

Global Soccer Analytics

Daniel Oman and Pranu Prakash





Points of Interest & Purpose

What factors make a club successful?

What role do finances play in the success of a team?



Background - Soccer

Position	Code
Right Wing	RW
Left Wing	LW
Center Forward	CF
Striker	ST
Center Midfield	CM
Center Attacking Mid	CAM
Center Defensive Mid	CDM
Goalkeeper	GK
Right Back	RB
Left Back	LB
Center Back	CB
Left Midfield	LM
Right Midfield	RM
Right Wingback	RWB
Left Wingback	LWB



Background - Terminology

Wage: how much a player is paid yearly

Value: how much a player would be worth on the transfer market at any given time

Overall: numerical rating given to each player on the videogame FIFA 22 according to their real life performance across multiple different relevant categories (attacking, mentality, skill, defending, etc). Ranges from 0-100.





Insights

1. What average characteristics do successful teams share?
2. Does average age of squad, average overall rating, and club value determine position on table?
3. Which position on the field has the most valuable players?
4. What countries have the most talented and valuable players?
5. Which state is the most profitable for collegiate soccer in the US?

Data Collection Process

Data from:

Web-scraped: International player data from the FIFA 22 database + additional country data

Downloaded CSV: US collegiate soccer data

API Database: Data from the 'Big Five' leagues



'PL'
,



'PD'



'BL1'









'FL1'



'SA'

Data Collection: Web Scrapping (player data)

NAME	AGE	OVA	POT	TEAM & CONTRACT	HEIGHT	WEIGHT	FOOT	VALUE	WAGE	ATTACKING	SKILL	MOVEMENT	POWER	MENTALITY	DEFENDING	GOALKEEPING	TOTAL
 L. Messi RW ST CF	34	93	93	Paris Saint-Germain 2021 ~ 2023	170cm	72kg	Left	€78M	€320K	429	470	451	389	347	79	54	2219
 R. Lewandowski ST	32	92	92	FC Bayern München 2014 ~ 2023	185cm	81kg	Right	€119.5M	€270K	430	407	407	424	396	96	51	2211
 K. Mbappé ST LW	22	91	95	Paris Saint-Germain 2018 ~ 2022	182cm	73kg	Right	€194M	€230K	411	404	462	411	353	92	42	2175
 M. Salah RW	29	91	91	Liverpool 2017 ~ 2023	175cm	71kg	Left	€129M	€350K	402	412	457	399	382	122	62	2236
 K. De Bruyne CM CAM	30	91	91	Manchester City 2015 ~ 2025	181cm	70kg	Right	€125.5M	€350K	406	439	400	408	406	186	56	2301
 Neymar Jr LW CAM	29	91	91	Paris Saint-Germain 2017 ~ 2025	175cm	68kg	Right	€129M	€270K	403	446	451	359	367	96	59	2181

	name	nationality	team	positions	age	overall	potential	contract	contract_start	contract_end	...	value	wage	attacking	skill	movement	power	mentality	defending	goalkeeping	total
0	L. Messi	Argentina	Paris Saint-Germain	RW ST CF	34	93	93	2021 ~ 2023	2021	2023	...	78000000.0	320000.0	429	470	451	389	347	79	54	2219
1	R. Lewandowski	Poland	FC Bayern München	ST	32	92	92	2014 ~ 2023	2014	2023	...	119500000.0	270000.0	430	407	407	424	396	96	51	2211
2	K. Mbappé	France	Paris Saint-Germain	ST LW	22	91	95	2018 ~ 2022	2018	2022	...	194000000.0	230000.0	411	404	462	411	353	92	42	2175
3	M. Salah	Egypt	Liverpool	RW	29	91	91	2017 ~ 2023	2017	2023	...	129000000.0	350000.0	402	412	457	399	382	122	62	2236
4	K. De Bruyne	Belgium	Manchester City	CM CAM	30	91	91	2015 ~ 2025	2015	2025	...	125500000.0	350000.0	406	439	400	408	406	186	56	2301
...
369	L. Rudden	Republic of Ireland	Finn Harps	ST	19	47	60	2021 ~ 2022	2021	2022	...	110000.0	500.0	190	190	329	229	184	35	53	1210
870	R. Gallagher	Republic of Ireland	Finn Harps	CAM	20	47	61	2018 ~ 2022	2018	2022	...	110000.0	500.0	195	193	290	221	207	97	52	1255
871	G. Singh	India	Mumbai City FC	ST LM	17	47	62	2021 ~ 2024	2021	2024	...	110000.0	500.0	195	196	340	251	222	135	50	1389

Data Collection: Web Scrapping (country data)

