

Impact of Covid-19 on Airline Industry and Unemployment Rate. Analysis of Covid-19 by Race and maximum number of Unemployment claims received in USA.

```
pip install -U plotly
```

```
Requirement already up-to-date: plotly in /usr/local/lib/python3.6/dist-packages (4.14.1)
Requirement already satisfied, skipping upgrade: six in /usr/local/lib/python3.6/dist-packages (from plotly) (1.15.0)
Requirement already satisfied, skipping upgrade: retrying>=1.3.3 in /usr/local/lib/python3.6/dist-packages (from plotly) (1.3.3)
```

```
pip install bar_chart_race
```

```
Requirement already satisfied: bar_chart_race in /usr/local/lib/python3.6/dist-packages (0.1.0)
Requirement already satisfied: matplotlib>=3.1 in /usr/local/lib/python3.6/dist-packages (from bar_chart_race) (3.2.2)
Requirement already satisfied: pandas>=0.24 in /usr/local/lib/python3.6/dist-packages (from bar_chart_race) (1.1.4)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.6/dist-packages (from matplotlib>=3.1->bar_chart_race) (1.18)
Requirement already satisfied: numpy>=1.11 in /usr/local/lib/python3.6/dist-packages (from matplotlib>=3.1->bar_chart_race) (1.18)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.6/dist-packages (from matplotlib>=3.1->bar_chart_race) (2.8)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.6/dist-packages (from matplotlib>=3.1->bar_chart_race) (1.4.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.6/dist-packages (from matplotlib>=3.1->bar_chart_race) (0.1)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.24->bar_chart_race) (2018.9)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.6/dist-packages (from python-dateutil>=2.1->matplotlib>=3.1->ba
```

```
pip install altair vega_datasets
```

```
Requirement already satisfied: altair in /usr/local/lib/python3.6/dist-packages (4.1.0)
Requirement already satisfied: vega_datasets in /usr/local/lib/python3.6/dist-packages (0.8.0)
Requirement already satisfied: pandas>=0.18 in /usr/local/lib/python3.6/dist-packages (from altair) (1.1.4)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.6/dist-packages (from altair) (2.11.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages (from altair) (1.18.5)
Requirement already satisfied: toolz in /usr/local/lib/python3.6/dist-packages (from altair) (0.11.1)
Requirement already satisfied: jsonschema in /usr/local/lib/python3.6/dist-packages (from altair) (2.6.0)
Requirement already satisfied: entrypoints in /usr/local/lib/python3.6/dist-packages (from altair) (0.3)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.18->altair) (2018.9)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.18->altair) (2.8)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.6/dist-packages (from jinja2->altair) (1.1.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.6/dist-packages (from python-dateutil>=2.7.3->pandas>=0.18->alt
```

Hypothesis - Initially looking at Media Reports we have assumed NewYork to be the most effected State in USA by Covid-19 hence maximum unemployment rate

```
import pandas as pd
```

```
df1 = pd.read_csv("https://storage.googleapis.com/analytics3/time_series_covid19_confirmed_US.csv")
df1.head(5)
```

	UID	iso2	iso3	code3	FIPS	Admin2	Province_State	Country_Region	Lat	Long_	Combined_Key	1/22/20	1/23
0	84001001	US	USA	840	1001.0	Autauga	Alabama	US	32.539527	-86.644082	Autauga, Alabama, US	0	
1	84001003	US	USA	840	1003.0	Baldwin	Alabama	US	30.727750	-87.722071	Baldwin, Alabama, US	0	
2	84001005	US	USA	840	1005.0	Barbour	Alabama	US	31.868263	-85.387129	Barbour, Alabama, US	0	
3	84001007	US	USA	840	1007.0	Bibb	Alabama	US	32.996421	-87.125115	Bibb, Alabama, US	0	
4	84001009	US	USA	840	1009.0	Blount	Alabama	US	33.982109	-86.567906	Blount, Alabama, US	0	

5 rows × 319 columns

```
pd.set_option('display.max_rows', 5000)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 5000)
```

```

df1.columns = df1.columns.str.lower() #Change all the column names to lowercase
df1.columns = df1.columns.str.replace('long_','long') #Change column name from "long_" to "long"
df1.drop(['iso2','iso3', "code3", "country_region"] , axis=1 , inplace=True) #To the Copied data frame drop all unwanted columns
df1.head(4)

```

	uid	fips	admin2	province_state	lat	long	combined_key	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20
0	84001001	1001.0	Autauga	Alabama	32.539527	-86.644082	Autauga, Alabama, US	0	0	0	0	0	0
1	84001003	1003.0	Baldwin	Alabama	30.727750	-87.722071	Baldwin, Alabama, US	0	0	0	0	0	0
2	84001005	1005.0	Barbour	Alabama	31.868263	-85.387129	Barbour, Alabama, US	0	0	0	0	0	0
3	84001007	1007.0	Bibb	Alabama	32.996421	-87.125115	Bibb, Alabama, US	0	0	0	0	0	0

```

df1.dropna(subset=["fips"], inplace=True) #Drop null values in fips, county code cannot be NULL
df1['fips'] = df1['fips'].astype(int).astype(str) #Convert fips datatype from float to string
df1['fips'] = df1['fips'].map(lambda x: f'{x:0>5}') #Append 0 in the front of fips column to have length of 5
df1.head(4)

```

	uid	fips	admin2	province_state	lat	long	combined_key	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20
0	84001001	01001	Autauga	Alabama	32.539527	-86.644082	Autauga, Alabama, US	0	0	0	0	0	0
1	84001003	01003	Baldwin	Alabama	30.727750	-87.722071	Baldwin, Alabama, US	0	0	0	0	0	0
2	84001005	01005	Barbour	Alabama	31.868263	-85.387129	Barbour, Alabama, US	0	0	0	0	0	0
3	84001007	01007	Bibb	Alabama	32.996421	-87.125115	Bibb, Alabama, US	0	0	0	0	0	0

```

dates = df1.columns[7:]
confirmed_df1 = df1.melt(
    id_vars=['province_state', 'lat', 'long'],
    value_vars=dates,
    var_name='date',
    value_name='confirmed'
)
confirmed_grp_df1 = confirmed_df1.groupby(['province_state']).sum('confirmed').reset_index() #Group by country and sum the Covid count
confirmed_grp_df1.head(5)

```

	province_state	lat	long	confirmed
0	Alabama	678600.009377	-1.789350e+06	21316944
1	Alaska	538130.751740	-1.322266e+06	1402531
2	American Samoa	-4395.468000	-5.240066e+04	0
3	Arizona	155582.674826	-5.149601e+05	31878288
4	Arkansas	806464.644915	-2.135225e+06	11513705

```

df1_date = df1.copy(deep=True) #Copy the data frame to another dataframe to sum all the date values
df1_date.drop(['uid', "fips", "admin2", "province_state",
               "lat", "long", "combined_key"] , axis=1 , inplace=True) #To the Copied data frame drop all unwanted columns
df1['confirmed_covid'] = df1_date.sum(axis = 1, skipna = True) #Sum all the date column values and add it to the initial dataframe as a new column
df1[ (df1.fips == '06001')] #Check value for a county in California

```

	uid	fips	admin2	province_state	lat	long	combined_key	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20
195	84006001	06001	Alameda	California	37.646294	-121.892927	Alameda, California, US	0	0	0	0	0	0

1. Maximum per day count of Covid-19 Cases from Jan-20 to Nov-20 in USA

```

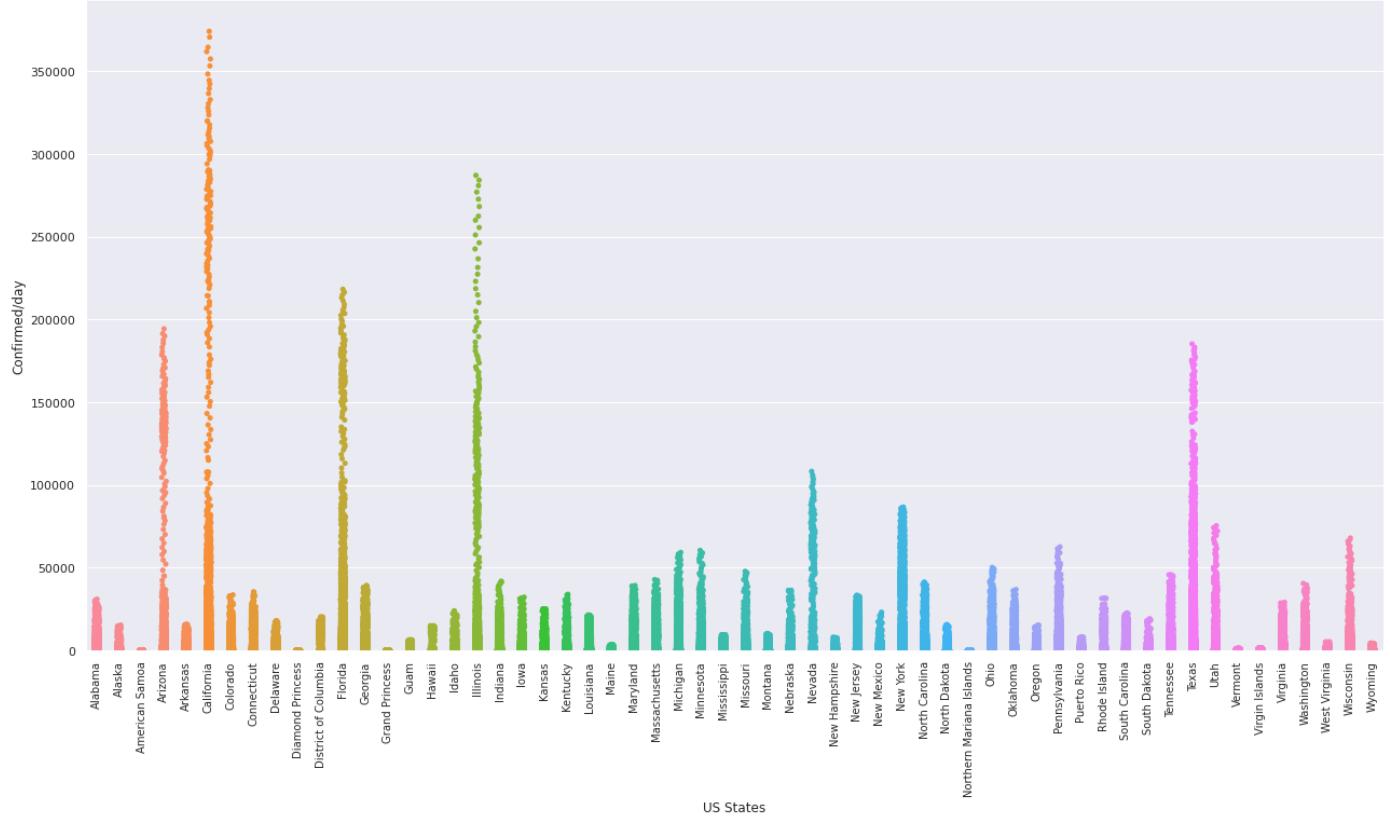
import seaborn as sns

sns.set(rc={'figure.figsize':(10,7)})
ax = sns.catplot(data=confirmed_df1, x='province_state', y='confirmed', height=9, aspect=8/4)
ax.set_xticklabels(rotation=90, fontsize = 10)
ax.set(xlabel='US States', ylabel='Confirmed/day')
ax.set(title="USA per day Confirmed count")
ax.set(ylim=(0, None))

```

<seaborn.axisgrid.FacetGrid at 0x7f75b67047b8>

USA per day Confirmed count



California is leading with the highest number of Covid-19 deaths per day followed by Illinois and Florida.

