In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
sns.set()

# for plots
import matplotlib.pyplot as plt
from matplotlib import cm
from matplotlib.dates import date2num
from mpl_toolkits.basemap import Basemap

# for date and time processing
import datetime

# for statistical graphs
import seaborn as sns
```

```
C:\Users\91974\anaconda3\lib\site-packages\scipy\__init__.py:146: UserWarning: A NumPy version >
=1.16.5 and <1.23.0 is required for this version of SciPy (detected version 1.23.2
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

In [2]:
```
cities=pd.read_csv('cities_r2 (1).csv')
```

In [3]:
```
cities
```

Out[3]:

	name_of_city	state_code	state_name	dist_code	population_total	population_male	population_female
0	Abohar	3	PUNJAB	9	145238	76840	68398
1	Achalpur	27	MAHARASHTRA	7	112293	58256	54037
2	Adilabad	28	ANDHRA PRADESH	1	117388	59232	58156
3	Adityapur	20	JHARKHAND	24	173988	91495	82493
4	Adoni	28	ANDHRA PRADESH	21	166537	82743	83794
...	...	...	...	...	...	...	...
488	Vizianagaram	28	ANDHRA PRADESH	12	227533	111596	115937
489	Warangal	28	ANDHRA PRADESH	9	620116	310400	309716
490	Wardha	27	MAHARASHTRA	8	105543	53241	52302
491	Yamunanagar	6	HARYANA	3	216628	115404	101224
492	Yavatmal	27	MAHARASHTRA	14	116714	58717	57997

493 rows × 22 columns

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

Out[4]:
```
(493, 22)
```

In [5]:
```
cities.info ()
# there is no null values anywhere in the dataset
```

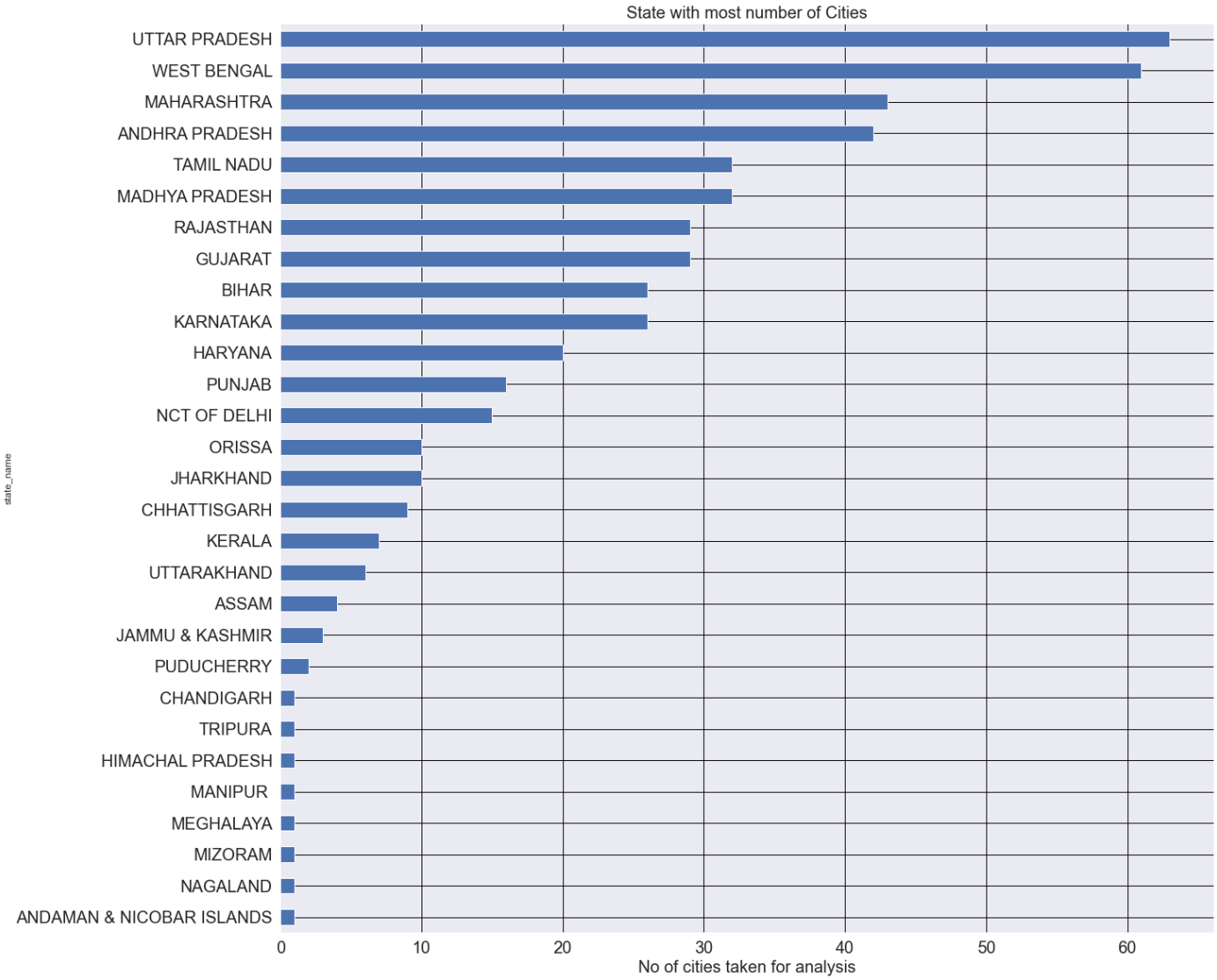
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 493 entries, 0 to 492
Data columns (total 22 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   name_of_city                    493 non-null    object
 1   state_code                      493 non-null    int64
 2   state_name                      493 non-null    object
 3   dist_code                       493 non-null    int64
 4   population_total                493 non-null    int64
 5   population_male                 493 non-null    int64
 6   population_female               493 non-null    int64
 7   0-6_population_total            493 non-null    int64
 8   0-6_population_male             493 non-null    int64
 9   0-6_population_female           493 non-null    int64
 10  literates_total                 493 non-null    int64
 11  literates_male                  493 non-null    int64
 12  literates_female                493 non-null    int64
 13  sex_ratio                       493 non-null    int64
 14  child_sex_ratio                 493 non-null    int64
 15  effective_literacy_rate_total   493 non-null    float64
 16  effective_literacy_rate_male    493 non-null    float64
 17  effective_literacy_rate_female  493 non-null    float64
 18  location                        493 non-null    object
 19  total_graduates                 493 non-null    int64
 20  male_graduates                  493 non-null    int64
 21  female_graduates                493 non-null    int64
dtypes: float64(3), int64(16), object(3)
memory usage: 84.9+ KB
```

In [6]:
```
cities.describe ()
```

Out[6]:

	state_code	dist_code	population_total	population_male	population_female	0-6_population_total	6_popula
count	493.000000	493.000000	4.930000e+02	4.930000e+02	4.930000e+02	4.930000e+02	
mean	18.643002	16.782961	4.481124e+05	2.343468e+05	2.137656e+05	4.709285e+04	248
std	9.297168	15.566131	1.033228e+06	5.487786e+05	4.848622e+05	1.050279e+05	55
min	1.000000	1.000000	1.000360e+05	5.020100e+04	4.512600e+04	6.547000e+03	34
25%	9.000000	7.000000	1.261420e+05	6.638400e+04	6.041100e+04	1.363900e+04	72
50%	19.000000	13.000000	1.841330e+05	9.665500e+04	8.776800e+04	1.944000e+04	10
75%	27.000000	21.000000	3.490330e+05	1.750550e+05	1.700260e+05	3.794500e+04	19
max	35.000000	99.000000	1.247845e+07	6.736815e+06	5.741632e+06	1.209275e+06	647

In [7]:
```
# A bar chart to show from which states, how many cities are taken for examination.
fig = plt.figure(figsize=(20,20))
states = cities.groupby('state_name')['name_of_city'].count().sort_values(ascending=True)
states.plot(kind="barh", fontsize = 20)
plt.grid(b=True, which='both', color='Black',linestyle='-')
plt.title('State with most number of Cities',fontsize = 20)
plt.xlabel('No of cities taken for analysis', fontsize = 20)
plt.show ()
# we can see states like UP and WB are given high priority by taking more than 60 cities.
```



Top 10 populous cities¶

In [8]:
```
# Extracting Co-ordinates details from the provided data
cities['latitude'] = cities['location'].apply(lambda x: x.split(',')[0])
cities['longitude'] = cities['location'].apply(lambda x: x.split(',')[1])
cities.head(1)
```

Out[8]:

	name_of_city	state_code	state_name	dist_code	population_total	population_male	population_female	6_popu
0	Abohar	3	PUNJAB	9	145238	76840	68398	

1 rows × 24 columns

The Top 10 Cities sorted according to the Total Population (Descending Order)

In [9]:
```
# A table to show top 10 cities with most population
print("The Top 10 Cities sorted according to the Total Population (Descending Order)")
top_pop_cities = cities.sort_values(by='population_total',ascending=False)
top10_pop_cities=top_pop_cities.head(10)
top10_pop_cities
```