Countries or Areas						
No ^	Country or Area	ISO-alpha3 Code	M49 Code	Region 1	Region 2	Continent
1	Afghanistan	AFG	004	Southern Asia		Asia
2	Åland Islands	ALA	248	Northern Europe		Europe
3	Albania	ALB	008	Southern Europe		Europe
4	Algeria	DZA	012	Northern Africa		Africa
5	American Samoa	ASM	016	Polynesia		Oceania
6	Andorra	AND	020	Southern Europe		Europe
7	Angola	AGO	024	Middle Africa	Sub-Saharan Africa	Africa
8	Anguilla	ALA	660	Caribbean	Latin America and the Caribbean	North America
9	Antarctica	ATA	010	Antarctica		Antarctica
10	Antigua and Barbuda	ATG	028	Caribbean	Latin America and the Caribbean	North America
11	Argentina	ARG	032	South America	Latin America and the Caribbean	South America
12	Armenia	ARM	051	Western Asia		Asia
13	Aruba	ABW	533	Caribbean	Latin America and the Caribbean	North America
14	Australia	AUS	036	Australia and New Zealand		Oceania
15	Austria	AUT	040	Western Europe		Europe
16	Azerbaijan	AZE	031	Western Asia		Asia
17	Bahamas	BHS	044	Caribbean	Latin America and the Caribbean	North America
18	Bahrain	BHR	048	Western Asia		Asia
19	Bangladesh	BGD	050	Southern Asia		Asia

	Country or Area	ISO-alpha3 Code	M49 Code	Region 1	Region 2	Continent
0	Afghanistan	AFG	004	Southern Asia	None	Asia
1	Åland Islands	ALA	248	Northern Europe	None	Europe
2	Albania	ALB	008	Southern Europe	None	Europe
3	Algeria	DZA	012	Northern Africa	None	Africa
4	American Samoa	ASM	016	Polynesia	None	Oceania
...
244	Wallis and Futuna Islands	WLF	876	Polynesia	None	Oceania
245	Western Sahara	ESH	732	Northern Africa	None	Africa
246	Yemen	YEM	887	Western Asia	None	Asia
247	Zambia	ZMB	894	Eastern Africa	Sub-Saharan Africa	Africa
248	Zimbabwe	ZWE	716	Eastern Africa	Sub-Saharan Africa	Africa

Data Collection: CSV

state_cd	ClassificationCode	classification_name	EFMaleCount	EFFemaleCount	EFTotalCount	PARTIC_MEN	...	WOMEN_FTHDCOACH_MALE	WOMEN_FTHDCOACH_FEM	REV_MEN	REV_WOMEN	EXP_MEN	EXP_WOMEN
AL	2	NCAA Division I-FCS	1951	3024	4975	0.0	...	1.0	0.0	0.0	579985.0	0.0	579989.0
AL	1	NCAA Division I-FBS	4020	6252	10272	33.0	...	0.0	1.0	856140.0	865816.0	856140.0	865816.0
AL	5	NCAA Division II without football	3915	2818	6733	31.0	...	1.0	0.0	289306.0	342040.0	289306.0	342040.0
AL	2	NCAA Division I-FCS	1258	2188	3446	0.0	...	1.0	0.0	0.0	435889.0	0.0	435889.0
AL	1	NCAA Division I-FBS	13112	15902	29014	0.0	...	1.0	0.0	0.0	816094.0	0.0	1883398.0
...
ME	6	NCAA Division III with football	1072	1332	2404	42.0	...	0.0	0.0	170721.0	60025.0	128510.0	58790.0

```
'institution_name', 'addr1_txt', 'city_txt', 'state_cd',
'ClassificationCode', 'classification_name', 'EFMaleCount',
'EFFemaleCount', 'EFTotalCount', 'PARTIC_MEN', 'PARTIC_WOMEN',
'SUM_FTHDCOACH_MALE', 'SUM_FTHDCOACH_FEM', 'MEN_FTHDCOACH_MALE',
'MEN_FTHDCOACH_FEM', 'WOMEN_FTHDCOACH_MALE', 'WOMEN_FTHDCOACH_FEM',
'REV_MEN', 'REV_WOMEN', 'EXP_MEN', 'EXP_WOMEN', 'division',
'profit_men', 'profit_women', 'net_profits'],
```

