In [ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

1 Loading Data

In [ ]:
```
df = pd.read_csv('NCVS_Select_-_Personal_Population.csv')
df.head()
```

Out[ ]:

	idper	yearq	year	ager	sex	hispanic	race	race_ethnicity	h
0	20418276993093879999916201	2001.4	2001	6	2	2	1	1	
1	20418276993653049999916201	2001.4	2001	5	1	2	1	1	
2	20418276993662879999926201	2001.4	2001	3	2	2	1	1	
3	20418280990253879999916201	2001.4	2001	4	2	1	1	6	
4	20418280990253879999916202	2001.4	2001	4	1	1	1	6	

2 Data Clean

In [ ]:
```
for columns in df:
    print(df[columns].value_counts())
```

```
idper
264207125606308229282813401    8
262974897319650429208811401    8
262974225891225329202011402    8
264239755691453729533715402    8
264239755691453729533715401    8
                              ..
194988345501077499999926303    1
194988345501077499999926304    1
195002973091888299999924601    1
195002973091888299999924602    1
206049355841652356822515202    1
Name: count, Length: 1777471, dtype: int64
yearq
2019.1    76103
2021.1    76006
2019.2    75633
2019.3    75332
2021.2    74782
          ...
2008.3    38209
2005.3    38168
2011.4    25483
1993.1    23724
1993.2    23663
Name: count, Length: 116, dtype: int64
year
2019    301100
2018    296017
2021    291878
2017    285904
2020    270566
1995    197366
1994    196865
2016    196186
2015    189711
1996    188010
2012    187684
2013    182699
2014    181178
2001    179143
2000    177923
1998    177654
1997    177603
1999    175524
2002    174252
2004    173796
2003    172703
2007    170869
2010    167444
2011    162867
2005    158988
2009    157796
2006    157108
2008    155704
1993    146593
```

```
Name: count, dtype: int64
ager
4    1440655
5    1244619
6    1007951
3     886380
2     539938
1     531588
Name: count, dtype: int64
sex
2    2950937
1    2700194
Name: count, dtype: int64
hispanic
2     4909536
1      717448
88      24147
Name: count, dtype: int64
race
1    4678655
2     619139
4     267524
5      53254
3      32559
Name: count, dtype: int64
race_ethnicity
1    4000367
6     717448
2     602114
4     260286
5      43835
3      27081
Name: count, dtype: int64
hincome1
7     1460567
88     882150
6      860947
5      766472
4      570543
3      533634
2      360791
1      216027
Name: count, dtype: int64
hincome2
-1    4205666
 3     478849
 2     345778
 4     292545
 1     229044
 5      99249
Name: count, dtype: int64
marital
2     2950463
1     1695642
4      516249
3      341099
```

```
5      104437
88      43241
Name: count, dtype: int64
popsize
 1    2353128
 0    1455249
-1     527521
 2     471606
 5     344389
 3     262802
 4     236436
Name: count, dtype: int64
region
 3    1801407
 2    1283231
 4    1157128
 1     881844
-1     527521
Name: count, dtype: int64
msa
2    3049509
1    1718204
3     883418
Name: count, dtype: int64
locality
-1    5088687
 2     388429
 3     121633
 1      52382
Name: count, dtype: int64
educatn1
5     2721439
4     2243527
3      443496
88     127299
2       98658
1       16712
Name: count, dtype: int64
educatn2
-1     1790933
 5      980979
 6      942885
 7      689055
 4      459147
 8      366533
 3      265919
 98      89260
 2       56182
 1       10238
Name: count, dtype: int64
veteran
-2    4205666
 0    1023869
 9     201793
-1     116797
 1      95265
```

```
 8       7741
Name: count, dtype: int64
citizen
-1    4205666
 1    1044721
 9     252866
 2      86865
 3      54225
 8       6788
Name: count, dtype: int64
wgtpercy
0.000000       730019
391.124280         33
314.834460         30
357.007232         29
330.599376         29
                ...
961.645275          1
1194.627330         1
1244.333535         1
1123.328660         1
1215.185743         1
Name: count, Length: 2870617, dtype: int64
```