```
The Top 10 Cities sorted according to the Total Population (Descending Order)
```

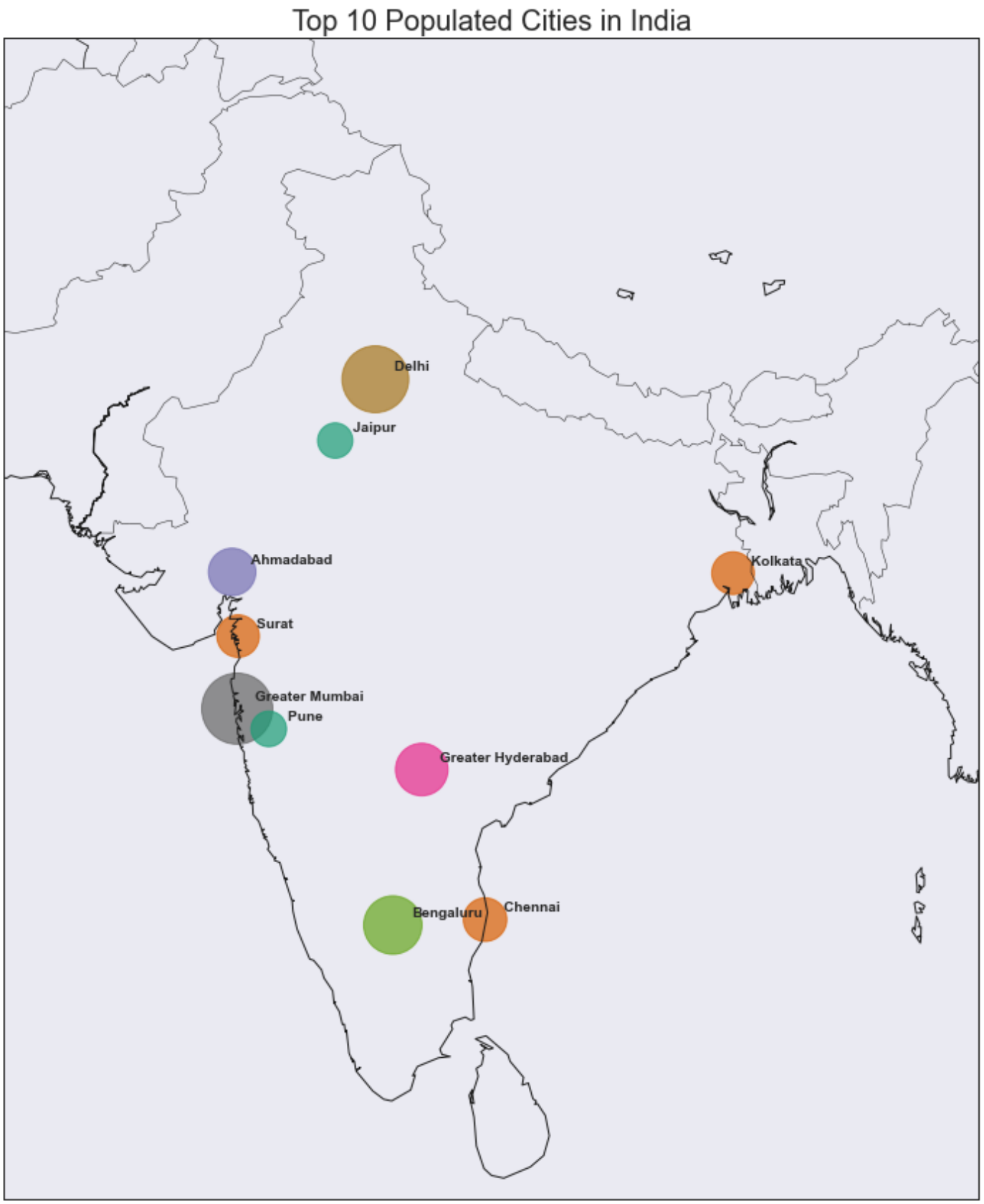
Out[9]:

	name_of_city	state_code	state_name	dist_code	population_total	population_male	population_female
185	Greater Mumbai	27	MAHARASHTRA	99	12478447	6736815	5741632
141	Delhi	7	NCT OF DELHI	99	11007835	5871362	5136473
72	Bengaluru	29	KARNATAKA	18	8425970	4401299	4024671
184	Greater Hyderabad	28	ANDHRA PRADESH	99	6809970	3500802	3309168
7	Ahmadabad	24	GUJARAT	7	5570585	2935869	2634716
119	Chennai	33	TAMIL NADU	2	4681087	2357633	2323454
274	Kolkata	19	WEST BENGAL	16	4486679	2362662	2124017
449	Surat	24	GUJARAT	25	4462002	2538243	1923759
380	Pune	27	MAHARASHTRA	25	3115431	1602137	1513294
225	Jaipur	8	RAJASTHAN	12	3073350	1619280	1454070

10 rows × 24 columns

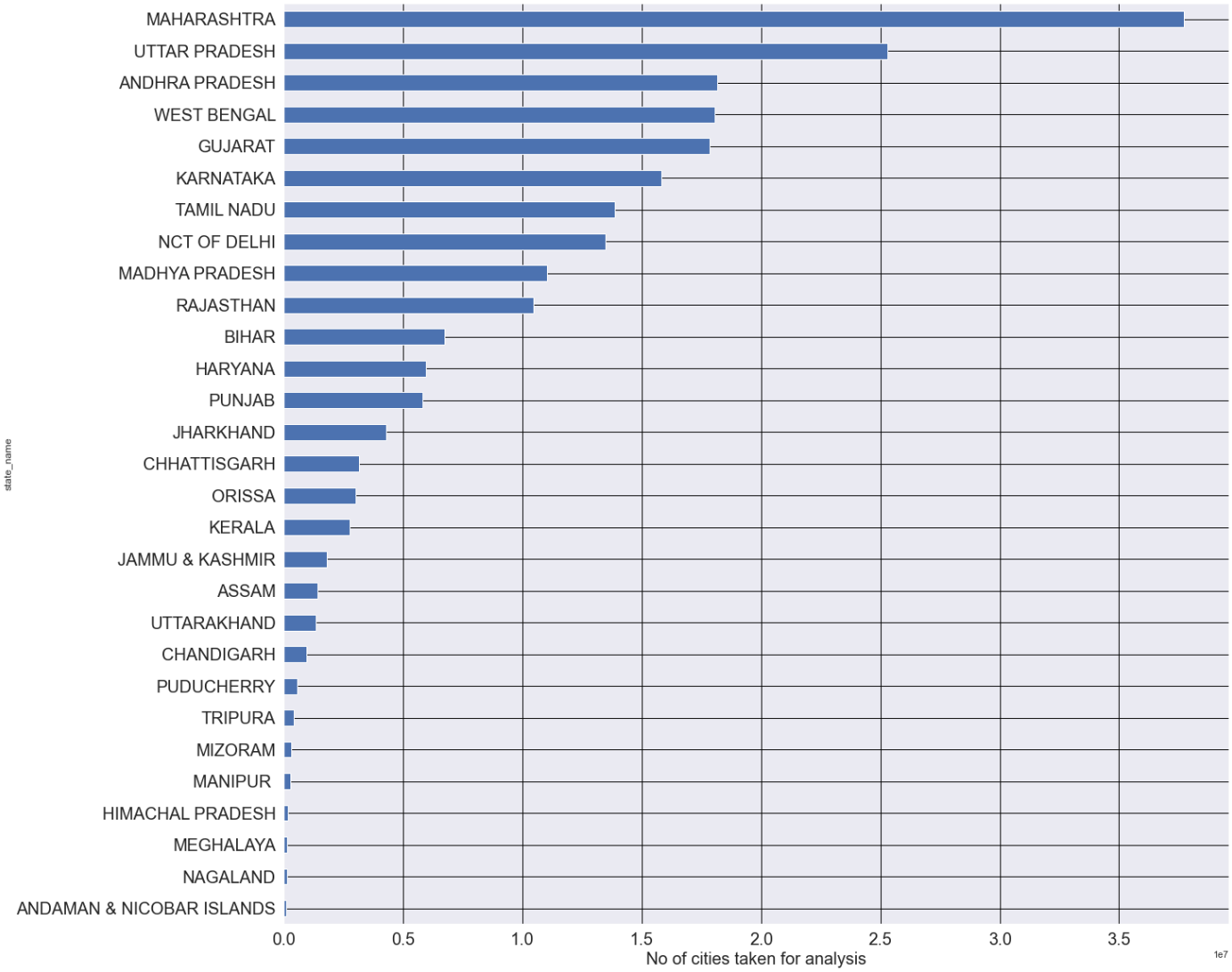
In [10]:
```
# Plotting these top 10 populous cities on India map. Circles are sized according to the
# population of the respective city

plt.subplots(figsize=(20, 15))
map = Basemap(width=1200000,height=900000,projection='lcc',resolution='l',
                    llcrnrlon=67,llcrnrlat=5,urcrnrlon=99,urcrnrlat=37,lat_0=28,lon_0=77)

map.drawmapboundary ()
map.drawcountries ()
map.drawcoastlines ()

lg=np.array(top10_pop_cities['longitude'])
lt=np.array(top10_pop_cities['latitude'])
pt=np.array(top10_pop_cities['population_total'])
nc=np.array(top10_pop_cities['name_of_city'])

x, y = map(lg, lt)
population_sizes = top10_pop_cities["population_total"].apply(lambda x: int(x / 5000))
plt.scatter(x, y, s=population_sizes, marker="o", c=population_sizes, cmap=cm.Dark2, alpha=0.7)


for ncs, xpt, ypt in zip(nc, x, y):
    plt.text(xpt+60000, ypt+30000, ncs, fontsize=10, fontweight='bold')

plt.title('Top 10 Populated Cities in India',fontsize=20)
```

Out[10]:
```
Text(0.5, 1.0, 'Top 10 Populated Cities in India')
```



Plotting Statewise cities to check which state have most population living in urban areas

In [11]:
```
# A bar chart to show the population of the states
fig = plt.figure(figsize=(20,20))
states = cities.groupby('state_name')['population_total'].sum().sort_values(ascending=True)
states.plot(kind="barh", fontsize = 20)
plt.grid(b=True, which='both', color='Black',linestyle='-')
plt.xlabel('No of cities taken for analysis', fontsize = 20)
plt.show ()
# we can see states like Maharashtra and UP have huge urban population
```



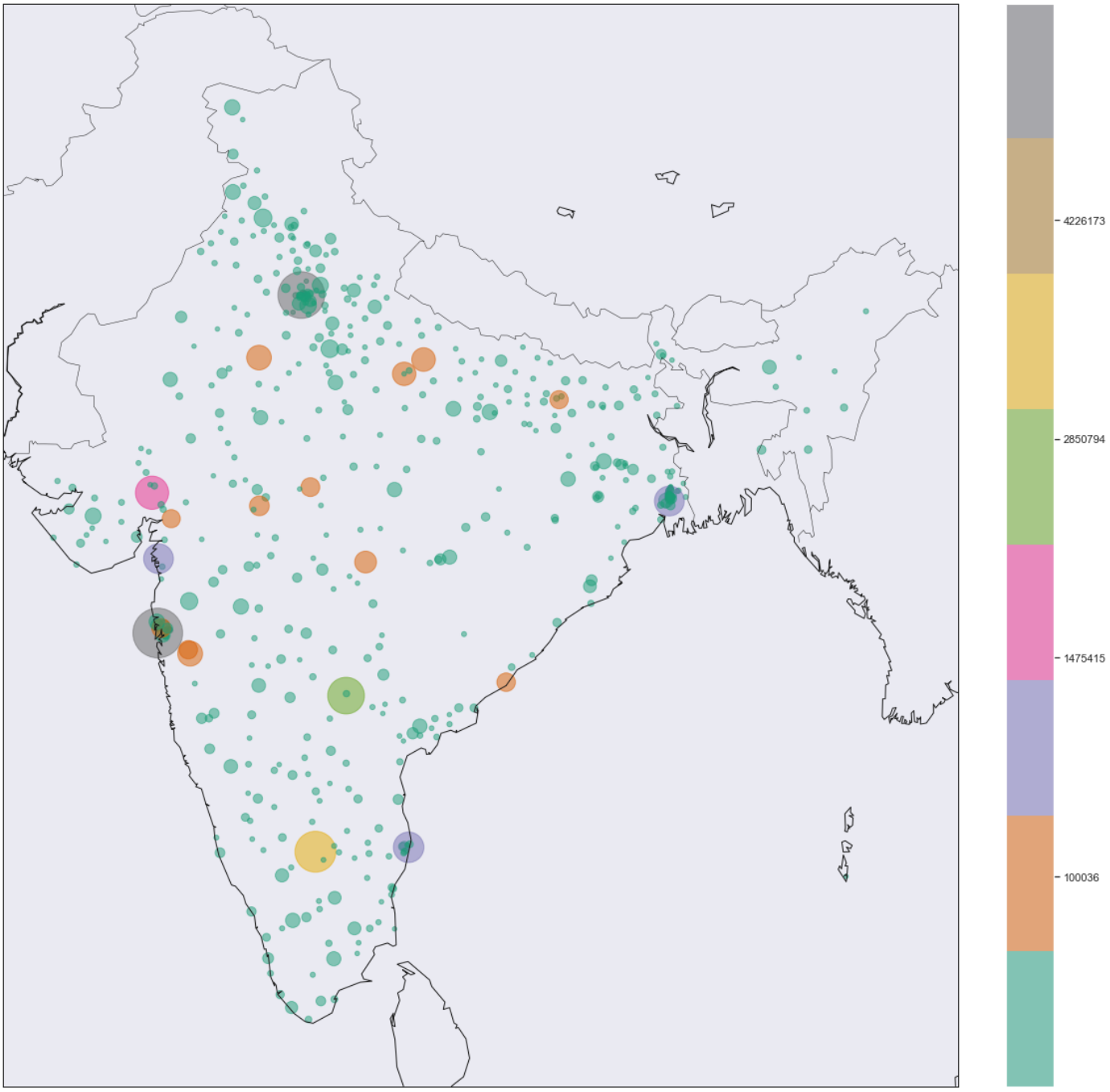
Plotting every city on India map according to population

In [12]:
```
# Creating a function to plot the population data on real India map

def plot_map(sizes, colorbarValue):

    plt.figure(figsize=(19,20))
    f, ax = plt.subplots(figsize=(19, 20))