state_cd	ClassificationCode	classification_name	EFMaleCount	EFFemaleCount	EFTotalCount	PARTIC_MEN	...	WOMEN_FTHDCOACH_MALE	WOMEN_FTHDCOACH_FEM	REV_MEN	REV_WOMEN	EXP_MEN	EXP_WOMEN
AL	2	NCAA Division I-FCS	1951	3024	4975	0.0	...	1.0	0.0	0.0	579985.0	0.0	579989.0
AL	1	NCAA Division I-FBS	4020	6252	10272	33.0	...	0.0	1.0	856140.0	865816.0	856140.0	865816.0
AL	5	NCAA Division II without football	3915	2818	6733	31.0	...	1.0	0.0	289306.0	342040.0	289306.0	342040.0
AL	2	NCAA Division I-FCS	1258	2188	3446	0.0	...	1.0	0.0	0.0	435889.0	0.0	435889.0
AL	1	NCAA Division I-FBS	13112	15902	29014	0.0	...	1.0	0.0	0.0	816094.0	0.0	1883398.0
...
ME	6	NCAA Division III with football	1072	1332	2404	42.0	...	0.0	0.0	170721.0	60025.0	128510.0	58790.0

Data Collection: API

API									
Competitions			Matches		Teams		Areas		
Competition			Match		Team		Area		
Matches	Standings	Teams	Lineups	Scorer	Squad / Staff				
			Refs	Bookings					

	id	team	position	points	playedGames	won	lost	draw	goalsFor	goalsAgainst	goalDifference	form	league
0	5	FC Bayern München	1	75	31	24	4	3	92	30	62	Good	BL1
1	4	Borussia Dortmund	2	63	31	20	8	3	77	46	31	Good	BL1
2	3	Bayer 04 Leverkusen	3	55	31	16	8	7	72	44	28	Good	BL1
3	721	RB Leipzig	4	54	31	16	9	6	66	33	33	Good	BL1
4	17	Sport-Club Freiburg	5	52	31	14	7	10	52	37	15	Good	BL1
...
93	584	U.C. Sampdoria	16	30	34	8	20	6	41	57	-16	Bad	SA
94	104	Cagliari	17	28	34	6	18	10	31	62	-31	Bad	SA
95	455	US Salernitana 1919	18	25	33	6	20	7	28	70	-42	Bad	SA
96	107	Genoa	19	25	34	3	15	16	25	54	-29	Bad	SA
97	454	Venezia FC	20	22	33	5	21	7	27	61	-34	Bad	SA

98 rows x 13 columns

```

"squad": [
  {
    "id": 3176,
    "name": "Matthias Ginter",
    "position": "Defender",
    "dateOfBirth": "1994-01-19T00:00:00Z",
    "countryOfBirth": "Germany",
    "nationality": "Germany",
    "role": "PLAYER"
  },
  {
    "id": 3185,
    "name": "Lars Stindl",
    "position": "Midfielder",
    "dateOfBirth": "1988-08-26T00:00:00Z",
    "countryOfBirth": "Germany",
    "nationality": "Germany",
    "role": "PLAYER"
  }
]

```

Insight 1

What average characteristics do successful teams share?

- Compare many average variables a team has to the team's standing in the league table.
- Calculate correlations between each variable and standing across leagues

Function to return slope of best fit, intercept of best fit, and correlation, between x and y variables:

```
def linear_data(x, y):  
    """Returns tuple of slope, intercept, pearson's correlation coefficient"""  
    theta = np.polyfit(x, y, 1)  
    correlation = np.corrcoef(x, y)[0,1]  
    return theta[0], theta[1], correlation
```

✓ 0.3s

Pyth

Function to return correlation data per league (df). Returns slope, intercept, correlation between any two given variables per league. Variable names as strings, they are column headers from league_data and player_data, and aggregate function to apply to Y variable

```
def league_correlation_data(X, Y, X_agg, Y_agg):  
    df = player_data.merge(league_data, on="team").groupby(["league", "team"]).aggregate(x_agg = (X, X_agg), y_agg = (Y, Y_agg))  
    slope = df.groupby("league").apply(lambda d: linear_data(x = d["x_agg"], y = d["y_agg"])[0])  
    intercept = df.groupby("league").apply(lambda d: linear_data(x = d["x_agg"], y = d["y_agg"])[1])  
    correlation = df.groupby("league").apply(lambda d: linear_data(x = d["x_agg"], y = d["y_agg"])[2])  
    linear_stats = pd.DataFrame({"slope":slope, "intercept":intercept, "correlation":correlation})  
    linear_stats["x_name"], linear_stats["x_agg"], linear_stats["y_name"], linear_stats["y_agg"] = X, X_agg, Y, Y_agg  
    linear_stats.reset_index(inplace = True)  
    return linear_stats
```

Insight 1

Heat maps:

- league on one axis, variable on the other
 - variable vs variable, will create symmetric heat map
- We are plotting correlations between each pair of variables here

