Bellevue University

DSC-540 Data Preparation

Name: Tai Ngo

Date: 05/29/2020

Final Project

Milestone 2 - Prepare data from the adult.csv file

In [1507]:
```
# Load required libraries
import numpy as np
import pandas as pd
```

In [1508]:
```
# Load the file into a dataframe
df = pd.read_csv('adult.csv')
```

In [1509]:
```
# quick look into the dataset
df.head()
```

Out[1509]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	sex	capital.gain	capital.loss	hours.per.week	native.c
0	90	?	77053	HS-grad	9	Widowed	?	Not-in-family	White	Female	0	4356	40	United
1	82	Private	132870	HS-grad	9	Widowed	Exec-managerial	Not-in-family	White	Female	0	4356	18	United
2	66	?	186061	Some-college	10	Widowed	?	Unmarried	Black	Female	0	4356	40	United
3	54	Private	140359	7th-8th	4	Divorced	Machine-op-inspct	Unmarried	White	Female	0	3900	40	United
4	41	Private	264663	Some-college	10	Separated	Prof-specialty	Own-child	White	Female	0	3900	40	United

In [1510]:
```
df.shape
```

Out[1510]:
```
(32561, 15)
```

Remove unncessary columns

In [1511]:
```
# drop columns: final weight, workclass, capital.gain, capital.loss, marital status and relationship
df = df.drop(columns=['fnlwgt', 'workclass', 'capital.gain', 'capital.loss', 'marital.status','relationship','native.country'], a
df.head()
```

Out[1511]:

	age	education	education.num	occupation	race	sex	hours.per.week	income
0	90	HS-grad	9	?	White	Female	40	<=50K
1	82	HS-grad	9	Exec-managerial	White	Female	18	<=50K
2	66	Some-college	10	?	Black	Female	40	<=50K
3	54	7th-8th	4	Machine-op-inspct	White	Female	40	<=50K
4	41	Some-college	10	Prof-specialty	White	Female	40	<=50K

In [1512]:
```
df.shape
```

Out[1512]:
```
(32561, 8)
```

Change the names of columns

In [1513]:
```
df.columns
```

Out[1513]:
```
Index(['age', 'education', 'education.num', 'occupation', 'race', 'sex',
       'hours.per.week', 'income'],
      dtype='object')
```

In [1514]:
```
# Edit the name of columns of interests
df = df.rename(columns={
    'age':'Age',
    'sex': 'Gender',
    'race': 'Race',
    'hours.per.week':'Work_hrs/Week',
    'relationship': 'Relationship',
    'education': 'Education',
    'education.num': 'Years_in_School',
    'occupation':'Occupation',
    'income':'Annual_Income_above50K'})
```

Remove rows with missing data

In [1515]:
```
# Drop rows that have values with a question mark
df = df[(df != '?').all(axis=1)]

# check the number of missing values
df.isnull().sum()
```

```
C:\Users\Tai\anaconda3\lib\site-packages\pandas\core\ops\array_ops.py:253: FutureWarning:

elementwise comparison failed; returning scalar instead, but in the future will perform elementwise comparison
```

Out[1515]:
```
Age                       0
Education                 0
Years_in_School           0
Occupation                0
Race                      0
Gender                    0
Work_hrs/Week             0
Annual_Income_above50K    0
dtype: int64
```

Format the values for annual income greater than or less than 50K a year

In [1516]:
```
df['Annual_Income_above50K'] = df['Annual_Income_above50K'].replace({'<=50K': 'No', '>50K': 'Yes'})
```

In [1517]:
```
# The final dataset after cleaning
df1 = df
df1.head()
```

Out[1517]:

	Age	Education	Years_in_School	Occupation	Race	Gender	Work_hrs/Week	Annual_Income_above50K
1	82	HS-grad	9	Exec-managerial	White	Female	18	No
3	54	7th-8th	4	Machine-op-inspct	White	Female	40	No
4	41	Some-college	10	Prof-specialty	White	Female	40	No
5	34	HS-grad	9	Other-service	White	Female	45	No
6	38	10th	6	Adm-clerical	White	Male	40	No

In [1518]:
```
df1=df1.reset_index().drop(columns=['index'])
```

Fuzzy Matching

In [1519]:
```
# load library to run fuzzy matching
from fuzzywuzzy import fuzz
from fuzzywuzzy import process
```

In [1520]:
```
# perform ratio() function to see the score
fuzz.ratio('Exec-managerial','Adm-clerical')
```

Out[1520]: 44

In [1521]:
```
# perform partial_ratio() function to see the score
fuzz.partial_ratio('Exec-managerial','Adm-clerical')
```

Out[1521]: 52

In [1522]:
```
# perform token_set_ratio() function to see the score
fuzz.token_set_ratio('Exec-managerial','Adm-clerical')
```

Out[1522]: 44

Milestone 3 - Pull the data from a website and Prepare the data

In [1523]:
```
# load necessary libraries
import requests
from bs4 import BeautifulSoup
```

In [1524]:
```
# get the response from the website
# and parse by using BeautifulSoup
url = 'https://www.worldometers.info/gdp/gdp-per-capita/'
page = requests.get(url)
soup = BeautifulSoup(page.content, 'html.parser')
```

In [1525]:
```
# examine the structure of the website
print(soup.prettify())
```

```
        Gross Domestic Product (GDP) per capita shows a country's GDP divided by its total population. The table below lists
countries in the world ranked by GDP at Purchasing Power Parity (PPP) per capita, along with the Nominal GDP per capita. PPP
takes into account the relative cost of living, rather than using only exchange rates, therefore providing a more accurate pi
cture of the real differences in income.
       </p>
       <p>
        See also:
        <a href="/gdp/gdp-by-country/">
         <strong>
          GDP by Country
         </strong>
        </a>
       </p>
       <div class="table-responsive" style="font-size:16px; text-align:left; width:100%; max-width:850px;">
        <table cellspacing="0" class="table table-striped table-bordered" id="example2" text-align:left"="">
         <thead>
          <tr>
           <th>

            #
           </th>
```

In [1526]:
```
# check the type of the website
def encoding_check(r):
    return (r.encoding)
encoding_check(page)
```

Out[1526]: 'UTF-8'

In [1527]:
```
# check the type of the content
def decode_content(r, encoding):
    return (r.content.decode(encoding))
contents = decode_content(page, encoding_check(page))
type(contents)
```

Out[1527]: str

In [1528]:
```
# length of the content
len(contents)
```

Out[1528]: 70962

In [1529]:
```
countries = soup.find('table', id='example2')
country = countries.find('td')
a = countries.text
a = str(a)

# select only relevant information
print(a[120:])
```

```
1 Qatar  $128,647  $61,264  752%   2 Macao  $115,367  $80,890  675%   3 Luxembourg  $107,641  $105,280  629%   4 Singapore  $9
4,105  $56,746  550%   5 Brunei  $79,003  $28,572  462%   6 Ireland  $76,745  $69,727  449%   7 United Arab Emirates  $74,035
$40,325  433%   8 Kuwait  $72,096  $29,616  422%   9 Switzerland  $66,307  $80,296  388%   10 San Marino  $63,549  $48,495  37
2%   11 Norway  $62,183  $75,428  364%   12 Hong Kong  $61,671  $46,733  361%   13 United States  $59,928  $59,939  350%   14 I
celand  $55,322  $73,233  324%   15 Netherlands  $54,422  $48,796  318%   16 Denmark  $54,356  $57,545  318%   17 Saudi Arabia
$53,893  $20,747  315%   18 Austria  $53,879  $47,261  315%   19 Germany  $52,556  $44,680  307%   20 Sweden  $51,405  $54,075
301%   21 Australia  $49,378  $53,831  289%   22 Belgium  $49,367  $43,325  289%   23 Bahrain  $47,708  $23,715  279%   24 Cana
da  $46,510  $44,841  272%   25 Finland  $46,344  $45,778  271%   26 United Kingdom  $44,920  $39,532  263%   27 France  $44,03
```