2. Trend of Covid-19 Cases in USA from Jan 2020 to Nov 2020

```

import bar_chart_race as bcr

df11 = df1.copy(deep=True) #Copy the data frame to another dataframe to sum all the date values

df11.drop(["uid", "fips", "admin2", "lat", "long",
           "combined_key"] , axis=1 , inplace=True) #To the Copied data frame drop all unwanted columns other than date values

cols=df11.filter(regex=':/20').columns #Fetch all the date column names

df12 = df11.melt(id_vars=['province_state'], value_vars=(cols), value_name='count', var_name='date') #Convert date columns to rows
df13 = df12.groupby(['province_state', 'date']).sum('count').reset_index() #Group by state and date to sum the death counts for each s
df13['date'] = pd.to_datetime(df13['date'], format='%m/%d/%y') #Change the date values to date format
df14 = df13.pivot_table(index=['date'], columns=['province_state']) #Convert province state as rows to columns

```

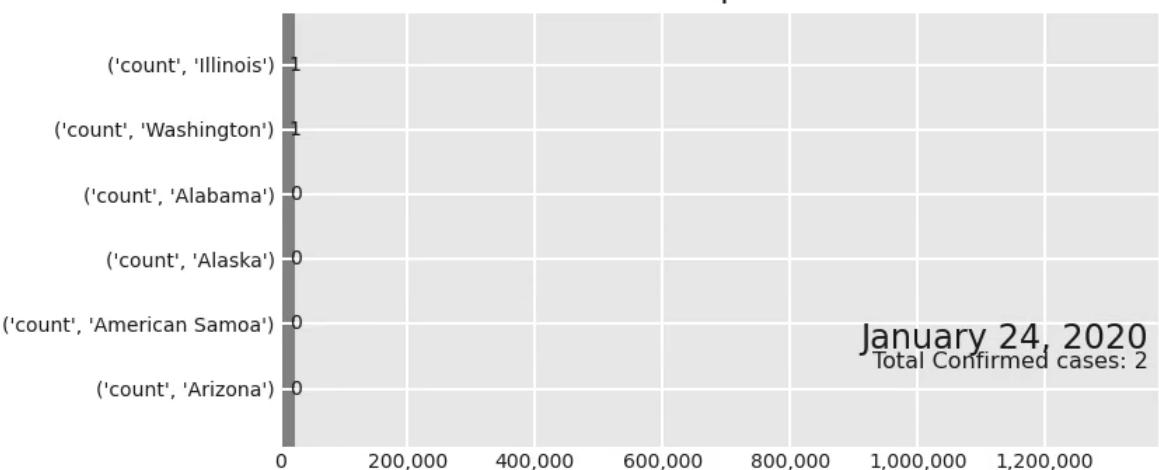
```

bcr.bar_chart_race(
    df=df14,
    filename=None,
    orientation='h',
    sort='desc',
    n_bars=6,
    fixed_order=False,
    fixed_max=True,
    steps_per_period=10,
    interpolate_period=False,
    label_bars=True,
    bar_size=.95,
    period_label={'x': .99, 'y': .25, 'ha': 'right', 'va': 'center'},
    period_fmt='%B %d, %Y',
    period_summary_func=lambda v, r: {'x': .99, 'y': .18,
                                       's': f'Total Confirmed cases: {v.nlargest(6).sum():,.0f}',
                                       'ha': 'right', 'size': 8, 'family': 'DejaVu Sans'},
    perpendicular_bar_func='median',
    period_length=500,
    figsize=(5, 3),
    dpi=144,
    cmap='dark12',
    title='COVID-19 impacts in USA',
    title_size='',
    bar_label_size=7,
    tick_label_size=7,
    shared_fontdict={'family': 'DejaVu Sans', 'color': '.1'},
    scale='linear',
    writer=None,
    fig=None,
    bar_kwargs={'alpha': .7},
    filter_column_colors=True)

/usr/local/lib/python3.6/dist-packages/matplotlib/backends/backend_agg.py:214: RuntimeWarning: Glyph 127 missing from current font
  font.set_text(s, 0.0, flags=flags)
/usr/local/lib/python3.6/dist-packages/matplotlib/backends/backend_agg.py:183: RuntimeWarning: Glyph 127 missing from current font
  font.set_text(s, 0, flags=flags)

```

COVID-19 impacts in USA



Initially, **New York** recorded maximum confirmed cases but later in July and August confirmed cases in **California** were boosted. As of Nov 2020, **Texas** tops the list.

3. Top 5 States with maximum Covid-19 Cases.

```

import numpy as np
import matplotlib.pyplot as plt

state covid = confirmed.groupby('province state'. as index=False).agg({"confirmed": "sum"}) #Sum covid deaths based on states

```

```

state_covid_top = state_covid.nlargest(5,'confirmed') #Pick top 5 states

def func(pct, allvalues):
    absolute = int(pct / 100.*np.sum(allvalues))
    return "{:.1f}\n({:d} #cnt)".format(pct, absolute)

fig = plt.figure(figsize =(10, 7))
ax = fig.add_axes([0,0,1,1])
ax.axis('equal')

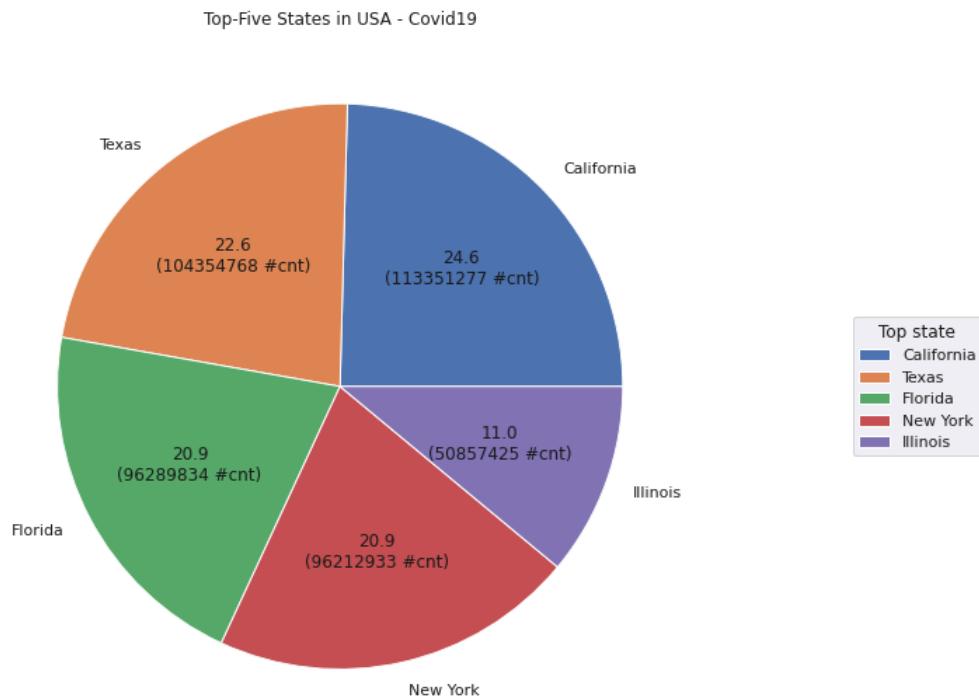
ax.pie(state_covid_top.confirmed, labels = state_covid_top.province_state, autopct = lambda pct: func(pct, state_covid_top.confirmed))

ax.legend(state_covid_top.province_state,
          title ="Top state",
          loc ="center left",
          bbox_to_anchor =(1, 0, 0.5, 1))

ax.set_title("Top-Five States in USA - Covid19")

plt.show()

```



The above plot shows **California** Tops the list with 1.13M confirmed cases followed by **Texas** 1.04M.

```

import pandas as pd

df2 = pd.read_csv("https://storage.googleapis.com/analytics3/time_series_covid19_deaths_US.csv") #Reading Covid-19 Death USA data file
df2.head(5)

```

```
df2.columns = df2.columns.str.lower() #Change all the column names to lowercase
df2.head(3)
```

	uid	iso2	iso3	code3	fips	admin2	province_state	country_region	lat	long_	combined_key	population	1
0	84001001	US	USA	840	1001.0	Autauga	Alabama		US	32.539527	-86.644082	Autauga, Alabama, US	55869
1	84001003	US	USA	840	1003.0	Baldwin	Alabama		US	30.727750	-87.722071	Baldwin, Alabama, US	223234
2	84001005	US	USA	840	1005.0	Barbour	Alabama		US	31.868263	-85.387129	Barbour, Alabama, US	24686

```
df2.columns = df2.columns.str.replace('long_', 'long') #Change column name from "long_" to "long"
df2.describe()
```

	uid	code3	fips	lat	long	population	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20
count	3.340000e+03	3340.000000	3330.000000	3340.000000	3340.000000	3.340000e+03	3340.0	3340.0	3340.0	3340.0	3340.0
mean	8.342958e+07	834.491617	33061.684685	36.707212	-88.601474	9.966039e+04	0.0	0.0	0.0	0.0	0.0
std	4.315345e+06	36.498055	18638.940791	9.062922	21.718982	3.242549e+05	0.0	0.0	0.0	0.0	0.0
min	1.600000e+01	16.000000	60.000000	-14.271000	-174.159600	0.000000e+00	0.0	0.0	0.0	0.0	0.0
25%	8.401811e+07	840.000000	19079.500000	33.895587	-97.790204	9.928500e+03	0.0	0.0	0.0	0.0	0.0
50%	8.402921e+07	840.000000	31014.000000	38.002344	-89.486710	2.491100e+04	0.0	0.0	0.0	0.0	0.0
75%	8.404612e+07	840.000000	47130.500000	41.573069	-82.311265	6.499800e+04	0.0	0.0	0.0	0.0	0.0
max	8.410000e+07	850.000000	99999.000000	69.314792	145.673900	1.003911e+07	0.0	0.0	0.0	0.0	0.0

```
df2.drop(['uid','iso2',"iso3", "code3", "country_region"] , axis=1 , inplace=True) #To the Copied data frame drop all unwanted columns
```

```
df2.dropna(subset=["fips"], inplace=True) #Drop null values in fips, county code cannot be NULL
df2.describe()
```