We are only looking at mean as the aggregates to compare (for now). Create data for all possible combinations:

```
columns = ['overall', 'contract_length', 'height', 'weight', 'value', 'attacking', 'skill', 'mentality', 'defending', 'position', 'wage']
correlation_data = pd.DataFrame(columns=["league", "slope", "intercept", "correlation", "x_name", "x_agg", "y_name", "y_agg"])
for x in columns:
    for y in columns:
        correlation_data = pd.concat([correlation_data, league_correlation_data(X = x, Y = y, X_agg="mean", Y_agg="mean")], axis = 0)
3.3s
```

Now plot league vs variables correlation with position heatmap:

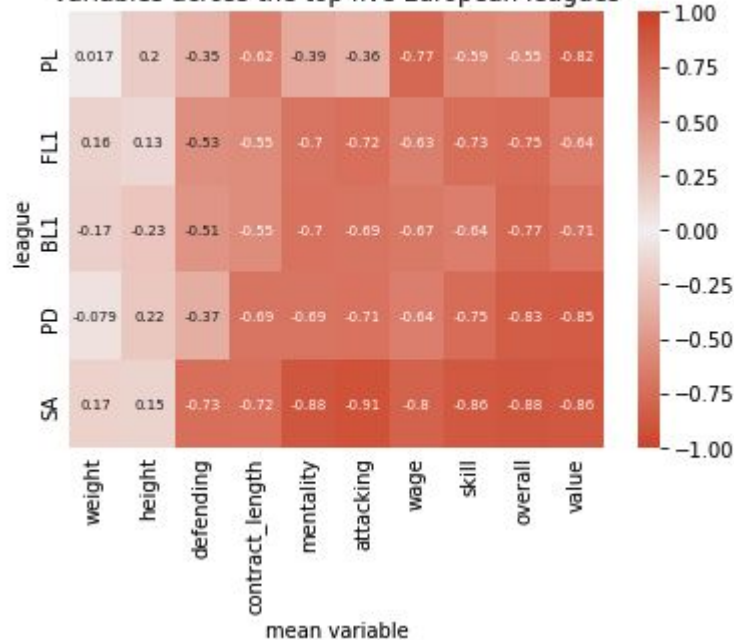
```
p = correlation_data[(correlation_data["x_name"] == "position") & (correlation_data["y_name"] != "position")].pivot(index = "league", columns = "y_name", values = "correlation")
p["abs_sum_1"] = p.abs().sum(axis=1) # create dummy column just for sorting
p.sort_values(by="abs_sum_1", ascending=True, inplace=True, axis=0) # values sorted on y-axis, so that cells with stronger correlations are at the bottom
p.loc["abs_sum_0",:] = p.abs().sum(axis=0) # create dummy column just for sorting
p.sort_values(by="abs_sum_0", ascending=True, inplace=True, axis=1) # values sorted on x-axis so that cells with stronger correlations are towards the right
p.drop("abs_sum_1", axis=1, inplace=True) # drop dummy columns
p.drop("abs_sum_0", axis=0, inplace=True)
ax = sns.heatmap(p, annot=True, center = 0, cmap = sns.diverging_palette(20,20, as_cmap=True), vmax=1, vmin=-1, annot_kws={"size": 7})
ax.tick_params(axis='both', which='both', length=0)
ax.set(title = "Correlation between position on table and other average \nvariables across the top five European leagues", xlabel="mean variable")
ax
```