In [ ]:
```
for columns in df:
    print(df[columns].unique())
```

```
['20418276993093879999916201' '20418276993653049999916201'
 '20418276993662879999926201' ... '20604413154692755644151 4101'
 '20604935584165235682251 5201' '20604935584165235682251 5202']
[2001.4 2001.2 2000.1 2004.4 1997.1 1995.2 1999.2 1996.2 1994.2 2003.1
 2007.1 2005.4 2005.3 1998.4 1993.1 1997.4 2005.2 2007.3 2004.1 2003.3
 2000.4 2002.4 2002.2 2008.1 2004.3 2003.4 2007.4 2006.2 2006.3 2005.1
 2007.2 2006.4 2006.1 2004.2 1998.1 2008.2 1999.3 1993.2 1994.3 1997.2
 1993.3 2000.2 1995.3 2003.2 1996.3 1999.1 1998.2 2002.1 2001.3 1994.4
 1993.4 1999.4 2002.3 2001.1 1997.3 1995.4 2000.3 1996.4 1995.1 1994.1
 1998.3 1996.1 2012.1 2013.3 2017.2 2010.2 2016.2 2015.1 2018.1 2018.3
 2013.4 2012.2 2010.3 2016.3 2008.3 2015.2 2018.4 2010.4 2008.4 2014.1
 2012.3 2017.3 2015.3 2016.4 2009.1 2018.2 2011.1 2012.4 2014.2 2009.2
 2011.2 2015.4 2013.1 2014.3 2009.3 2017.1 2017.4 2011.3 2009.4 2013.2
 2016.1 2014.4 2011.4 2010.1 2019.4 2019.3 2020.2 2020.1 2020.3 2020.4
 2021.1 2021.2 2021.3 2021.4 2019.2 2019.1]
[2001 2000 2004 1997 1995 1999 1996 1994 2003 2007 2005 1998 1993 2002
 2008 2006 2012 2013 2017 2010 2016 2015 2018 2014 2009 2011 2019 2020
 2021]
[6 5 3 4 1 2]
[2 1]
[ 2  1 88]
[1 4 2 3 5]
[1 6 4 2 3 5]
[88  7  6  5  3  4  1  2]
[-1  2  5  4  3  1]
[ 4  1  2  5  3 88]
[ 1  5  0  4  3  2 -1]
[ 1  4  2  3 -1]
[3 1 2]
[-1  3  2  1]
[ 4  5  3 88  2  1]
[-1  6  7  5  3  4 98  8  2  1]
[-2  0  9  1 -1  8]
[-1  2  9  1  3  8]
[1938.505545  1085.292835  1065.50968   ... 1148.070153  1318.3525315
 1215.1857425]
```

In [ ]:
```
for columns in df:
    print(columns,df[columns].isnull().sum())
```

```
idper 0
yearq 0
year 0
ager 0
sex 0
hispanic 0
race 0
race_ethnicity 0
hincome1 0
hincome2 0
marital 0
popsize 0
region 0
msa 0
locality 0
educatn1 0
educatn2 0
veteran 0
citizen 0
wgtpercy 0
```

2 Data Drop

In [ ]:
```
percentageMissing1 = (df['hincome2'] == -1).sum()/df['hincome2'].count()* 100
percentageMissing2 = (df['locality'] == -1).sum()/df['locality'].count()* 100
percentageMissing3 = (df['veteran'] == -2).sum()/df['veteran'].count()* 100
percentageMissing4 = (df['citizen'] == -1).sum()/df['citizen'].count()* 100

print("hincome2 : ",percentageMissing1)
print("locality : ",percentageMissing2)
print("veteran : ",percentageMissing3)
print("citizen : ",percentageMissing4)
```

```
hincome2 :  74.42166886593144
locality :  90.04723125335443
veteran :  74.42166886593144
citizen :  74.42166886593144
```

In [ ]:
```
df = df.drop(columns=['idper','yearq','hispanic','race_ethnicity','hincome2','local
```

In [ ]:
```
df = df.drop(df[df['sex'] == 1].index)
```

Dropped Columns:

No.	Columns	Reason
1	idper	ID used to identify row, seems unnessecary
2	yearq	yearq is irrelevant as we have year
3	hispanic	irrelevant
4	race_ethnicity	irrelevant, using race
5	hincome2	74% rows invalid data
6	locality	90% rows invalid data
7	msa	irrelevant
8	veteran	74% rows invalid data
9	citizen	74% rows invalid data
10	wgtpercy	not sure how to use it in this analysis
11	sex == 1	male population

3 Data Visualizations

In [ ]:
```
len(df)
```

Out[ ]:
```
2950937
```

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

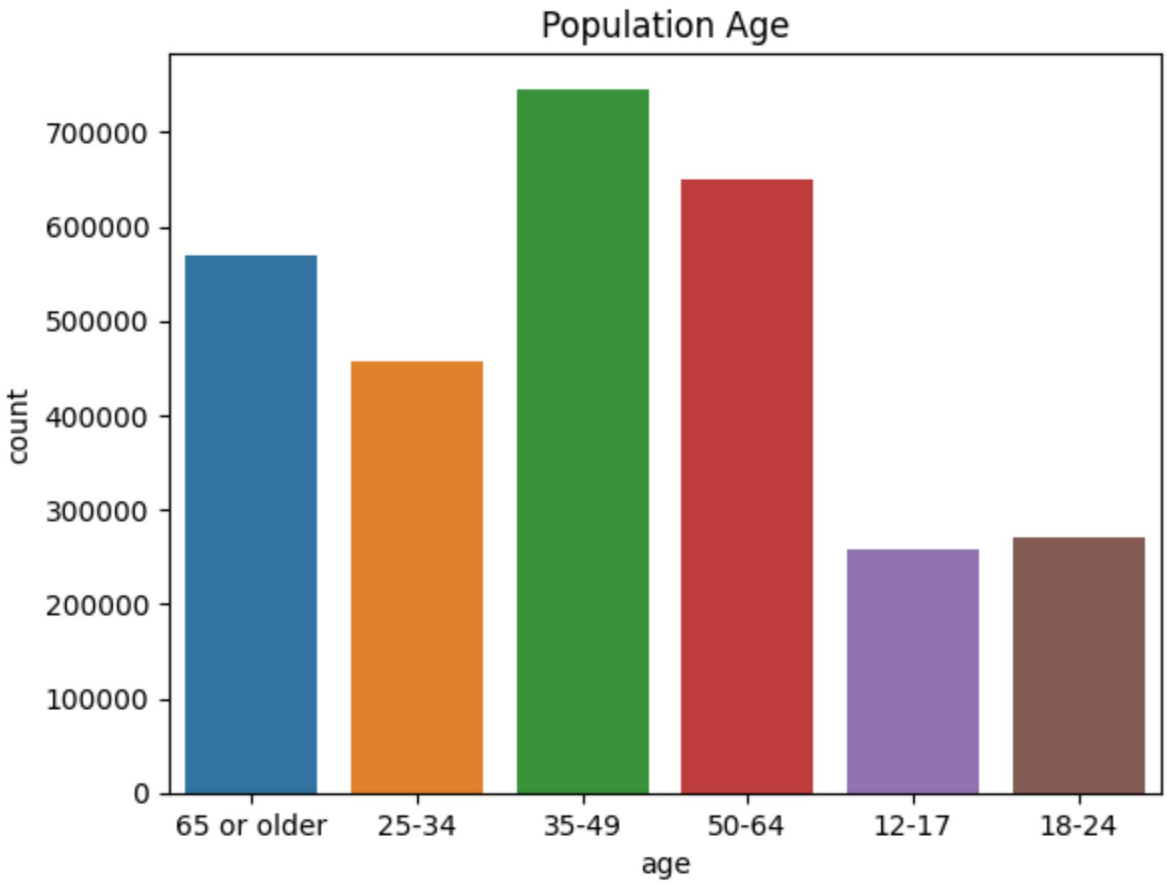
Out[ ]:
```
Index(['year', 'ager', 'sex', 'race', 'hincome1', 'marital', 'popsize',
       'region', 'educatn1', 'educatn2'],
      dtype='object')
```

In [ ]:
```
title = 'Population Age'
plt.title(title)
plt.xlabel('Age Group')
temp = []
for row in df['ager']:
    if row == 1:
        temp.append("12-17")
    if row == 2:
        temp.append("18-24")
    if row == 3:
        temp.append("25-34")
    if row == 4:
        temp.append("35-49")
    if row == 5:
        temp.append("50-64")
    if row == 6:
        temp.append("65 or older")

dfv = pd.DataFrame()
dfv['age'] = temp

sns.countplot(x=dfv['age'])
#plt.savefig(f'{title}.png',format='png')
```

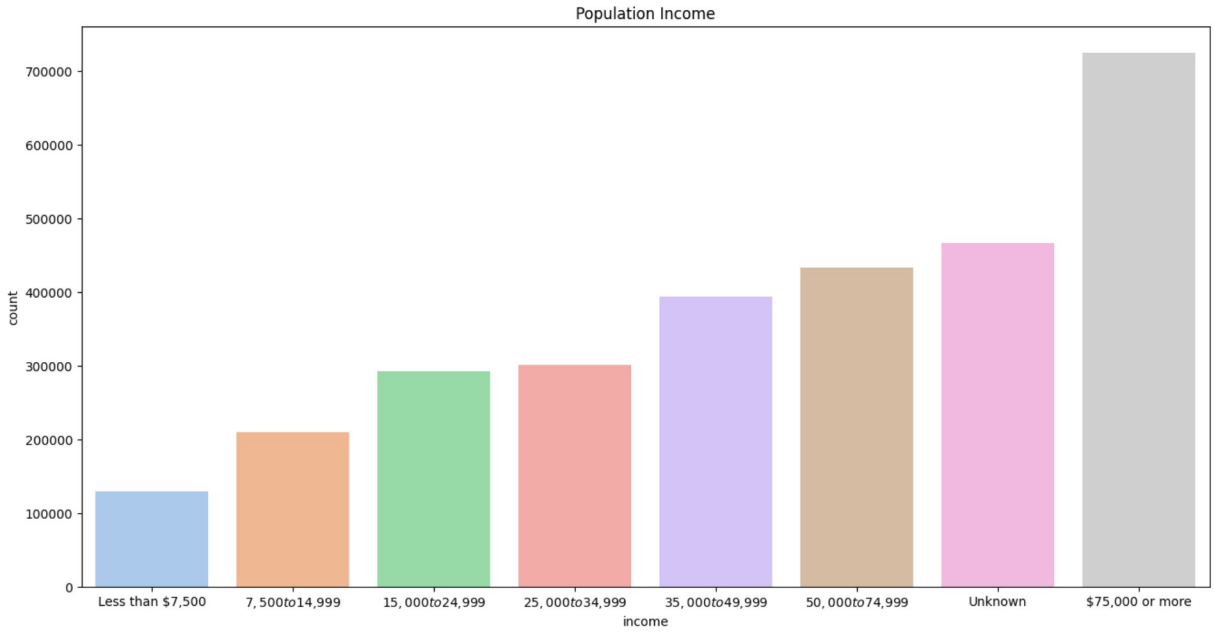
Out[ ]:
```
<Axes: title={'center': 'Population Age'}, xlabel='age', ylabel='count'>
```



Population follows a regular population distribution

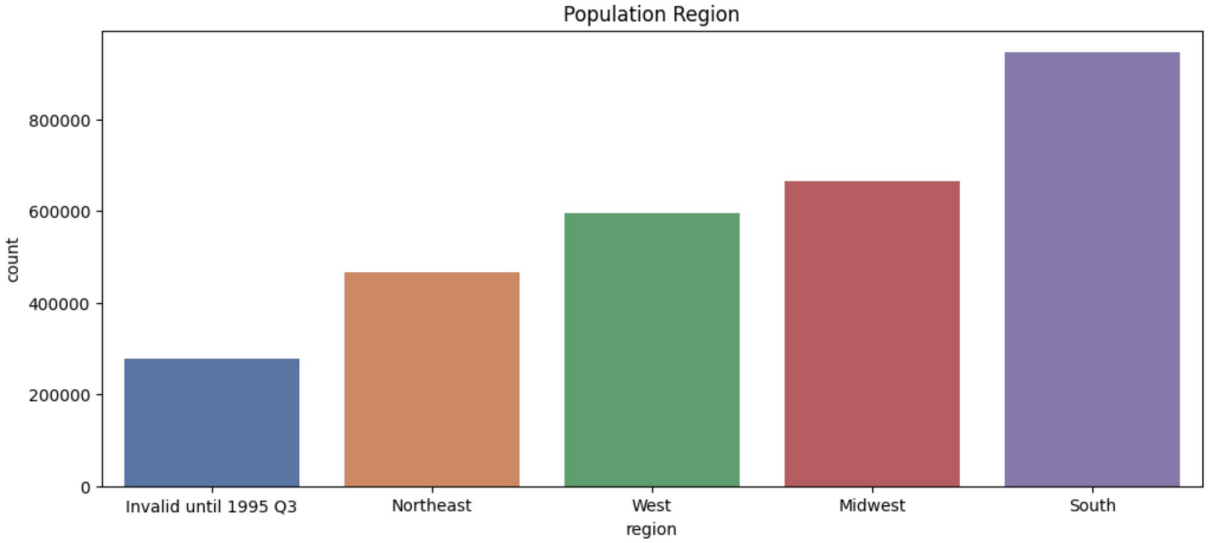
In [ ]:
```
title = "Population Income"
plt.figure(figsize=(16,8))
plt.title(title)
colors = sns.color_palette('pastel')
dict = {1:'Less than $7,500',2:'$7,500 to $14,999',3:'$15,000 to $24,999',4:'$25,00
        5:'$35,000 to $49,999',6:'$50,000 to $74,999',7:'$75,000 or more',88:'Unkno
dfv['income'] = [dict[row] for row in df['hincome1']]
sns.countplot(x=dfv['income'], order = dfv['income'].value_counts(ascending=True).i
```

Out[ ]: <Axes: title={'center': 'Population Income'}, xlabel='income', ylabel='count'>



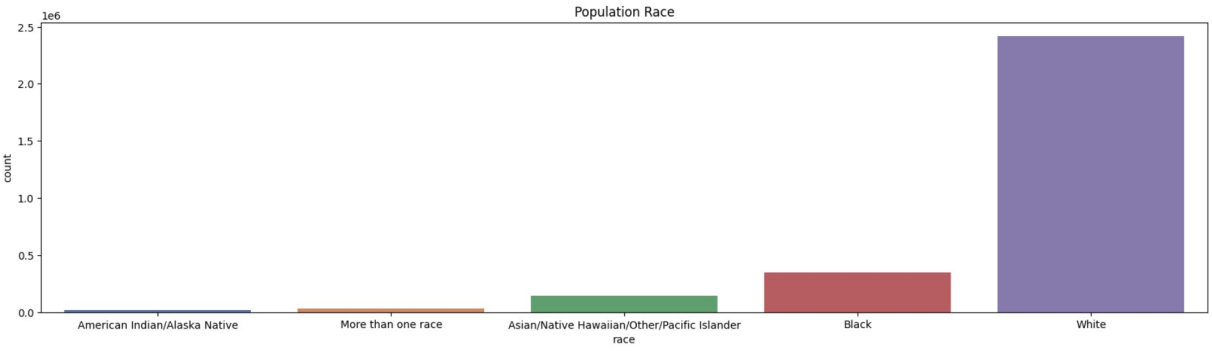
In [ ]:
```
title = "Population Region"
plt.figure(figsize=(12,5))
plt.title(title)
colors = sns.color_palette('deep')
dict = {-1:'Invalid until 1995 Q3',1:'Northeast',2:'Midwest',3:'South',
        4:'West'}
dfv['region'] = [dict[row] for row in df['region']]
sns.countplot(x=dfv['region'], order = dfv['region'].value_counts(ascending=True).i
```

Out[ ]: <Axes: title={'center': 'Population Region'}, xlabel='region', ylabel='count'>



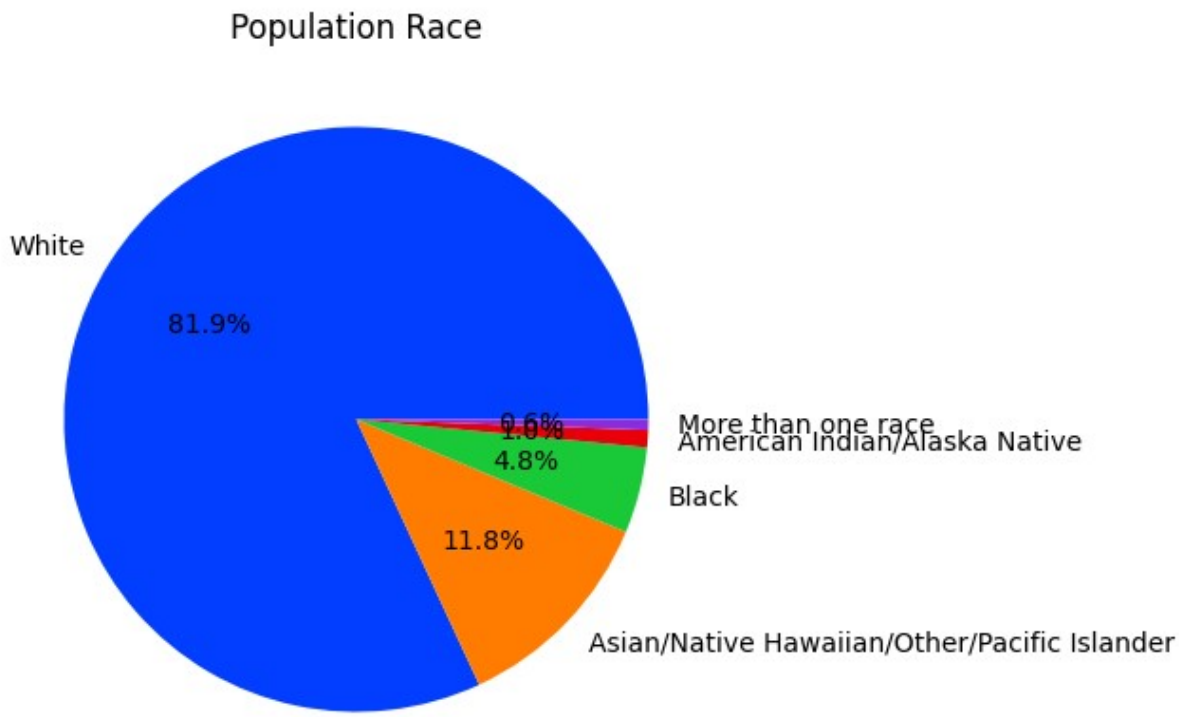
In [ ]:
```
title = "Population Race"
plt.figure(figsize=(20,5))
plt.title(title)
colors = sns.color_palette('deep')
dict = {1:'White',2:'Black',3:'American Indian/Alaska Native',
        4:'Asian/Native Hawaiian/Other/Pacific Islander', 5:'More than one race'}
dfv['race'] = [dict[row] for row in df['race']]
sns.countplot(x=dfv['race'], order = dfv['race'].value_counts(ascending=True).index
```

Out[ ]:
```
<Axes: title={'center': 'Population Race'}, xlabel='race', ylabel='count'>
```



Majority of respondents are white

In [ ]:
```
palette_color = sns.color_palette('bright')
plt.figure(figsize=(5,5))
plt.pie(dfv['race'].value_counts().tolist(), labels=dfv['race'].unique().tolist(),
plt.title(title)
plt.show()
```


Population Race
White
81.9%
0.6%
More than one race
1.0%
American Indian/Alaska Native
4.8%
Black
11.8%
Asian/Native Hawaiian/Other/Pacific Islander