3 $39,827 258% 28 Japan $42,067 $38,214 246% 29 Oman $41,331 $15,170 242% 30 Italy $40,924 $32,038 239% 31 M
alta $40,797 $28,585 239% 32 New Zealand $40,748 $43,415 238% 33 Aruba $39,493 $25,630 231% 34 Spain $39,037
$28,175 228% 35 Israel $38,868 $42,852 227% 36 South Korea $38,824 $29,958 227% 37 Czech Republic (Czechia) $38,0
20 $20,291 222% 38 Slovenia $36,387 $23,488 213% 39 Cyprus $36,012 $18,695 211% 40 Estonia $33,448 $20,170 19
6% 41 Lithuania $33,253 $16,709 194% 42 Portugal $32,554 $21,316 190% 43 Slovakia $32,371 $17,551 189% 44 Trin
idad and Tobago $31,645 $15,952 185% 45 Bahamas $30,495 $31,858 178% 46 Poland $29,924 $13,871 175% 47 Malaysia
$29,511 $10,118 173% 48 Seychelles $29,328 $15,536 172% 49 Hungary $28,799 $14,364 168% 50 Saint Kitts & Nevis
$28,636 $19,061 167% 51 Greece $28,583 $19,214 167% 52 Latvia $28,362 $15,613 166% 53 Turkey $28,002 $10,498 1
64% 54 Romania $26,660 $10,781 156% 55 Kazakhstan $26,491 $9,009 155% 56 Croatia $26,296 $13,200 154% 57 Russi
a $25,763 $10,846 151% 58 Chile $24,747 $15,001 145% 59 Panama $24,521 $15,166 143% 60 Equatorial Guinea $24,43
9 $9,741 143% 61 Antigua and Barbuda $23,522 $15,825 138% 62 Uruguay $22,610 $16,341 132% 63 Mauritius $22,356
$10,491 131% 64 Bulgaria $20,948 $8,197 123% 65 Iran $20,885 $5,628 122% 66 Argentina $20,829 $14,508 122% 67
Libya $19,673 $5,791 115% 68 Montenegro $19,355 $7,720 113% 69 Belarus $18,896 $5,762 111% 70 Mexico $18,656
$9,224 109% 71 Barbados $18,559 $16,328 109% 72 Gabon $18,113 $7,271 106% 73 Turkmenistan $18,031 $6,587 105%
74 Thailand $17,910 $6,579 105% 75 Azerbaijan $17,450 $4,139 102% 76 Costa Rica $17,110 $11,573 100% 77 Botswana
$17,024 $7,894 100% 78 Iraq $16,935 $5,114 99% 79 China $16,842 $8,612 98% 80 Maldives $16,688 $9,802 98% 81
Dominican Republic $16,064 $7,223 94% 82 Brazil $15,553 $9,881 91% 83 Serbia $15,432 $4,692 90% 84 Algeria $15,
293 $4,048 89% 85 North Macedonia $15,290 $5,418 89% 86 Suriname $15,191 $5,251 89% 87 Grenada $15,156 $10,164
89% 88 Palau $14,854 $16,275 87% 89 Lebanon $14,513 $7,857 85% 90 Colombia $14,503 $6,429 85% 91 Saint Lucia
$13,986 $9,602 82% 92 South Africa $13,526 $6,120 79% 93 Peru $13,463 $6,723 79% 94 Paraguay $13,109 $5,776 7
7% 95 Bosnia and Herzegovina $13,108 $5,387 77% 96 Mongolia $12,946 $3,672 76% 97 Albania $12,943 $4,521 76% 9
8 Sri Lanka $12,863 $4,135 75% 99 Indonesia $12,310 $3,837 72% 100 Tunisia $11,936 $3,494 70% 101 St. Vincent &
Grenadines $11,769 $7,150 69% 102 Ecuador $11,612 $6,214 68% 103 Egypt $11,608 $2,441 68% 104 Georgia $10,674
$3,762 62% 105 Namibia $10,471 $5,516 61% 106 Dominica $10,037 $6,951 59% 107 Armenia $9,668 $3,918 57% 108 F
iji $9,575 $5,768 56% 109 Bhutan $9,392 $3,391 55% 110 Jordan $9,173 $4,095 54% 111 Jamaica $9,066 $5,061 53%
112 Ukraine $8,699 $2,521 51% 113 Eswatini $8,659 $3,942 51% 114 Belize $8,525 $4,957 50% 115 Philippines $8,36
1 $2,982 49% 116 Morocco $8,225 $3,083 48% 117 Guyana $8,180 $4,671 48% 118 Guatemala $8,168 $4,471 48% 119
El Salvador $8,023 $3,883 47% 120 Bolivia $7,576 $3,351 44% 121 Timor-Leste $7,228 $2,377 42% 122 India $7,166
$1,980 42% 123 Laos $7,038 $2,424 41% 124 Cabo Verde $6,913 $3,298 40% 125 Uzbekistan $6,880 $1,554 40% 126 V
ietnam $6,790 $2,366 40% 127 Angola $6,658 $4,096 39% 128 Samoa $6,641 $4,305 39% 129 Myanmar $6,174 $1,256 3
6% 130 Tonga $5,969 $4,193 35% 131 Nigeria $5,887 $1,969 34% 132 Nicaragua $5,855 $2,164 34% 133 Moldova $5,7
11 $2,002 33% 134 Pakistan $5,539 $1,467 32% 135 Congo $5,454 $1,703 32% 136 Honduras $4,997 $2,437 29% 137
Sudan $4,914 $2,879 29% 138 State of Palestine $4,896 $3,054 29% 139 Ghana $4,502 $2,026 26% 140 Marshall Island
s $4,247 $3,517 25% 141 Papua New Guinea $4,208 $2,434 25% 142 Zambia $4,033 $1,535 24% 143 Cambodia $4,018
$1,384 23% 144 Mauritania $3,958 $1,173 23% 145 Côte d'Ivoire $3,945 $1,529 23% 146 Tuvalu $3,933 $3,494 23%
147 Bangladesh $3,877 $1,564 23% 148 Kyrgyzstan $3,735 $1,222 22% 149 Cameroon $3,722 $1,422 22% 150 Senegal
$3,458 $1,366 20% 151 Sao Tome & Principe $3,359 $1,896 20% 152 Kenya $3,292 $1,578 19% 153 Vanuatu $3,215 $3,0
22 19% 154 Tajikistan $3,202 $805 19% 155 Tanzania $2,948 $975 17% 156 Lesotho $2,932 $1,233 17% 157 Comoros
$2,751 $1,312 16% 158 Nepal $2,702 $900 16% 159 Yemen $2,606 $1,123 15% 160 Zimbabwe $2,434 $1,548 14% 161 S
olomon Islands $2,427 $2,049 14% 162 Benin $2,276 $827 13% 163 Guinea $2,247 $868 13% 164 Mali $2,218 $828 1

$657 11% 169 Ethiopia $1,903 $757 11% 170 Uganda $1,868 $631 11% 171 Burkina Faso $1,866 $642 11% 172 Haiti
$1,819 $766 11% 173 Guinea-Bissau $1,704 $737 10% 174 Gambia $1,699 $673 10% 175 Togo $1,663 $618 10% 176 Ma
dagascar $1,558 $450 9% 177 Sierra Leone $1,530 $504 9% 178 Liberia $1,285 $699 8% 179 Mozambique $1,250 $441
7% 180 Malawi $1,205 $357 7% 181 Niger $1,019 $376 6% 182 DR Congo $889 $462 5% 183 Burundi $735 $293 4%
184 Central African Republic $727 $424 4% 185 American Samoa N.A. $11,399 N.A. 186 Cuba N.A. $8,541 N.A. 187 Nor
thern Mariana Islands N.A. $28,164 N.A. 188 Guam N.A. $35,665 N.A. 189 Andorra N.A. $39,128 N.A.

In [1530]:
```
# Need to put the web data into a dataframe in python
# It is easier to pull the data as a list instead of a string
# split text
b = a.split()
c = b[19:1281]
print(c)
type(c)
```