✓ 0.6s

Python

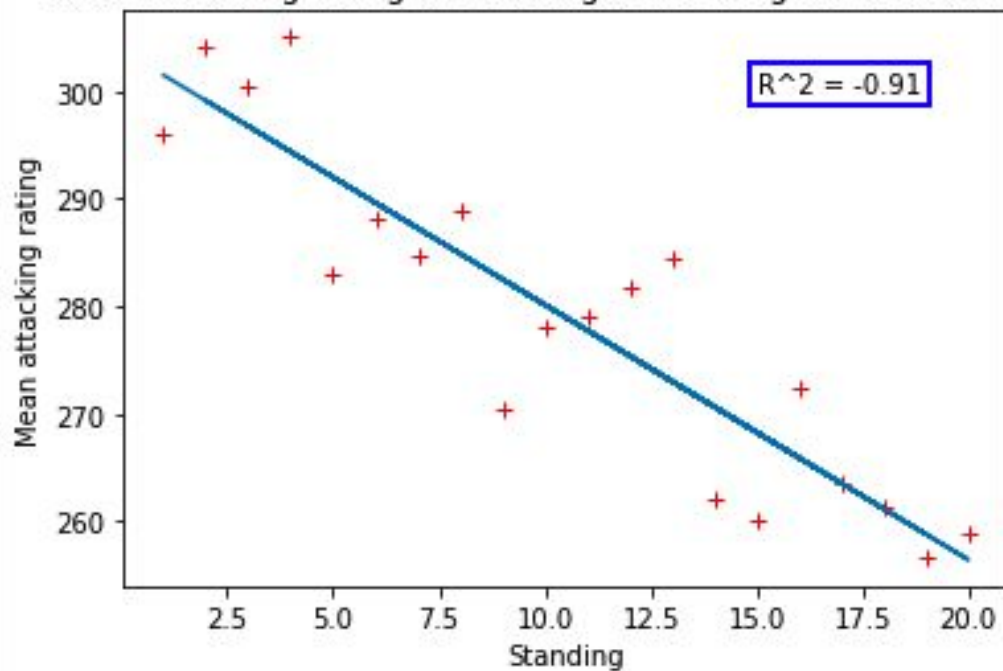
Visualization for insight 1

Correlation between position on table and other average variables across the top five European leagues



```
y_name  
value           0.778014  
overall         0.757060  
skill           0.713166  
wage            0.702148  
attacking       0.676871  
mentality       0.672674  
contract_length 0.626713  
defending       0.497023  
height          0.184613  
weight          0.119941
```

Mean attacking rating versus league standing for Serie A teams



Insight 2

```
[ ] def insight2():
    import numpy as np
    import pandas as pd

    web_scrape = pd.read_csv("player_data.csv", index_col=0).reset_index().drop("index", axis = 1)
    api = pd.read_csv("league_data.csv", index_col=0)
    final_api = api.loc[:, ["team", "position", "points", "won", "goalsFor", "form"]]
    final_api.loc[:, "form"] = np.where(final_api.loc[:, "form"] == "Good", 1, 0) # rating changes to binary

    # return final_api

    final_wc = web_scrape.groupby("team").agg({"age": "mean", "overall": "mean", "value": "mean"})

    # print(final_api)
    # print(final_wc)

    df = final_api.merge(final_wc, how="outer", on="team")
    updated_df = df.dropna(axis=0)
    # return updated_df

    bl1_df = updated_df.iloc[0:18] # bundesliga
    fl1_df = updated_df.iloc[18:38] # ligue 1
    pd_df = updated_df.iloc[38:58] # spain
    pl_df = updated_df.iloc[58:78] # premier league
    sa_df = updated_df.iloc[78:98] # serie a

    return bl1_df, fl1_df, pd_df, pl_df, sa_df
```

```
def visual2():
[ ] import matplotlib.pyplot as plt
    from pandas.plotting import andrews_curves

    # Data
    bl1_df, fl1_df, pd_df, pl_df, sa_df = insight2()
    final_list = [bl1_df, fl1_df, pd_df, pl_df, sa_df]

    ## Test Plot (Andrew Curve)

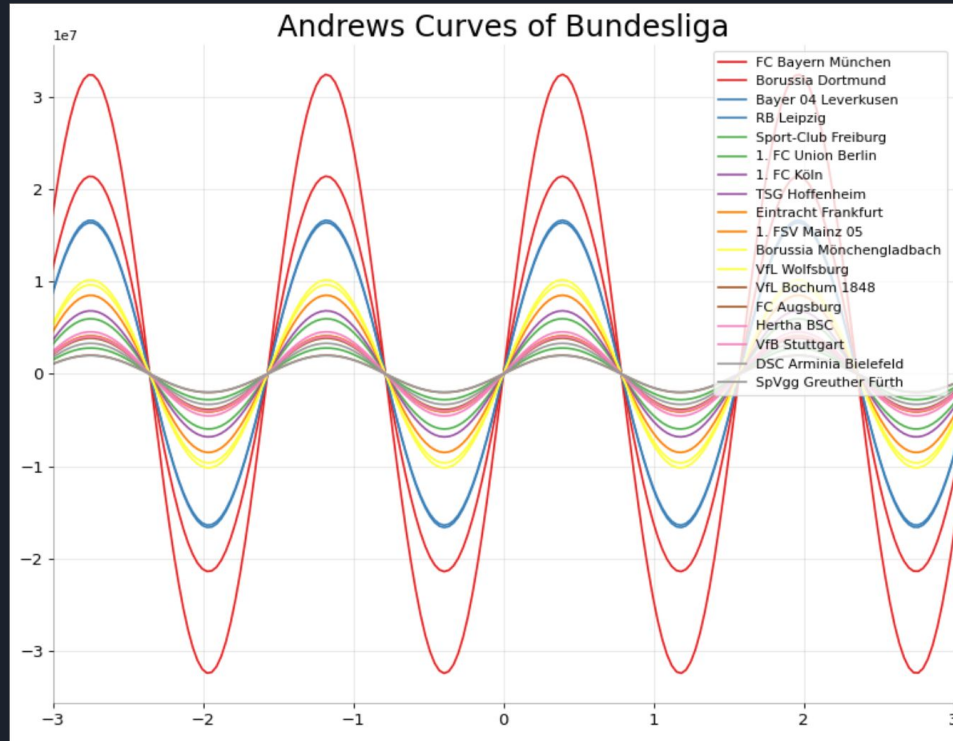
    for i in range(len(final_list)):
        # Plot
        plt.figure(figsize=(12,9), dpi= 80)
        andrews_curves(final_list[i], 'team', colormap='Set1')

        # Lighten borders
        plt.gca().spines["top"].set_alpha(0)
        plt.gca().spines["bottom"].set_alpha(.3)
        plt.gca().spines["right"].set_alpha(0)
        plt.gca().spines["left"].set_alpha(.3)

        if i == 0:
            plt.title('Andrews Curves of Bundesliga', fontsize=22)
        elif i == 1:
            plt.title('Andrews Curves of Ligue 1', fontsize=22)
        elif i == 2:
            plt.title('Andrews Curves of Liga BBVA', fontsize=22)
        elif i == 3:
            plt.title('Andrews Curves of Premier League', fontsize=22)
        elif i == 4:
            plt.title('Andrews Curves of Serie A', fontsize=22)

        plt.xlim(-3,3)
        plt.grid(alpha=0.3)
        plt.xticks(fontsize=12)
        plt.yticks(fontsize=12)
        plt.show()
```