	fips	lat	long	population	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	1/29/20
count	3330.000000	3330.000000	3330.000000	3.330000e+03	3330.0	3330.0	3330.0	3330.0	3330.0	3330.0	3330.0	3330.0
mean	33061.684685	36.721454	-88.617244	9.981284e+04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
std	18638.940791	9.030193	21.618761	3.246300e+05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
min	60.000000	-14.271000	-174.159600	0.000000e+00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
25%	19079.500000	33.893441	-97.780500	1.006775e+04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
50%	31014.000000	37.995839	-89.456721	2.496500e+04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
75%	47130.500000	41.587421	-82.313398	6.507675e+04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
max	99999.000000	69.314792	145.673900	1.003911e+07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

```
df2['fips'] = df2['fips'].astype(int).astype(str) #Convert fips datatype from float to string
print(df2.head(4))
```

	fips	admin2	province_state	lat	long	combined_key	population	1/22/20	1/23/20	1/24/20	1/25/20	1/26
0	1001	Autauga	Alabama	32.539527	-86.644082	Autauga, Alabama, US	55869	0	0	0	0	0
1	1003	Baldwin	Alabama	30.727750	-87.722071	Baldwin, Alabama, US	223234	0	0	0	0	0
2	1005	Barbour	Alabama	31.868263	-85.387129	Barbour, Alabama, US	24686	0	0	0	0	0
3	1007	Bibb	Alabama	32.996421	-87.125115	Bibb, Alabama, US	22394	0	0	0	0	0

```
df2['fips'] = df2['fips'].map(lambda x: f'{x:0>5}') #Append 0 in the front of fips column to have length of 5
print(df2.head(4))
```

	fips	admin2	province_state	lat	long	combined_key	population	1/22/20	1/23/20	1/24/20	1/25/20	1/2
0	01001	Autauga	Alabama	32.539527	-86.644082	Autauga, Alabama, US	55869	0	0	0	0	0
1	01003	Baldwin	Alabama	30.727750	-87.722071	Baldwin, Alabama, US	223234	0	0	0	0	0

```

2 01005 Barbour      Alabama 31.868263 -85.387129 Barbour, Alabama, US    24686     0     0     0     0
3 01007 Bibb        Alabama 32.996421 -87.125115 Bibb, Alabama, US    22394     0     0     0     0

```

```

i = df2[ df2.population == 0 ].index #Find index of records, where population is equal to 0
df2.drop(i, inplace=True) #Since population can not be 0 & county codes is not valid too for these records, drop all the records, when
df2.describe()

```

	lat	long	population	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	1/29/20	1/30/20	1/31/20
count	3224.000000	3224.000000	3.224000e+03	3224.0	3224.0	3224.0	3224.0	3224.0	3224.0	3224.0	3224.0	3224.0	3224.0
mean	37.928797	-91.530839	1.030945e+05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
std	6.198305	14.696106	3.294119e+05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
min	-14.271000	-174.159600	8.600000e+01	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
25%	34.338308	-98.086287	1.115200e+04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
50%	38.205416	-89.944637	2.621500e+04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
75%	41.688491	-82.977893	6.749075e+04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
max	69.314792	145.673900	1.003911e+07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

```

dates = df2.columns[7:] #Fetch the date column names
death_df2 = df2.melt( #Convert Dates column to rows
    id_vars=['province_state'],
    value_vars=dates,
    var_name='date',
    value_name='death'
)
death_grp_df2 = death_df2.groupby(['province_state']).sum('death').reset_index() #Group by state and sum all the Covid-19 deaths
death_grp_df2.head(5)

```

	province_state	death
0	Alabama	367813
1	Alaska	7880
2	American Samoa	0
3	Arizona	770643
4	Arkansas	175805

```

df2_date = df2.copy(deep=True) #Copy the data frame to another dataframe to sum all the date values to have consolidated no of deaths
df2_date.drop(["fips", "admin2", "province_state",
               "lat", "long", "combined_key", "population"], axis=1, inplace=True) #To the Copied data frame drop all the columns
df2['death_covid'] = df2_date.sum(axis = 1, skipna = True) #Sum all the date column values and add it to the initial dataframe as a new column
df2['death_percentage'] = (df2['death_covid']/df2['population'])*100 #Calculating the Mortality rate
df2[ (df2.fips == '06001')] #Check value for a county in California

```

	fips	admin2	province_state	lat	long	combined_key	population	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20
195	06001	Alameda	California	37.646294	-121.892927	Alameda, California, US	1671329	0	0	0	0	0	0

4. Covid-19 Death Rate in USA.

```

import plotly
from urllib.request import urlopen
import json
with urlopen('https://raw.githubusercontent.com/plotly/datasets/master/geojson-counties-fips.json') as response:
    counties = json.load(response)

import pandas as pd
import plotly.express as px

```

```

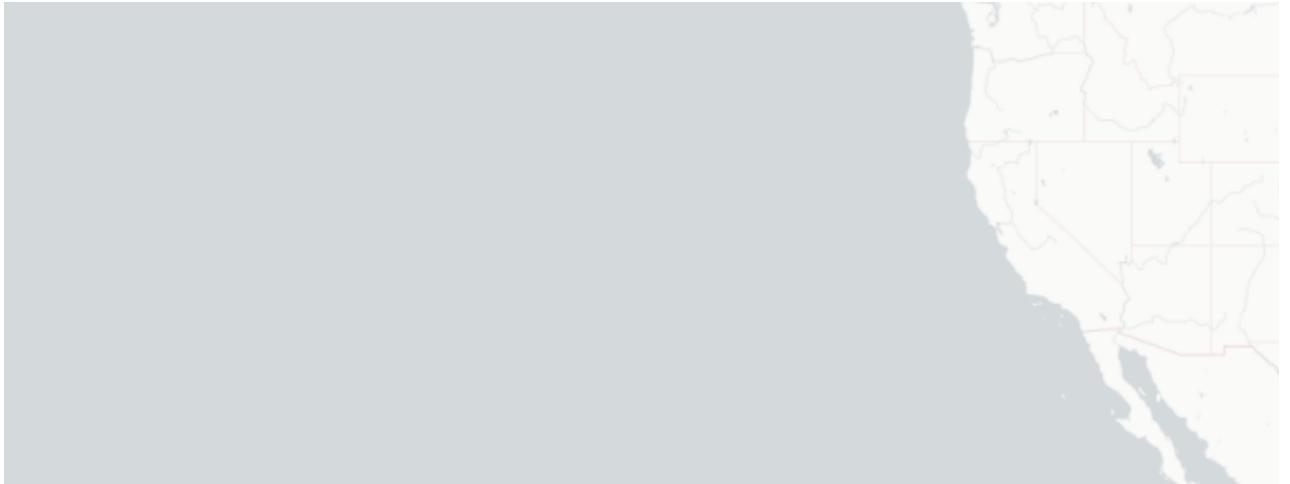
import plotly.graph_objs as go

fig = px.choropleth_mapbox(df2, geojson=counties, locations='fips', color='death_percentage',
                           color_continuous_scale="Viridis",
                           title =("2020 Covid19 Death Rate by County") ,
                           range_color=(0, 10),
                           mapbox_style="carto-positron",
                           zoom=3, center = {"lat": 37.0902, "lon": -95.7129},
                           opacity=0.5,
                           labels={'death_in_percentage':'Death Rate'})

fig.show()

```

2020 Covid19 Death Rate by County



We can conclude from the above that **Southern and Eastern USA** has more number of Covid-19 deaths.

```

df2_sum = df2.copy(deep=True)
df2_sum.drop(["fips", "admin2", "lat", "long", "combined_key", "population"] , axis=1 , inplace=True) #Drop unwanted columns
df2_sum['jan_d'] = df2_sum[list(df2_sum.filter(regex='^1.*/20'))].sum(axis = 1, skipna = True) #Sum all January dates
df2_sum['feb_d'] = df2_sum[list(df2_sum.filter(regex='^2.*/20'))].sum(axis = 1, skipna = True) #Sum all February dates
df2_sum['mar_d'] = df2_sum[list(df2_sum.filter(regex='^3.*/20'))].sum(axis = 1, skipna = True) #Sum all March dates
df2_sum['apr_d'] = df2_sum[list(df2_sum.filter(regex='^4.*/20'))].sum(axis = 1, skipna = True)
df2_sum['may_d'] = df2_sum[list(df2_sum.filter(regex='^5.*/20'))].sum(axis = 1, skipna = True)
df2_sum['jun_d'] = df2_sum[list(df2_sum.filter(regex='^6.*/20'))].sum(axis = 1, skipna = True)
df2_sum['jul_d'] = df2_sum[list(df2_sum.filter(regex='^7.*/20'))].sum(axis = 1, skipna = True)
df2_sum['aug_d'] = df2_sum[list(df2_sum.filter(regex='^8.*/20'))].sum(axis = 1, skipna = True)
df2_sum['sep_d'] = df2_sum[list(df2_sum.filter(regex='^9.*/20'))].sum(axis = 1, skipna = True)
df2_sum['oct_d'] = df2_sum[list(df2_sum.filter(regex='^10.*/20'))].sum(axis = 1, skipna = True)
df2_sum['nov_d'] = df2_sum[list(df2_sum.filter(regex='^11.*/20'))].sum(axis = 1, skipna = True)
df2_sum.drop(list(df2_sum.filter(regex='.*/20')) , axis=1 , inplace=True)

df2_grp_state = df2_sum.groupby(['province_state']).sum('jan_d').reset_index()

df2_grp_state.head(5)

```

province_state death_covid death_percentage jan_d feb_d mar_d apr_d may_d jun_d jul_d aug_d sep_d oct_d nov_d

5. Covid-19 Confirmed and Death Cases by State in USA

```

import altair as alt

chart = alt.Chart(confirmed_grp_df1)

base = alt.Chart(confirmed_grp_df1).transform_calculate(
    Confirmed='"Confirmed"',
    Death='"Death"'
)

scale = alt.Scale(domain=["Confirmed", "Death"], range=['darkblue', 'red'])

conf_line = alt.Chart(confirmed_grp_df1).mark_line().encode(
    alt.X('province_state:N'),
    alt.Y('confirmed:Q')).properties(title='Covid-19 Confirmed and Death cases over States')

conf_point1 = alt.Chart(confirmed_grp_df1).mark_circle().encode(
    alt.X('province_state:N'),
    alt.Y('confirmed:Q'), tooltip=['province_state', 'confirmed'],
    color=alt.Color('Confirmed:N', scale=scale)
)

conf_point2 = alt.Chart(confirmed_grp_df1).mark_circle().encode(
    alt.X('province_state:N'),
    alt.Y('confirmed:Q'), tooltip=['province_state', 'confirmed'],
    )

red = alt.value('#f54242')
brush = alt.selection_interval()

death_line = alt.Chart(death_grp_df2).mark_line().encode(
    alt.X('province_state:N'),
    alt.Y('death:Q'), color=red)

death_point = alt.Chart(death_grp_df2).mark_circle().encode(
    alt.X('province_state:N', title="US States"),
    alt.Y('death:Q', title="Covid-19 Trend"), color=red, tooltip=['province_state', 'death'],
    size=alt.Size('death'),
    scale=alt.Scale(range=[0, 1000]))

chart.configure_legend(strokeColor='gray', fillColor="#EEEEEE", padding=10, cornerRadius=10, orient='top-right')

conf_line + conf_point1 + conf_point2 + death_line + death_point

```

Maximum confirmed Covid-19 cases are reported in **California** followed by **Texas** whereas **NewYork** tops the list with maximum Covid-19 Deaths followed by **NewJersey**.