    # Setting up Basemap
    map = Basemap(width=5000000, height=3500000, resolution='l', projection='aea', llcrnrlon=69,
                  llcrnrlat=6, urcrnrlon=99, urcrnrlat=36, lon_0=78, lat_0=20, ax=ax)

    # draw map boundaries
    map.drawmapboundary()
    map.drawcountries()
    map.drawcoastlines()

    # plotting cities on map using previously derived coordinates
    x, y = map(np.array(cities["longitude"]),np.array(cities["latitude"]))
    cs = map.scatter(x, y, s=sizes, marker="o", c=sizes, cmap=cm.Dark2, alpha=0.5)

    # adding colorbar
    cbar = map.colorbar(cs, location='right',pad="5%")
    cbar.ax.set_yticklabels(colorbarValue)

    plt.show()
```

In [13]:
```
# Using the function created in the previous cell, we are plotting the population data

population_sizes = cities["population_total"].apply(lambda x: int(x / 5000))
colorbarValue = np.linspace(cities["population_total"].min(), cities["population_total"].max(),
                            num=10)
colorbarValue = colorbarValue.astype(int)

plot_map(population_sizes, colorbarValue)
```

```
C:\Users\91974\AppData\Local\Temp/ipykernel_67156/707578470.py:23: UserWarning: FixedFormatter s
hould only be used together with FixedLocator
  cbar.ax.set_yticklabels(colorbarValue)
```

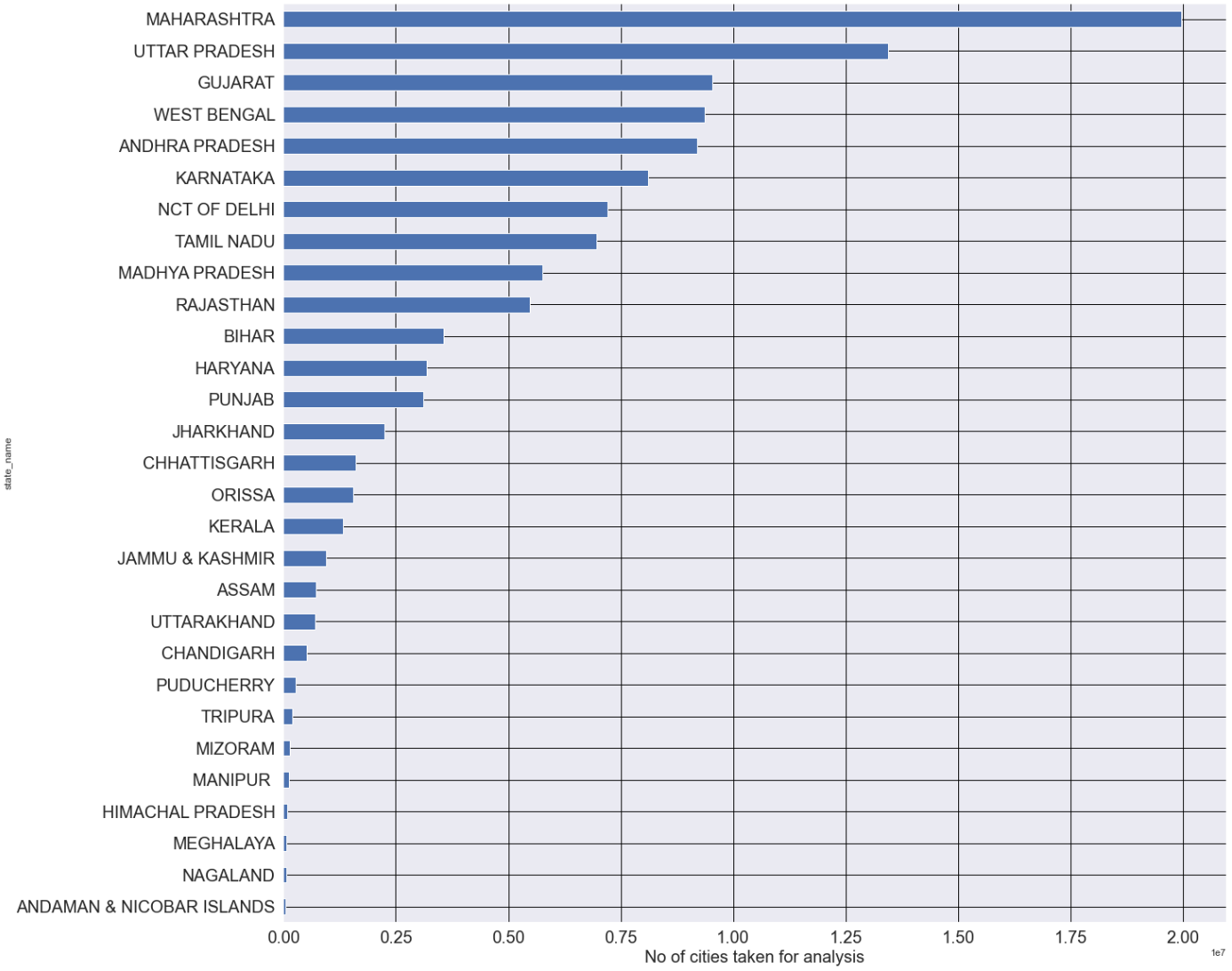
```
<Figure size 1368x1440 with 0 Axes>
```



plotting Statewise cities to check which state have most male population¶

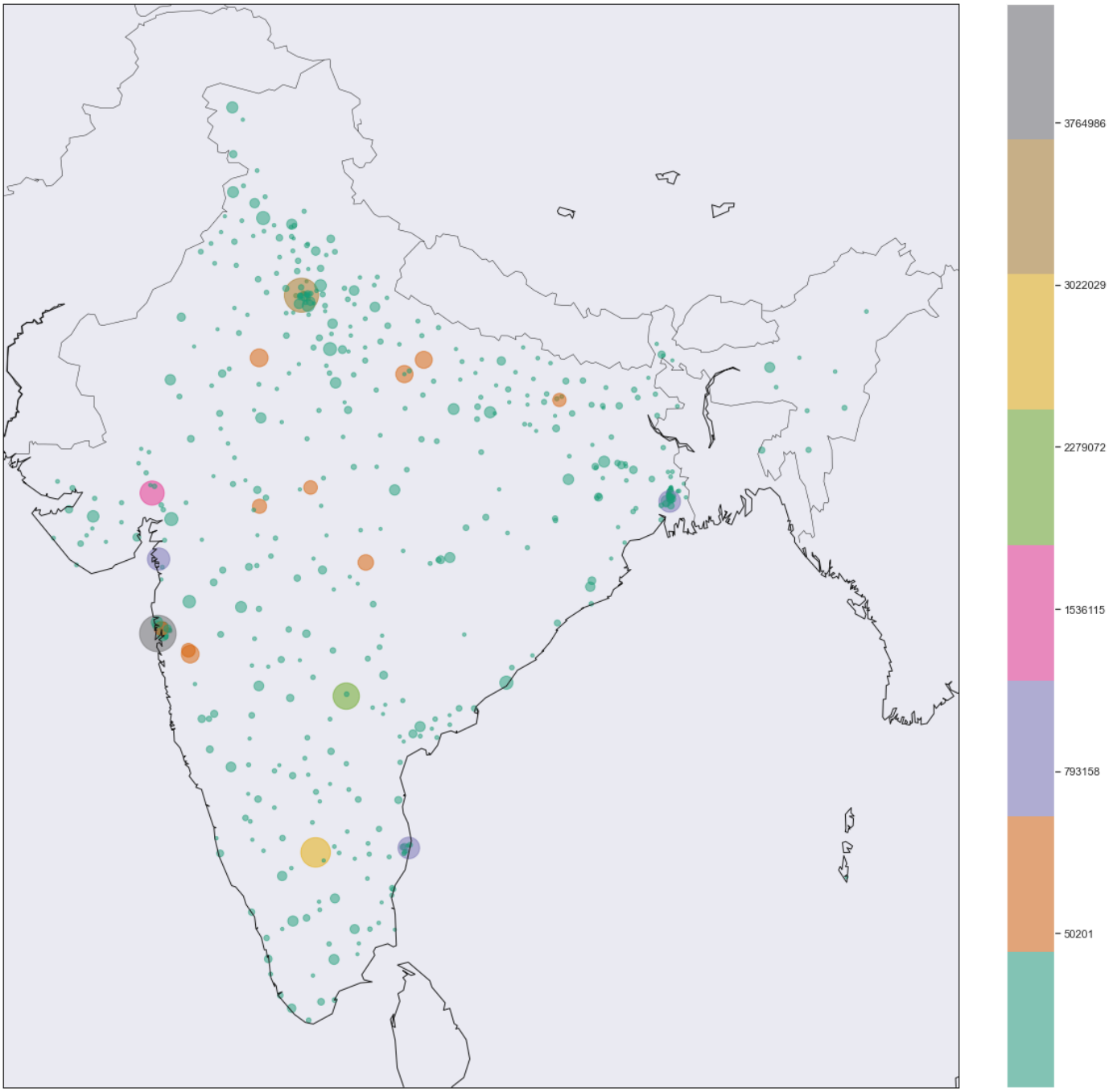
In [14]:
```
# A bar chart to show the male population of the states
fig = plt.figure(figsize=(20,20))
states = cities.groupby('state_name')['population_male'].sum().sort_values(ascending=True)
states.plot(kind="barh", fontsize = 20)
plt.grid(b=True, which='both', color='Black',linestyle='-')
```

```
plt.xlabel('No of cities taken for analysis', fontsize = 20)
plt.show ()
# we can see states like Maharashtra and UP have huge male population
```



In [15]:
```
# Plotting the same on the map
population_sizes = cities["population_male"].apply(lambda x: int(x / 5000))
colorbarValue = np.linspace(cities["population_male"].min(), cities["population_male"].max(),
                            num=10)
colorbarValue = colorbarValue.astype(int)

plot_map(population_sizes, colorbarValue)
```

```
C:\Users\91974\AppData\Local\Temp/ipykernel_67156/707578470.py:23: UserWarning: FixedFormatter s
hould only be used together with FixedLocator
  cbar.ax.set_yticklabels(colorbarValue)
<Figure size 1368x1440 with 0 Axes>
```



These are the top 10 cities with high male population

In [16]:
```
# A table to show top 10 cities with most male population
print("The Top 10 Cities sorted according to the Total Male Population (Descending Order)")
top_male_cities = cities.sort_values(by='population_male',ascending=False)
top10_male_pop_cities=top_male_cities.head(10)
top10_male_pop_cities
```