Visualization for Insight 2





Insight 3

Which position on the field has the most valuable players?

- Explode player data's dataframes positions column into multiple rows for each position (some players have more than 1 position listed)

	name	nationality	wage	value	overall	positions
0	L. Messi	Argentina	320000.0	78000000.0	93	RW
1	L. Messi	Argentina	320000.0	78000000.0	93	ST
2	L. Messi	Argentina	320000.0	78000000.0	93	CF

- Group by country, find mean stats
- Group by position, find mean stats

Insight 3

Group by nationality and the position, and then find mean wage, value, and overall. The sort by nationality and mean wage.

```
positions_data.groupby(["nationality", "positions"]).aggregate(mean_wage = ("wage", "mean"), mean_value = ("value", "mean"),  
mean_overall = ("overall", "mean"), count = ("overall", "count")).sort_values(by=["nationality", "mean_wage"], ascending=[True, False])
```

✓ 0.1s

		mean_wage	mean_value	mean_overall	count
nationality	positions				
Afghanistan	RM	2000.000000	8.750000e+05	64.000000	1
Albania	GK	35000.000000	9.500000e+06	76.000000	2
	RB	25666.666667	3.933333e+06	72.666667	3
	ST	19750.000000	3.412500e+06	67.500000	8
	CF	17000.000000	4.450000e+06	69.500000	2
...
Zimbabwe	CB	8000.000000	2.041667e+06	68.000000	3
	LB	5500.000000	1.750000e+06	67.500000	2
	RB	2000.000000	9.000000e+05	67.000000	1
	CF	950.000000	3.000000e+06	73.000000	1
	LW	500.000000	1.200000e+06	67.000000	1

Insight 3

Top 10 highest paid average positions and nationality.

```
positions_data.groupby(["nationality", "positions"]).aggregate(mean_wage = ("wage", "mean"), mean_value = ("value", "mean"),  
mean_overall = ("overall", "mean"), count = ("overall", "count")).nlargest(10, columns = ["count", "mean_wage"])
```

✓ 0.9s

			mean_wage	mean_value	mean_overall	count
	nationality	positions				
United Kingdom of Great Britain and Northern Ireland		CM	9712.840909	2.201909e+06	63.877273	440
		CB	10700.883838	2.290114e+06	64.671717	396
		ST	8919.300912	2.229331e+06	63.449848	329
	Germany	CB	8112.500000	2.524560e+06	66.052817	284
	Spain	CM	15056.054688	5.917988e+06	69.753906	256
	France	CM	13659.836066	5.074508e+06	68.344262	244
	Germany	CDM	9568.907563	3.366681e+06	66.516807	238
		CM	11876.059322	4.321758e+06	67.080508	236
United Kingdom of Great Britain and Northern Ireland		CDM	9972.127660	2.069319e+06	65.153191	235
	Spain	CB	12163.938053	3.718938e+06	68.694690	226

Insight 3

General positions, not by country:

+ Code

+ Markdown

```
pdf = positions_data.groupby("positions").aggregate(mean_wage = ("wage", "mean"), mean_value = ("value", "mean"), mean_overall = ("overall", "mean"), count = ("overall", "count")).sort_values(by = "mean_wage", ascending=False)
pdf
```