6. Top 5 States with maximum Covid-19 deaths.

```

import numpy as np
import matplotlib.pyplot as plt

state_covid = death_grp_df2.groupby('province_state', as_index=False).agg({"death": "sum"}) #Sum covid deaths based on states
state_covid_top = state_covid.nlargest(5,'death') #Pick top 5 states

def func(pct, allvalues):
    absolute = int(pct / 100.*np.sum(allvalues))
    return "{:.1f}\n({:d} #cnt)".format(pct, absolute)

fig = plt.figure(figsize =(10, 7))
ax = fig.add_axes([0,0,1,1])
ax.axis('equal')

ax.pie(state_covid_top.death, labels = state_covid_top.province_state, autopct = lambda pct: func(pct, state_covid_top.death))

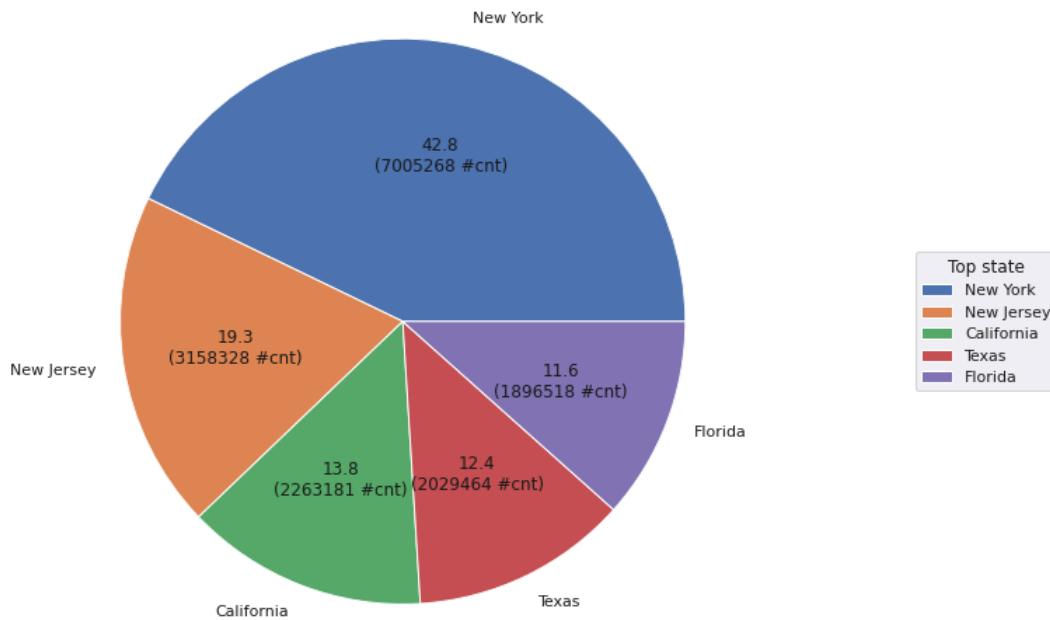
ax.legend(state_covid_top.province_state,
          title ="Top state",
          loc ="center left",
          bbox_to_anchor =(1, 0, 0.5, 1))

ax.set_title("Top-Five States in USA - Covid19 Death")

plt.show()

```

Top-Five States in USA - Covid19 Death



Maximum Covid-19 Deaths in USA are recorded by **NewYork** followed by **NewJersey**.

7. Top 5 Counties with maximum Covid-19 deaths.

```

import matplotlib.pyplot as plt
from matplotlib.pyplot import figure

```

```

county_covid = df2.groupby('combined_key', as_index=False).agg({"death_covid": "sum"}) #Sum covid deaths based on county, states and countries
county_covid_top = county_covid.nlargest(5,'death_covid') #Select top 5 Counties

df23 = pd.DataFrame()

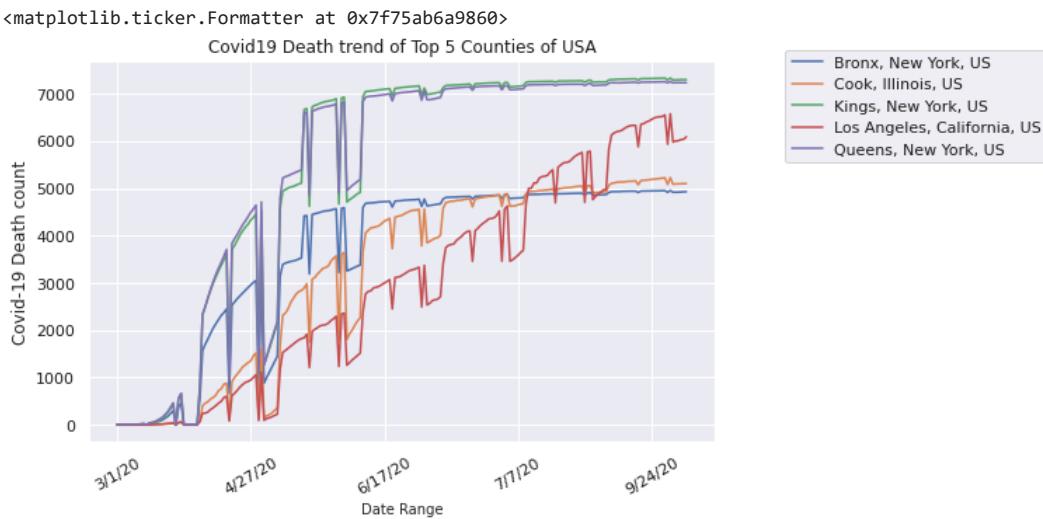
for col in county_covid_top.combined_key: #For each county fetch the records from existing dataframe and create new dataframe
    county_data = df2[df2.combined_key == col ]
    df23.append(county_data)

df23.drop(["fips", "admin2", "province_state", "lat", "long",
           "population", "death_covid", "death_percentage"] , axis=1 , inplace=True) #To the Copied data frame drop all columns
df23.drop(list(df23.filter(regex='^2|^1.*/20')), axis=1 , inplace=True)

df25 = df23.pivot_table( columns=['combined_key']) #Pivot the dataframe

df25.plot(kind='line', figsize=(8,5))
plt.title("Covid19 Death trend of Top 5 Counties of USA")
plt.xlabel('Date Range', fontsize=10)
plt.ylabel('Covid-19 Death count', fontsize='medium')
plt.legend(bbox_to_anchor=(1.1, 1.05))
plt.xticks(ticks=None, rotation=30)
plt.Formatter()

```



Supporting the Analysis shown above, 3 counties of the top 5 belong to **NewYork**, Surprisingly **Cook County from IL** shows up in this list.

8. Top 5 States with maximum Mortality Rate

```

dates = df2.columns[7:]
death_pop_df2 = df2.melt(
    id_vars=['province_state', 'population'],
    value_vars=dates,
    var_name='date',
    value_name='death'
)
death_pop_grp_df2 = death_pop_df2.groupby(['province_state']).sum('death').reset_index()

race_state_chart = death_pop_grp_df2[death_pop_grp_df2['province_state'].isin(['New York', 'New Jersey', 'California', 'Florida', 'Texas'])]
race_state_chart['death_percentage'] = (race_state_chart['death']/race_state_chart['population'])*100
race_state_chart

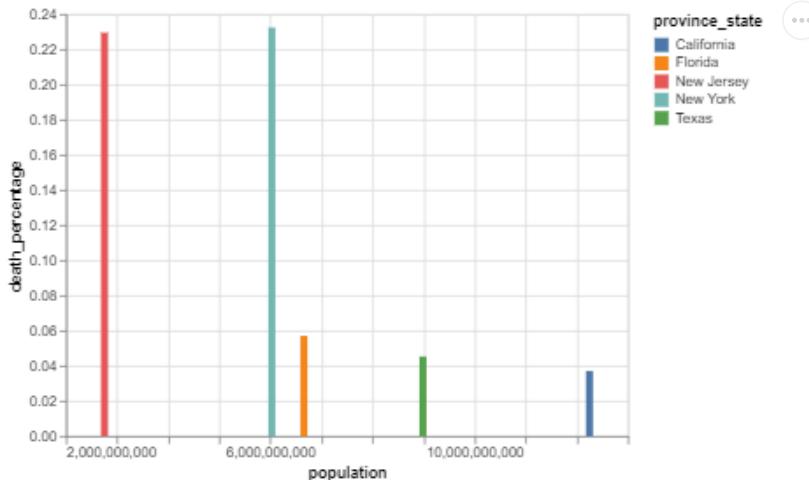
alt.Chart(race_state_chart).mark_bar().encode(
    x='population:Q',
    y='death_percentage:Q',
    color='province_state:N',
    tooltip=['population', 'death_percentage', 'province_state'] # show Name and Origin in a tooltip
).interactive()

```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:11: SettingWithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-vers



From the above analysis, **NewYork** records maximum deaths but **NewJersey** reports higher Mortality rate.