```
The Top 10 Cities sorted according to the Total Male Population (Descending Order)
```

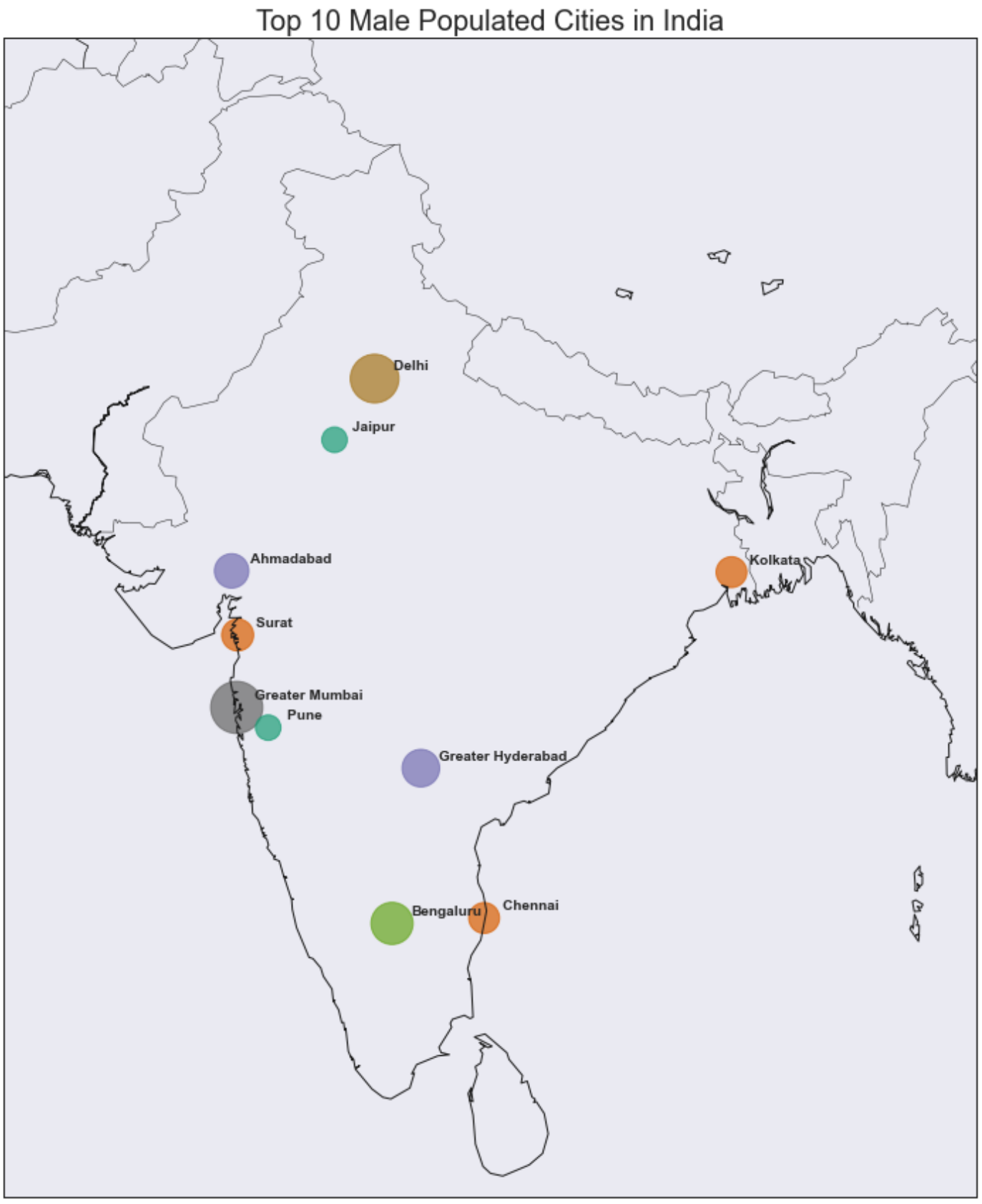
Out[16]:

	name_of_city	state_code	state_name	dist_code	population_total	population_male	population_female
185	Greater Mumbai	27	MAHARASHTRA	99	12478447	6736815	5741632
141	Delhi	7	NCT OF DELHI	99	11007835	5871362	5136473
72	Bengaluru	29	KARNATAKA	18	8425970	4401299	4024671
184	Greater Hyderabad	28	ANDHRA PRADESH	99	6809970	3500802	3309168
7	Ahmadabad	24	GUJARAT	7	5570585	2935869	2634716
449	Surat	24	GUJARAT	25	4462002	2538243	1923759
274	Kolkata	19	WEST BENGAL	16	4486679	2362662	2124017
119	Chennai	33	TAMIL NADU	2	4681087	2357633	2323454
225	Jaipur	8	RAJASTHAN	12	3073350	1619280	1454070
380	Pune	27	MAHARASHTRA	25	3115431	1602137	1513294

10 rows × 24 columns

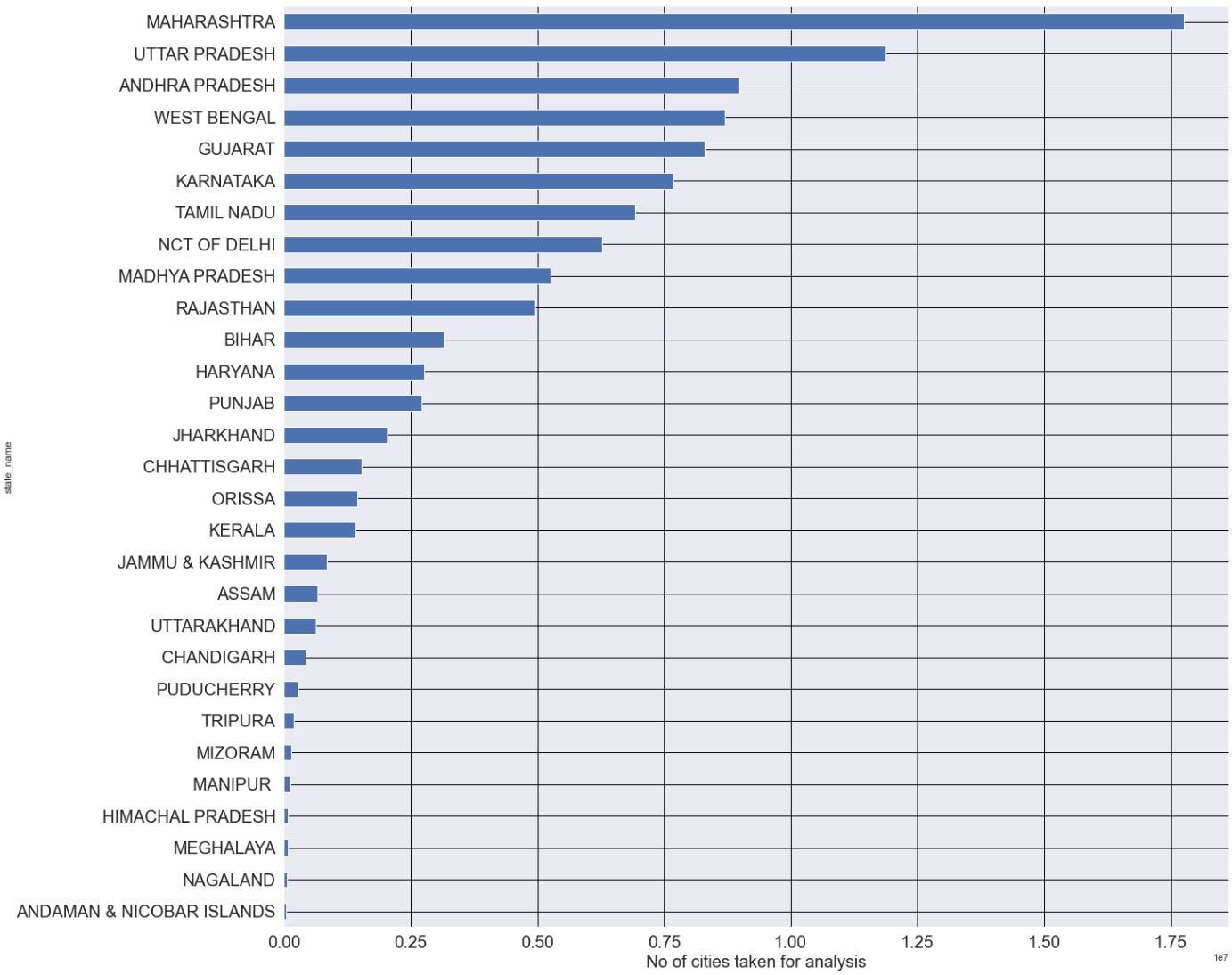
In [17]:
```
# Plotting these top 10 male populous cities on India map. Circles are sized according to the
# male population of the respective city

plt.subplots(figsize=(20, 15))
map = Basemap(width=1200000,height=900000,projection='lcc',resolution='l',
                    llcrnrlon=67,llcrnrlat=5,urcrnrlon=99,urcrnrlat=37,lat_0=28,lon_0=77)

map.drawmapboundary ()
map.drawcountries ()
map.drawcoastlines ()

lg=np.array(top10_male_pop_cities['longitude'])
lt=np.array(top10_male_pop_cities['latitude'])
pt=np.array(top10_male_pop_cities['population_male'])
nc=np.array(top10_male_pop_cities['name_of_city'])

x, y = map(lg, lt)
population_sizes_male = top10_male_pop_cities["population_male"].apply(lambda x: int(x / 5000))
plt.scatter(x, y, s=population_sizes_male, marker="o", c=population_sizes_male, cmap=cm.Dark2, a

for ncs, xpt, ypt in zip(nc, x, y):
    plt.text(xpt+60000, ypt+30000, ncs, fontsize=10, fontweight='bold')

