Import modules

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import math
from matplotlib.ticker import FuncFormatter
import plotly
import plotly.figure_factory as ff
from pandas.plotting import parallel_coordinates
import numpy as np

%matplotlib inline
```

Data load and transformation

In [2]:
```
education = pd.read_csv('education.csv')
crime = pd.read_csv('crimeratesbystate-formatted.csv')
birthrate = pd.read_csv('birth-rate.csv')

# remove whitespaces from crime dataset (sine we have already encounte
education = education.applymap(lambda x: x.strip() if type(x) is str e
crime = crime.applymap(lambda x: x.strip() if type(x) is str else x)
birthrate = birthrate.applymap(lambda x: x.strip() if type(x) is str e
```

Histogram

In [3]:
```
# Distribution of birth rate
birthrate_hist = pd.melt(birthrate, id_vars="Country", var_name="Year"
birthrate_hist["BirthRate_int"] = birthrate_hist["BirthRate"].apply(la
birthrate_hist.head()
```

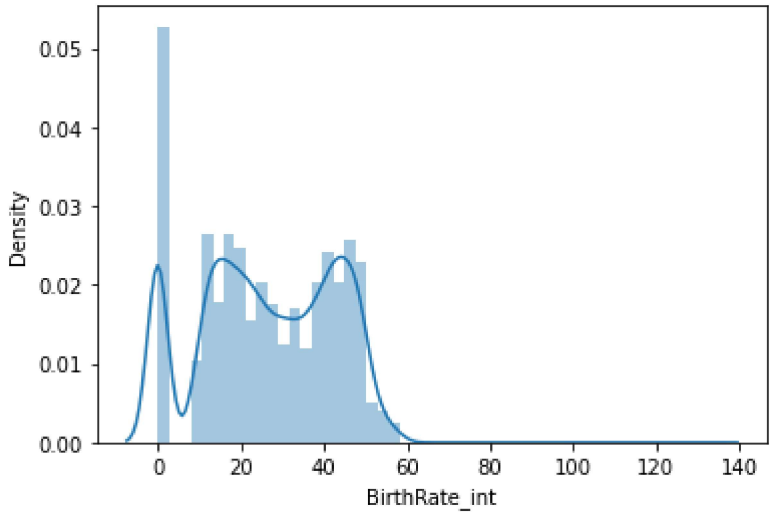
Out[3]:

	Country	Year	BirthRate	BirthRate_int
0	Aruba	1960	36.400	37
1	Afghanistan	1960	52.201	53
2	Angola	1960	54.432	55
3	Albania	1960	40.886	41
4	Netherlands Antilles	1960	32.321	33

In [4]:
```
1 sns.distplot( birthrate_hist["BirthRate_int"] )
```

```
C:\Users\debas\anaconda3\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figur
e-level function with similar flexibility) or `histplot` (an axes-level f
unction for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[4]: <AxesSubplot:xlabel='BirthRate_int', ylabel='Density'>

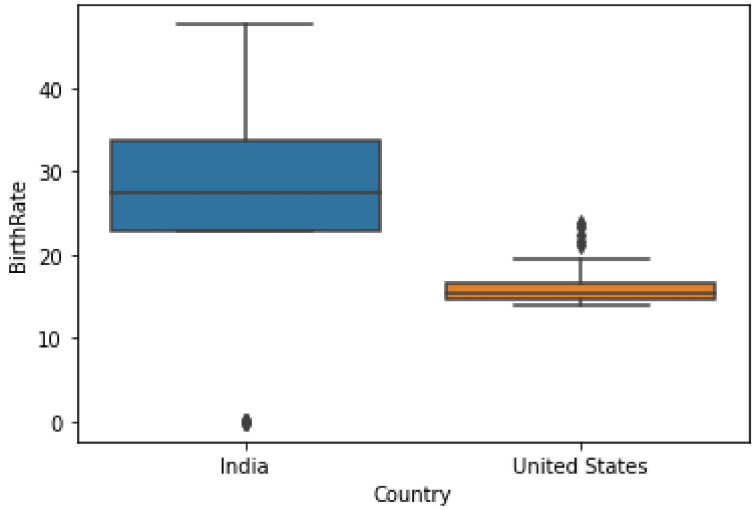


Box plot

Comparison of birthrate betwen India and USA

In [5]:
```
1 birthrate_box = birthrate_hist[(birthrate_hist["Country"]=="United Sta
2 sns.boxplot(x = birthrate_box["Country"], y=birthrate_box["BirthRate"]
```