Analysis of New York and New Jersey by zones(Dark Red - High Risk, Light Red - Medium Risk, Light Green - Low Risk and Dark Green - Not affected).

```
#Segmentation
segments = [-1, 50, 300, 2000, pd.np.inf]
labels = ['dark_green_zone', 'green_zone', 'light_red_zone', 'dark_red_zone']
df2_newyork = df2[(df2.province_state == 'New York')].copy(deep=True)
df2_newyork.drop(["fips", "province_state", "lat", "long", "combined_key", "population", "death_percentage", "death_covid"], axis=1, inplace=True) #Drop all the date columns
df2_newyork.drop(list(df2_newyork.filter(regex='^2|^3|^4|^5|^6|^7|^8|^9|^10|^1.*/20')), axis=1, inplace=True) #Drop all the date columns
df2_newyork['death_nov'] = df2_newyork.sum(axis = 1, skipna = True)
zone = pd.cut(df2_newyork.death_nov, segments, labels=labels) #Segement and mark the zones based on Death count
df2_newyork['zone'] = zone #Add the column to the copied dataset
df2_newyork.drop(list(df2_newyork.filter(regex='^1.*/20')), axis=1, inplace=True) #Drop all the November date columns
df2_newyork[(df2_newyork.zone == 'dark_green_zone')]
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:2: FutureWarning:
```

The pandas.np module is deprecated and will be removed from pandas in a future version. Import numpy directly instead

	admin2	death_nov	zone
1911	Cortland	24	dark_green_zone
1920	Hamilton	24	dark_green_zone
1922	Jefferson	30	dark_green_zone
1947	Schoharie	48	dark_green_zone
1955	Tompkins	0	dark_green_zone

```
#Segmentation
segments = [-1, 50, 300, 2000, pd.np.inf]
labels = ['dark_green_zone', 'green_zone', 'light_red_zone', 'dark_red_zone']
df2_newjersey = df2[(df2.province_state == 'New Jersey')].copy(deep=True)
df2_newjersey.drop(["fips", "province_state", "lat", "long", "combined_key", "population", "death_percentage", "death_covid"], axis=1, inplace=True) #Drop all the date columns
df2_newjersey.drop(list(df2_newjersey.filter(regex='^2|^3|^4|^5|^6|^7|^8|^9|^10|^1.*/20')), axis=1, inplace=True) #Drop all the date columns
df2_newjersey['death_nov'] = df2_newjersey.sum(axis = 1, skipna = True)
zone = pd.cut(df2_newjersey.death_nov, segments, labels=labels) #Segement and mark the zones based on Death count
df2_newjersey['zone'] = zone #Add the column to the copied dataset
df2_newjersey.drop(list(df2_newjersey.filter(regex='^1.*/20')), axis=1, inplace=True) #Drop all the November date columns
df2_newjersey
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:2: FutureWarning:
```

```
The pandas.np module is deprecated and will be removed from pandas in a future version. Import numpy directly instead
```

	admin2	death_nov	zone
1842	Atlantic	6533	dark_red_zone
1843	Bergen	49949	dark_red_zone
1844	Burlington	12588	dark_red_zone
1845	Camden	15551	dark_red_zone
1846	Cape May	2475	dark_red_zone
1847	Cumberland	3988	dark_red_zone
1848	Essex	52088	dark_red_zone
1849	Gloucester	6137	dark_red_zone
1850	Hudson	37048	dark_red_zone
1851	Hunterdon	3066	dark_red_zone
1852	Mercer	15486	dark_red_zone
1853	Middlesex	35080	dark_red_zone
1854	Monmouth	21093	dark_red_zone
1855	Morris	20323	dark_red_zone
1856	Ocean	26020	dark_red_zone
1858	Passaic	30687	dark_red_zone
1859	Salem	2165	dark_red_zone
1860	Somerset	14436	dark_red_zone
1861	Sussex	4749	dark_red_zone

New Jersey is totally in **Dark red zone** whereas New York has few counties which are **Dark green zone** i.e. least affected by Covid-19.

```
df1_sum = df1.copy(deep=True)
df1_sum.drop(["fips", "uid", "admin2", "lat", "long", "combined_key"] , axis=1 , inplace=True)
df1_sum['jan_c'] = df1_sum[list(df1_sum.filter(regex='^1.*/20'))].sum(axis = 1, skipna = True)
df1_sum['feb_c'] = df1_sum[list(df1_sum.filter(regex='^2.*/20'))].sum(axis = 1, skipna = True)
df1_sum['mar_c'] = df1_sum[list(df1_sum.filter(regex='^3.*/20'))].sum(axis = 1, skipna = True)
df1_sum['apr_c'] = df1_sum[list(df1_sum.filter(regex='^4.*/20'))].sum(axis = 1, skipna = True)
df1_sum['may_c'] = df1_sum[list(df1_sum.filter(regex='^5.*/20'))].sum(axis = 1, skipna = True)
df1_sum['jun_c'] = df1_sum[list(df1_sum.filter(regex='^6.*/20'))].sum(axis = 1, skipna = True)
df1_sum['jul_c'] = df1_sum[list(df1_sum.filter(regex='^7.*/20'))].sum(axis = 1, skipna = True)
df1_sum['aug_c'] = df1_sum[list(df1_sum.filter(regex='^8.*/20'))].sum(axis = 1, skipna = True)
df1_sum['sep_c'] = df1_sum[list(df1_sum.filter(regex='^9.*/20'))].sum(axis = 1, skipna = True)
df1_sum['oct_c'] = df1_sum[list(df1_sum.filter(regex='^10.*/20'))].sum(axis = 1, skipna = True)
df1_sum['nov_c'] = df1_sum[list(df1_sum.filter(regex='^11.*/20'))].sum(axis = 1, skipna = True)
df1_sum.drop(list(df1_sum.filter(regex='.*/20')) , axis=1 , inplace=True)

df1_grp_state = df1_sum.groupby(['province_state']).sum('jan_c').reset_index()

df1_grp_state.head(5)
```

	province_state	confirmed_covid	jan_c	feb_c	mar_c	apr_c	may_c	jun_c	jul_c	aug_c	sep_c	oct_c	nov_c
0	Alabama	21316944	0	0	6735	127476	377501	812738	1938938	3358162	4214797	5351525	5129072
1	Alaska	1402531	0	0	808	8403	12586	23313	68368	156848	222916	370781	538508
2	American Samoa	0	0	0	0	0	0	0	0	0	0	0	0
3	Arizona	31878288	6	29	6783	128753	417640	1294975	4103132	5950207	6319715	7141160	6515888
4	Arkansas	11513705	0	0	3721	52677	150164	402154	972505	1633912	2167050	3025302	3106220

```
c_d_merge = pd.merge(left=df1_grp_state, right=df2_grp_state, how='outer', left_on='province_state', right_on='province_state')
c_d_merge.head(5)
```

	province_state	confirmed_covid	jan_c	feb_c	mar_c	apr_c	may_c	jun_c	jul_c	aug_c	sep_c	oct_c	nov_c	de
0	Alabama	21316944	0	0	6735	127476	377501	812738	1938938	3358162	4214797	5351525	5129072	
1	Alaska	1402531	0	0	808	8403	12586	23313	68368	156848	222916	370781	538508	
2	American Samoa	0	0	0	0	0	0	0	0	0	0	0	0	0
3	Arizona	31878288	6	29	6783	128753	417640	1294975	4103132	5950207	6319715	7141160	6515888	
4	Arkansas	11513705	0	0	3721	52677	150164	402154	972505	1633912	2167050	3025302	3106220	

```
c_d_grp_merge = pd.merge(left=confirmed_grp_df1, right=death_grp_df2, how='outer', left_on='province_state', right_on='province_state')
c_d_grp_merge.head(5)
```

	province_state	lat	long	confirmed	death
0	Alabama	678600.009377	-1.789350e+06	21316944	367813.0
1	Alaska	538130.751740	-1.322266e+06	1402531	7880.0
2	American Samoa	-4395.468000	-5.240066e+04	0	0.0
3	Arizona	155582.674826	-5.149601e+05	31878288	770643.0
4	Arkansas	806464.644915	-2.135225e+06	11513705	175805.0

```
import pandas as pd
```

```
race_df = pd.read_csv("https://storage.googleapis.com/analytics3/Provisional_COVID-19_Death_Counts_by_County_and_Race.csv")
race_df.head(5)
```

	Data as of	Start week	End Week	State	County Name	Urban Rural Code	FIPS State	FIPS County	FIPS Code	Indicator	Total deaths	COVID-19 Deaths	Non-Hispanic White	Non-Hispanic Black	Hi Am
0	11/18/2020	02/01/2020	11/14/2020	AL	Calhoun County	4	1	15	1015	Distribution of all-cause deaths (%)	1274	117	0.835	0.152	
1	11/18/2020	02/01/2020	11/14/2020	AL	Calhoun County	4	1	15	1015	Distribution of COVID-19 deaths	1274	117	0.778	0.205	

```
race_df.columns = race_df.columns.str.lower()
race_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1152 entries, 0 to 1151
Data columns (total 21 columns):
 #   Column           Non-Null Count  Dtype  
 ---  --  
 0   data as of      1152 non-null   object  
 1   start week     1152 non-null   object  
 2   end week       1152 non-null   object  
 3   state          1152 non-null   object  
 4   county name    1152 non-null   object  
 5   urban rural code 1152 non-null   int64  
 6   fips state     1152 non-null   int64  
 7   fips county    1152 non-null   int64  
 8   fips code      1152 non-null   int64  
 9   indicator      1152 non-null   object  
 10  total deaths   1152 non-null   int64  
 11  covid-19 deaths 1152 non-null   int64  
 12  non-hispanic white 1149 non-null   float64 
 13  non-hispanic black 1055 non-null   float64 
 14  non-hispanic american indian or alaska native 529 non-null   float64 
 15  non-hispanic asian        804 non-null   float64 
 16  non-hispanic native hawaiian or other pacific islander 431 non-null   float64 
 17  hispanic        1010 non-null   float64 
 18  other           627 non-null    float64 
 19  urban rural description 1152 non-null   object
```

```

20 footnote
dtypes: float64(7), int64(6), object(8)
memory usage: 189.1+ KB

race_col = race_df.columns.str.replace('\s+', '_')
race_df.columns = race_col
race_df.head(5)

   data_as_of  start_week  end_week  state county_name urban_rural_code  fips_state  fips_county  fips_code indicator  total_
0  11/18/2020  02/01/2020  11/14/2020    AL  Calhoun County             4          1         15      1015 Distribution of all-cause deaths (%)
1  11/18/2020  02/01/2020  11/14/2020    AL  Calhoun County             4          1         15      1015 Distribution of COVID-19 deaths (%)
2  11/18/2020  02/01/2020  11/14/2020    AL  Calhoun County             4          1         15      1015 Distribution of population (%)
3  11/18/2020  02/01/2020  11/14/2020    AL  Etowah County              4          1         55      1055 Distribution of all-cause deaths (%)
4  11/18/2020  02/01/2020  11/14/2020    AL  Etowah County              4          1         55      1055 Distribution of COVID-19 deaths (%)

race_df['non-hispanic_white'] = race_df['non-hispanic_white'].fillna(0)
race_df['non-hispanic_black'] = race_df['non-hispanic_black'].fillna(0)
race_df['non-hispanic_american_indian_or_alaska_native'] = race_df['non-hispanic_american_indian_or_alaska_native'].fillna(0)
race_df['non-hispanic_asian'] = race_df['non-hispanic_asian'].fillna(0)
race_df['non-hispanic_native_hawaiian_or_other_pacific_islander'] = race_df['non-hispanic_native_hawaiian_or_other_pacific_islander'].fillna(0)
race_df['hispanic'] = race_df['hispanic'].fillna(0)
race_df['other'] = race_df['other'].fillna(0)
race_df.head(4)

   data_as_of  start_week  end_week  state county_name urban_rural_code  fips_state  fips_county  fips_code indicator  total_
0  11/18/2020  02/01/2020  11/14/2020    AL  Calhoun County             4          1         15      1015 Distribution of all-cause deaths (%)
1  11/18/2020  02/01/2020  11/14/2020    AL  Calhoun County             4          1         15      1015 Distribution of COVID-19 deaths (%)
2  11/18/2020  02/01/2020  11/14/2020    AL  Calhoun County             4          1         15      1015 Distribution of population (%)
3  11/18/2020  02/01/2020  11/14/2020    AL  Etowah County              4          1         55      1055 Distribution of all-cause deaths (%)

race_df.drop(['data_as_of','start_week','end_week', 'urban_rural_code', 'fips_state', 'fips_county','footnote'] , axis=1 , inplace=True)
race_df['fips_code'] = race_df['fips_code'].astype(int).astype(str) #Convert fips datatype from float to string
race_df['fips_code'] = race_df['fips_code'].map(lambda x: f'{x:0>5}')