plt.title('Top 10 Male Populated Cities in India',fontsize=20)
```

Out[17]:
```
Text(0.5, 1.0, 'Top 10 Male Populated Cities in India')
```



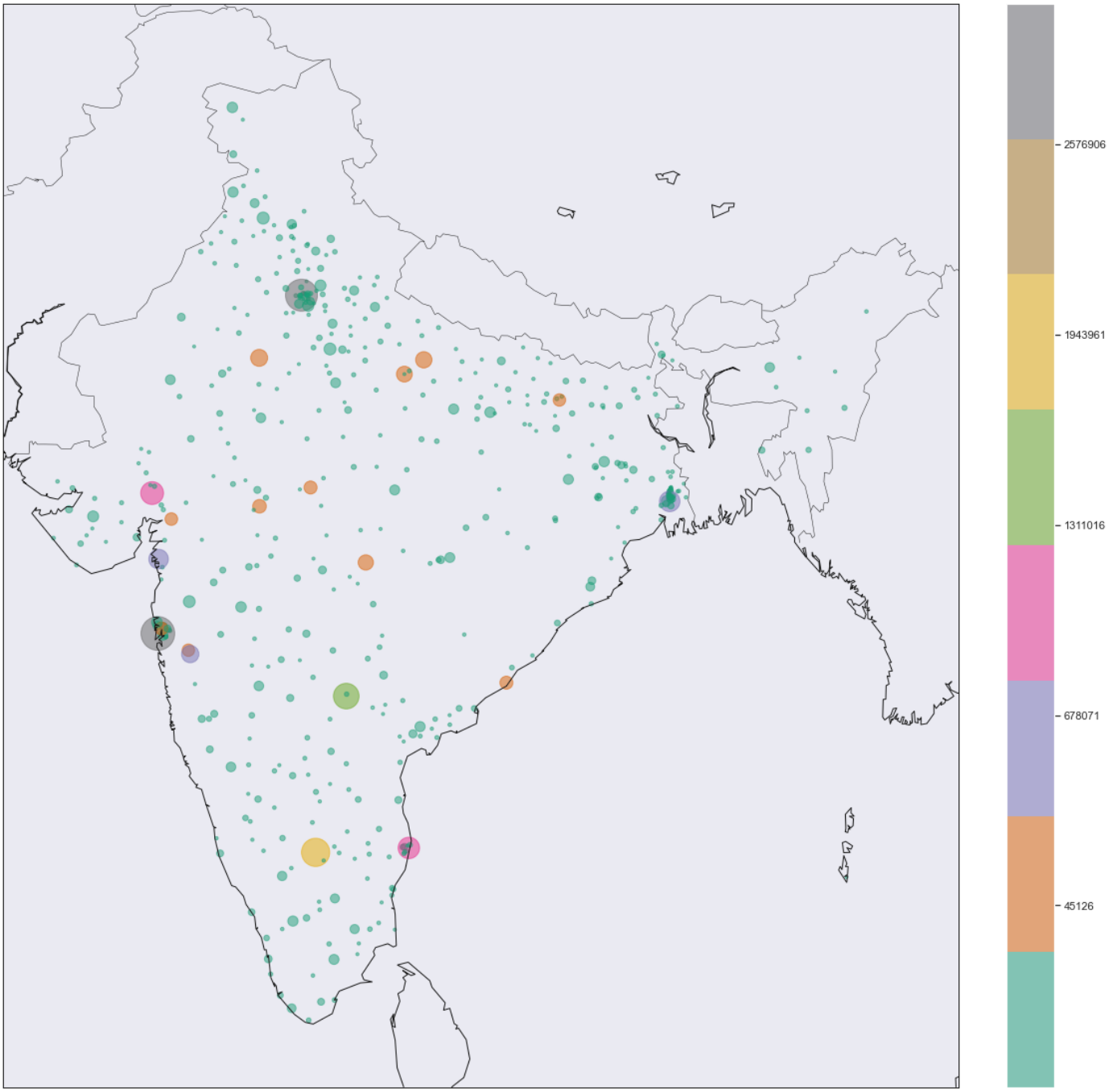
Plotting Statewise cities to check which state have most female population

In [18]:
```
# A bar chart to show the female population of the states
fig = plt.figure(figsize=(20,20))
states = cities.groupby('state_name')['population_female'].sum().sort_values(ascending=True)
states.plot(kind="barh", fontsize = 20)
plt.grid(b=True, which='both', color='Black',linestyle='-')
plt.xlabel('No of cities taken for analysis', fontsize = 20)
plt.show ()
# we can see again states like Maharashtra and UP have huge female population
```



In [19]:
```
# Plotting the same on the map
population_sizes = cities["population_female"].apply(lambda x: int(x / 5000))
colorbarValue = np.linspace(cities["population_female"].min(), cities["population_female"].max()
                            num=10)
colorbarValue = colorbarValue.astype(int)

plot_map(population_sizes, colorbarValue)
```

```
C:\Users\91974\AppData\Local\Temp/ipykernel_67156/707578470.py:23: UserWarning: FixedFormatter s
hould only be used together with FixedLocator
  cbar.ax.set_yticklabels(colorbarValue)
<Figure size 1368x1440 with 0 Axes>
```



These are the top 10 cities with high female population¶

In [20]:
```
# A table to show top 10 cities with most female population
print("The Top 10 Cities sorted according to the Total Female Population (Descending Order)")
top_female_cities = cities.sort_values(by='population_female',ascending=False)
top10_female_pop_cities=top_female_cities.head(10)
top10_female_pop_cities
```

```
The Top 10 Cities sorted according to the Total Female Population (Descending Order)
```

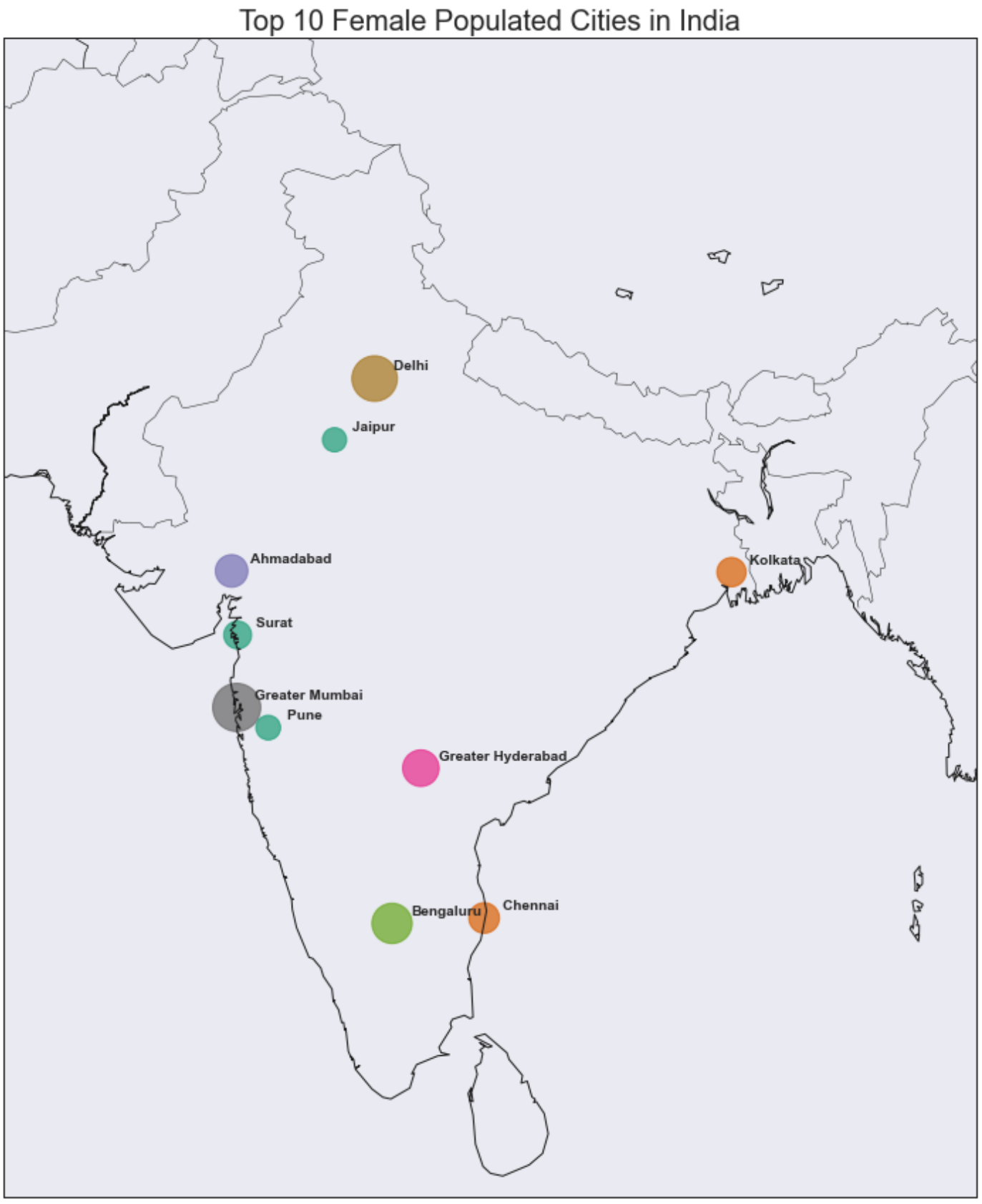
Out[20]:

	name_of_city	state_code	state_name	dist_code	population_total	population_male	population_female
185	Greater Mumbai	27	MAHARASHTRA	99	12478447	6736815	5741632
141	Delhi	7	NCT OF DELHI	99	11007835	5871362	5136473
72	Bengaluru	29	KARNATAKA	18	8425970	4401299	4024671
184	Greater Hyderabad	28	ANDHRA PRADESH	99	6809970	3500802	3309168
7	Ahmadabad	24	GUJARAT	7	5570585	2935869	2634716
119	Chennai	33	TAMIL NADU	2	4681087	2357633	2323454
274	Kolkata	19	WEST BENGAL	16	4486679	2362662	2124017
449	Surat	24	GUJARAT	25	4462002	2538243	1923759
380	Pune	27	MAHARASHTRA	25	3115431	1602137	1513294
225	Jaipur	8	RAJASTHAN	12	3073350	1619280	1454070

10 rows × 24 columns

In [21]:
```
# Plotting these top 10 female populous cities on India map. Circles are sized according to the
# female population of the respective city

plt.subplots(figsize=(20, 15))
map = Basemap(width=1200000,height=900000,projection='lcc',resolution='l',
                    llcrnrlon=67,llcrnrlat=5,urcrnrlon=99,urcrnrlat=37,lat_0=28,lon_0=77)

map.drawmapboundary ()
map.drawcountries ()
map.drawcoastlines ()

lg=np.array(top10_female_pop_cities['longitude'])
lt=np.array(top10_female_pop_cities['latitude'])
pt=np.array(top10_female_pop_cities['population_female'])
nc=np.array(top10_female_pop_cities['name_of_city'])

x, y = map(lg, lt)
population_sizes_female = top10_female_pop_cities["population_female"].apply(lambda x: int(x / 5
plt.scatter(x, y, s=population_sizes_female, marker="o", c=population_sizes_female, cmap=cm.Dark


for ncs, xpt, ypt in zip(nc, x, y):
    plt.text(xpt+60000, ypt+30000, ncs, fontsize=10, fontweight='bold')

plt.title('Top 10 Female Populated Cities in India',fontsize=20)
```

Out[21]:
```
Text(0.5, 1.0, 'Top 10 Female Populated Cities in India')
```



Top 10 cities where most of the kids live

In [22]:
```
# Lets find the top ten cities in which large number of kids live
print("The Top 10 Cities sorted according to the Total Kids Population (Descending Order)")
top_kids_cities = cities.sort_values(by='0-6_population_total',ascending=False)
top10_kids_pop_cities=top_kids_cities.head(10)
top10_kids_pop_cities
```