Out[5]: <AxesSubplot:xlabel='Country', ylabel='BirthRate'>



Bullet chart

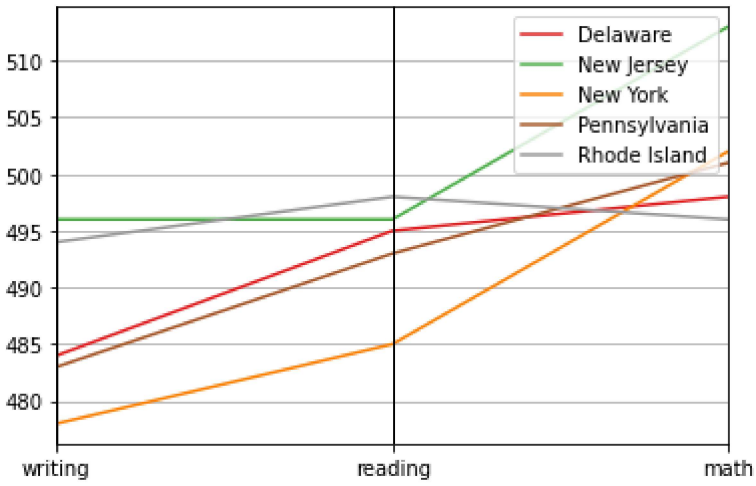
In [7]:

```
## US burglary statistics against some dummy benchmark
# transform data
crime_bullet = crime[crime["state"]=="United States"][["state","burgla
crime_bullet['target'] = 500
crime_bullet_tuple = [tuple(x) for x in crime_bullet.values][0]


# set parameter for bullet chart
limits = [300, 500, 1000]
palette = sns.color_palette("Blues_r", len(limits))
fig, ax = plt.subplots()
ax.set_aspect('equal')
ax.set_yticks([1])
ax.set_yticklabels='United States'

prev_limit = 0
for idx, lim in enumerate(limits):
    ax.barh([1], lim-prev_limit, left=prev_limit, height=75, color=pal
    prev_limit = lim

# draw the value we're measuring
ax.barh([1], crime_bullet_tuple[1], color='black', height=45)

ax.axvline(crime_bullet_tuple[2], color="gray", ymin=0.10, ymax=0.9)
```

Out[7]: <matplotlib.lines.Line2D at 0x1b9e031af70>

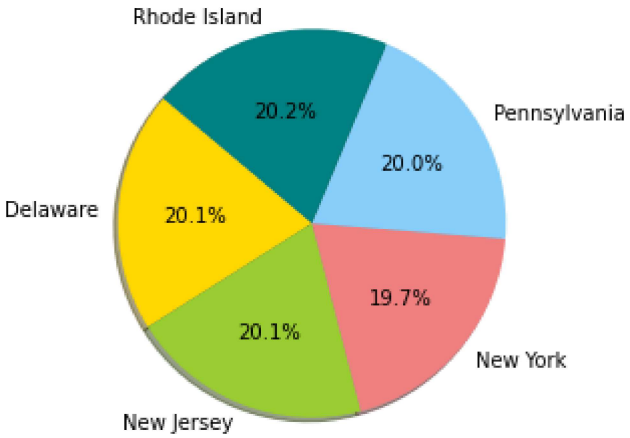
Parallel Coordinate plot

In [8]:
```
##Comparison of reading, writing and math numbers between 5 states
# transform data
education_parallel = education[education['state'].isin(['New York','Ne

# make the plot
parallel_coordinates(education_parallel, 'state', colormap=plt.get_cma
plt.show()
```



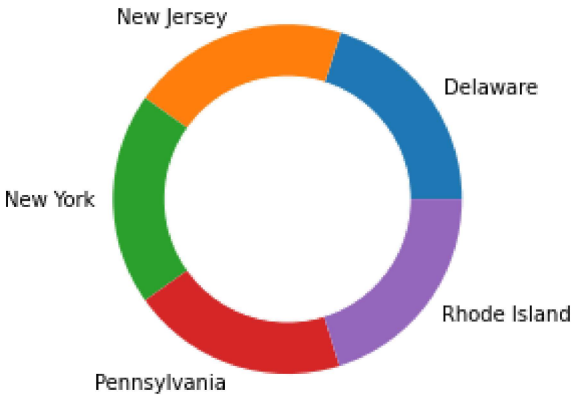
Pie chart

In [9]:
```
##Comparison of reading numbers between 5 states
# transform data
education_pie = education_parallel[['state','reading']]

# set colors
colors = ['gold', 'yellowgreen', 'lightcoral', 'lightskyblue','teal']

# plot
plt.pie(education_pie['reading'], labels=education_pie['state'], color
autopct='%1.1f%%', shadow=True, startangle=140)

plt.axis('equal')
plt.show()
```



Donought chart

In [11]:
```
#Comparison of reading, writing and math numbers between 5 states
# transform data
education_donut = education_pie

# create a pieplot
plt.pie(education_donut['reading'], labels=education_donut['state'])

# add a circle at the center
my_circle=plt.Circle( (0,0), 0.7, color='white')
p=plt.gcf()
p.gca().add_artist(my_circle)

plt.show()
```



In [ ]:
```
1
```