WELCOME TO THE NOTEBOOK

Importing the modules

import numpy as np import pandas as pd import plotly.graph_objects as go import plotly.express as px import plotly import re print('modules are imported')

In [1]:
```
import numpy as np
import pandas as pd
import plotly.graph_objects as go
import plotly.express as px
import plotly
import re
import os
print('modules are imported')
```

```
modules are imported
```

Loading the fifa 2020 dataset

In [2]:
```
df_20=pd.read_csv('players_20.csv')
```

In [3]:
```
df_20.head()
```

Out[3]:

	sofifa_id	player_url	short_name	long_name	age	dob	height_cm	weight_kg	nationality	club	...	lwb	ldm	cdm	rdm	rwb	lb	lcb	cb	rcb	rb
0	158023	https://sofifa.com/player/158023/lionel-messi/...	L. Messi	Lionel Andrés Messi Cuccittini	32	1987-06-24	170	72	Argentina	FC Barcelona	...	68+2	66+2	66+2	66+2	68+2	63+2	52+2	52+2	52+2	63+2
1	20801	https://sofifa.com/player/20801/c-ronaldo-dos-...	Cristiano Ronaldo	Cristiano Ronaldo dos Santos Aveiro	34	1985-02-05	187	83	Portugal	Juventus	...	65+3	61+3	61+3	61+3	65+3	61+3	53+3	53+3	53+3	61+3
2	190871	https://sofifa.com/player/190871/neymar-da-sil...	Neymar Jr	Neymar da Silva Santos Junior	27	1992-02-05	175	68	Brazil	Paris Saint-Germain	...	66+3	61+3	61+3	61+3	66+3	61+3	46+3	46+3	46+3	61+3
3	200389	https://sofifa.com/player/200389/jan-oblak/20/...	J. Oblak	Jan Oblak	26	1993-01-07	188	87	Slovenia	Atlético Madrid	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	183277	https://sofifa.com/player/183277/eden-hazard/2...	E. Hazard	Eden Hazard	28	1991-01-07	175	74	Belgium	Real Madrid	...	66+3	63+3	63+3	63+3	66+3	61+3	49+3	49+3	49+3	61+3

5 rows × 104 columns

In [4]:
```
df_20.shape
```

Out[4]:
```
(18278, 104)
```

In [5]:
```
cols=list(df_20.columns)
print(cols)
```

```
['sofifa_id', 'player_url', 'short_name', 'long_name', 'age', 'dob', 'height_cm', 'weight_kg', 'nationality', 'club', 'overall', 'potential', 'value_eur', 'wage_eur', 'player_positions', 'preferred_foot', 'international_reputation', 'weak_foot', 'skill_moves', 'work_rate', 'body_type', 'real_face', 'release_clause_eur', 'player_tags', 'team_position', 'team_jersey_number', 'loaned_from', 'joined', 'contract_valid_until', 'nation_position', 'nation_jersey_number', 'pace', 'shooting', 'passing', 'dribbling', 'defending', 'physic', 'gk_diving', 'gk_handling', 'gk_kicking', 'gk_reflexes', 'gk_speed', 'gk_positioning', 'player_traits', 'attacking_crossing', 'attacking_finishing', 'attacking_heading_accuracy', 'attacking_short_passing', 'attacking_volleys', 'skill_dribbling', 'skill_curve', 'skill_fk_accuracy', 'skill_long_passing', 'skill_ball_control', 'movement_acceleration', 'movement_sprint_speed', 'movement_agility', 'movement_reactions', 'movement_balance', 'power_shot_power', 'power_jumping', 'power_stamina', 'power_strength', 'power_long_shots', 'mentality_aggression', 'mentality_interceptions', 'mentality_positioning', 'mentality_vision', 'mentality_penalties', 'mentality_composure', 'defending_marking', 'defending_standing_tackle', 'defending_sliding_tackle', 'goalkeeping_diving', 'goalkeeping_handling', 'goalkeeping_kicking', 'goalkeeping_positioning', 'goalkeeping_reflexes', 'ls', 'st', 'rs', 'lw', 'lf', 'cf', 'rf', 'rw', 'lam', 'cam', 'ram', 'lm', 'lcm', 'cm', 'rcm', 'rm', 'lwb', 'ldm', 'cdm', 'rdm', 'rwb', 'lb', 'lcb', 'cb', 'rcb', 'rb']
```

Data Preprocessing:

Dropping some useless columns

In [6]:
```
useless_cols=['dob','sofifa_id','player_url','long_name','body_type','real_face','loaned_from','nation_position', 'nation_jersey_number']
```

In [7]:
```
df_20 = df_20.drop(useless_cols, axis=1)
```

In [8]:
```
df_20.head()
```

Out[8]:

	short_name	age	height_cm	weight_kg	nationality	club	overall	potential	value_eur	wage_eur	...	lwb	ldm	cdm	rdm	rwb	lb	lcb	cb	rcb	rb
0	L. Messi	32	170	72	Argentina	FC Barcelona	94	94	95500000	565000	...	68+2	66+2	66+2	66+2	68+2	63+2	52+2	52+2	52+2	63+2
1	Cristiano Ronaldo	34	187	83	Portugal	Juventus	93	93	58500000	405000	...	65+3	61+3	61+3	61+3	65+3	61+3	53+3	53+3	53+3	61+3
2	Neymar Jr	27	175	68	Brazil	Paris Saint-Germain	92	92	105500000	290000	...	66+3	61+3	61+3	61+3	66+3	61+3	46+3	46+3	46+3	61+3
3	J. Oblak	26	188	87	Slovenia	Atlético Madrid	91	93	77500000	125000	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	E. Hazard	28	175	74	Belgium	Real Madrid	91	91	90000000	470000	...	66+3	63+3	63+3	63+3	66+3	61+3	49+3	49+3	49+3	61+3

5 rows × 95 columns

Calculating BMI

calculating body max index of each Player

In [9]:
```
df_20['BMI']=df_20['weight_kg']/(df_20['height_cm']/100)**2
```

In [10]:
```
df_20['BMI'].head()
```

Out[10]:
```
0    24.913495
1    23.735308
2    22.204082
3    24.615211
4    24.163265
Name: BMI, dtype: float64
```

Player's Position

Converting the categorical values in Player's Position column in integer values.