```

```
race_dt['indicator'] = race_dt['indicator'].astype(str)
race_df.head(4)
```

					covid-19_deaths	non-hispanic_white	non-hispanic_black	non-hispanic_american_indian_or
0	AL	Calhoun County	01015	Distribution of all-cause deaths (%)	1274	117	0.835	0.152
1	AL	Calhoun County	01015	Distribution of COVID-19 deaths (%)	1274	117	0.778	0.205

```
df_melted = race_df.melt(id_vars=['fips_code','state','county_name','indicator','total_deaths','covid-19_deaths'], value_vars=['non-hispanic_white','non-hispanic_black','non-hispanic_american_indian_or'])
```

	fips_code	state	county_name	indicator	total_deaths	covid-19_deaths	race	percentage
0	01015	AL	Calhoun County	Distribution of all-cause deaths (%)	1274	117	non-hispanic_white	0.835
1	01015	AL	Calhoun County	Distribution of COVID-19 deaths (%)	1274	117	non-hispanic_white	0.778
2	01015	AL	Calhoun County	Distribution of population (%)	1274	117	non-hispanic_white	0.716
3	01055	AL	Etowah County	Distribution of all-cause deaths (%)	1349	150	non-hispanic_white	0.847
4	01055	AL	Etowah County	Distribution of COVID-19 deaths (%)	1349	150	non-hispanic_white	0.840

```
population_filter = race_df['indicator'].str.contains('population', case=False, na=False)
total_death_filter = race_df['indicator'].str.contains('all-cause deaths', case=False, na=False)
covid19_death_filter = race_df['indicator'].str.contains('COVID-19 deaths', case=False, na=False)
```

```
population_filter.sum()
```

```
384
```

```
total_death_filter.sum()
```

```
384
```

```
covid19_death_filter.sum()
```

```
384
```

```
population_df = race_df[population_filter]
```

```
population_df.drop(['indicator'] , axis=1 , inplace=True)
population_df.head(4)
```

```
/usr/local/lib/python3.6/dist-packages/pandas/core/frame.py:4174: SettingWithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-vers

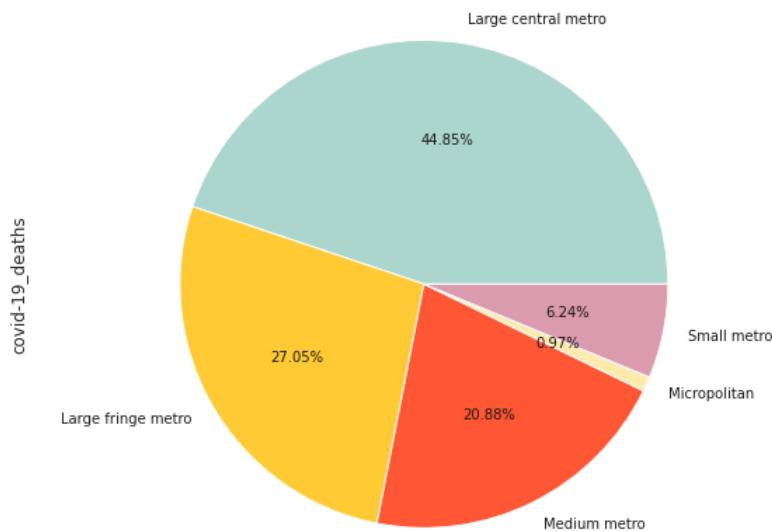
state	county_name	fips_code	total_deaths	covid-19_deaths	non-hispanic_white	non-hispanic_black	non-hispanic_american_indian_or_alaska_native
2	AL	Calhoun	01015	1274	117	0.716	0.210
3	AL	Etowah	01055	1349	150	0.847	0.139

9. Geographical Communities with maximum Covid-19 Deaths in USA

```
country

urban_rural = population_df.groupby('urban_rural_description')['covid-19_deaths'].sum()
pie = urban_rural.plot.pie(colors=['#abd6ce', '#FFCA33', '#FF5733', '#ffeaac', '#db9bae'],
                           autopct=(lambda p : '{:.2f}%'.format(p)),
                           fontsize=10,
                           figsize=(8, 8))
pie.yaxis.set_label_coords(-0.15, 0.5)
pie.set_title('Covid-19 Deaths by Areas')

Text(0.5, 1.0, 'Covid-19 Deaths by Areas')
Covid-19 Deaths by Areas
```



Large Central Metro areas account to nearly 45% of Covid-19 deaths in USA, which supports the analysis above (NewYork, NewJersey California are all Metro Areas). Micropolitan Areas are least effected by Covid-19.

```
total_death_df = race_df[total_death_filter]
total_death_df.drop(['indicator'] , axis=1 , inplace=True)
total_death_df.head(4)
```

```
/usr/local/lib/python3.6/dist-packages/pandas/core/frame.py:4174: SettingWithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-vers

state	county_name	fips_code	total_deaths	covid-19_deaths	non-hispanic_white	non-hispanic_black	non-hispanic_american_indian_or_alaska_native
0	AL	Calhoun County	01015	1274	117	0.835	0.152
3	AL	Etowah County	01055	1349	150	0.847	0.139

```

covid19_death_df = race_df[covid19_death_filter]
covid19_death_df.drop(['indicator'] , axis=1 , inplace=True)
covid19_death_df.head(4)

```

/usr/local/lib/python3.6/dist-packages/pandas/core/frame.py:4174: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-vers

	state	county_name	fips_code	total_deaths	covid-19_deaths	non-hispanic_white	non-hispanic_black	non-hispanic_american_indian_or_alaska_n
1	AL	Calhoun County	01015	1274	117	0.778	0.205	
4	AL	Etowah County	01055	1349	150	0.840	0.153	

```

df_merge = pd.merge(left=population_df, right=total_death_df, how='inner', left_on='fips_code', right_on='fips_code')
df_merge.head(5)

```

	state_x	county_name_x	fips_code	total_deaths_x	covid-19_deaths_x	non-hispanic_white_x	non-hispanic_black_x	non-hispanic_american_indian_c
0	AL	Calhoun County	01015	1274	117	0.716	0.210	
1	AL	Etowah County	01055	1349	150	0.774	0.155	
2	AL	Houston County	01069	1669	278	0.660	0.271	
3	AL	Jefferson County	01073	8141	724	0.495	0.431	
4	AL	Lee County	01081	1071	158	0.676	0.226	

```

df_merge.drop(['state_y','county_name_y','total_deaths_y', 'covid-19_deaths_y', 'urban_rural_description_y'] , axis=1 , inplace=True)
df_merge.head(2)

```

	state_x	county_name_x	fips_code	total_deaths_x	covid-19_deaths_x	non-hispanic_white_x	non-hispanic_black_x	non-hispanic_american_indian_c
0	AL	Calhoun County	01015	1274	117	0.716	0.210	
1	AL	Etowah County	01055	1349	150	0.774	0.155	

```

race_df_final = pd.merge(left=df_merge, right=covid19_death_df, how='inner', left_on='fips_code', right_on='fips_code')
race_df_final.head(2)

```

	state_x	county_name_x	fips_code	total_deaths_x	covid-19_deaths_x	non-hispanic_white_x	non-hispanic_black_x	non-hispanic_american_indian_c
0	AL	Calhoun County	01015	1274	117	0.716	0.210	
1	AL	Etowah County	01055	1349	150	0.774	0.155	

```

race_df_final.drop(['state','county_name','total_deaths', 'covid-19_deaths', 'urban_rural_description_x'] , axis=1 , inplace=True)
race_df_final.head(2)

```

	state_x	county_name_x	fips_code	total_deaths_x	covid-19_deaths_x	non-hispanic_white_x	non-hispanic_black_x	non-hispanic_american_indian_c
0	AL	Calhoun County	01015	1274	117	0.716	0.210	
1	AL	Etowah County	01055	1349	150	0.774	0.155	

```

race_df_final.rename(columns = {'non-hispanic_white_x':'non-hispanic_white_population'}, inplace = True)
race_df_final.rename(columns = {'non-hispanic_black_x':'non-hispanic_black_population'}, inplace = True)
race_df_final.rename(columns = {'non-hispanic_american_indian_or_alaska_native_x':'non-hispanic_american_indian_or_alaska_native_population'})
race_df_final.rename(columns = {'non-hispanic_asian_x':'non-hispanic_asian_population'}, inplace = True)
race_df_final.rename(columns = {'non-hispanic_native_hawaiian_or_other_pacific_islander_x':'non-hispanic_native_hawaiian_or_other_pacific_islander_population'})
race_df_final.rename(columns = {'hispanic_x':'hispanic_population'}, inplace = True)
race_df_final.rename(columns = {'other_x':'other_population'}, inplace = True)
race_df_final.rename(columns = {'non-hispanic_white_y':'non-hispanic_white_total_death'}, inplace = True)
race_df_final.rename(columns = {'non-hispanic_black_y':'non-hispanic_black_total_death'}, inplace = True)
race_df_final.rename(columns = {'non-hispanic_american_indian_or_alaska_native_y':'non-hispanic_american_indian_or_alaska_native_total_death'})
race_df_final.rename(columns = {'non-hispanic_asian_y':'non-hispanic_asian_total_death'}, inplace = True)
race_df_final.rename(columns = {'non-hispanic_native_hawaiian_or_other_pacific_islander_y':'non-hispanic_native_hawaiian_or_other_pacific_islander_total_death'})
race_df_final.rename(columns = {'hispanic_y':'hispanic_total_death'}, inplace = True)
race_df_final.rename(columns = {'other_y':'other_total_death'}, inplace = True)
race_df_final.rename(columns = {'non-hispanic_white':'non-hispanic_white_covid19_death'}, inplace = True)
race_df_final.rename(columns = {'non-hispanic_black':'non-hispanic_black_covid19_death'}, inplace = True)
race_df_final.rename(columns = {'non-hispanic_american_indian_or_alaska_native':'non-hispanic_american_indian_or_alaska_native_covid19_death'})
race_df_final.rename(columns = {'non-hispanic_asian':'non-hispanic_asian_covid19_death'}, inplace = True)
race_df_final.rename(columns = {'non-hispanic_native_hawaiian_or_other_pacific_islander':'non-hispanic_native_hawaiian_or_other_pacific_islander_covid19_death'})
race_df_final.rename(columns = {'hispanic':'hispanic_covid19_death'}, inplace = True)
race_df_final.rename(columns = {'other':'other_covid19_death'}, inplace = True)
race_df_final.rename(columns = {'state_x':'state'}, inplace = True)
race_df_final.rename(columns = {'county_name_x':'county_name'}, inplace = True)
race_df_final.rename(columns = {'total_deaths_x':'total_deaths'}, inplace = True)
race_df_final.rename(columns = {'covid-19_deaths_x':'covid-19_deaths'}, inplace = True)
race_df_final.rename(columns = {'fips_code':'fips'}, inplace = True)
race_df_final.head(2)