['1', 'Qatar', '$128,647', '$61,264', '752%', '2', 'Macao', '$115,367', '$80,890', '675%', '3', 'Luxembourg', '$107,641', '$10
5,280', '629%', '4', 'Singapore', '$94,105', '$56,746', '550%', '5', 'Brunei', '$79,003', '$28,572', '462%', '6', 'Ireland',
'$76,745', '$69,727', '449%', '7', 'United', 'Arab', 'Emirates', '$74,035', '$40,325', '433%', '8', 'Kuwait', '$72,096', '$29,6
16', '422%', '9', 'Switzerland', '$66,307', '$80,296', '388%', '10', 'San', 'Marino', '$63,549', '$48,495', '372%', '11', 'Norw
ay', '$62,183', '$75,428', '364%', '12', 'Hong', 'Kong', '$61,671', '$46,733', '361%', '13', 'United', 'States', '$59,928', '$5
9,939', '350%', '14', 'Iceland', '$55,322', '$73,233', '324%', '15', 'Netherlands', '$54,422', '$48,796', '318%', '16', 'Denmar
k', '$54,356', '$57,545', '318%', '17', 'Saudi', 'Arabia', '$53,893', '$20,747', '315%', '18', 'Austria', '$53,879', '$47,261',
'315%', '19', 'Germany', '$52,556', '$44,680', '307%', '20', 'Sweden', '$51,405', '$54,075', '301%', '21', 'Australia', '$49,37
8', '$53,831', '289%', '22', 'Belgium', '$49,367', '$43,325', '289%', '23', 'Bahrain', '$47,708', '$23,715', '279%', '24', 'Can
ada', '$46,510', '$44,841', '272%', '25', 'Finland', '$46,344', '$45,778', '271%', '26', 'United', 'Kingdom', '$44,920', '$39,5
32', '263%', '27', 'France', '$44,033', '$39,827', '258%', '28', 'Japan', '$42,067', '$38,214', '246%', '29', 'Oman', '$41,33
1', '$15,170', '242%', '30', 'Italy', '$40,924', '$32,038', '239%', '31', 'Malta', '$40,797', '$28,585', '239%', '32', 'New',
'Zealand', '$40,748', '$43,415', '238%', '33', 'Aruba', '$39,493', '$25,630', '231%', '34', 'Spain', '$39,037', '$28,175', '22
8%', '35', 'Israel', '$38,868', '$42,852', '227%', '36', 'South', 'Korea', '$38,824', '$29,958', '227%', '37', 'Czech', 'Republ
ic', '(Czechia)', '$38,020', '$20,291', '222%', '38', 'Slovenia', '$36,387', '$23,488', '213%', '39', 'Cyprus', '$36,012', '$1
8,695', '211%', '40', 'Estonia', '$33,448', '$20,170', '196%', '41', 'Lithuania', '$33,253', '$16,709', '194%', '42', 'Portuga
l', '$32,554', '$21,316', '190%', '43', 'Slovakia', '$32,371', '$17,551', '189%', '44', 'Trinidad', 'and', 'Tobago', '$31,645',
'$15,952', '185%', '45', 'Bahamas', '$30,495', '$31,858', '178%', '46', 'Poland', '$29,924', '$13,871', '175%', '47', 'Malaysi
a', '$29,511', '$10,118', '173%', '48', 'Seychelles', '$29,328', '$15,536', '172%', '49', 'Hungary', '$28,799', '$14,364', '16
8%', '50', 'Saint', 'Kitts', '&', 'Nevis', '$28,636', '$19,061', '167%', '51', 'Greece', '$28,583', '$19,214', '167%', '52', 'L
atvia', '$28,362', '$15,613', '166%', '53', 'Turkey', '$28,002', '$10,498', '164%', '54', 'Romania', '$26,660', '$10,781', '15
6%', '55', 'Kazakhstan', '$26,491', '$9,009', '155%', '56', 'Croatia', '$26,296', '$13,200', '154%', '57', 'Russia', '$25,763',
'$10,846', '151%', '58', 'Chile', '$24,747', '$15,001', '145%', '59', 'Panama', '$24,521', '$15,166', '143%', '60', 'Equatoria

```
l', 'Guinea', '$24,439', '$9,741', '143%', '61', 'Antigua', 'and', 'Barbuda', '$23,522', '$15,825', '138%', '62', 'Uruguay',
'$22,610', '$16,341', '132%', '63', 'Mauritius', '$22,356', '$10,491', '131%', '64', 'Bulgaria', '$20,948', '$8,197', '123%',
'65', 'Iran', '$20,885', '$5,628', '122%', '66', 'Argentina', '$20,829', '$14,508', '122%', '67', 'Libya', '$19,673', '$5,791',
'115%', '68', 'Montenegro', '$19,355', '$7,720', '113%', '69', 'Belarus', '$18,896', '$5,762', '111%', '70', 'Mexico', '$18,65
6', '$9,224', '109%', '71', 'Barbados', '$18,559', '$16,328', '109%', '72', 'Gabon', '$18,113', '$7,271', '106%', '73', 'Turkme
nistan', '$18,031', '$6,587', '105%', '74', 'Thailand', '$17,910', '$6,579', '105%', '75', 'Azerbaijan', '$17,450', '$4,139',
'102%', '76', 'Costa', 'Rica', '$17,110', '$11,573', '100%', '77', 'Botswana', '$17,024', '$7,894', '100%', '78', 'Iraq', '$16,
935', '$5,114', '99%', '79', 'China', '$16,842', '$8,612', '98%', '80', 'Maldives', '$16,688', '$9,802', '98%', '81', 'Dominica
n', 'Republic', '$16,064', '$7,223', '94%', '82', 'Brazil', '$15,553', '$9,881', '91%', '83', 'Serbia', '$15,432', '$4,692', '9
0%', '84', 'Algeria', '$15,293', '$4,048', '89%', '85', 'North', 'Macedonia', '$15,290', '$5,418', '89%', '86', 'Suriname', '$1
5,191', '$5,251', '89%', '87', 'Grenada', '$15,156', '$10,164', '89%', '88', 'Palau', '$14,854', '$16,275', '87%', '89', 'Leban
on', '$14,513', '$7,857', '85%', '90', 'Colombia', '$14,503', '$6,429', '85%', '91', 'Saint', 'Lucia', '$13,986', '$9,602', '8
2%', '92', 'South', 'Africa', '$13,526', '$6,120', '79%', '93', 'Peru', '$13,463', '$6,723', '79%', '94', 'Paraguay', '$13,10
9', '$5,776', '77%', '95', 'Bosnia', 'and', 'Herzegovina', '$13,108', '$5,387', '77%', '96', 'Mongolia', '$12,946', '$3,672',
'76%', '97', 'Albania', '$12,943', '$4,521', '76%', '98', 'Sri', 'Lanka', '$12,863', '$4,135', '75%', '99', 'Indonesia', '$12,3
10', '$3,837', '72%', '100', 'Tunisia', '$11,936', '$3,494', '70%', '101', 'St.', 'Vincent', '&', 'Grenadines', '$11,769', '$7,
150', '69%', '102', 'Ecuador', '$11,612', '$6,214', '68%', '103', 'Egypt', '$11,608', '$2,441', '68%', '104', 'Georgia', '$10,6
74', '$3,762', '62%', '105', 'Namibia', '$10,471', '$5,516', '61%', '106', 'Dominica', '$10,037', '$6,951', '59%', '107', 'Arme
nia', '$9,668', '$3,918', '57%', '108', 'Fiji', '$9,575', '$5,768', '56%', '109', 'Bhutan', '$9,392', '$3,391', '55%', '110',
'Jordan', '$9,173', '$4,095', '54%', '111', 'Jamaica', '$9,066', '$5,061', '53%', '112', 'Ukraine', '$8,699', '$2,521', '51%',
'113', 'Eswatini', '$8,659', '$3,942', '51%', '114', 'Belize', '$8,525', '$4,957', '50%', '115', 'Philippines', '$8,361', '$2,9
82', '49%', '116', 'Morocco', '$8,225', '$3,083', '48%', '117', 'Guyana', '$8,180', '$4,671', '48%', '118', 'Guatemala', '$8,16
8', '$4,471', '48%', '119', 'El', 'Salvador', '$8,023', '$3,883', '47%', '120', 'Bolivia', '$7,576', '$3,351', '44%', '121', 'T
imor-Leste', '$7,228', '$2,377', '42%', '122', 'India', '$7,166', '$1,980', '42%', '123', 'Laos', '$7,038', '$2,424', '41%', '1
24', 'Cabo', 'Verde', '$6,913', '$3,298', '40%', '125', 'Uzbekistan', '$6,880', '$1,554', '40%', '126', 'Vietnam', '$6,790',
'$2,366', '40%', '127', 'Angola', '$6,658', '$4,096', '39%', '128', 'Samoa', '$6,641', '$4,305', '39%', '129', 'Myanmar', '$6,1
74', '$1,256', '36%', '130', 'Tonga', '$5,969', '$4,193', '35%', '131', 'Nigeria', '$5,887', '$1,969', '34%', '132', 'Nicaragu
a', '$5,855', '$2,164', '34%', '133', 'Moldova', '$5,711', '$2,002', '33%', '134', 'Pakistan', '$5,539', '$1,467', '32%', '13
5', 'Congo', '$5,454', '$1,703', '32%', '136', 'Honduras', '$4,997', '$2,437', '29%', '137', 'Sudan', '$4,914', '$2,879', '2
9%', '138', 'State', 'of', 'Palestine', '$4,896', '$3,054', '29%', '139', 'Ghana', '$4,502', '$2,026', '26%', '140', 'Marshal
l', 'Islands', '$4,247', '$3,517', '25%', '141', 'Papua', 'New', 'Guinea', '$4,208', '$2,434', '25%', '142', 'Zambia', '$4,03
3', '$1,535', '24%', '143', 'Cambodia', '$4,018', '$1,384', '23%', '144', 'Mauritania', '$3,958', '$1,173', '23%', '145', 'Côt
e', "d'Ivoire", '$3,945', '$1,529', '23%', '146', 'Tuvalu', '$3,933', '$3,494', '23%', '147', 'Bangladesh', '$3,877', '$1,564',
'23%', '148', 'Kyrgyzstan', '$3,735', '$1,222', '22%', '149', 'Cameroon', '$3,722', '$1,422', '22%', '150', 'Senegal', '$3,45
8', '$1,366', '20%', '151', 'Sao', 'Tome', '&', 'Principe', '$3,359', '$1,896', '20%', '152', 'Kenya', '$3,292', '$1,578', '1
9%', '153', 'Vanuatu', '$3,215', '$3,022', '19%', '154', 'Tajikistan', '$3,202', '$805', '19%', '155', 'Tanzania', '$2,948',
'$975', '17%', '156', 'Lesotho', '$2,932', '$1,233', '17%', '157', 'Comoros', '$2,751', '$1,312', '16%', '158', 'Nepal', '$2,70
2', '$900', '16%', '159', 'Yemen', '$2,606', '$1,123', '15%', '160', 'Zimbabwe', '$2,434', '$1,548', '14%', '161', 'Solomon',
'Islands', '$2,427', '$2,049', '14%', '162', 'Benin', '$2,276', '$827', '13%', '163', 'Guinea', '$2,247', '$868', '13%', '164',
'Mali', '$2,218', '$828', '13%', '165', 'Kiribati', '$2,185', '$1,626', '13%', '166', 'Rwanda', '$2,043', '$762', '12%', '167',

0', 'Uganda', '$1,868', '$631', '11%', '171', 'Burkina', 'Faso', '$1,866', '$642', '11%', '172', 'Haiti', '$1,819', '$766', '1
1%', '173', 'Guinea-Bissau', '$1,704', '$737', '10%', '174', 'Gambia', '$1,699', '$673', '10%', '175', 'Togo', '$1,663', '$61
8', '10%', '176', 'Madagascar', '$1,558', '$450', '9%', '177', 'Sierra', 'Leone', '$1,530', '$504', '9%', '178', 'Liberia',
'$1,285', '$699', '8%', '179', 'Mozambique', '$1,250', '$441', '7%', '180', 'Malawi', '$1,205', '$357', '7%', '181', 'Niger',
'$1,019', '$376', '6%', '182', 'DR', 'Congo', '$889', '$462', '5%', '183', 'Burundi', '$735', '$293', '4%', '184', 'Central',
'African', 'Republic', '$727', '$424', '4%', '185', 'American', 'Samoa', 'N.A.', '$11,399', 'N.A.', '186', 'Cuba', 'N.A.', '$8,
541', 'N.A.', '187', 'Northern', 'Mariana', 'Islands', 'N.A.', '$28,164', 'N.A.', '188', 'Guam', 'N.A.', '$35,665', 'N.A.', '18
9', 'Andorra', 'N.A.', '$39,128', 'N.A.']
```

Out[1530]: list

In [1531]:
```
import pandas as pd
```

In [1532]:
```
# put the data into a dataframe
f = pd.DataFrame(c)
f
```

Out[1532]:

	0
0	1
1	Qatar
2	$128,647
3	$61,264
4	752%
...	...
990	189
991	Andorra
992	N.A.
993	$39,128
994	N.A.

995 rows × 1 columns

In [1533]:
```
# need to use numpy to form an array
import numpy as np
```

In [1534]:
```
# since there are five variables per country, there are five columns
d = np.array(c)
d.reshape(199,5)
```

Out[1534]:
```
array([['1', 'Qatar', '$128,647', '$61,264', '752%'],
       ['2', 'Macao', '$115,367', '$80,890', '675%'],
       ['3', 'Luxembourg', '$107,641', '$105,280', '629%'],
       ['4', 'Singapore', '$94,105', '$56,746', '550%'],
       ['5', 'Brunei', '$79,003', '$28,572', '462%'],
       ['6', 'Ireland', '$76,745', '$69,727', '449%'],
       ['7', 'United', 'Arab', 'Emirates', '$74,035'],
       ['$40,325', '433%', '8', 'Kuwait', '$72,096'],
       ['$29,616', '422%', '9', 'Switzerland', '$66,307'],
       ['$80,296', '388%', '10', 'San', 'Marino'],
       ['$63,549', '$48,495', '372%', '11', 'Norway'],
       ['$62,183', '$75,428', '364%', '12', 'Hong'],
       ['Kong', '$61,671', '$46,733', '361%', '13'],
       ['United', 'States', '$59,928', '$59,939', '350%'],
       ['14', 'Iceland', '$55,322', '$73,233', '324%'],
       ['15', 'Netherlands', '$54,422', '$48,796', '318%'],
       ['16', 'Denmark', '$54,356', '$57,545', '318%'],
       ['17', 'Saudi', 'Arabia', '$53,893', '$20,747'],
       ['315%', '18', 'Austria', '$53,879', '$47,261'],
       ['315%', '19', 'Germany', '$52,556', '$44,680'],
```

In [1535]:
```
# create a dataframe for each country
c1 = f[0:5]
c2 = f[5:10].reset_index(drop=True)
c3 = f[10:15].reset_index(drop=True)
c4 = f[15:20].reset_index(drop=True)
c5 = f[20:25].reset_index(drop=True)
c6 = f[25:30].reset_index(drop=True)
c7 = f[37:42].reset_index(drop=True)
c8 = f[42:47].reset_index(drop=True)
c9 = f[53:58].reset_index(drop=True)
c10 = f[70:75].reset_index(drop=True)
c11 = f[75:80].reset_index(drop=True)
c12 = f[80:85].reset_index(drop=True)
c13 = f[91:96].reset_index(drop=True)
c14 = f[96:101].reset_index(drop=True)
c15 = f[101:106].reset_index(drop=True)
c16 = f[106:111].reset_index(drop=True)
c17 = f[111:116].reset_index(drop=True)
c18 = f[116:121].reset_index(drop=True)
c19 = f[121:126].reset_index(drop=True)
c20 = f[126:131].reset_index(drop=True)
c21 = f[137:142].reset_index(drop=True)
c22 = f[142:147].reset_index(drop=True)
c23 = f[147:152].reset_index(drop=True)
c24 = f[152:157].reset_index(drop=True)
c25 = f[157:162].reset_index(drop=True)
c26 = f[168:173].reset_index(drop=True)
c27 = f[173:178].reset_index(drop=True)
c28 = f[178:183].reset_index(drop=True)
c29 = f[196:201].reset_index(drop=True)
c30 = f[201:206].reset_index(drop=True)
```

In [1536]:
```
# each 'c' value represent a country
c1
```

Out[1536]:

	0
0	1
1	Qatar
2	$128,647
3	$61,264
4	752%

Merge the dataframes together

In [1537]:
```
df = [c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11, c12, c13, c14, c15, c16, c17, c18, c19, c20, c21, c22, c23, c24, c25,
    c26, c27, c28, c29, c30]
```

In [1538]:
```
df = pd.concat(df, sort=False, axis=1, join='outer', verify_integrity=False, copy=True)
```

In [1539]:
```
df.shape
```

Out[1539]: (5, 30)

In [1540]:
```
df.head()
```

Out[1540]:

	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
0	1	2	3	4	5	6	8	9	11	14	...	27	28	29	30	31	33
1	Qatar	Macao	Luxembourg	Singapore	Brunei	Ireland	Kuwait	Switzerland	Norway	Iceland	...	France	Japan	Oman	Italy	Malta	Aruba
2	$128,647	$115,367	$107,641	$94,105	$79,003	$76,745	$72,096	$66,307	$62,183	$55,322	...	$44,033	$42,067	$41,331	$40,924	$40,797	$39,493
3	$61,264	$80,890	$105,280	$56,746	$28,572	$69,727	$29,616	$80,296	$75,428	$73,233	...	$39,827	$38,214	$15,170	$32,038	$28,585	$25,630
4	752%	675%	629%	550%	462%	449%	422%	388%	364%	324%	...	258%	246%	242%	239%	239%	231%

5 rows × 30 columns

Change the orientation of the table from horizontal to vertical

In [1541]:
```
df = df.transpose()
```

The columns have no labels - Labeling each column

In [1542]:
```
df.rename(columns={
    0: 'Rank',
    1: 'Country_',
    2: 'GDP_PPP_per_Capita',
    3: 'GDP_Nominal_per_Capita',
    4: 'vs_World_PPP_GDP_per_Capita'
}, inplace=True)
```

In [1543]:
```
df.head(10)
```

Out[1543]:

	Rank	Country_	GDP_PPP_per_Capita	GDP_Nominal_per_Capita	vs_World_PPP_GDP_per_Capita
0	1	Qatar	$128,647	$61,264	752%
0	2	Macao	$115,367	$80,890	675%
0	3	Luxembourg	$107,641	$105,280	629%
0	4	Singapore	$94,105	$56,746	550%
0	5	Brunei	$79,003	$28,572	462%
0	6	Ireland	$76,745	$69,727	449%
0	8	Kuwait	$72,096	$29,616	422%
0	9	Switzerland	$66,307	$80,296	388%
0	11	Norway	$62,183	$75,428	364%
0	14	Iceland	$55,322	$73,233	324%

Remove the column that shows only 0

In [1544]:
```
# after being transposed, there is a column with only 0 values
# remove the column with only 0 values
df.reset_index(drop=True, inplace=True)
```

In [1545]:
```
# how the final data looks
df2 = df
df2.head()
```

Out[1545]:

	Rank	Country_	GDP_PPP_per_Capita	GDP_Nominal_per_Capita	vs_World_PPP_GDP_per_Capita
0	1	Qatar	$128,647	$61,264	752%
1	2	Macao	$115,367	$80,890	675%
2	3	Luxembourg	$107,641	$105,280	629%
3	4	Singapore	$94,105	$56,746	550%
4	5	Brunei	$79,003	$28,572	462%

Milestone 4 - Connect to an API and Pull data

In [1546]:
```
# load libraries
import urllib.request as request
import json
```

In [1547]:
```
# connect to an api and obtain data
with request.urlopen('http://api.worldbank.org/v2/country/all?format=json') as response:
    if response.getcode() == 200:
        source = response.read()
        data = json.loads(source)
    else:
        print('An Error Occurred')
```

In [1548]:
```
type(data)
```

Out[1548]: list

In [1549]:
```
print(data)
```


```
[{'page': 1, 'pages': 7, 'per_page': '50', 'total': 304}, [{'id': 'ABW', 'iso2Code': 'AW', 'name': 'Aruba', 'region': {'id':
'LCN', 'iso2code': 'ZJ', 'value': 'Latin America & Caribbean '}, 'adminregion': {'id': '', 'iso2code': '', 'value': ''}, 'inc
omeLevel': {'id': 'HIC', 'iso2code': 'XD', 'value': 'High income'}, 'lendingType': {'id': 'LNX', 'iso2code': 'XX', 'value':
'Not classified'}, 'capitalCity': 'Oranjestad', 'longitude': '-70.0167', 'latitude': '12.5167'}, {'id': 'AFG', 'iso2Code': 'A
F', 'name': 'Afghanistan', 'region': {'id': 'SAS', 'iso2code': '8S', 'value': 'South Asia'}, 'adminregion': {'id': 'SAS', 'is
o2code': '8S', 'value': 'South Asia'}, 'incomeLevel': {'id': 'LIC', 'iso2code': 'XM', 'value': 'Low income'}, 'lendingType':
{'id': 'IDX', 'iso2code': 'XI', 'value': 'IDA'}, 'capitalCity': 'Kabul', 'longitude': '69.1761', 'latitude': '34.5228'}, {'i
d': 'AFR', 'iso2Code': 'A9', 'name': 'Africa', 'region': {'id': 'NA', 'iso2code': 'NA', 'value': 'Aggregates'}, 'adminregio
n': {'id': '', 'iso2code': '', 'value': ''}, 'incomeLevel': {'id': 'NA', 'iso2code': 'NA', 'value': 'Aggregates'}, 'lendingTy
pe': {'id': '', 'iso2code': '', 'value': 'Aggregates'}, 'capitalCity': '', 'longitude': '', 'latitude': ''}, {'id': 'AGO', 'i
so2Code': 'AO', 'name': 'Angola', 'region': {'id': 'SSF', 'iso2code': 'ZG', 'value': 'Sub-Saharan Africa '}, 'adminregion':
{'id': 'SSA', 'iso2code': 'ZF', 'value': 'Sub-Saharan Africa (excluding high income)'}, 'incomeLevel': {'id': 'LMC', 'iso2cod
e': 'XN', 'value': 'Lower middle income'}, 'lendingType': {'id': 'IBD', 'iso2code': 'XF', 'value': 'IBRD'}, 'capitalCity': 'L
uanda', 'longitude': '13.242', 'latitude': '-8.81155'}, {'id': 'ALB', 'iso2Code': 'AL', 'name': 'Albania', 'region': {'id':
'ECS', 'iso2code': 'Z7', 'value': 'Europe & Central Asia'}, 'adminregion': {'id': 'ECA', 'iso2code': '7E', 'value': 'Europe &
Central Asia (excluding high income)'}, 'incomeLevel': {'id': 'UMC', 'iso2code': 'XT', 'value': 'Upper middle income'}, 'lend
ingType': {'id': 'IBD', 'iso2code': 'XF', 'value': 'IBRD'}, 'capitalCity': 'Tirane', 'longitude': '19.8172', 'latitude': '41.
3317'}, {'id': 'AND', 'iso2Code': 'AD', 'name': 'Andorra', 'region': {'id': 'ECS', 'iso2code': 'Z7', 'value': 'Europe & Centr
al Asia'}, 'adminregion': {'id': '', 'iso2code': '', 'value': ''}, 'incomeLevel': {'id': 'HIC', 'iso2code': 'XD', 'value': 'H
igh income'}, 'lendingType': {'id': 'LNX', 'iso2code': 'XX', 'value': 'Not classified'}, 'capitalCity': 'Andorra la Vella'
```


In [1550]:
```
# remove the first line because it blocks the list from being converted to the dataframe
data = data.pop(1)
data
```

Out[1550]:

```
[{'id': 'ABW',
  'iso2Code': 'AW',
  'name': 'Aruba',
  'region': {'id': 'LCN',
   'iso2code': 'ZJ',
   'value': 'Latin America & Caribbean '},
  'adminregion': {'id': '', 'iso2code': '', 'value': ''},
  'incomeLevel': {'id': 'HIC', 'iso2code': 'XD', 'value': 'High income'},
  'lendingType': {'id': 'LNX', 'iso2code': 'XX', 'value': 'Not classified'},
  'capitalCity': 'Oranjestad',
```

```
 'longitude': '-70.0167',
 'latitude': '12.5167'},
{'id': 'AFG',
 'iso2Code': 'AF',
 'name': 'Afghanistan',
 'region': {'id': 'SAS', 'iso2code': '8S', 'value': 'South Asia'},
 'adminregion': {'id': 'SAS', 'iso2code': '8S', 'value': 'South Asia'},
 'incomeLevel': {'id': 'LIC', 'iso2code': 'XM', 'value': 'Low income'},
 'lendingType': {'id': 'IDX', 'iso2code': 'XI', 'value': 'IDA'},
 'capitalCity': 'Kabul'
```

In [1551]:
```
# the world bank does not provide the api with the full list of all countries
len(data)
```

Out[1551]: 50

In [1552]:
```
# convert the list to dataframe
import pandas as pd
df = pd.DataFrame(data)
```

In [1553]:
```
df
```

Out[1553]:

	id	iso2Code	name	region	adminregion	incomeLevel	lendingType	capitalCity	longitude	latitude
0	ABW	AW	Aruba	{'id': 'LCN', 'iso2code': 'ZJ', 'value': 'Lati...	{'id': '', 'iso2code': '', 'value': ''}	{'id': 'HIC', 'iso2code': 'XD', 'value': 'High...	{'id': 'LNX', 'iso2code': 'XX', 'value': 'Not ...	Oranjestad	-70.0167	12.5167
1	AFG	AF	Afghanistan	{'id': 'SAS', 'iso2code': '8S', 'value': 'Sout...	{'id': 'SAS', 'iso2code': '8S', 'value': 'Sout...	{'id': 'LIC', 'iso2code': 'XM', 'value': 'Low ...	{'id': 'IDX', 'iso2code': 'XI', 'value': 'IDA'}	Kabul	69.1761	34.5228
2	AFR	A9	Africa	{'id': 'NA', 'iso2code': 'NA', 'value': 'Aggre...	{'id': '', 'iso2code': '', 'value': ''}	{'id': 'NA', 'iso2code': 'NA', 'value': 'Aggre...	{'id': '', 'iso2code': '', 'value': 'Aggregates'}			
3	AGO	AO	Angola	{'id': 'SSF', 'iso2code': 'ZG', 'value': 'Sub-...	{'id': 'SSA', 'iso2code': 'ZF', 'value': 'Sub-...	{'id': 'LMC', 'iso2code': 'XN', 'value': 'Lowe...	{'id': 'IBD', 'iso2code': 'XF', 'value': 'IBRD'}	Luanda	13.242	-8.81155
4	ALB	AL	Albania	{'id': 'ECS', 'iso2code': 'Z7', 'value': 'Euro...	{'id': 'ECA', 'iso2code': '7E', 'value': 'Euro...	{'id': 'UMC', 'iso2code': 'XT', 'value': 'Uppe...	{'id': 'IBD', 'iso2code': 'XF', 'value': 'IBRD'}	Tirane	19.8172	41.3317
5	AND	AD	Andorra	{'id': 'ECS', 'iso2code': 'Z7', 'value': 'Euro...	{'id': '', 'iso2code': '', 'value': ''}	{'id': 'HIC', 'iso2code': 'XD', 'value': 'High...	{'id': 'LNX', 'iso2code': 'XX', 'value': 'Not ...	Andorra la	1.5218	42.5075

Cleaning/Formatting data

In [1554]:
```
# need to do some data cleaning to extract useful info out of these columns
income = df['incomeLevel']
region = df['region']
adminregion = df['adminregion']
lending = df['lendingType']
income.head()
```

Out[1554]:
```
0    {'id': 'HIC', 'iso2code': 'XD', 'value': 'High...
1    {'id': 'LIC', 'iso2code': 'XM', 'value': 'Low ...
2    {'id': 'NA', 'iso2code': 'NA', 'value': 'Aggre...
3    {'id': 'LMC', 'iso2code': 'XN', 'value': 'Lowe...
4    {'id': 'UMC', 'iso2code': 'XT', 'value': 'Uppe...
Name: incomeLevel, dtype: object
```

In [1555]:
```
# while accessing this column, we have 3 pairs
# need to remove the first two pairs
income[1]
```

Out[1555]: {'id': 'LIC', 'iso2code': 'XM', 'value': 'Low income'}

In [1556]:
```
# remove the 'id' key
i = 0
while i < 50:
    del income[i]['id']
    i += 1

i = 0
while i < 50:
    del region[i]['id']
    i += 1

i = 0
while i < 50:
    del adminregion[i]['id']
    i += 1
```

```
i = 0
while i < 50:
    del lending[i]['id']
    i += 1
```

In [1557]:
```
# remove the 'iso2code' key
i = 0
while i < 50:
    del income[i]['iso2code']
    i += 1

i = 0
while i < 50:
    del region[i]['iso2code']
    i += 1

i = 0
while i < 50:
    del adminregion[i]['iso2code']
    i += 1

i = 0
while i < 50:
    del lending[i]['iso2code']
    i += 1
```

In [1558]:
```
# check the result
income.head()
```

Out[1558]:
```
0            {'value': 'High income'}
1             {'value': 'Low income'}
2             {'value': 'Aggregates'}
3    {'value': 'Lower middle income'}
4    {'value': 'Upper middle income'}
Name: incomeLevel, dtype: object
```

In [1559]:
```
# apply the changes into the dataframe
df['incomeLevel'] = income
df['region'] = region
df['adminregion'] = adminregion
df['lendingType'] = lending
```

In [1560]:
```
df.head()
```

Out[1560]:

	id	iso2Code	name	region	adminregion	incomeLevel	lendingType	capitalCity	longitude	latitude
0	ABW	AW	Aruba	{'value': 'Latin America & Caribbean '}	{'value': ''}	{'value': 'High income'}	{'value': 'Not classified'}	Oranjestad	-70.0167	12.5167
1	AFG	AF	Afghanistan	{'value': 'South Asia'}	{'value': 'South Asia'}	{'value': 'Low income'}	{'value': 'IDA'}	Kabul	69.1761	34.5228
2	AFR	A9	Africa	{'value': 'Aggregates'}	{'value': ''}	{'value': 'Aggregates'}	{'value': 'Aggregates'}			
3	AGO	AO	Angola	{'value': 'Sub-Saharan Africa '}	{'value': 'Sub-Saharan Africa (excluding high ...	{'value': 'Lower middle income'}	{'value': 'IBRD'}	Luanda	13.242	-8.81155
4	ALB	AL	Albania	{'value': 'Europe & Central Asia'}	{'value': 'Europe & Central Asia (excluding hi...	{'value': 'Upper middle income'}	{'value': 'IBRD'}	Tirane	19.8172	41.3317

Remove unncessary columns

In [1561]:
```
df = df.drop(columns=['id','iso2Code','adminregion'], axis=1)
df.head()
```

Out[1561]:

	name	region	incomeLevel	lendingType	capitalCity	longitude	latitude
0	Aruba	{'value': 'Latin America & Caribbean '}	{'value': 'High income'}	{'value': 'Not classified'}	Oranjestad	-70.0167	12.5167
1	Afghanistan	{'value': 'South Asia'}	{'value': 'Low income'}	{'value': 'IDA'}	Kabul	69.1761	34.5228
2	Africa	{'value': 'Aggregates'}	{'value': 'Aggregates'}	{'value': 'Aggregates'}			
3	Angola	{'value': 'Sub-Saharan Africa '}	{'value': 'Lower middle income'}	{'value': 'IBRD'}	Luanda	13.242	-8.81155
4	Albania	{'value': 'Europe & Central Asia'}	{'value': 'Upper middle income'}	{'value': 'IBRD'}	Tirane	19.8172	41.3317

Rename columns

In [1562]:
```
df.columns
```

Out[1562]:
```
Index(['name', 'region', 'incomeLevel', 'lendingType', 'capitalCity',
       'longitude', 'latitude'],
      dtype='object')
```

In [1563]:
```
df = df.rename(columns = {
    'name': 'Country',
    'region': 'Region',
    'incomeLevel': 'Income_Level',
    'lendingType': 'Lending_Type',
    'capitalCity': 'Capital_City',
    'longitude': 'Longitude',
    'latitude': 'Latitude'})
df.head()
```

Out[1563]:

	Country	Region	Income_Level	Lending_Type	Capital_City	Longitude	Latitude
0	Aruba	{'value': 'Latin America & Caribbean '}	{'value': 'High income'}	{'value': 'Not classified'}	Oranjestad	-70.0167	12.5167
1	Afghanistan	{'value': 'South Asia'}	{'value': 'Low income'}	{'value': 'IDA'}	Kabul	69.1761	34.5228
2	Africa	{'value': 'Aggregates'}	{'value': 'Aggregates'}	{'value': 'Aggregates'}			
3	Angola	{'value': 'Sub-Saharan Africa '}	{'value': 'Lower middle income'}	{'value': 'IBRD'}	Luanda	13.242	-8.81155
4	Albania	{'value': 'Europe & Central Asia'}	{'value': 'Upper middle income'}	{'value': 'IBRD'}	Tirane	19.8172	41.3317

Remove rows with empty values

In [1564]:
```
df = df[(df != '').all(axis=1)]
df3 = df
df.head(5)
```

Out[1564]:

	Country	Region	Income_Level	Lending_Type	Capital_City	Longitude	Latitude
0	Aruba	{'value': 'Latin America & Caribbean '}	{'value': 'High income'}	{'value': 'Not classified'}	Oranjestad	-70.0167	12.5167
1	Afghanistan	{'value': 'South Asia'}	{'value': 'Low income'}	{'value': 'IDA'}	Kabul	69.1761	34.5228
3	Angola	{'value': 'Sub-Saharan Africa '}	{'value': 'Lower middle income'}	{'value': 'IBRD'}	Luanda	13.242	-8.81155
4	Albania	{'value': 'Europe & Central Asia'}	{'value': 'Upper middle income'}	{'value': 'IBRD'}	Tirane	19.8172	41.3317
5	Andorra	{'value': 'Europe & Central Asia'}	{'value': 'High income'}	{'value': 'Not classified'}	Andorra la Vella	1.5218	42.5075

In [ ]:
```
```

Fuzzy Matching

In [1565]:
```
# load library to run fuzzy matching
from fuzzywuzzy import fuzz
from fuzzywuzzy import process
```

In [1566]:
```
# perform ratio() function to see the score
fuzz.ratio('Low income','High income')
```

Out[1566]: 67

In [1567]:
```
# perform partial_ratio() function to see the score
fuzz.partial_ratio('Low income','High income')
```

Out[1567]: 70

In [1568]:
```
# perform token_set_ratio() function to see the score
fuzz.token_set_ratio('Low income','High income')
```

Out[1568]: 75

Mileston 5 - Merge the Data / Store in a Database / Visualize the Data

Check the preceded datasets

In [1569]:
```
df1.head()
```

Out[1569]:

	Age	Education	Years_in_School	Occupation	Race	Gender	Work_hrs/Week	Annual_Income_above50K
0	82	HS-grad	9	Exec-managerial	White	Female	18	No
1	54	7th-8th	4	Machine-op-inspct	White	Female	40	No
2	41	Some-college	10	Prof-specialty	White	Female	40	No
3	34	HS-grad	9	Other-service	White	Female	45	No
4	38	10th	6	Adm-clerical	White	Male	40	No

In [1570]:
```
df2.head()
```

Out[1570]:

	Rank	Country_	GDP_PPP_per_Capita	GDP_Nominal_per_Capita	vs_World_PPP_GDP_per_Capita
0	1	Qatar	$128,647	$61,264	752%
1	2	Macao	$115,367	$80,890	675%
2	3	Luxembourg	$107,641	$105,280	629%
3	4	Singapore	$94,105	$56,746	550%
4	5	Brunei	$79,003	$28,572	462%

In [1571]:
```
df3.head()
```

Out[1571]:

	Country	Region	Income_Level	Lending_Type	Capital_City	Longitude	Latitude
0	Aruba	{'value': 'Latin America & Caribbean '}	{'value': 'High income'}	{'value': 'Not classified'}	Oranjestad	-70.0167	12.5167
1	Afghanistan	{'value': 'South Asia'}	{'value': 'Low income'}	{'value': 'IDA'}	Kabul	69.1761	34.5228
3	Angola	{'value': 'Sub-Saharan Africa '}	{'value': 'Lower middle income'}	{'value': 'IBRD'}	Luanda	13.242	-8.81155
4	Albania	{'value': 'Europe & Central Asia'}	{'value': 'Upper middle income'}	{'value': 'IBRD'}	Tirane	19.8172	41.3317
5	Andorra	{'value': 'Europe & Central Asia'}	{'value': 'High income'}	{'value': 'Not classified'}	Andorra la Vella	1.5218	42.5075

Merge the dataframes together

In [1572]:
```
df = [df1, df2, df3]
```

In [1573]:
```
#df = [df1, df2, df3]
df = pd.concat(df, sort=False, axis=1, join='outer', verify_integrity=False, copy=True)
```

In [1574]:
```
df.head(30)
```

k_hrs/Week	Annual_Income_above50K	Rank	Country_	GDP_PPP_per_Capita	GDP_Nominal_per_Capita	vs_World_PPP_GDP_per_Capita	Country	Region	Income_Lev
18	No	1	Qatar	$128,647	$61,264	752%	Aruba	{'value': 'Latin America & Caribbean '}	{'value': 'Hig income
40	No	2	Macao	$115,367	$80,890	675%	Afghanistan	{'value': 'South Asia'}	{'value': 'Lo income
40	No	3	Luxembourg	$107,641	$105,280	629%	NaN	NaN	Na
45	No	4	Singapore	$94,105	$56,746	550%	Angola	{'value': 'Sub-Saharan Africa '}	{'value': 'Low midd income
								{'value': 'Europe &	{'value': 'Upp

In [1575]:
```
df.dtypes
```

Out[1575]:
```
Age                          int64
Education                   object
Years_in_School              int64
Occupation                  object
Race                        object
Gender                      object
Work_hrs/Week                int64
Annual_Income_above50K      object
Rank                        object
Country_                    object
```

```
GDP_PPP_per_Capita             object
GDP_Nominal_per_Capita         object
vs_World_PPP_GDP_per_Capita    object
Country                        object
Region                         object
Income_Level                   object
Lending_Type                   object
Capital_City                   object
Longitude                      object
Latitude                       object
dtype: object
```

In [1576]:
```
# convert all columns to type str to store in the database
df=df.astype(str)
```

Store the data into a database (SQLLite)

In [1577]:
```
# load library
import sqlite3
```

In [1578]:
```
# create and connect to the database file name 'Milestone5'
conn = sqlite3.connect('Milestone5.sqlite')
```

In [1579]:
```
df.to_sql('Milestone5', conn, if_exists='replace', index=False)
pd.read_sql('SELECT * FROM Milestone5', conn).head()
```

Out[1579]:

	Age	Education	Years_in_School	Occupation	Race	Gender	Work_hrs/Week	Annual_Income_above50K	Rank	Country_	GDP_PPP_per_Capita	GDP_Nor
0	82	HS-grad	9	Exec-managerial	White	Female	18	No	1	Qatar	$128,647	
1	54	7th-8th	4	Machine-op-inspct	White	Female	40	No	2	Macao	$115,367	
2	41	Some-college	10	Prof-specialty	White	Female	40	No	3	Luxembourg	$107,641	
3	34	HS-grad	9	Other-service	White	Female	45	No	4	Singapore	$94,105	
4	38	10th	6	Adm-clerical	White	Male	40	No	5	Brunei	$79,003	

Visualizing Data

In [1580]:
```
# load libraries
import plotly.express as px
import plotly.graph_objects as go
import matplotlib.pyplot as plt
```

From the first dataset

In [1581]:
```
# check the columns from the database
pd.read_sql("SELECT * FROM Milestone5", conn).columns
```

Out[1581]:
```
Index(['Age', 'Education', 'Years_in_School', 'Occupation', 'Race', 'Gender',
       'Work_hrs/Week', 'Annual_Income_above50K', 'Rank', 'Country_',
       'GDP_PPP_per_Capita', 'GDP_Nominal_per_Capita',
       'vs_World_PPP_GDP_per_Capita', 'Country', 'Region', 'Income_Level',
       'Lending_Type', 'Capital_City', 'Longitude', 'Latitude'],
      dtype='object')
```

In [1599]:
```
# assign variables to read the columns
gender = pd.read_sql("SELECT Gender FROM Milestone5", conn)
income_1 = pd.read_sql("SELECT Annual_Income_above50K FROM Milestone5", conn)
schoolyr = pd.read_sql("SELECT Years_in_School FROM Milestone5", conn)
country_2 = pd.read_sql("SELECT Country_ FROM Milestone5", conn)
gdp_21 = pd.read_sql("SELECT GDP_Nominal_per_Capita FROM Milestone5", conn)
gdp_22 = pd.read_sql("SELECT vs_World_PPP_GDP_per_Capita FROM Milestone5", conn)
gdp_23 = pd.read_sql("SELECT GDP_PPP_per_Capita FROM Milestone5", conn)
```

In [1583]:
```
# create a dataframe to plot the first dataset
df_db1 = [gender, income_1]
df_db1 = pd.concat(df_db1, sort=False, axis=1, join='outer', verify_integrity=False, copy=True)
```

In [1584]:
```
df_db1.columns
```

Out[1584]:
```
Index(['Gender', 'Annual_Income_above50K'], dtype='object')
```

In [1585]:
```
# examine the data
x = df_db1.groupby('Annual_Income_above50K')['Gender'].value_counts()
x
```

Out[1585]:
```
Annual_Income_above50K  Gender
No                      Male      14265
                        Female     8803
Yes                     Male       6523
                        Female     1127
Name: Gender, dtype: int64
```

In [1586]:
```
# create a dataframe to plot the data
plot_df1 = pd.DataFrame({'Above 50k':{'Female':'1127','Male':'6523'},
                         'Below 50k':{'Female':'8803', 'Male': '14265'}})
```

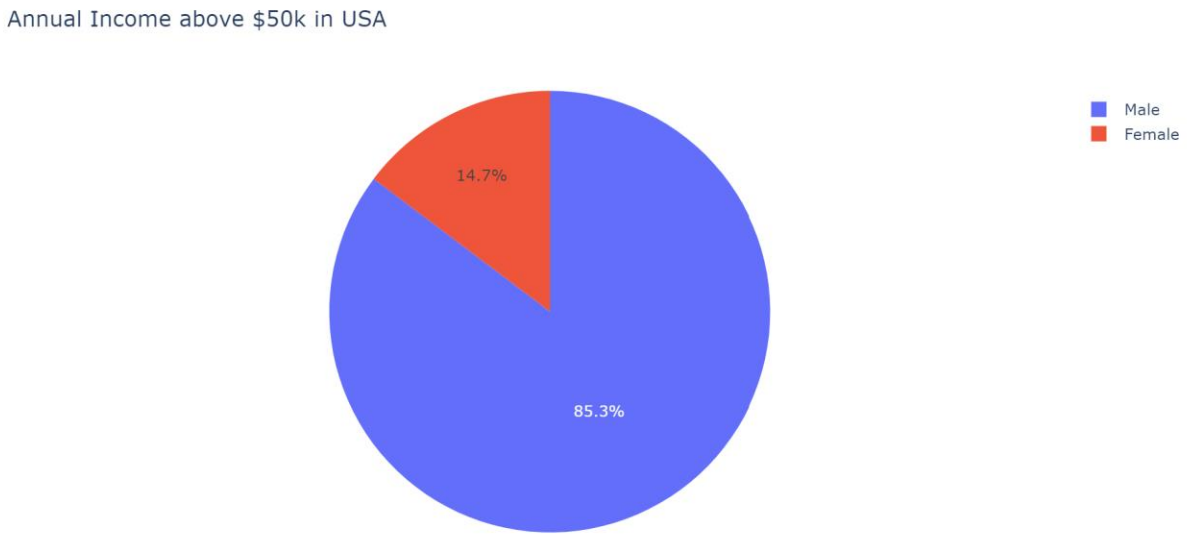
In [1587]:
```
plot_df1
```

Out[1587]:

	Above 50k	Below 50k
Female	1127	8803
Male	6523	14265

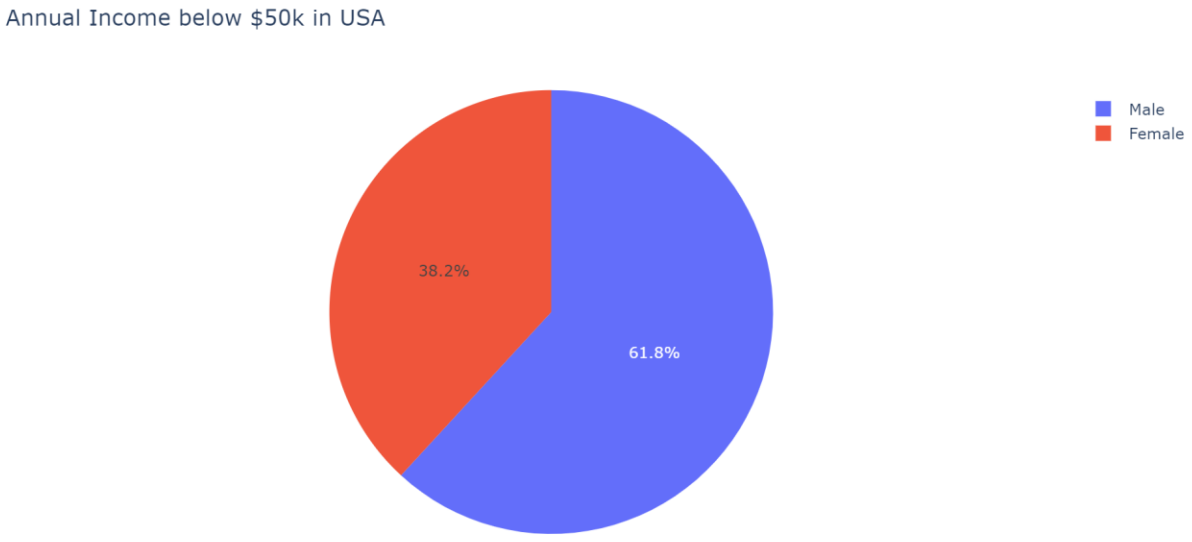
In [1588]:
```
# create a pie chart
labels = ['Female','Male']
values = plot_df1['Above 50k']

fig = go.Figure(data=[go.Pie(labels=labels, values=values)])
fig.update_layout(title_text='Annual Income above $50k in USA')
fig.show()
```



In [1589]:
```
# create a pie chart
labels = ['Female','Male']
values = plot_df1['Below 50k']

fig = go.Figure(data=[go.Pie(labels=labels, values=values)])
fig.update_layout(title_text='Annual Income below $50k in USA')
fig.show()
```



According to the data, for both high income and low income, there are more males than females.

There are many reasons to explain this: more males responded to the survey, or more males in the workforce.

However, the interesting fact about this data is that at low-income level, the difference between male and female is much smaller. As both male and female make over 50k a year, we see far more male in this range.

In [1590]:
```
# merge two columns
df_db2 = [schoolyr, income_1]
```

In [1591]:
```
df_db2 = pd.concat(df_db2, sort=True, axis=1)
```

In [1592]:
```
df_db2.head()
```

Out[1592]:

	Years_in_School	Annual_Income_above50K
0	9	No
1	4	No
2	10	No
3	9	No
4	6	No

In [1594]:
```
# examine the data
x = df_db2.groupby('Years_in_School')['Annual_Income_above50K'].value_counts()
x
```

Out[1594]:

Years_in_School	Annual_Income_above50K	
1	No	46
10	No	5423
	Yes	1352
11	No	973
	Yes	348
12	No	761
	Yes	259
13	No	3006
	Yes	2176
14	Yes	941
	No	734
15	Yes	415
	No	143
16	Yes	295
	No	103
2	No	150
	Yes	6
3	No	289
	Yes	14
4	No	535
	Yes	38
5	No	437
	Yes	26
6	No	771
	Yes	60
7	No	996
	Yes	60
8	No	362
	Yes	31
9	No	8339
	Yes	1629

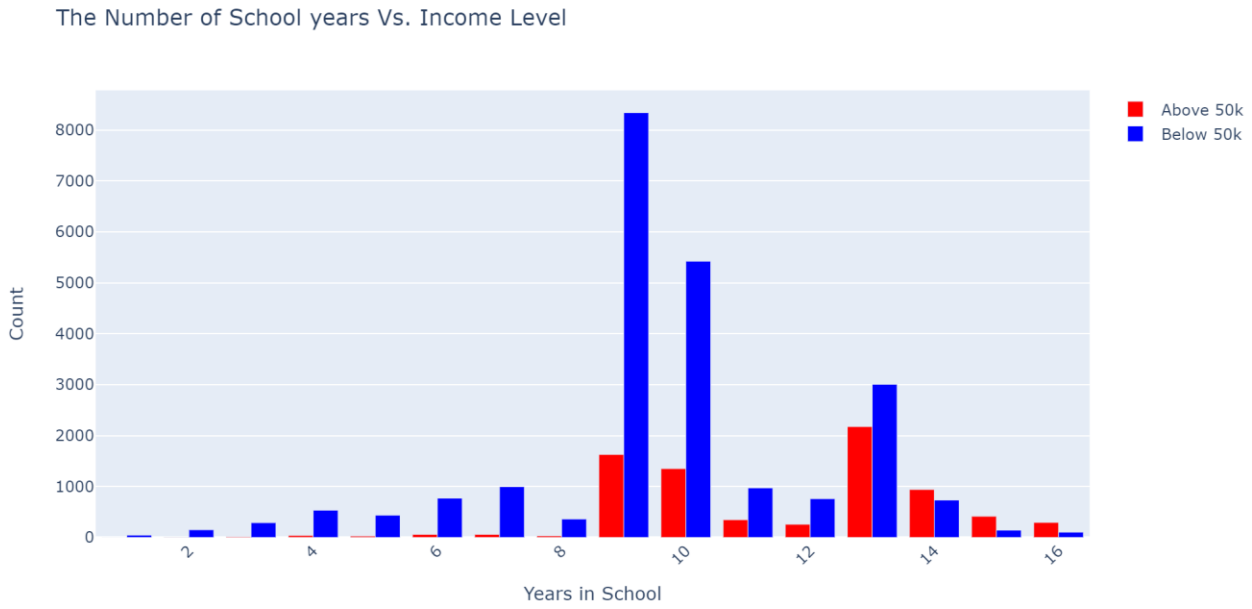
Name: Annual_Income_above50K, dtype: int64

In [1637]:

```
# draw a bar chart
years = ['1', '2', '3', '4', '5', '6',
         '7', '8', '9', '10', '11', '12', '13', '14', '15', '16']

fig = go.Figure()
fig.add_trace(go.Bar(
    x=years,
    y=[0, 6, 14, 38, 26, 60, 60, 31, 1629, 1352, 348, 259, 2176, 941, 415, 295],
    name='Above 50k',
    marker_color='rgb(255,0,0)'
))
fig.add_trace(go.Bar(
    x=years,
    y=[46, 150, 289, 535, 437, 771, 996, 362, 8339, 5423, 973, 761, 3006, 734, 143, 103],
    name='Below 50k',
    marker_color='rgb(0,0,255)'
))

# Here we modify the tickangle of the xaxis, resulting in rotated labels.
fig.update_layout(barmode='group', xaxis_tickangle=-45,
                  yaxis_title="Count",
                  xaxis_title="Years in School",
                  title='The Number of School years Vs. Income Level')
fig.show()
```



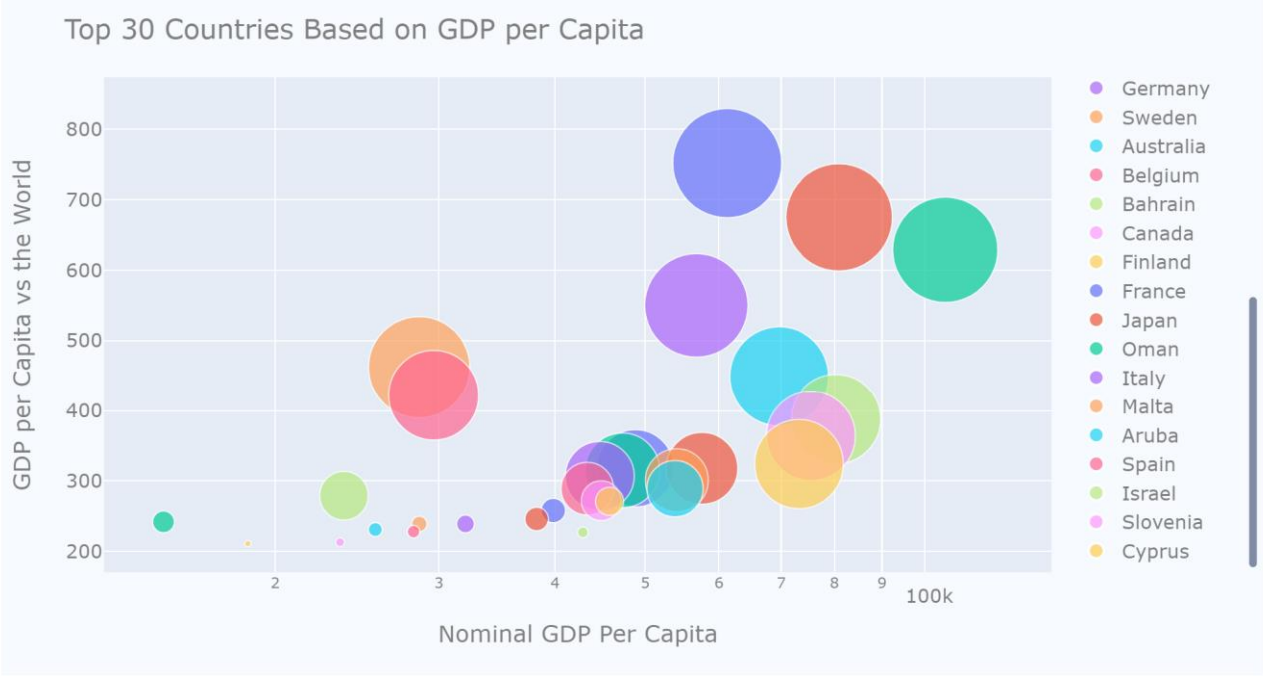
Another look at the income gap, for people who complete less than 14 years in school, there are far more people who make below 50k a year compared to those who make more than 50k a year.

As the years of education goes from 14 to 16, more people report to make above 50k a year than those who make less than 50k a year. This data shows that college makes a considerable impact in people's income.

In [1618]:
```
# prepare the data to draw the second dataset
gdp_21=gdp_21[:30]
gdp_22=gdp_22[:30]
country_2=country_2[:30]
gdp_23=gdp_23[:30].astype(object)
rank = [300,290,280,270,260,250,204,203,202,201,
       150,130,140,120,100,80,70,60,40,20,
       15,14,12,8,6,5,4,3,2,1,]
```

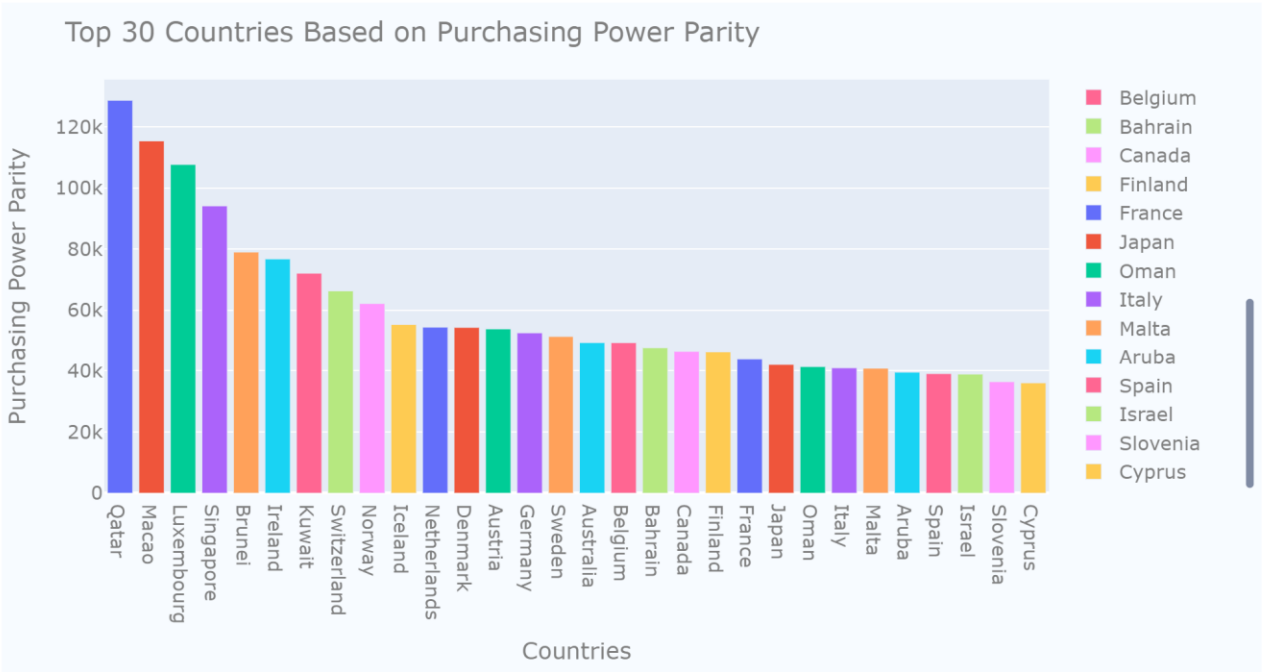
In [1619]:
```
# create a dataframe to plot the data
plot_df2 = [country_2, gdp_21, gdp_22, gdp_23]
plot_df2 = pd.concat(plot_df2, sort=True, axis=1)
```

In [1620]:
```
# draw a scatter plot
fig = px.scatter(plot_df2, x="GDP_Nominal_per_Capita", y="vs_World_PPP_GDP_per_Capita", color="Country_", size=rank,
          hover_name="Country_", log_x=True, size_max=60)
fig.update_layout(
    paper_bgcolor="rgb(247,251,255)",
    title="Top 30 Countries Based on GDP per Capita",
    xaxis_title="Nominal GDP Per Capita",
    yaxis_title="GDP per Capita vs the World",
    font=dict(
        size=15,
        color="#7f7f7f"))
fig.show()
```



According to this graph, Qatar, Macao and Luxembourg have the highest nominal GDP and GDP per capita versus the average of the world.

In [1636]:
```
# draw a bar chart
fig = px.bar(plot_df2, x='Country_', y='GDP_PPP_per_Capita', color='Country_')
fig.update_layout(
    paper_bgcolor="rgb(247,251,255)",
    title="Top 30 Countries Based on Purchasing Power Parity",
    xaxis_title="Countries",
    yaxis_title="Purchasing Power Parity",
    font=dict(
        size=15,
        color="#7f7f7f"))
fig.show()
```



Purchasing Power Parity is a powerful indicator about the finance of an individual because it takes into account the living standard of a country. However, this is only relative to their own country.

Once again, this shows that Qatar, Macao and Luxembourg dominate in this category.

What I have learned and had to do to complete this project

In this project, I have gone over five milestones over the course of 12 weeks to get here. Although the journey has not been too long, but it been fun, exciting and challenging. Along the road, there was not a single easy line of code. Everything needs to be figured out and used properly. Starting with the least technical work and also the most time-consuming task, finding an appropriate dataset with at least 1000 rows and 10 columns. In addition, I had to find a website and API that should be workable for my level. I end up not using the ones I picked because eventually I figured out that the API was not free for public and the website gave 403 error when I tried to access. I have learned most basic tasks about cleaning/formating/organizing data to make them look easier to understand. I have learned about Fuzzy Matching, pulled data from a website and from an API. Processing the data after pulling it from the website and API has not been easy, but when I look back, it is not hard; but it requires some clever technique to work around the data. The last milestone teaches me about the basic of database, such as creating a database, storing data on the database and pulling the data. I learned to manipulate the type of data before I store it in the database. Lastly, I learn about visualizing data. I have used matplotlib and seaborn before, so this time I want to try the plotly package.