```
The Top 10 Cities sorted according to the Total Kids Population (Descending Order)
```

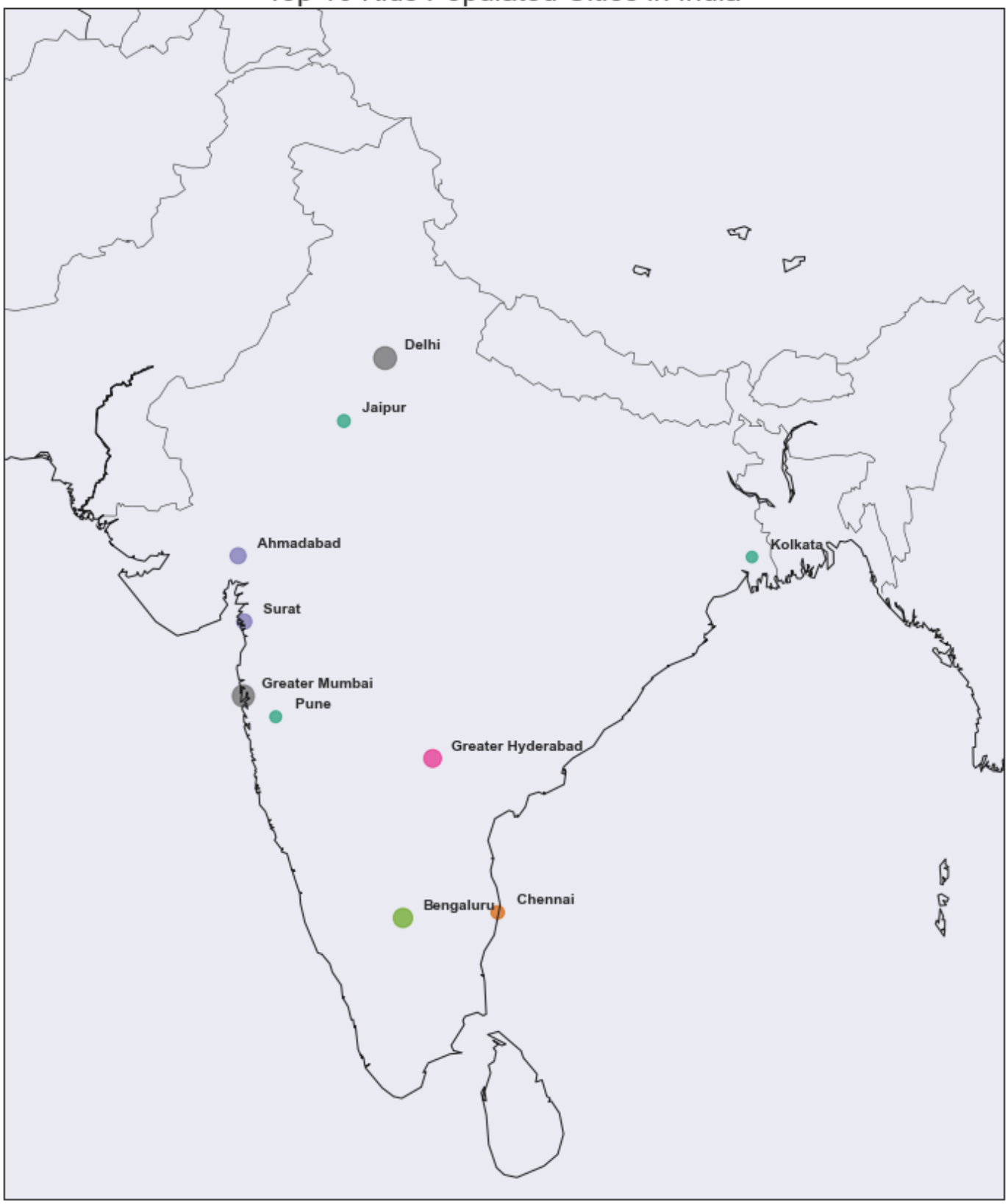
Out[22]:

	name_of_city	state_code	state_name	dist_code	population_total	population_male	population_female
141	Delhi	7	NCT OF DELHI	99	11007835	5871362	5136473
185	Greater Mumbai	27	MAHARASHTRA	99	12478447	6736815	5741632
72	Bengaluru	29	KARNATAKA	18	8425970	4401299	4024671
184	Greater Hyderabad	28	ANDHRA PRADESH	99	6809970	3500802	3309168
7	Ahmadabad	24	GUJARAT	7	5570585	2935869	2634716
449	Surat	24	GUJARAT	25	4462002	2538243	1923759
119	Chennai	33	TAMIL NADU	2	4681087	2357633	2323454
225	Jaipur	8	RAJASTHAN	12	3073350	1619280	1454070
380	Pune	27	MAHARASHTRA	25	3115431	1602137	1513294
274	Kolkata	19	WEST BENGAL	16	4486679	2362662	2124017

10 rows × 24 columns

In [23]:
```
# Lets find the top ten cities in which large number of kids live

plt.subplots(figsize=(20, 15))
map = Basemap(width=1200000,height=900000,projection='lcc',resolution='l',
                    llcrnrlon=67,llcrnrlat=5,urcrnrlon=99,urcrnrlat=37,lat_0=28,lon_0=77)

map.drawmapboundary ()
map.drawcountries ()
map.drawcoastlines ()

lg=np.array(top10_kids_pop_cities['longitude'])
lt=np.array(top10_kids_pop_cities['latitude'])
pt=np.array(top10_kids_pop_cities['0-6_population_total'])
nc=np.array(top10_kids_pop_cities['name_of_city'])

x, y = map(lg, lt)
population_sizes_kids = top10_kids_pop_cities["0-6_population_total"].apply(lambda x: int(x / 50
plt.scatter(x, y, s=population_sizes_kids, marker="o", c=population_sizes_kids, cmap=cm.Dark2, a


for ncs, xpt, ypt in zip(nc, x, y):
    plt.text(xpt+60000, ypt+30000, ncs, fontsize=10, fontweight='bold')

plt.title('Top 10 Kids Populated Cities in India',fontsize=20)
```

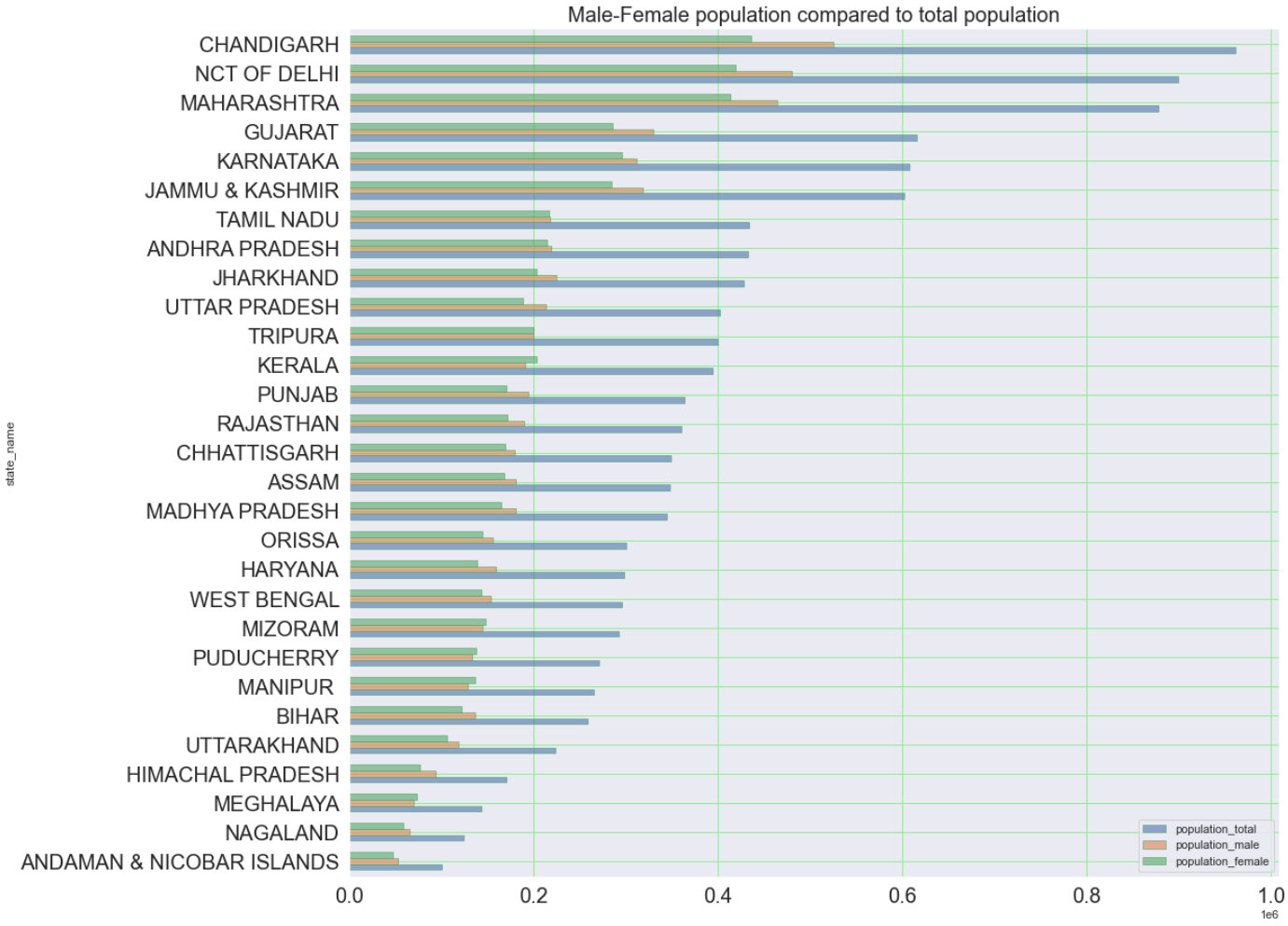
Out[23]:
```
Text(0.5, 1.0, 'Top 10 Kids Populated Cities in India')
```



In [24]:
```
cities.columns
```

Out[24]:
```
Index(['name_of_city', 'state_code', 'state_name', 'dist_code',
       'population_total', 'population_male', 'population_female',
       '0-6_population_total', '0-6_population_male', '0-6_population_female',
       'literates_total', 'literates_male', 'literates_female', 'sex_ratio',
       'child_sex_ratio', 'effective_literacy_rate_total',
       'effective_literacy_rate_male', 'effective_literacy_rate_female',
       'location', 'total_graduates', 'male_graduates', 'female_graduates',
       'latitude', 'longitude'],
      dtype='object')
```

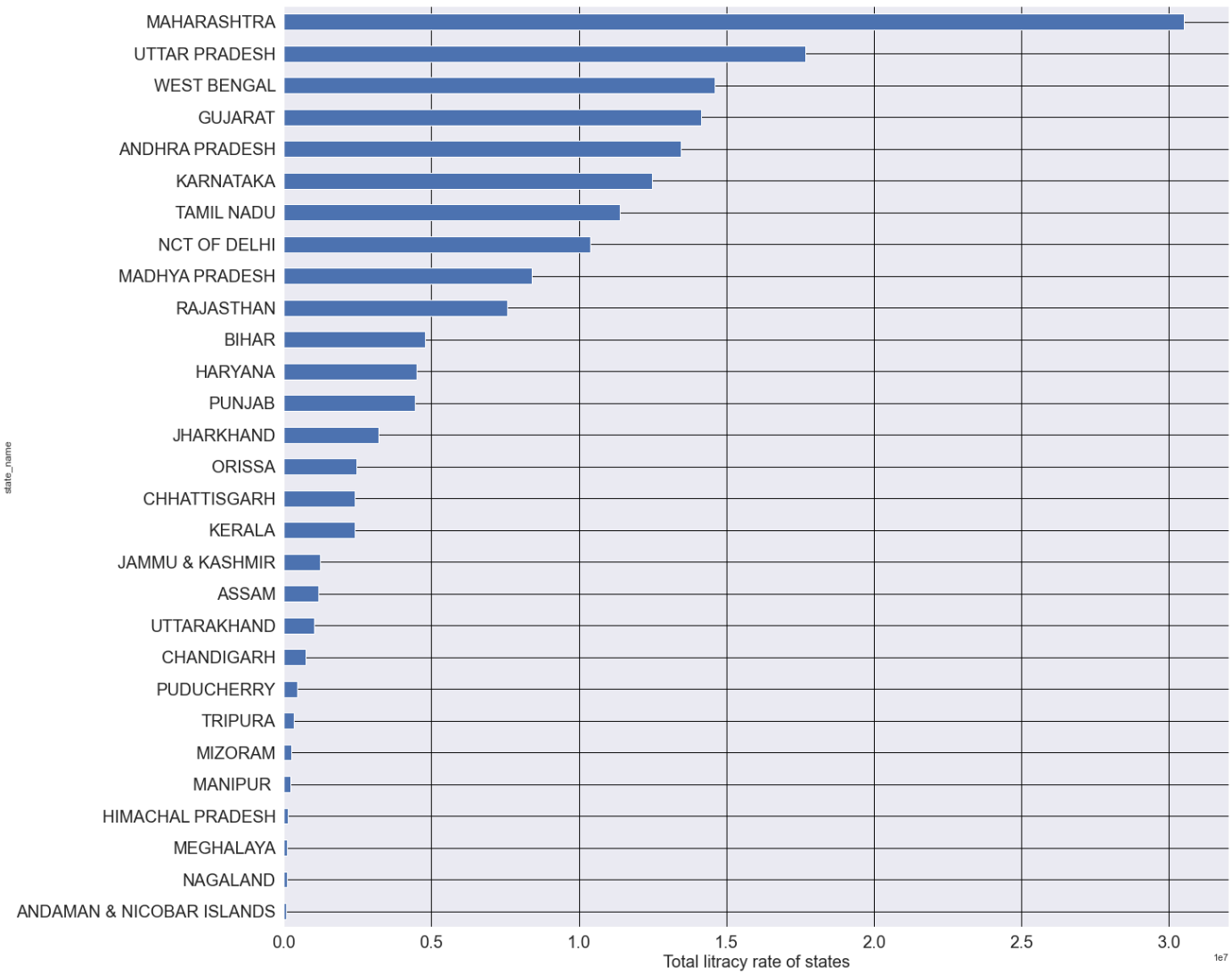
In [33]:
```
state_literacy_effective = cities[["state_name","population_total", "population_male", "populatio


state_literacy_effective.sort_values("population_total", ascending=True).plot(kind="barh",
                        grid=True,
                        figsize=(16,15),
                        alpha = 0.6,
                        width=0.6,
                        stacked = False,
                        edgecolor="g",
                        fontsize = 20)
plt.title("Male-Female population compared to total population",fontsize=20)
plt.grid(b=True, which='both', color='lightGreen',linestyle='-')
plt.show ()
```



In [ ]:
```
```

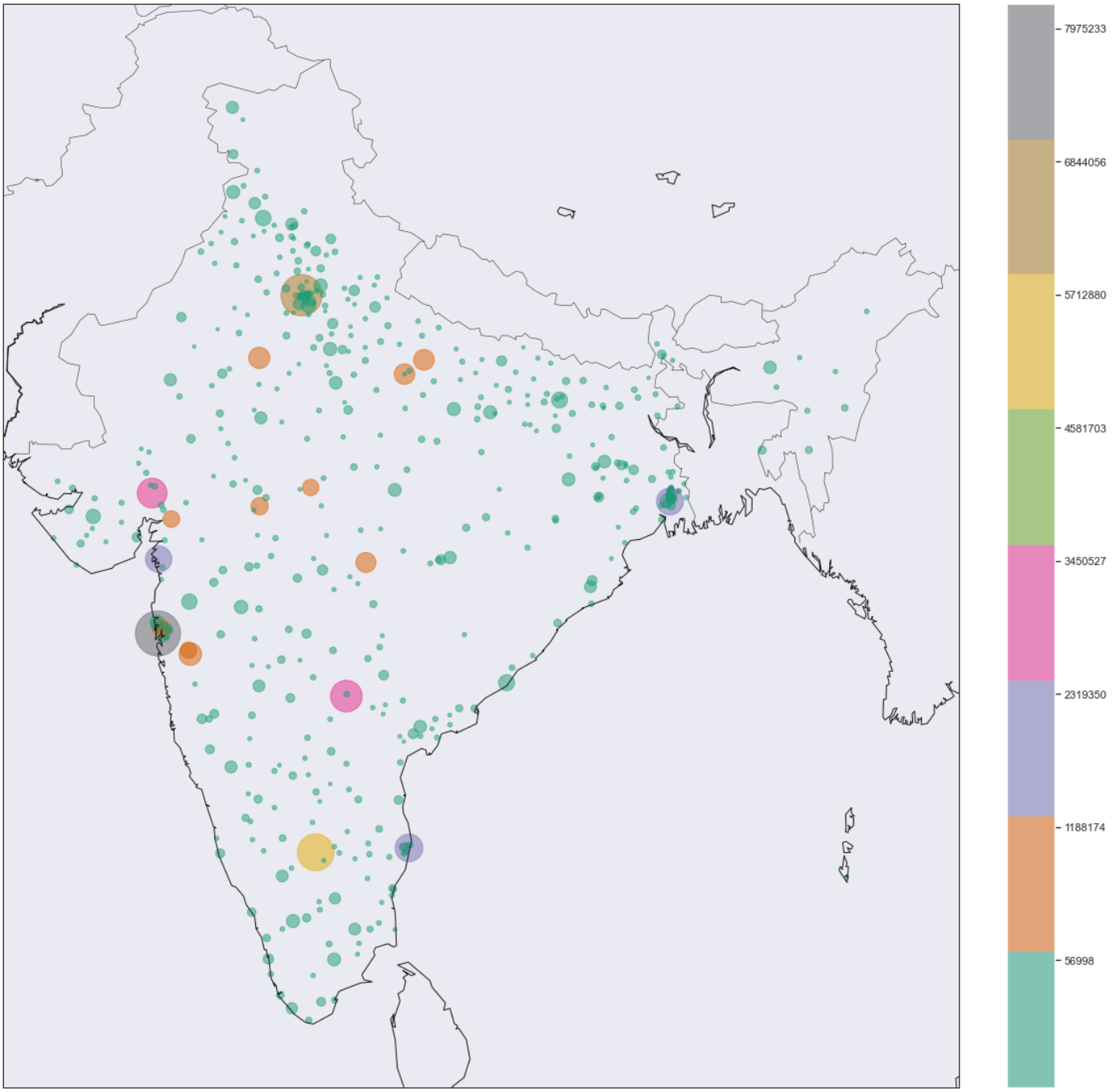
Analysing Litracy rate of the states

In [26]:
```
# A bar chart to show the total literates of the states
fig = plt.figure(figsize=(20,20))
states = cities.groupby('state_name')['literates_total'].sum().sort_values(ascending=True)
states.plot(kind="barh", fontsize = 20)
plt.grid(b=True, which='both', color='Black',linestyle='-')
plt.xlabel('Total litracy rate of states', fontsize = 20)
plt.show ()
# we can see again states like Maharashtra and UP have huge litrate population living in cities
```



In [27]:
```
# Plotting the same on the map
population_sizes = cities["literates_total"].apply(lambda x: int(x / 5000))
colorbarValue = np.linspace(cities["literates_total"].min(), cities["literates_total"].max(),
                            num=10)
colorbarValue = colorbarValue.astype(int)

plot_map(population_sizes, colorbarValue)
# Major metro cities again shows higher litracy rates
```

```
C:\Users\91974\AppData\Local\Temp/ipykernel_67156/707578470.py:23: UserWarning: FixedFormatter s
hould only be used together with FixedLocator
  cbar.ax.set_yticklabels(colorbarValue)
<Figure size 1368x1440 with 0 Axes>
```



Top 10 cities where most of the literates live

In [28]:
```
# Lets find the top ten cities in which large number of literates live
print("The Top 10 Cities sorted according to the Total litrate Population (Descending Order)")
top10_literate_cities = cities.sort_values(by='literates_total',ascending=False)
top10_literate_cities=top10_literate_cities.head(10)
top10_literate_cities
```