In [11]:
```
df_20[['short_name','player_positions']]
```

Out[11]:

	short_name	player_positions
0	L. Messi	RW, CF, ST
1	Cristiano Ronaldo	ST, LW
2	Neymar Jr	LW, CAM
3	J. Oblak	GK
4	E. Hazard	LW, CF
...	...	...
18273	Shao Shuai	CB
18274	Xiao Mingjie	CB
18275	Zhang Wei	CM
18276	Wang Haijian	CM
18277	Pan Ximing	CM

18278 rows × 2 columns

In [12]:
```
new_player_positions=df_20['player_positions'].str.get_dummies(sep=',').add_prefix('Position')
new_player_positions.head()
```

Out[12]:

	Position CAM	Position CB	Position CDM	Position CF	Position CM	Position LB	Position LM	Position LW	Position LWB	Position RB	...	PositionGK	PositionLB	PositionLM	PositionLW	PositionLWB	PositionRB	PositionRM	PositionRW	PositionRWB	PositionST
0	0	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	1	0	0	...	0	0	0	0	0	0	0	0	0	1
2	1	0	0	0	0	0	0	0	0	0	...	0	0	0	1	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	1	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	...	0	0	0	1	0	0	0	0	0	0

5 rows × 29 columns

In [13]:
```
df_20=pd.concat([df_20,new_player_positions], axis=1)
df_20.head()
```

Out[13]:

	short_name	age	height_cm	weight_kg	nationality	club	overall	potential	value_eur	wage_eur	...	PositionGK	PositionLB	PositionLM	PositionLW	PositionLWB	PositionRB	PositionRM	PositionRW	PositionRWB	PositionST
0	L. Messi	32	170	72	Argentina	FC Barcelona	94	94	95500000	565000	...	0	0	0	0	0	0	0	1	0	0
1	Cristiano Ronaldo	34	187	83	Portugal	Juventus	93	93	58500000	405000	...	0	0	0	0	0	0	0	0	0	1
2	Neymar Jr	27	175	68	Brazil	Paris Saint-Germain	92	92	105500000	290000	...	0	0	0	1	0	0	0	0	0	0
3	J. Oblak	26	188	87	Slovenia	Atlético Madrid	91	93	77500000	125000	...	1	0	0	0	0	0	0	0	0	0
4	E. Hazard	28	175	74	Belgium	Real Madrid	91	91	90000000	470000	...	0	0	0	1	0	0	0	0	0	0

5 rows × 125 columns

Positioning Columns ratings

Cleaning, Processing and Assigning the new attributes to columns listed below.

In [14]:
```
columns = ['ls', 'st', 'rs', 'lw', 'lf', 'cf', 'rf', 'rw', 'lam', 'cam', 'ram',
       'lm', 'lcm', 'cm', 'rcm', 'rm', 'lwb', 'ldm', 'cdm', 'rdm', 'rwb', 'lb',
       'lcb', 'cb', 'rcb', 'rb']
```

In [15]:
```
df_20[columns].head()
```

Out[15]:

	ls	st	rs	lw	lf	cf	rf	rw	lam	cam	...	lwb	ldm	cdm	rdm	rwb	lb	lcb	cb	rcb	rb
0	89+2	89+2	89+2	93+2	93+2	93+2	93+2	93+2	93+2	93+2	...	68+2	66+2	66+2	66+2	68+2	63+2	52+2	52+2	52+2	63+2
1	91+3	91+3	91+3	89+3	90+3	90+3	90+3	89+3	88+3	88+3	...	65+3	61+3	61+3	61+3	65+3	61+3	53+3	53+3	53+3	61+3
2	84+3	84+3	84+3	90+3	89+3	89+3	89+3	90+3	90+3	90+3	...	66+3	61+3	61+3	61+3	66+3	61+3	46+3	46+3	46+3	61+3
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	83+3	83+3	83+3	89+3	88+3	88+3	88+3	89+3	89+3	89+3	...	66+3	63+3	63+3	63+3	66+3	61+3	49+3	49+3	49+3	61+3

5 rows × 26 columns

Now, lets omit the '+' sign

In [18]:
```
for col in columns:
    df_20[col]=df_20[col].str.split('+',n=1,expand=True)[0]
    df_20[columns]
```

In [19]:
```
df_20[columns]
```

Out[19]:

	ls	st	rs	lw	lf	cf	rf	rw	lam	cam	...	lwb	ldm	cdm	rdm	rwb	lb	lcb	cb	rcb	rb
0	89	89	89	93	93	93	93	93	93	93	...	68	66	66	66	68	63	52	52	52	63
1	91	91	91	89	90	90	90	89	88	88	...	65	61	61	61	65	61	53	53	53	61
2	84	84	84	90	89	89	89	90	90	90	...	66	61	61	61	66	61	46	46	46	61
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	83	83	83	89	88	88	88	89	89	89	...	66	63	63	63	66	61	49	49	49	61
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
18273	32	32	32	31	31	31	31	31	31	31	...	43	42	42	42	43	45	46	46	46	45
18274	33	33	33	33	32	32	32	33	33	33	...	44	43	43	43	44	46	47	47	47	46
18275	43	43	43	43	43	43	43	43	44	44	...	47	49	49	49	47	47	49	49	49	47
18276	43	43	43	45	44	44	44	45	46	46	...	48	48	48	48	48	48	49	49	49	48
18277	42	42	42	44	43	43	43	44	46	46	...	48	49	49	49	48	48	50	50	50	48

18278 rows × 26 columns

Deleting the Nan values, lets replace Nan with 0

In [20]:
```
df_20[columns] = df_20[columns].fillna(0)
```

Converting columns into int here

In [ ]:

In [21]:
```
df_20[columns]
```

Out[21]:

	ls	st	rs	lw	lf	cf	rf	rw	lam	cam	...	lwb	ldm	cdm	rdm	rwb	lb	lcb	cb	rcb	rb
0	89	89	89	93	93	93	93	93	93	93	...	68	66	66	66	68	63	52	52	52	63
1	91	91	91	89	90	90	90	89	88	88	...	65	61	61	61	65	61	53	53	53	61
2	84	84	84	90	89	89	89	90	90	90	...	66	61	61	61	66	61	46	46	46	61
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	83	83	83	89	88	88	88	89	89	89	...	66	63	63	63	66	61	49	49	49	61
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
18273	32	32	32	31	31	31	31	31	31	31	...	43	42	42	42	43	45	46	46	46	45
18274	33	33	33	33	32	32	32	33	33	33	...	44	43	43	43	44	46	47	47	47	46
18275	43	43	43	43	43	43	43	43	44	44	...	47	49	49	49	47	47	49	49	49	47
18276	43	43	43	45	44	44	44	45	46	46	...	48	48	48	48	48	48	49	49	49	48
18277	42	42	42	44	43	43	43	44	46	46	...	48	49	49	49	48	48	50	50	50	48

18278 rows × 26 columns

Filling missing values

Let's fill "dribbling","defending", "physic", "passing", "shooting" and "pace" missing values of these columns by median

In [22]:
```
columns=["dribbling","defending", "physic", "passing", "shooting", "pace"]
df_20[columns]
```

Out[22]:

	dribbling	defending	physic	passing	shooting	pace
0	96.0	39.0	66.0	92.0	92.0	87.0
1	89.0	35.0	78.0	82.0	93.0	90.0
2	95.0	32.0	58.0	87.0	85.0	91.0
3	NaN	NaN	NaN	NaN	NaN	NaN
4	94.0	35.0	66.0	86.0	83.0	91.0
...	...	...	...	...	...	...
18273	33.0	47.0	51.0	28.0	23.0	57.0
18274	35.0	48.0	48.0	33.0	24.0	58.0
18275	45.0	48.0	51.0	44.0	35.0	54.0
18276	47.0	45.0	52.0	47.0	35.0	59.0
18277	45.0	47.0	55.0	51.0	32.0	60.0

18278 rows × 6 columns

In [23]:
```
df_20[columns].isnull().sum()
```

Out[23]:
```
dribbling    2036
defending    2036
physic       2036
passing      2036
shooting     2036
pace         2036
dtype: int64
```

In [24]:
```
for col in columns:
     df_20[col] = df_20[col].fillna(df_20[col].median())
df_20[columns]
```

Out[24]:

	dribbling	defending	physic	passing	shooting	pace
0	96.0	39.0	66.0	92.0	92.0	87.0
1	89.0	35.0	78.0	82.0	93.0	90.0
2	95.0	32.0	58.0	87.0	85.0	91.0
3	64.0	56.0	66.0	58.0	54.0	69.0
4	94.0	35.0	66.0	86.0	83.0	91.0
...	...	...	...	...	...	...
18273	33.0	47.0	51.0	28.0	23.0	57.0
18274	35.0	48.0	48.0	33.0	24.0	58.0
18275	45.0	48.0	51.0	44.0	35.0	54.0
18276	47.0	45.0	52.0	47.0	35.0	59.0
18277	45.0	47.0	55.0	51.0	32.0	60.0

18278 rows × 6 columns

Fill the remaning NaN values with 0

In [25]:
```
df_20=df_20.fillna(0)
```

Count the NaN's

In [26]:
```
df_20.isnull().sum()
```

Out[26]:
```
short_name     0
age            0
height_cm      0
weight_kg      0
nationality    0
              ..
PositionRB     0
PositionRM     0
PositionRW     0
PositionRWB    0
PositionST     0
Length: 125, dtype: int64
```

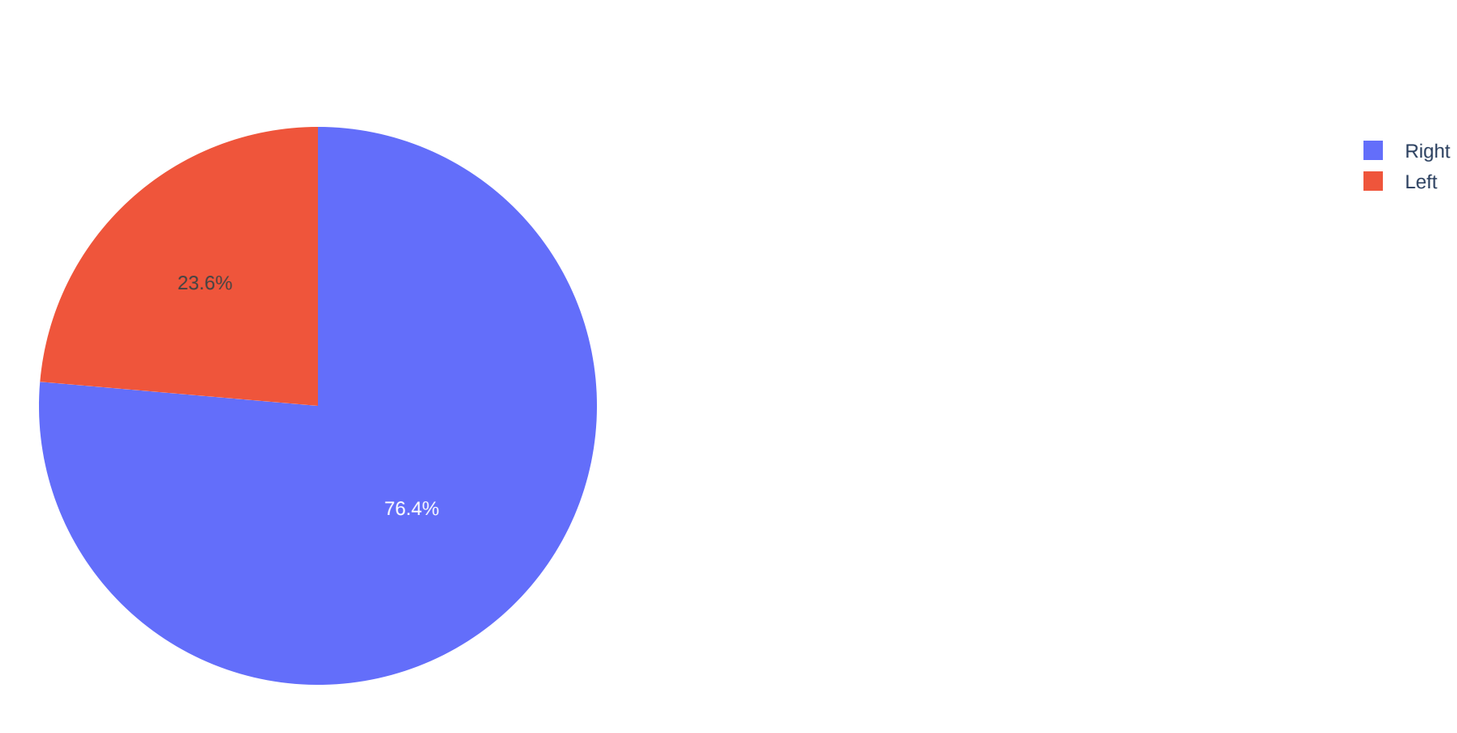
Exploratory Data Analysis

1- Scatter Plot(coloured by age) year 2020 - Overall Rating vs Value in Euros

In [27]:
```
fig=go.Figure(
data=go.Scatter(
x=df_20['overall'],
y=df_20['value_eur'],
    mode='markers',
    marker=dict(
    size=10,
    color=df_20['age'],
    showscale=True
    ),
    text=df_20['short_name']
)
)
fig.update_layout(title="Scatter Plot(coloured by age) year 2020 - Overall Rating vs Value in Euros",
                xaxis_title="Overall Rating",
                yaxis_title="Value in Euros")
fig.show()
```

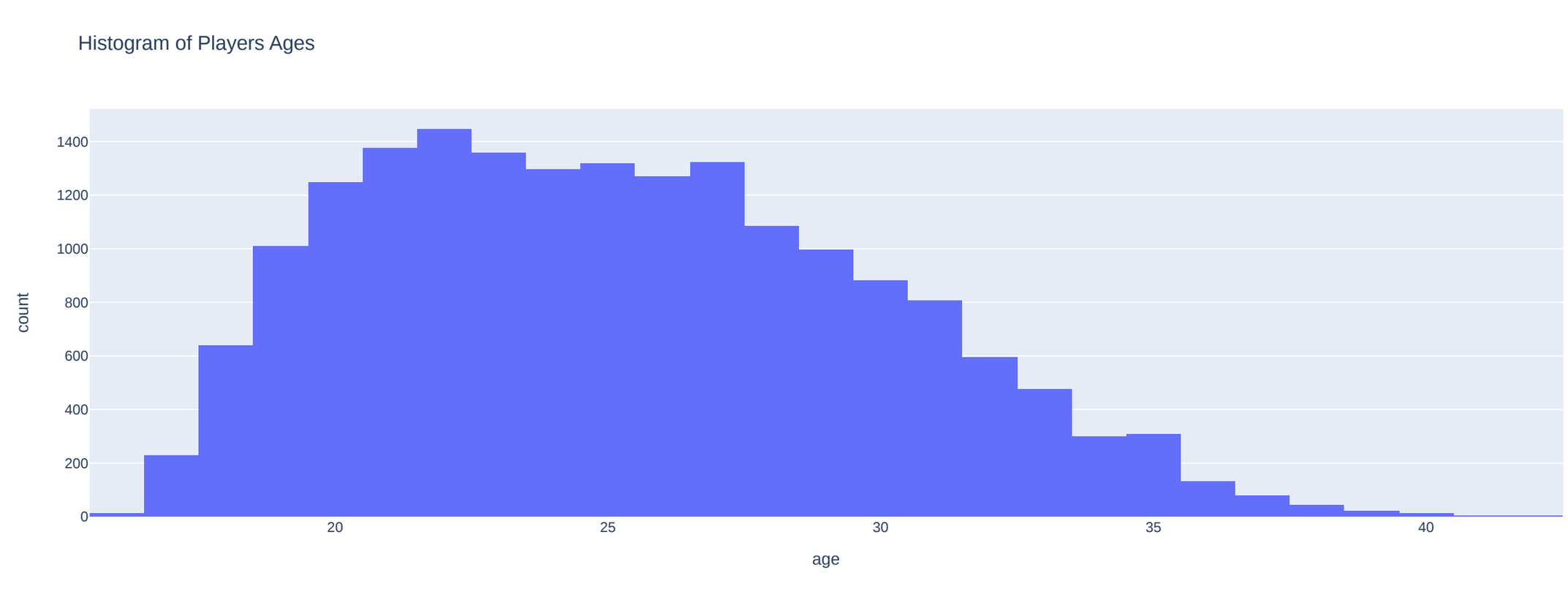
2- Pie chart proportion of right-foot players vs left-foot players

In [28]:
```
fig=px.pie(df_20,names="preferred_foot",title="Percentage of players preferred foot")
fig.show()
```



3- Histogram of Players Ages

In [29]:
```
fig=px.histogram(df_20,x="age",title=" Histogram of Players Ages")
fig.show()
```



4- Scatter-plot to compare a player's growth overtime

First loading datasets of players from 2016 to 2019

In [30]:
```
df_16=pd.read_csv('players_16.csv')
df_17=pd.read_csv('players_17.csv')
df_18=pd.read_csv('players_18.csv')
df_19=pd.read_csv('players_19.csv')
print("datsets imported!")
```

```
datsets imported!
```

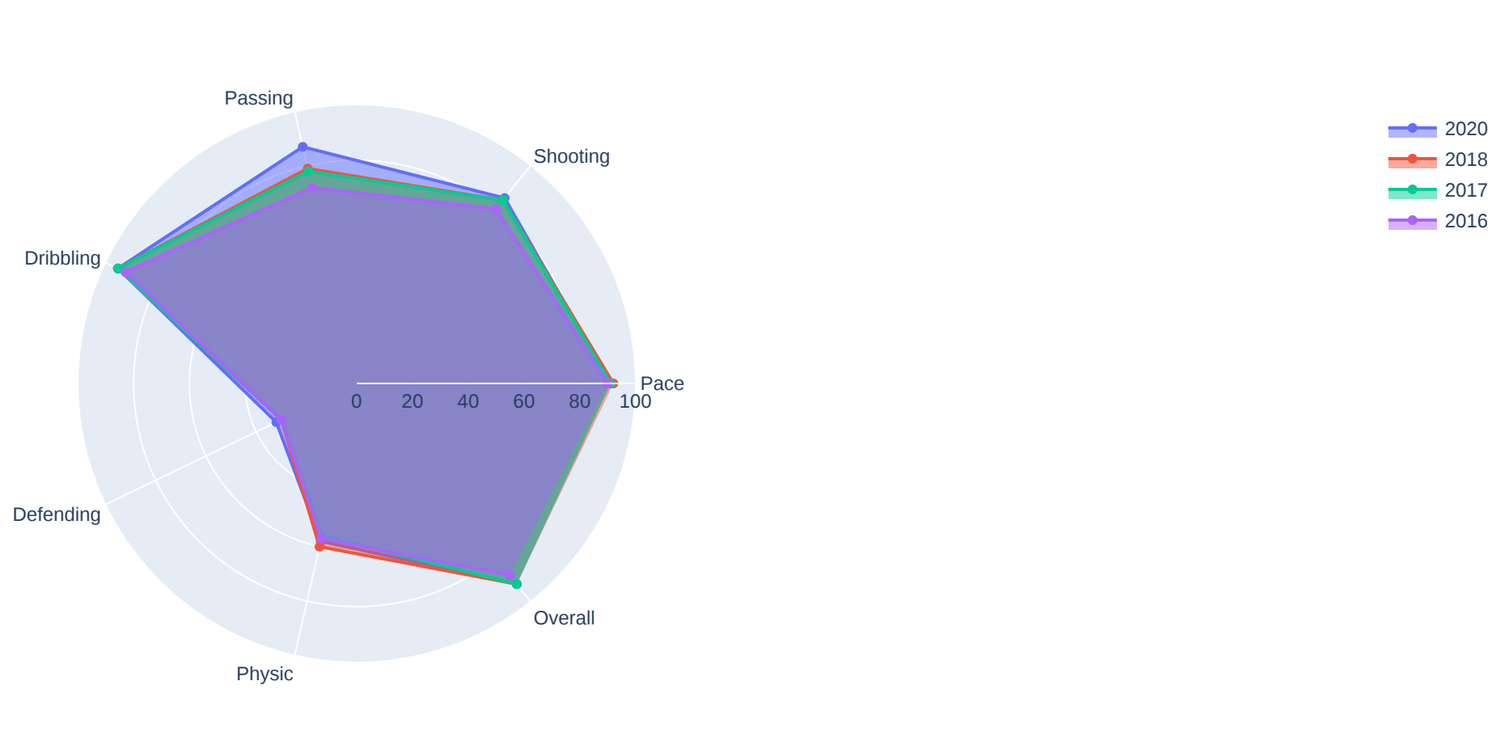
Player attributes column names

In [31]:
```
attributes=['Pace','Shooting','Passing','Dribbling','Defending','Physic','Overall']
```

Now creating a method for comparing players growth overtime

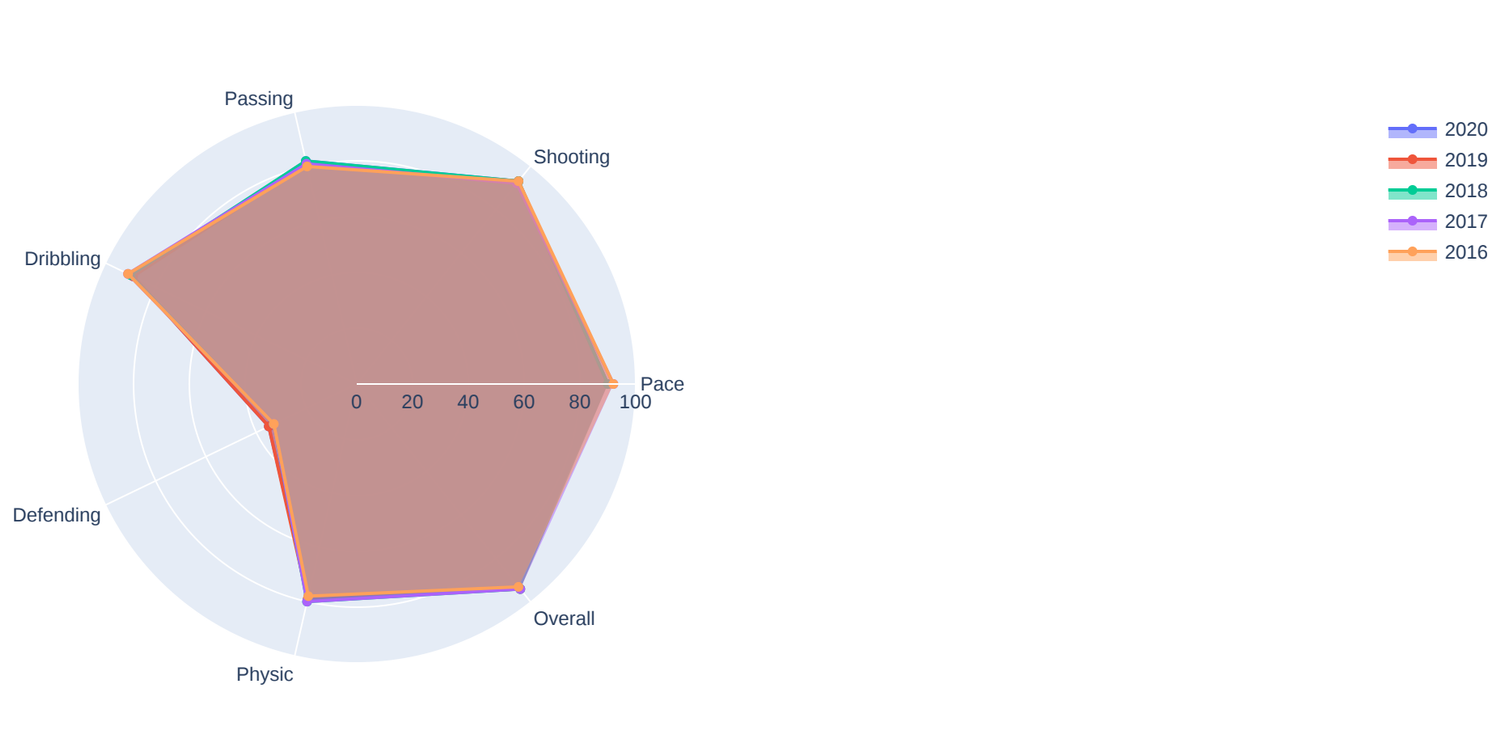
In [32]:
```
def playergrowth(name):
    data20 = df_20[df_20.short_name.str.startswith(name)]
    data19 = df_19[df_19.short_name.str.startswith(name)]
    data18 = df_18[df_18.short_name.str.startswith(name)]
    data17 = df_17[df_17.short_name.str.startswith(name)]
    data16 = df_16[df_16.short_name.str.startswith(name)]
    data = []

    trace0 = go.Scatterpolar(
        r = [data20['pace'].values[0], data20['shooting'].values[0]
            , data20['passing'].values[0], data20['dribbling'].values[0]
            , data20['defending'].values[0], data20['physic'].values[0]
            , data20['overall'].values[0]
            ]
        , theta = attributes
        , fill = 'toself'
        , name = '2020'
    )
    data.append(trace0)

    if name in df_19['short_name'].values:
        trace1 = go.Scatterpolar(
            r = [data19['pace'].values[0], data19['shooting'].values[0]
                , data19['passing'].values[0], data19['dribbling'].values[0]
                , data19['defending'].values[0], data19['physic'].values[0]
                , data19['overall'].values[0]
                ]
            , theta = attributes
            , fill = 'toself'
            , name = '2019'
        )
        data.append(trace1)

    if name in df_18['short_name'].values:
        trace2 = go.Scatterpolar(
            r = [data18['pace'].values[0], data18['shooting'].values[0]
                , data18['passing'].values[0], data18['dribbling'].values[0]
                , data18['defending'].values[0], data18['physic'].values[0]
                , data18['overall'].values[0]
                ]
            , theta = attributes
            , fill = 'toself'
            , name = '2018'
        )
        data.append(trace2)

    if name in df_17['short_name'].values:
        trace3 = go.Scatterpolar(
            r = [data17['pace'].values[0], data17['shooting'].values[0]
                , data17['passing'].values[0], data17['dribbling'].values[0]
                , data17['defending'].values[0], data17['physic'].values[0]
                , data17['overall'].values[0]
                ]
            , theta = attributes
            , fill = 'toself'
            , name = '2017'
        )
        data.append(trace3)

    if name in df_16['short_name'].values:
        trace4 = go.Scatterpolar(
            r = [data16['pace'].values[0], data16['shooting'].values[0]
                , data16['passing'].values[0], data16['dribbling'].values[0]
                 , data16['defending'].values[0], data16['physic'].values[0]
                , data16['overall'].values[0]
                ]
            , theta = attributes
            , fill = 'toself'
            , name = '2016'
        )
        data.append(trace4)


    layout = go.Layout(
        polar = dict(
            radialaxis = dict(
                visible = True
                , range = [0,100]
            )
        )
        , showlegend = True
        , title = 'Stat related to {} from 2016 to 2020'.format(name)
    )

    fig = go.Figure(data = data, layout = layout)
    fig.show()
```

In [33]:
```
playergrowth('Neymar')
```

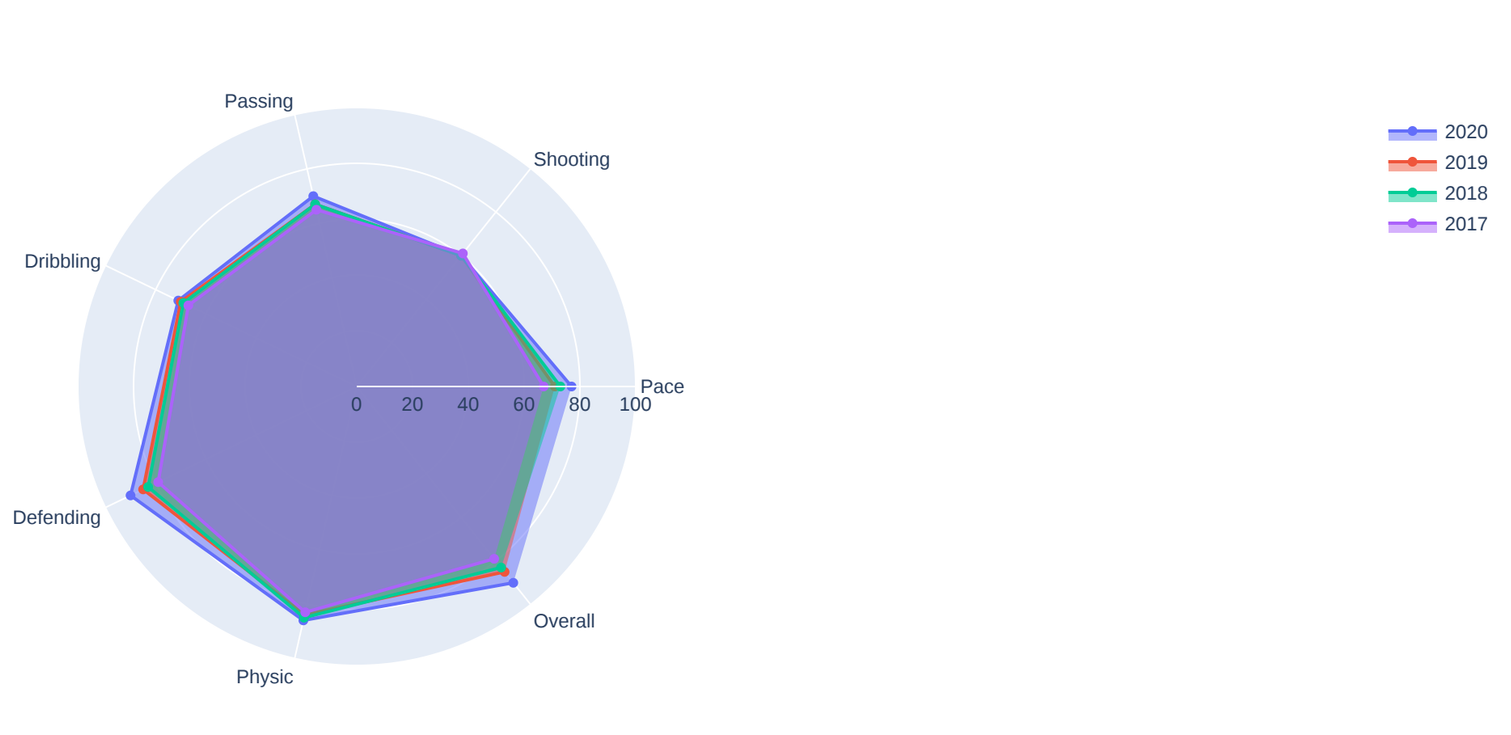


Now for Cristiano Ronaldo

In [34]:
```
playergrowth('Cristiano')
```

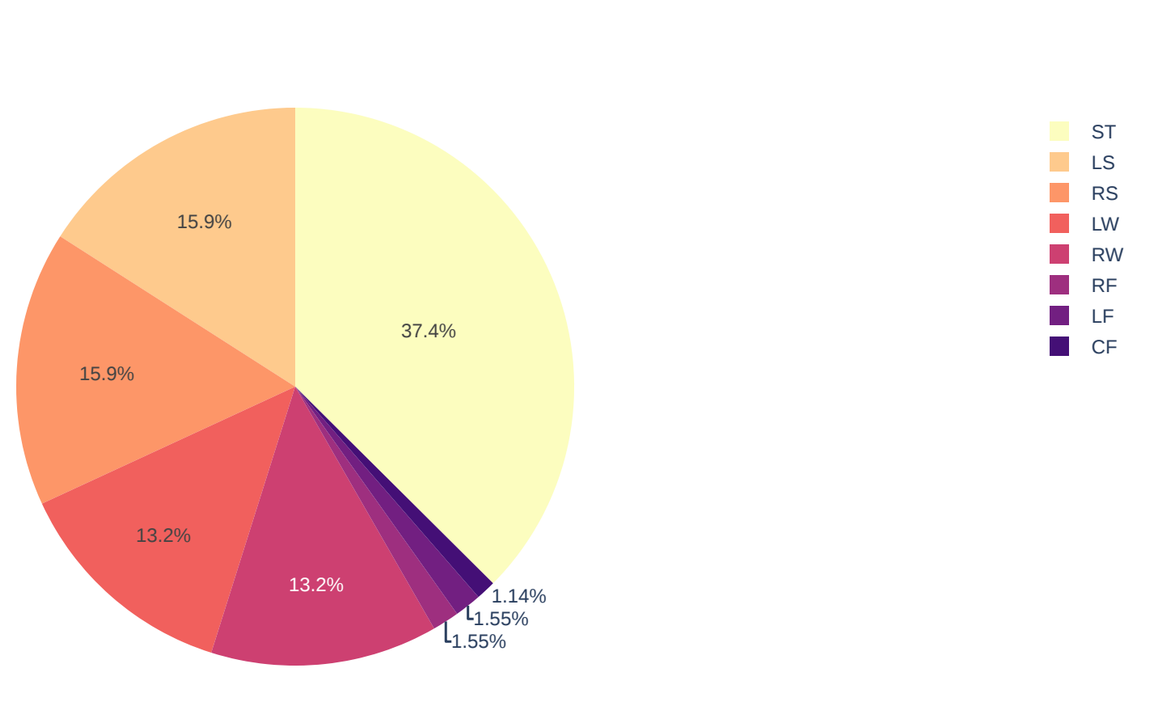


In [35]:
```
playergrowth('V. van Dijk')
```



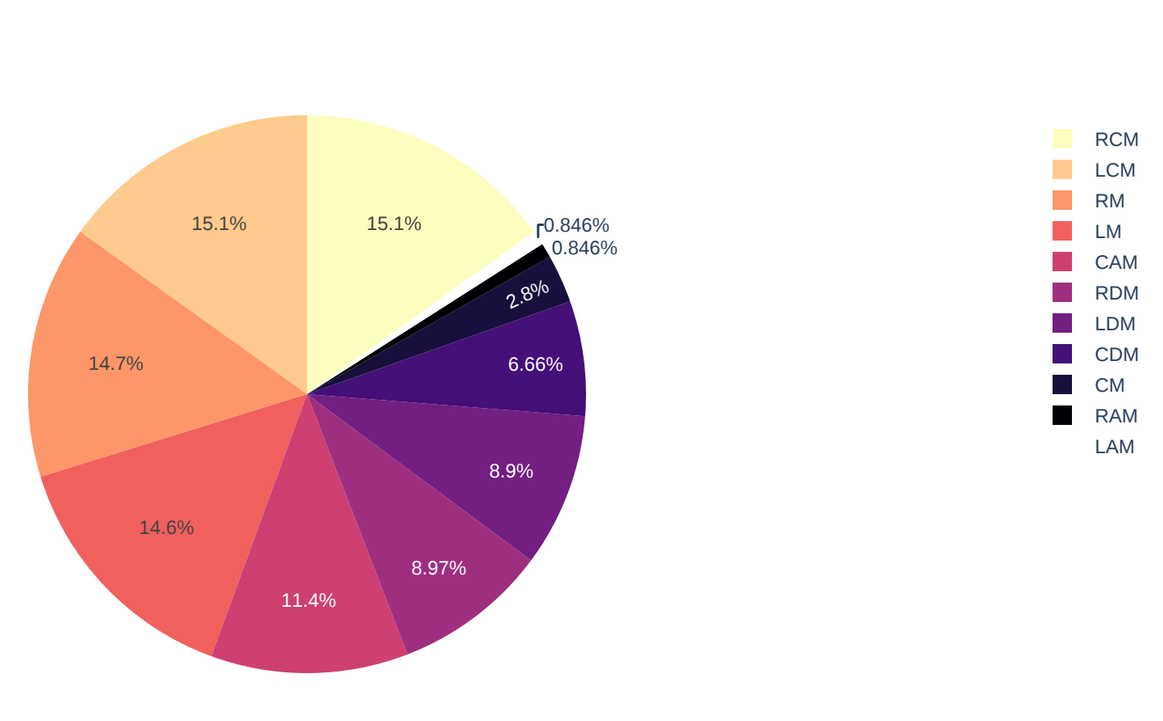
6- Pie-chart Describing the Percentage of Players in different Attacker positions

In [36]:
```
attack=['RW','LW','ST','CF','LS','RS','RF','LF']
Sample=df_20.query('team_position in @attack')
fig=px.pie(Sample,names='team_position',color_discrete_sequence=px.colors.sequential.Magma_r,
          title='Percentage of players in Attacker positions')
fig.show()
```



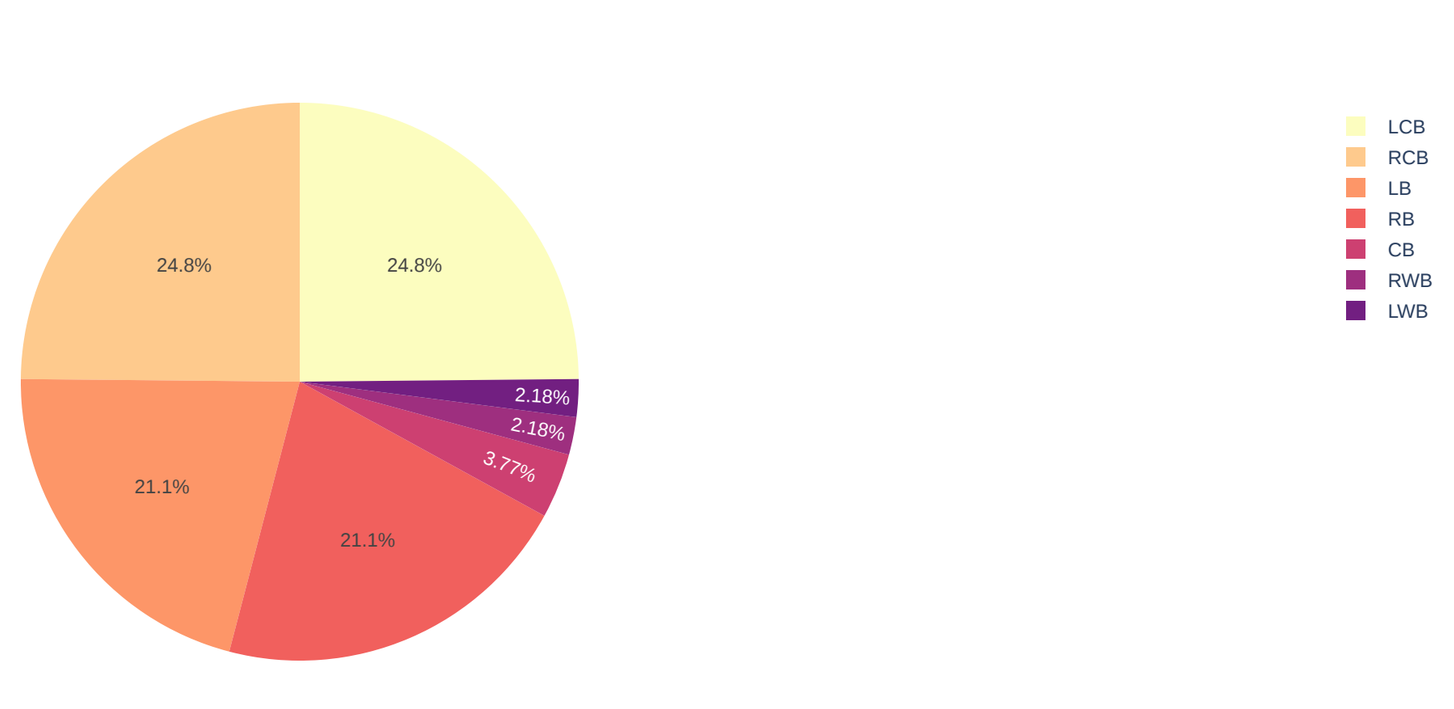
7- Pie-chart Describing the Percentage of Players in different Midfielder positions

In [37]:
```
mid=['CAM','RCM','CDM','LDM','RM','LCM','LM','RDM', 'RAM', 'CM','LAM']
Sample=df_20.query('team_position in @mid')
fig=px.pie(Sample,names='team_position',color_discrete_sequence=px.colors.sequential.Magma_r,
          title='Percentage of players in Midfielder positions')
fig.show()
```



8- Pie-chart Describing the Percentage of Players in different Defender positions

In [38]:
```
defence  = ['LCB','RCB','LB','RB','CB','RWB','LWB']

sample = df_20.query('team_position in  @defence')

fig = px.pie(sample, names='team_position', color_discrete_sequence= px.colors.sequential.Magma_r,
            title = 'Percentage  of  players in Defender position')
fig.show()
```



Pick Top 5 Players per Position

Creating a method to pick top 5 players based on player position and the player value in euro

In [39]:
```
def pick_top_players(pos , value):
    column  = str('Position')+str.upper(pos)
    target_players  =  df_20[(df_20[column]==1) & (df_20['value_eur'] <=value)][['short_name', 'age', 'overall', 'value_eur']].head(5)
    return target_players
```

In [40]:
```
pick_top_players('lb',34000000)
```

Out[40]:

	short_name	age	overall	value_eur
101	Marcelo	31	85	28000000
103	Alex Sandro	28	85	33000000
124	Alex Telles	26	84	33000000
174	Grimaldo	23	83	29500000
179	L. Digne	25	83	28500000

In [41]:
```
pick_top_players('st', 55000000)
```

Out[41]:

	short_name	age	overall	value_eur
19	L. Suárez	32	89	53000000
34	E. Cavani	32	88	47000000
71	C. Immobile	29	86	44500000
72	A. Lacazette	28	86	46000000
89	R. Lukaku	26	85	46000000

In [42]:
```
pick_top_players('cam', 100000000)
```

Out[42]:

	short_name	age	overall	value_eur
5	K. De Bruyne	28	91	90000000
23	P. Dybala	25	88	76500000
27	C. Eriksen	27	88	68000000
33	David Silva	33	88	36000000
37	M. Reus	30	88	56000000