✓ 0.8s

	mean_wage	mean_value	mean_overall	count
positions				
CF	17966.594828	6.354558e+06	68.452586	464
LW	12331.116071	4.448741e+06	66.964286	1120
RW	11687.724820	3.975139e+06	66.495504	1112
CAM	10214.833183	3.566465e+06	66.867899	2218
ST	9956.412978	3.155987e+06	66.053669	3298
RWB	9799.381625	2.765724e+06	66.489399	566
CM	9749.404617	3.198731e+06	66.241555	4115
CDM	9436.946750	3.016451e+06	66.780775	2892
RM	9284.663235	3.200663e+06	66.667525	2331
LM	9153.805497	3.275273e+06	66.727696	2365
LWB	9114.446367	2.700190e+06	66.435986	578
CB	8570.397384	2.572697e+06	66.031942	3976
RB	8484.245237	2.476925e+06	65.881290	2047
LB	8428.213563	2.492153e+06	66.075405	1976
GK	6335.662824	1.967351e+06	64.426513	2082

Attacking positions are higher paid on average. Attacking players are higher up the table than defensive players. This could be because they are the playmakers, they score goals!



Insight 4

What countries have the most talented and valuable players?

- Match nationalities with ISO codes.
- Group by nationality and continent
- Select european countries
- Get statistics (mean, standard deviation, 25/50/75%, max, min, count) for different variables
- Sort by mean overall rating in descending order

Insight 4

```
player_data = pd.read_csv("player_data.csv", index_col=0).reset_index().drop("index", axis = 1)
countries = pd.read_csv("countries.csv", index_col=0)
df = player_data.merge(countries, left_on="nationality",
                      right_on="Country or Area")["nationality",
                                                  "overall", "value",
                                                  "wage", "ISO-alpha3 Code",
                                                  "Continent"].groupby(["Continent",
                                                                           "nationality",
                                                                           "ISO-alpha3 Code"]).describe().reset_index()

df = df[df["Continent"] == "Europe"]
return df.sort_values(by = ("wage", "mean"), ascending=False)
```

	Continent	nationality	ISO-alpha3 Code	overall							...		value		wage						
				count	mean	std	min	25%	50%	75%	...	75%	max	count	mean	std	min	25%	50%	75%	max
108	Europe	Ukraine	UKR	64.0	70.312500	5.359682	54.0	66.75	71.5	74.00	...	47750000.0	320000000.0	64.0	7682.812500	18064.222332	500.0	500.0	800.0	3000.0	90000.0
88	Europe	Italy	ITA	327.0	70.027523	7.072966	50.0	65.00	70.0	75.00	...	48000000.0	1195000000.0	327.0	19664.831804	26848.650191	500.0	3000.0	8000.0	26000.0	170000.0
98	Europe	Portugal	PRT	368.0	69.798913	6.309268	52.0	65.00	69.0	74.00	...	42000000.0	1115000000.0	368.0	13892.934783	30532.577038	500.0	2000.0	5000.0	11000.0	270000.0
105	Europe	Spain	ESP	1086.0	69.439227	6.145208	50.0	65.00	69.0	74.00	...	40000000.0	900000000.0	1086.0	14273.664825	24738.737172	500.0	2000.0	5000.0	16000.0	210000.0
76	Europe	Bosnia and Herzegovina	BIH	63.0	69.396825	4.887574	58.0	66.00	69.0	72.00	...	25500000.0	250000000.0	63.0	12679.365079	24736.232135	500.0	2000.0	6000.0	12000.0	155000.0
102	Europe	Serbia	SRB	118.0	69.305085	6.048958	53.0	65.00	69.0	73.00	...	35000000.0	630000000.0	118.0	13395.762712	21220.923371	500.0	2000.0	6000.0	13000.0	130000.0
78	Europe	Croatia	HRV	154.0	69.240260	5.968386	51.0	65.00	69.0	73.00	...	37000000.0	480000000.0	154.0	13113.311688	26387.611003	500.0	950.0	3000.0	12000.0	190000.0