```
The Top 10 Cities sorted according to the Total litrate Population (Descending Order)
```

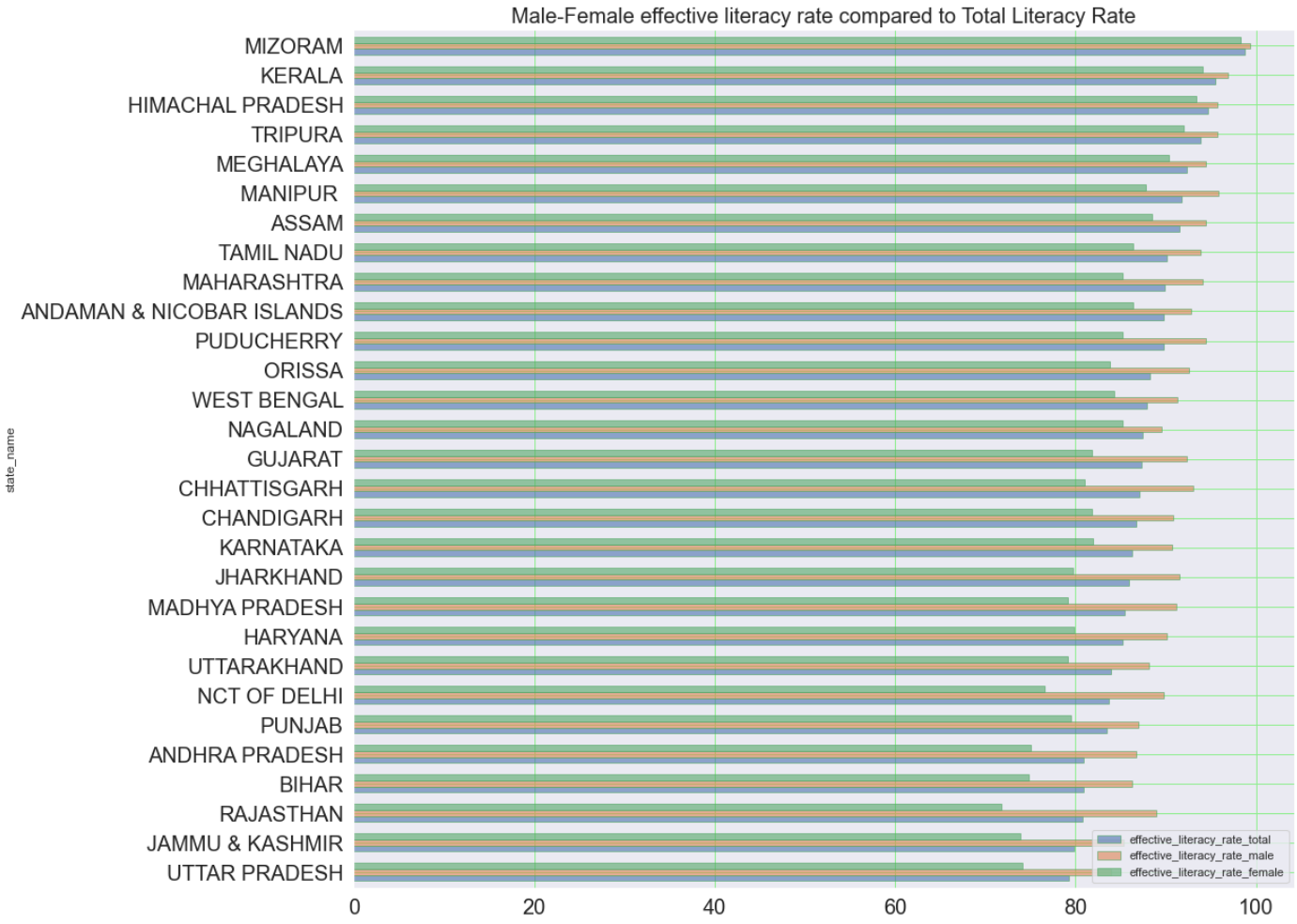
Out[28]:

	name_of_city	state_code	state_name	dist_code	population_total	population_male	population_female
185	Greater Mumbai	27	MAHARASHTRA	99	12478447	6736815	5741632
141	Delhi	7	NCT OF DELHI	99	11007835	5871362	5136473
72	Bengaluru	29	KARNATAKA	18	8425970	4401299	4024671
184	Greater Hyderabad	28	ANDHRA PRADESH	99	6809970	3500802	3309168
7	Ahmadabad	24	GUJARAT	7	5570585	2935869	2634716
119	Chennai	33	TAMIL NADU	2	4681087	2357633	2323454
274	Kolkata	19	WEST BENGAL	16	4486679	2362662	2124017
449	Surat	24	GUJARAT	25	4462002	2538243	1923759
380	Pune	27	MAHARASHTRA	25	3115431	1602137	1513294
225	Jaipur	8	RAJASTHAN	12	3073350	1619280	1454070

10 rows × 24 columns

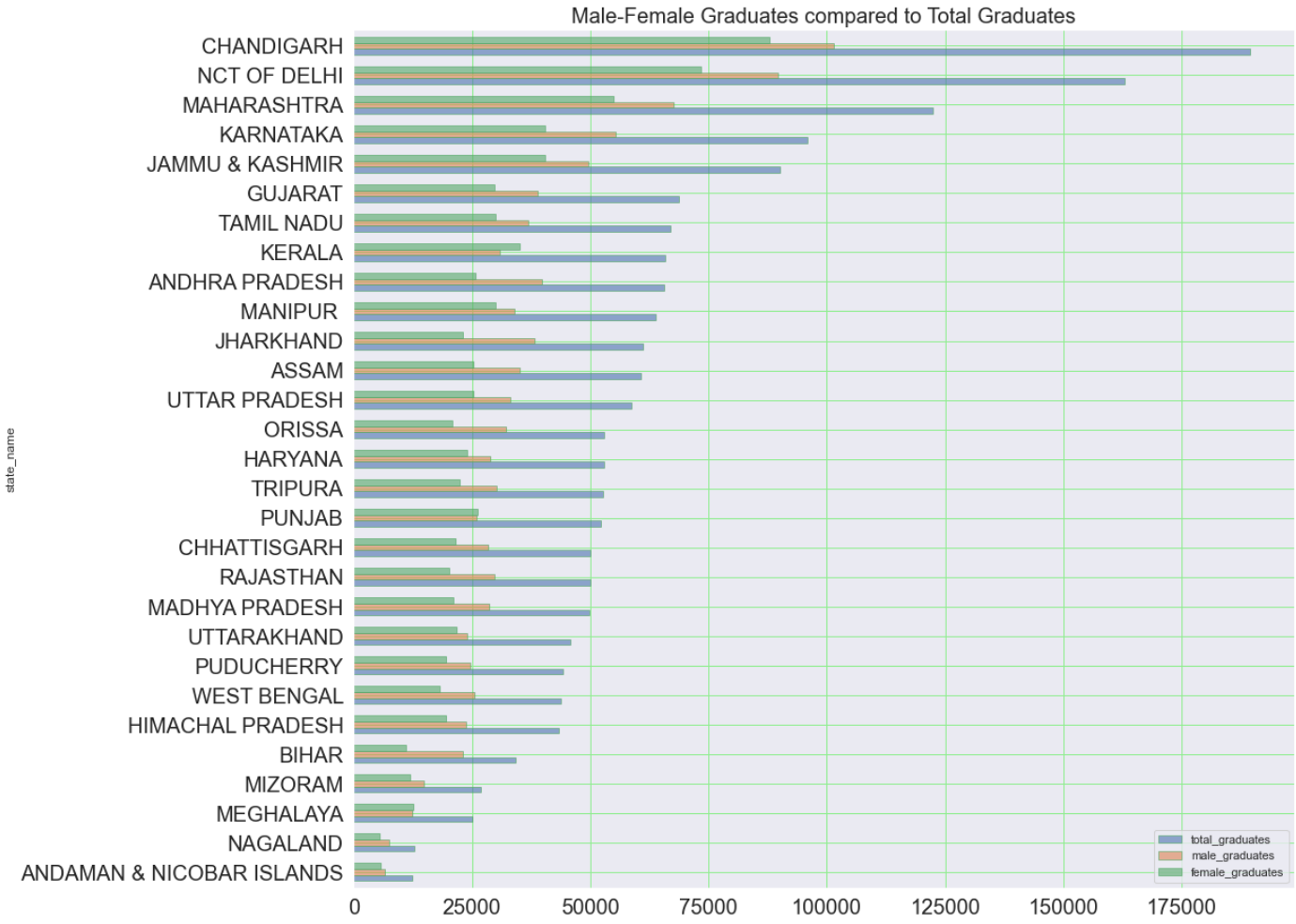
Analyzing effective literacy rate

In [36]:
```
# seperating effective literacy rate from the main dataset and sorting then in descending order
state_literacy_effective = cities[["state_name","effective_literacy_rate_total","effective_litera



state_literacy_effective.sort_values("effective_literacy_rate_total", ascending=True).plot(kind='
                        grid=True,
                        figsize=(16,15),
                        alpha = 0.6,
                        width=0.6,
                        stacked = False,
                        edgecolor="g",
                        fontsize = 20)
plt.title("Male-Female effective literacy rate compared to Total Literacy Rate",fontsize=20)
plt.grid(b=True, which='both', color='lightGreen',linestyle='-')
plt.show ()
# from the below chart, Mizoram, Kerala and HP have highest effective literacy rate across India
```



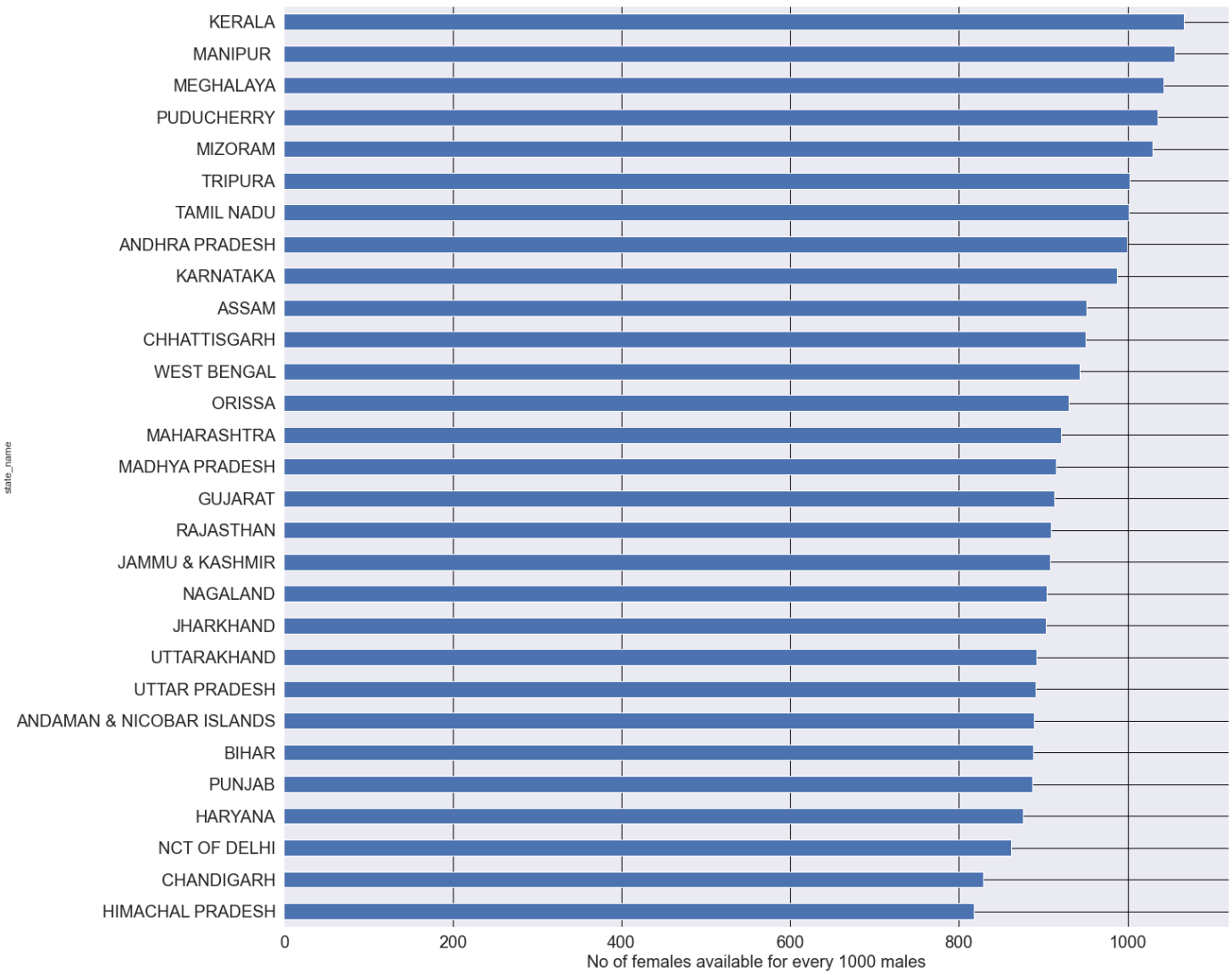
Analyzing graduates

In [38]:
```
# seperating Graduates from the main dataset and sorting then in descending order
state_graduates  = cities[["state_name",
                            "total_graduates",
                            "male_graduates",
                            "female_graduates"]].groupby("state_name").agg({"total_graduate
                                                                            "male_graduates
                                                                            "female_graduat
# Plotting the bar chart
state_graduates.sort_values("total_graduates", ascending=True).plot(kind="barh",
                      grid=True,
                      figsize=(16,15),
                      alpha = 0.6,
                      width=0.6,
                      stacked = False,
                      edgecolor="g",
                      fontsize = 20)
plt.title("Male-Female Graduates compared to Total Graduates",fontsize=20)
plt.grid(b=True, which='both', color='lightGreen',linestyle='-')
plt.show ()
# from the below Chandigarh, NCT of Delhi, Maharashta have most of their graduates living in citi
# we can note that Kerala and Meghalaya are the only states that have more number of female gradu
# male graduates
```



Analyzing Sex ratio across states

In [31]:
```
# A bar chart to show how many females are there for per 1000 males.
fig = plt.figure(figsize=(20,20))
states = cities.groupby('state_name')['sex_ratio'].mean().sort_values(ascending=True)
states.plot(kind="barh", fontsize = 20)
plt.grid(b=True, which='both', color='Black',linestyle='-')
plt.xlabel('No of females available for every 1000 males', fontsize = 20)
plt.show ()
# We can see that states of Kerala, Manipur, Meghalaya, Puducherry, Mizoram are having more femal
```



In [1]:

```
  File "C:\Users\91974\AppData\Local\Temp/ipykernel_72184/1585807553.py", line 1
    jupyter nbconvert --to webpdf --allow-chromium-download INDIAN.ipynb
            ^
SyntaxError: invalid syntax
```