```

	state	county_name	fips	total_deaths	covid-19_deaths	non-hispanic_white_population	non-hispanic_black_population	non-hispanic_american_ir
0	AL	Calhoun County	01015	1274	117	0.716	0.210	
1	AL	Etowah County	01055	1349	150	0.774	0.155	

```

age_gender_df = pd.read_csv("https://storage.googleapis.com/analytics3/Provisional_COVID-19_Death_Counts_by_Sex_Age_and_State_update")
age_gender_df.head(2)

```

Data as of	Start week	End Week	State	Sex	Age group	COVID-19 Deaths	Total Deaths	Pneumonia Deaths	Pneumonia and COVID-19 Deaths	Influenza Deaths	Pneumonia, Influenza, or COVID-19 Deaths	Footnote
One or more												

```
age_gender_df.columns = age_gender_df.columns.str.lower()
```

```
age_gender_df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 742 entries, 0 to 741
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   data as of       742 non-null    object  
 1   start week      742 non-null    object  
 2   end week        742 non-null    object  
 3   state           742 non-null    object  
 4   sex             742 non-null    object  
 5   age group       742 non-null    object  
 6   covid-19 deaths 597 non-null    float64 
 7   total deaths    741 non-null    float64 
 8   pneumonia deaths 593 non-null    float64 

```

```
9  pneumonia and covid-19 deaths           601 non-null   float64
10 influenza deaths                      368 non-null   float64
11 pneumonia, influenza, or covid-19 deaths 607 non-null   float64
12 footnote                                489 non-null    object
dtypes: float64(6), object(7)
memory usage: 75.5+ KB
```

```
age_gender_df.state.describe()
```

```
count          742
unique         53
top      Kentucky
freq            14
Name: state, dtype: object
```

```
age_gender_col = age_gender_df.columns.str.replace('\s+', '_')
age_gender_df.columns = age_gender_col
age_gender_df.head(5)
```

data_as_of	start_week	end_week	state	sex	age_group	covid-19_deaths	total_deaths	pneumonia_deaths	pneumonia_and_covid-19_deaths
------------	------------	----------	-------	-----	-----------	-----------------	--------------	------------------	-------------------------------

0	11/25/2020	2/1/2020	11/21/2020	Alabama	Male	0-17 years	NaN	310.0	NaN	0.0
---	------------	----------	------------	---------	------	------------	-----	-------	-----	-----

```
age_gender_df.drop(['data_as_of', 'start_week', 'end_week', 'pneumonia_deaths', 'pneumonia_and_covid-19_deaths', 'influenza_deaths', 'pn
age_gender_df.head(2)
```

state	sex	age_group	covid-19_deaths	total_deaths
-------	-----	-----------	-----------------	--------------

0	Alabama	Male	0-17 years	NaN	310.0
1	Alabama	Male	18-29 years	11.0	661.0

```
age_gender_df['covid-19_deaths'] = age_gender_df['covid-19_deaths'].fillna(0)
age_gender_df['total_deaths'] = age_gender_df['total_deaths'].fillna(0)
age_gender_df.head(2)
```

state	sex	age_group	covid-19_deaths	total_deaths
-------	-----	-----------	-----------------	--------------

0	Alabama	Male	0-17 years	0.0	310.0
1	Alabama	Male	18-29 years	11.0	661.0

```
race_df_final.head(2)
```

state	county_name	fips	total_deaths	covid-19_deaths	non-hispanic_white_population	non-hispanic_black_population	non-hispanic_american_ir
0	AL	Calhoun County	01015	1274	117	0.716	0.210
1	AL	Etowah County	01055	1349	150	0.774	0.155

```
df2_state = df2.loc[:, ['fips', 'province_state']]
df2_state.head(2)
```

fips	province_state
------	----------------

0	01001	Alabama
1	01003	Alabama

```
df2_state.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```

Int64Index: 3224 entries, 0 to 3339
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   fips        3224 non-null    object  
 1   province_state 3224 non-null    object  
dtypes: object(2)
memory usage: 75.6+ KB

```

```

#race_aggregated = pd.merge(left=race_df_final_aggregated_sum, right=race_df_final_aggregated_avg, how='inner', left_on='state', right_on='state')
#race_aggregated.head(2)

race_state = pd.merge(left=race_df_final, right=df2_state, how='inner', left_on='fips', right_on='fips')
race_state.head(2)

```

	state	county_name	fips	total_deaths	covid-19_deaths	non-hispanic_white_population	non-hispanic_black_population	non-hispanic_american_ir
0	AL	Calhoun County	01015	1274	117	0.716		0.210
1	AL	Etowah County	01055	1349	150	0.774		0.155

```
race_state.head(5)
```

	state	county_name	fips	total_deaths	covid-19_deaths	non-hispanic_white_population	non-hispanic_black_population	non-hispanic_american_ir
0	AL	Calhoun County	01015	1274	117	0.716		0.210
1	AL	Etowah County	01055	1349	150	0.774		0.155
2	AL	Houston County	01069	1669	278	0.660		0.271
3	AL	Jefferson County	01073	8141	724	0.495		0.431
4	AL	Lee County	01081	1071	158	0.676		0.226

```

print(race_df_final.info())
print(race_state.info())

```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 384 entries, 0 to 383
Data columns (total 27 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   state        384 non-null    object  
 1   county_name  384 non-null    object  
 2   fips         384 non-null    object  
 3   total_deaths 384 non-null    int64  
 4   covid-19_deaths 384 non-null    int64  
 5   non-hispanic_white_population 384 non-null    float64 
 6   non-hispanic_black_population 384 non-null    float64 
 7   non-hispanic_american_indian_or_alaska_native_population 384 non-null    float64 
 8   non-hispanic_asian_population 384 non-null    float64 
 9   non-hispanic_native_hawaiian_or_other_pacific_islander_population 384 non-null    float64 
 10  hispanic_population 384 non-null    float64 
 11  other_population 384 non-null    float64 
 12  non-hispanic_white_total_death 384 non-null    float64 
 13  non-hispanic_black_total_death 384 non-null    float64 
 14  non-hispanic_american_indian_or_alaska_native_total_death 384 non-null    float64 
 15  non-hispanic_asian_total_death 384 non-null    float64 
 16  non-hispanic_native_hawaiian_or_other_pacific_islander_total_death 384 non-null    float64 
 17  hispanic_total_death 384 non-null    float64 
 18  other_total_death 384 non-null    float64 
 19  non-hispanic_white_covid19_death 384 non-null    float64 
 20  non-hispanic_black_covid19_death 384 non-null    float64 
 21  non-hispanic_american_indian_or_alaska_native_covid19_death 384 non-null    float64 
 22  non-hispanic_asian_covid19_death 384 non-null    float64 
 23  non-hispanic_native_hawaiian_or_other_pacific_islander_covid19_death 384 non-null    float64 
 24  hispanic_covid19_death 384 non-null    float64

```

```

25 other_covid19_death          384 non-null   float64
26 urban_rural_description     384 non-null   object
dtypes: float64(21), int64(2), object(4)
memory usage: 84.0+ KB
None
<class 'pandas.core.frame.DataFrame'>
Int64Index: 384 entries, 0 to 383
Data columns (total 28 columns):
 #   Column           Non-Null Count Dtype  
 --- 
 0   state            384 non-null   object  
 1   county_name      384 non-null   object  
 2   fips             384 non-null   object  
 3   total_deaths     384 non-null   int64   
 4   covid-19_deaths  384 non-null   int64   
 5   non-hispanic_white_population 384 non-null   float64 
 6   non-hispanic_black_population 384 non-null   float64 
 7   non-hispanic_american_indian_or_alaska_native_population 384 non-null   float64 
 8   non-hispanic_asian_population 384 non-null   float64 
 9   non-hispanic_native_hawaiian_or_other_pacific_islander_population 384 non-null   float64 
 10  hispanic_population    384 non-null   float64 
 11  other_population     384 non-null   float64 
 12  non-hispanic_white_total_death 384 non-null   float64 
 13  non-hispanic_black_total_death 384 non-null   float64 
 14  non-hispanic_american_indian_or_alaska_native_total_death 384 non-null   float64 
 15  non-hispanic_asian_total_death 384 non-null   float64 
 16  non-hispanic_native_hawaiian_or_other_pacific_islander_total_death 384 non-null   float64 
 17  hispanic_total_death    384 non-null   float64 
 18  other_total_death     384 non-null   float64 
 19  non-hispanic_white_covid19_death 384 non-null   float64 

```

```

race_df_final_aggregated = race_state.loc[:, ['state', 'province_state', 'total_deaths', 'covid-19_deaths', 'non-hispanic_white_population', 'non-hispanic_black_population', 'non-hispanic_american_indian_or_alaska_native_population']]
race_df_final_aggregated.head(2)

```

	state	province_state	total_deaths	covid-19_deaths	non-hispanic_white_population	non-hispanic_black_population	non-hispanic_american_indian_or_alaska_native_population
0	AL	Alabama	1274	117	0.716		0.210
1	AL	Alabama	1349	150	0.774		0.155

```

race_df_final_aggregated_sum_groups = race_df_final_aggregated.groupby(['province_state'])['total_deaths', 'covid-19_deaths'].sum()
race_df_final_aggregated_sum = pd.DataFrame(race_df_final_aggregated_sum_groups)
race_df_final_aggregated_sum.head(2)

```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:1: FutureWarning:
```

```
Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.
```

	total_deaths	covid-19_deaths
province_state		
Alabama	25848	2710
Arizona	49863	5060

```

race_df_final_aggregated_avg_group = race_df_final_aggregated.groupby(['province_state'])['non-hispanic_white_population', 'non-hispanic_black_population', 'non-hispanic_american_indian_or_alaska_native_population'].mean()
race_df_final_aggregated_avg = pd.DataFrame(race_df_final_aggregated_avg_group)
race_df_final_aggregated_avg.head(2)

