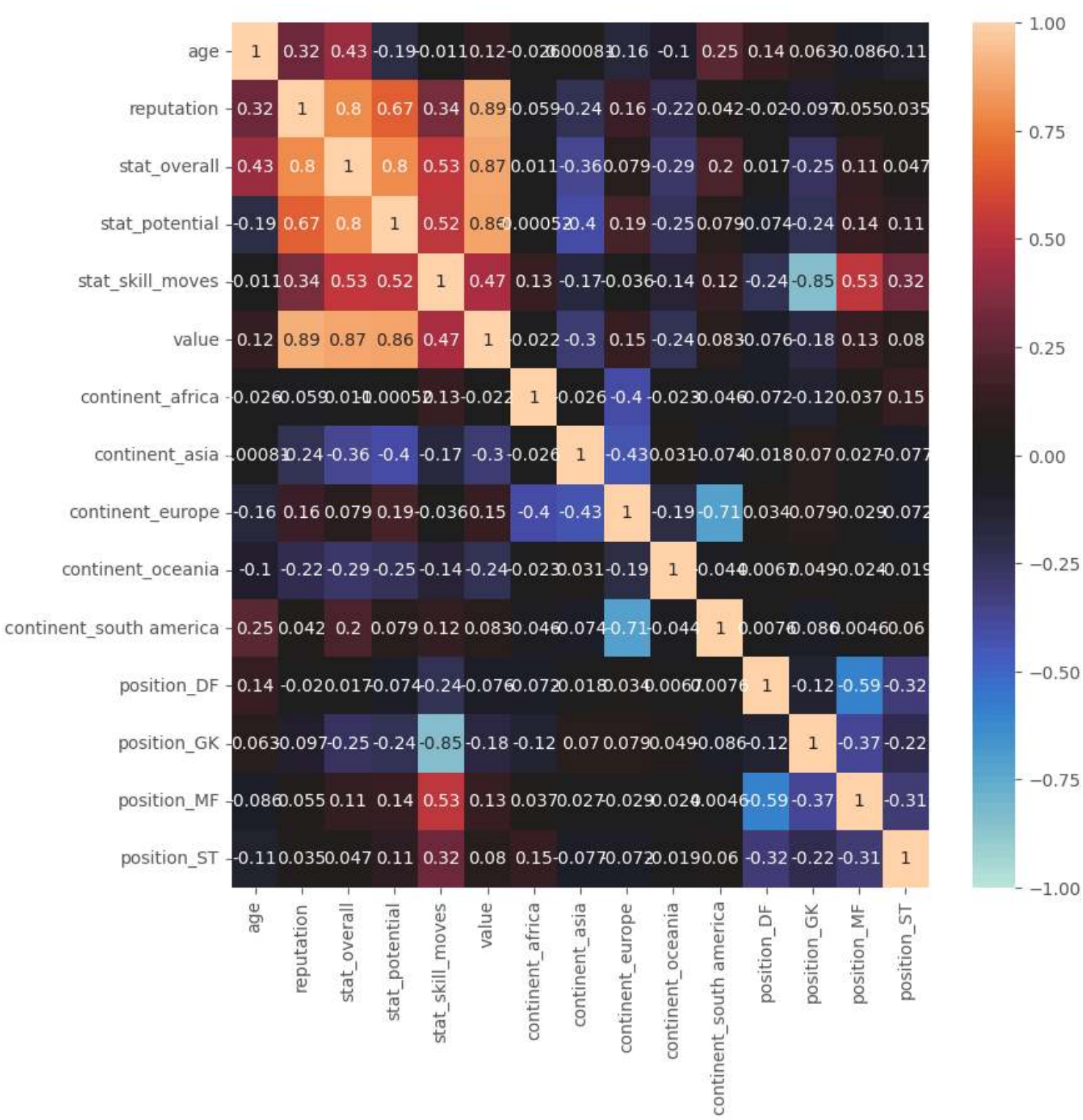
In [73]:
```
df_corr_res
```

Out[73]:

	age	reputation	stat_overall	stat_potential	stat_skill_moves	value	continent_africa	continent_asia	continent_europe	continent_oceania	continent_south america
age	1.000000	0.318879	0.427288	-0.192996	-0.010552	0.119474	-0.026449	0.000814	-0.158905	-0.100230	0.246510
reputation	0.318879	1.000000	0.796329	0.666404	0.340456	0.892277	-0.058821	-0.236849	0.160088	-0.221088	0.042426
stat_overall	0.427288	0.796329	1.000000	0.796941	0.532457	0.870486	0.010628	-0.364174	0.079079	-0.285272	0.204383
stat_potential	-0.192996	0.666404	0.796941	1.000000	0.518440	0.859533	-0.000519	-0.404287	0.194251	-0.248747	0.078714
stat_skill_moves	-0.010552	0.340456	0.532457	0.518440	1.000000	0.471352	0.131337	-0.172264	-0.036032	-0.143812	0.119332
value	0.119474	0.892277	0.870486	0.859533	0.471352	1.000000	-0.021999	-0.297886	0.145624	-0.239465	0.083332
continent_africa	-0.026449	-0.058821	0.010628	-0.000519	0.131337	-0.021999	1.000000	-0.026046	-0.402450	-0.023388	-0.046256
continent_asia	0.000814	-0.236849	-0.364174	-0.404287	-0.172264	-0.297886	-0.026046	1.000000	-0.433451	0.031325	-0.073622
continent_europe	-0.158905	0.160088	0.079079	0.194251	-0.036032	0.145624	-0.402450	-0.433451	1.000000	-0.191759	-0.714585
continent_oceania	-0.100230	-0.221088	-0.285272	-0.248747	-0.143812	-0.239465	-0.023388	0.031325	-0.191759	1.000000	-0.043977
continent_south america	0.246510	0.042426	0.204383	0.078714	0.119332	0.083332	-0.046256	-0.073622	-0.714585	-0.043977	1.000000
position_DF	0.136768	-0.020412	0.017436	-0.073528	-0.235570	-0.075726	-0.072209	-0.017613	0.034304	0.006717	0.007576
position_GK	0.063363	-0.096508	-0.252644	-0.242119	-0.848716	-0.180062	-0.124998	0.070472	0.078763	0.048855	-0.086243
position_MF	-0.086004	0.054971	0.112296	0.144545	0.526889	0.125620	0.037451	0.027012	-0.028847	-0.023787	0.004615
position_ST	-0.107087	0.034724	0.046502	0.105646	0.315237	0.079791	0.147801	-0.077061	-0.072095	-0.018638	0.060026

In [74]:
```
plt.figure(figsize = (9,9))
sns.heatmap(df_corr_res, vmax = 1, vmin = -1, center = 0, annot = True)
```

Out[74]: <Axes: >



In [ ]: