Spider plot

In [1]:
```
import pandas as pd
from math import pi
import matplotlib.pyplot as plt

vars = {'var1': 6, 'var2': 5, 'var3': 4}
```

In [2]:
```
data = pd.DataFrame([vars], index=['vars'])
data
```

Out[2]:

	var1	var2	var3
vars	6	5	4

In [3]:
```
Attributes = list(data)
Attributes
```

Out[3]:
```
['var1', 'var2', 'var3']
```

In [4]:
```
values = [data[i][0] for i in Attributes]
values
```

Out[4]:
```
[6, 5, 4]
```

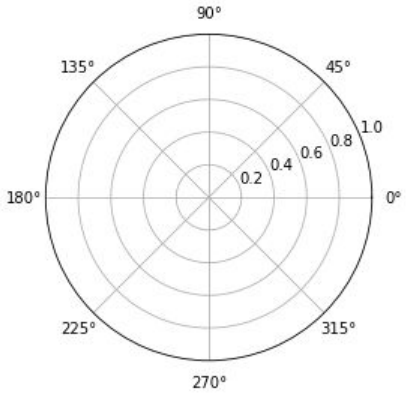
In [5]:
```
values += values[:1]
values
```

Out[5]:
```
[6, 5, 4, 6]
```

In [6]:
```
radians = [n / int(len(Attributes)) * 2 * pi for n in range(len(Attributes))]
radians += radians[:1]
radians
```

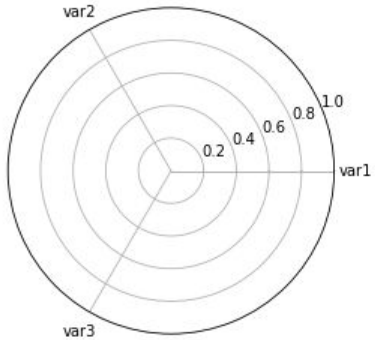
Out[6]:
```
[0.0, 2.0943951023931953, 4.1887902047863905, 0.0]
```

In [7]:
```
ax = plt.subplot(111, polar=True)
```



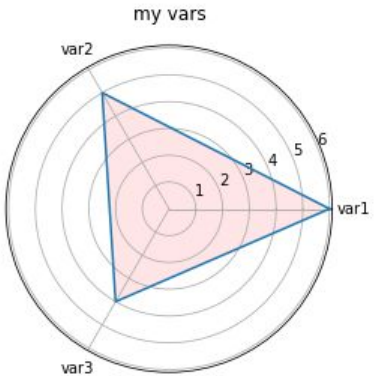
In [12]:
```
ax = plt.subplot(111, polar=True)

plt.xticks(radians, Attributes)
```

Out[12]:
```
([<matplotlib.projections.polar.ThetaTick at 0x7f080a257ef0>,
  <matplotlib.projections.polar.ThetaTick at 0x7f080a1fd240>,
  <matplotlib.projections.polar.ThetaTick at 0x7f080a29b198>,
  <matplotlib.projections.polar.ThetaTick at 0x7f080a29b630>],
 <a list of 3 Text xticklabel objects>)
```



In [13]:
```
ax = plt.subplot(111, polar=True)
plt.xticks(radians, Attributes)
ax.plot(radians, values)
ax.fill(radians, values, 'red', alpha=0.1)
ax.set_title('my vars')
```

Out[13]: Text(0.5, 1.05, 'my vars')



In [14]:
```
Messi = {'Pace': 89, 'Shooting': 90, 'Passing': 86, 'Dribbling': 95, 'Defending': 26, 'Physical': 61}
Ronaldo = {'Pace': 90, 'Shooting': 93, 'Passing': 82, 'Dribbling': 90, 'Defending': 33, 'Physical': 80}
```

In [15]:
```
data = pd.DataFrame([Messi, Ronaldo], index=['Messi', 'Ronaldo'])
data
```

Out[15]:

	Pace	Shooting	Passing	Dribbling	Defending	Physical
Messi	89	90	86	95	26	61
Ronaldo	90	93	82	90	33	80

In [16]:
```
Attributes = list(data)
```

In [17]:
```
values_R = [data[i][1] for i in Attributes]
values_M = [data[i][0] for i in Attributes]
```

In [18]:
```
values_R += values_R[:1]
values_M += values_M[:1]
```

In [20]:
```
radians = [n / int(len(Attributes)) * 2 * pi for n in range(len(Attributes))]
radians += radians[:1]
```

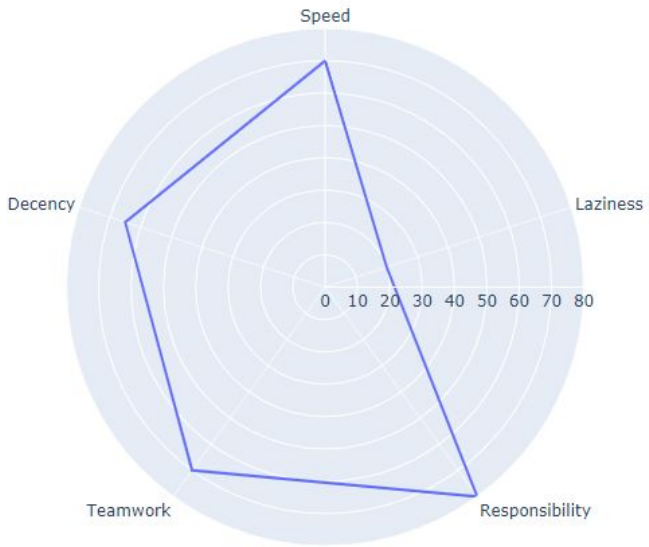
In [22]:
```
ax = plt.subplot(111, polar=True)

plt.xticks(radians, Attributes)

ax.plot(radians, values_R)
ax.plot(radians, values_M)

ax.fill(radians, values_R, 'red', alpha=0.1)
ax.fill(radians, values_M, 'blue', alpha=0.1)

plt.figtext(0.2, 0.9, 'Messi', color='blue')
plt.figtext(0.2, 0.85, 'vs')
plt.figtext(0.2, 0.8, 'Ronaldo', color='red')
```

Out[22]:
```
Text(0.2, 0.8, 'Ronaldo')
```

In [26]:
```
import plotly.express as px
import pandas as pd

data = pd.DataFrame(dict(
    skills = ['Speed', 'Laziness', 'Responsibility', 'Teamwork', 'Decency'],
    values = [70, 20, 80, 70, 65]))
```

In [32]:
```
fig = px.line_polar(data, r='values', theta='skills', line_close=True)
```

In [33]:
```
fig.show()
```



plotly.express.line_polar

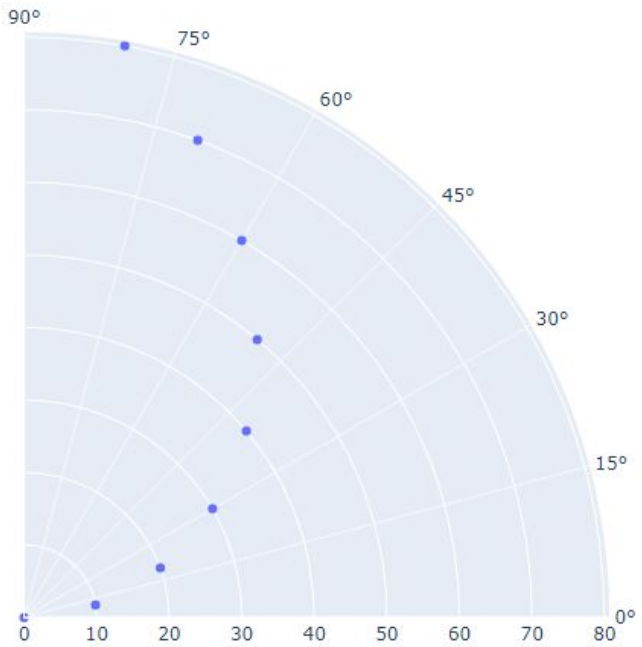
plotly.express. **line_polar** (*data_frame=None, r=None, theta=None, color=None, line_dash=None, hover_name=None, hover_data=None, custom_data=None, line_group=None, text=None, animation_frame=None, animation_group=None, category_orders={}, labels={}, color_discrete_sequence=None, color_discrete_map={}, line_dash_sequence=None, line_dash_map={}, direction='clockwise', start_angle=90, line_close=False, line_shape=None, render_mode='auto', range_r=None, range_theta=None, log_r=False, title=None, template=None, width=None, height=None*)

> In a polar line plot, each row of `data_frame` is represented as vertex of a polyline mark in polar coordinates.