Visualizations for insight 4

- 2 Choropleth maps of Europe, one for average overall rating and one for average wage.

```
cp1 = go.Choropleth(  
    locations=df['ISO-alpha3 Code'], # Spatial coordinates  
    z = df['value']['mean'], # Data to be color-coded  
    colorscale = 'Greens',  
    colorbar_title = "Value",  
    colorbar=dict(len=1, x=0.45, y=0.49),  
)  
cp2 = go.Choropleth(  
    locations=df['ISO-alpha3 Code'], # Spatial coordinates  
    z = df['overall']['mean'], # Data to be color-coded  
    colorscale = 'Reds',  
    colorbar_title = "Overall",  
    colorbar=dict(len=1, x=1, y=0.49)  
)  
fig = make_subplots(rows = 1, cols = 2, specs=[{"type":"choropleth"}, {"type":"choropleth"}], subplot_titles=('Mean Player Value Per Country', 'Mean Player Overall Rating Per Country'))  
fig.add_trace(cp1, 1, 1)  
fig.add_trace(cp2, 1, 2)  
fig.update_layout(height=400, width = 800, showlegend=False, **{'geo' + str(i) + '_scope': 'europe' for i in [''] + np.arange(2,3).tolist()})  
fig
```

Insight 4

Mean Player Value Per Country



Value

6M

4M

2M

Mean Player Overall Rating Per Country



Overall

70

68

66

64

62

Insight 5

```
[ ] def insight5():
    import pandas as pd
    import numpy as np

    player_data = pd.read_csv("collegiate_soccer_data.csv", index_col=0).reset_index().drop("index", axis = 1)

    updated_df = player_data.groupby("state_cd").mean()

    updated_df["net_profit"] = updated_df["profit_men"] + updated_df["profit_women"]

    final_df = updated_df.drop(["EFTotalCount", "ClassificationCode", "PARTIC_MEN", "PARTIC_WOMEN", "SUM_FTHDCOACH_MALE", "SUM_FTHDCOACH_FEM", 'MEN_FTHEADCOACH_MALE', 'MEN_FTHEADCOACH_FEM', 'WOMEN_FTHDCOACH_MALE', 'WOMEN_FTHDCOACH_FEM', "division"], axis=1)

    final_df["marginal_revenue"] = (final_df["REV_MEN"] - final_df["REV_WOMEN"]) / (final_df["EFMaleCount"] - final_df["EFFFemaleCount"])

    final_df["marginal_revenue"] = np.where(final_df["marginal_revenue"] <= 0, final_df["marginal_revenue"] * -1, final_df["marginal_revenue"])


    final_df["avg_revenue_men"] = final_df["REV_MEN"] / final_df["EFMaleCount"]

    final_df["avg_revenue_women"] = final_df["REV_WOMEN"] / final_df["EFFFemaleCount"]

    final_df = final_df.drop(["EFMaleCount", "EFFFemaleCount", "REV_MEN", "REV_WOMEN", "EXP_MEN", "EXP_WOMEN"], axis=1)

    final_df = final_df.sort_values(by=["net_profit", "marginal_revenue"], ascending=False)

    final_df.to_csv("final_school_df.csv")
    return final_df
```



	profit_men	profit_women	net_profit	marginal_revenue	avg_revenue_men	avg_revenue_women
state_cd						
WV	85274.500000	50113.500000	135388.000000	28.553980	354.529955	314.717211
DE	8050.200000	4802.200000	12852.400000	120.835129	134.893112	130.735483
DC	26833.250000	-16553.250000	10280.000000	34.090407	343.352203	193.010911
MO	5615.137255	2273.254902	7888.392157	103.719337	201.097153	180.663453
RI	1278.444444	3643.888889	4922.333333	52.452447	180.151971	156.424880
NV	2068.800000	1562.600000	3631.400000	211.340000	50.739544	82.100816
MA	1983.126984	1543.555556	3526.682540	122.879562	118.946673	119.325383
SD	2461.615385	-455.538462	2006.076923	1641.803644	75.938116	255.288353
ME	3172.888889	-1382.833333	1790.055556	240.955938	112.897106	129.849731
NH	418.083333	68.083333	486.166667	9.579648	143.489961	99.122789
NM	320.454545	111.818182	432.272727	475.672409	44.615643	139.969692
ND	0.000000	0.000000	0.000000	3690.077103	40.490647	284.314881
VI	0.000000	0.000000	0.000000	204.347826	239.795918	0.000000
PR	0.000000	0.000000	0.000000	10.008866	20.322839	11.944571
AK	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
NJ	-2301.888889	-2463.644444	-4765.533333	8.576301	113.913635	99.578865
NY	-4705.121622	-4548.114865	-9253.236486	30.022403	103.875644	91.196458



Conclusion

Overall Premise of Investigation: What makes a football team successful?

Conclusions from the data:

- Italy is a good place to be a professional footballer (Choropleth Map of Europe)
- Age, height, weight of squad doesn't dictate position on table
- Position on table is dictated by good standing/balance in many categories (Andrews Curve analysis)
- WV is the best state to play soccer, best institutions as best net profit for men and women

Challenges

- Country and team names did not align between datasets
 - Needed to write an algorithm to re-change names from api based on downloaded dataset names