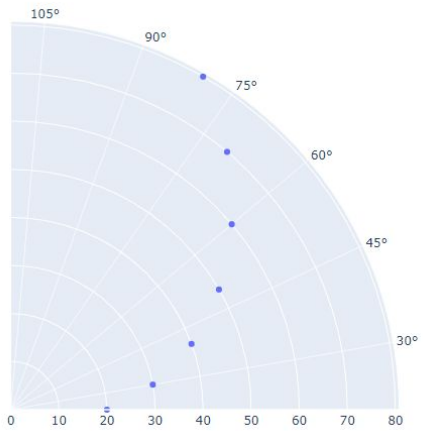
Parameters:

- **data_frame** (*DataFrame or array-like or dict*) – This argument needs to be passed for column names (and not keyword names) to be used. Array-like and dict are tranformed internally to a pandas DataFrame. Optional: if missing, a DataFrame gets constructed under the hood using the other arguments.
- **r** (*str or int or Series or array-like*) – Either a name of a column in `data_frame`, or a pandas Series or array_like object. Values from this column or array_like are used to position marks along the radial axis in polar coordinates.
- **theta** (*str or int or Series or array-like*) – Either a name of a column in `data_frame`, or a pandas Series or array_like object. Values from this column or array_like are used to position marks along the angular axis in polar coordinates.
- **color** (*str or int or Series or array-like*) – Either a name of a column in `data_frame`, or a pandas Series or array_like object. Values from this column or array_like are used to assign color to marks.
- **line_dash** (*str or int or Series or array-like*) – Either a name of a column in `data_frame`, or a pandas Series or array_like object. Values from this column or array_like are used to assign dash-patterns to lines.

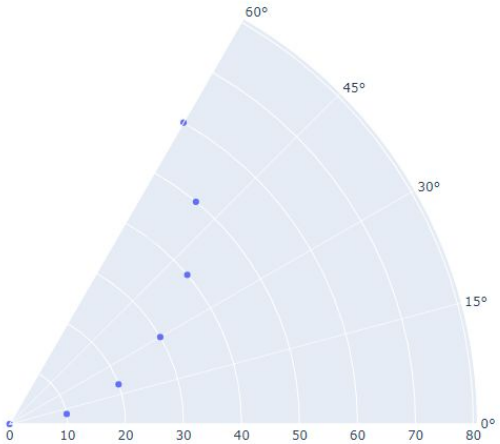
In [2]:
```
import plotly.express as px
fig = px.scatter_polar(r=range(0, 90, 10), theta=range(0, 90, 10),
                       range_theta=[0, 90], start_angle=0, direction="counterclockwise")
fig.show()
```



In [4]:
```
import plotly.express as px
fig = px.scatter_polar(r=range(0, 90, 10), theta=range(0, 90, 10),
                       range_theta=[0, 90], start_angle=-20, direction="counterclockwise")
fig.show()
```



In [5]:
```
import plotly.express as px
fig = px.scatter_polar(r=range(0, 90, 10), theta=range(0, 90, 10),
                       range_theta=[0, 60], start_angle=0, direction="counterclockwise")
fig.show()
```

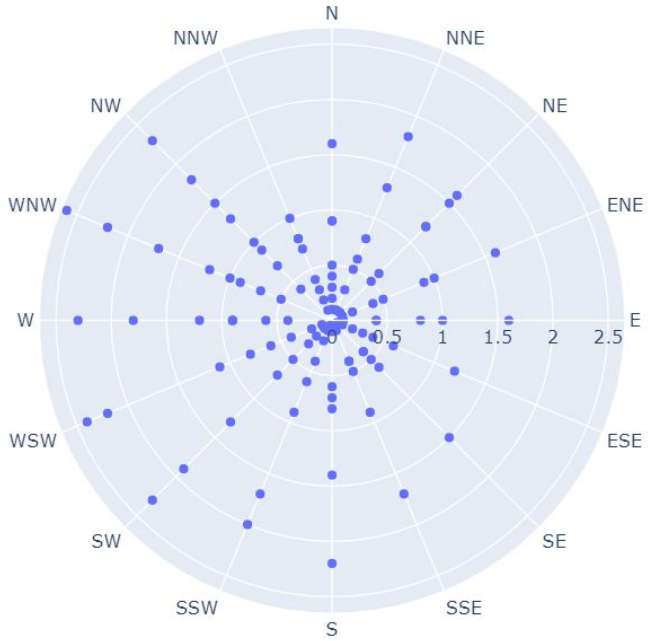


In [2]:
```
import plotly.express as px
wind = px.data.wind()
wind
```

Out[2]:

	direction	strength	frequency
0	N	0-1	0.50
1	NNE	0-1	0.60
2	NE	0-1	0.50
3	ENE	0-1	0.40
4	E	0-1	0.40
5	ESE	0-1	0.30
6	SE	0-1	0.40
7	SSE	0-1	0.40
8	S	0-1	0.60
9	SSW	0-1	0.40
10	SW	0-1	0.50
11	WSW	0-1	0.60

In [3]:
```
fig = px.scatter_polar(wind, r="frequency", theta="direction")
fig.show()
```



In [6]:
```
fig = px.line_polar(wind, r="frequency", theta="direction", color="strength", line_close=True)
fig.show()
```