```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:1: FutureWarning:
```

```
Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.
```

	non-hispanic_white_population	non-hispanic_black_population	non-hispanic_american_indian_or_alaska_native_population	hispanic_population
province_state				
Alabama	0.607222	0.311778		0.0040
Arizona	0.532400	0.026600		0.0658

```

race_aggregated_merge = pd.merge(right=race_df_final_aggregated_avg, left=race_df_final_aggregated_sum, how='inner', left_on='province_state')
race_aggregated_merge.head(2)

```

	total_deaths	covid-19_deaths	non-hispanic_white_population	non-hispanic_black_population	non-hispanic_american_indian_or_alaska_native_population
province_state					
Alabama	25848	2710	0.607222		0.311778
Arizona	49863	5060	0.532400		0.026600

```
race_aggregated_merge.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 47 entries, Alabama to Wisconsin
Data columns (total 23 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   total_deaths      47 non-null    int64  
 1   covid-19_deaths   47 non-null    int64  
 2   non-hispanic_white_population  47 non-null    float64 
 3   non-hispanic_black_population 47 non-null    float64 
 4   non-hispanic_american_indian_or_alaska_native_population 47 non-null    float64 
 5   non-hispanic_asian_population 47 non-null    float64 
 6   non-hispanic_native_hawaiian_or_other_pacific_islander_population 47 non-null    float64 
 7   hispanic_population      47 non-null    float64 
 8   other_population        47 non-null    float64 
 9   non-hispanic_white_total_death 47 non-null    float64 
 10  non-hispanic_black_total_death 47 non-null    float64 
 11  non-hispanic_american_indian_or_alaska_native_total_death 47 non-null    float64 
 12  non-hispanic_asian_total_death 47 non-null    float64 
 13  non-hispanic_native_hawaiian_or_other_pacific_islander_total_death 47 non-null    float64 
 14  hispanic_total_death     47 non-null    float64 
 15  other_total_death       47 non-null    float64 
 16  non-hispanic_white_covid19_death 47 non-null    float64 
 17  non-hispanic_black_covid19_death 47 non-null    float64 
 18  non-hispanic_american_indian_or_alaska_native_covid19_death 47 non-null    float64 
 19  non-hispanic_asian_covid19_death 47 non-null    float64 
 20  non-hispanic_native_hawaiian_or_other_pacific_islander_covid19_death 47 non-null    float64 
 21  hispanic_covid19_death    47 non-null    float64 
 22  other_covid19_death      47 non-null    float64 
dtypes: float64(21), int64(2)
memory usage: 8.8+ KB
```

```
age_gender_df.head(2)
```

	state	sex	age_group	covid-19_deaths	total_deaths
0	Alabama	Male	0-17 years	0.0	310.0
1	Alabama	Male	18-29 years	11.0	661.0

```
age_gender_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 742 entries, 0 to 741
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   state            742 non-null    object  
 1   sex              742 non-null    object  
 2   age_group        742 non-null    object  
 3   covid-19_deaths  742 non-null    float64 
 4   total_deaths     742 non-null    float64 
dtypes: float64(2), object(3)
memory usage: 29.1+ KB
```

```
race_state.head(2)
```

```

df2_st_pop = df2.loc[:, ['fips', 'province_state', 'population']]

      0   AL    01015     1274     117       0.716      0.210
race_state_pop = pd.merge(left=race_df_final, right=df2_st_pop, how='inner', left_on='fips', right_on='fips')
race_state_pop.head(2)

  state county_name   fips total_deaths covid-19_deaths non-hispanic_white_population non-hispanic_black_population non-hispanic_american_ir
  0   AL    Calhoun County 01015        1274          117             0.716                  0.210
  1   AL    Etowah County 01055        1349          150             0.774                  0.155

total_by_state_10 = race_aggregated_merge.nlargest(10,'covid-19_deaths')
total_by_state_10

  province_state total_deaths covid-19_deaths non-hispanic_white_population non-hispanic_black_population non-hispanic_american_indian_or_alask
  New York        137939      32265            0.592941           0.133706
  Texas          150444      18010            0.443500           0.110000
  California      214091      16967            0.386500           0.049667
  Florida         174329      16251            0.627484           0.130903
  New Jersey      73995       14616            0.589263           0.119263
  Pennsylvania    83655       8414             0.747944           0.095222
  Massachusetts   52339       8294             0.731900           0.070800
  Illinois         69830      8055             0.650300           0.135200
  Michigan         58601      6209             0.718909           0.153727
  Arizona          49863      5060             0.532400           0.026600

```

10. Total Deaths and Covid-19 Deaths by Age Group in USA

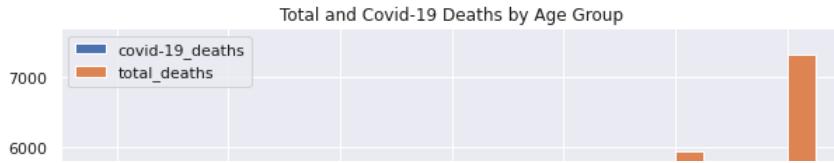
```

covid19_deaths_by_age_group = age_gender_df.groupby('age_group')[['covid-19_deaths', 'total_deaths']].mean()
sub = covid19_deaths_by_age_group.plot.bar( title='Total and Covid-19 Deaths by Age Group')
sub.set_ylabel('Covid-19 Deaths')
sub.set_xlabel('Age Group')

```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:1: FutureWarning:  
Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.
```

```
Text(0.5, 0, 'Age Group')
```

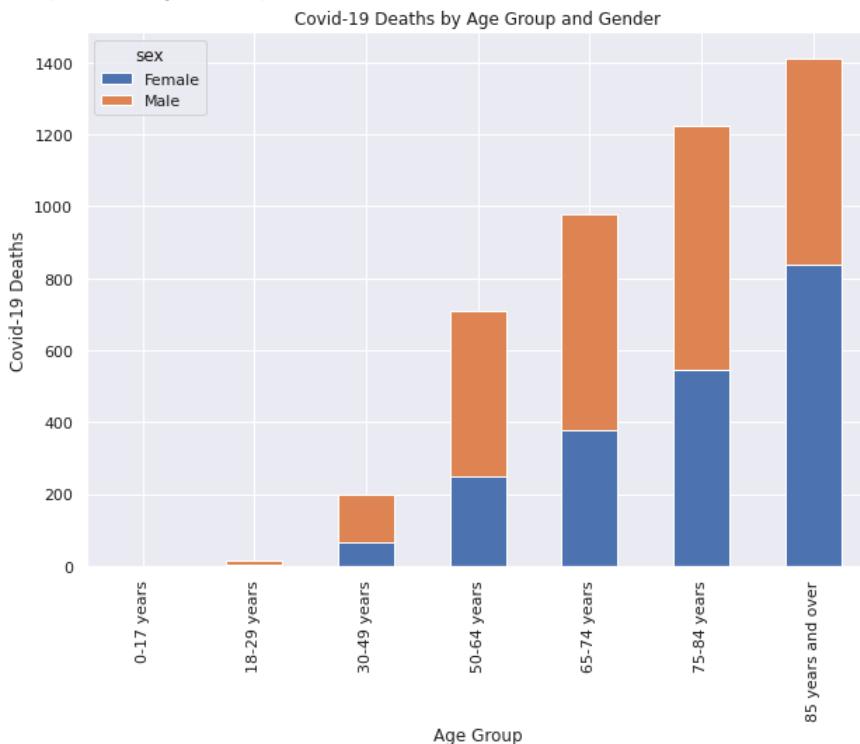


Highest Average Covid-19 Deaths are occurring in the age group **50 and above**.

11. Covid-19 Deaths by Age and Gender in USA

```
unstacked_age_gender = age_gender_df.groupby(['age_group', 'sex'])['covid-19_deaths'].mean().unstack()  
age_gender = unstacked_age_gender.plot(kind='bar', title='Covid-19 Deaths by Age Group and Gender', stacked=True)  
age_gender.set_ylabel('Covid-19 Deaths')  
age_gender.set_xlabel('Age Group')
```

```
Text(0.5, 0, 'Age Group')
```



The above analysis shows, Covid-19 deaths are maximum in the age group 50 and above. **Men** are most vulnerable while **Women** in the age group 85 and above.

```
age_gender_df.head()
```

	state	sex	age_group	covid-19_deaths	total_deaths
0	Alabama	Male	0-17 years	0.0	310.0
1	Alabama	Male	18-29 years	11.0	661.0
2	Alabama	Male	30-49 years	115.0	2193.0
3	Alabama	Male	50-64 years	425.0	5405.0
4	Alabama	Male	65-74 years	617.0	6033.0

```
unstacked_age_gender
```

sex	Female	Male
age_group		
0-17 years	0.245283	0.264151
18-29 years	4.924528	10.037736
30-49 years	64.094340	135.981132
50-64 years	247.754717	460.867925
65-74 years	379.415094	599.094340
75-84 years	546.415094	676.547170

```
pop_filter = df_melted['indicator'].str.contains('population', case=False, na=False)
death_filter = df_melted['indicator'].str.contains('all-cause deaths', case=False, na=False)
covid_death = df_melted['indicator'].str.contains('COVID-19 deaths', case=False, na=False)
```

```
df_melted_population = df_melted[pop_filter]
df_melted_population.drop(['indicator'] , axis=1 , inplace=True)
```

```
df_melted_deaths = df_melted[death_filter]
df_melted_deaths.drop(['indicator'] , axis=1 , inplace=True)
```

```
df_melted_covid_deaths = df_melted[covid_death]
df_melted_covid_deaths.drop(['indicator'] , axis=1 , inplace=True)
```

```
print(df_melted_population.head(2))
print(df_melted_deaths.head(2))
df_melted_covid_deaths.head(2)
```

fips_code	state	county_name	total_deaths	covid-19_deaths	race	percentage
2	01015	AL Calhoun County	1274	117	non-hispanic_white	0.716
5	01055	AL Etowah County	1349	150	non-hispanic_white	0.774
fips_code	state	county_name	total_deaths	covid-19_deaths	race	percentage
0	01015	AL Calhoun County	1274	117	non-hispanic_white	0.835
3	01055	AL Etowah County	1349	150	non-hispanic_white	0.847

/usr/local/lib/python3.6/dist-packages/pandas/core/frame.py:4174: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-vers

fips_code	state	county_name	total_deaths	covid-19_deaths	race	percentage
1	01015	AL Calhoun County	1274	117	non-hispanic_white	0.778
4	01055	AL Etowah County	1349	150	non-hispanic_white	0.840

```
race_melted_aggregated_merge = pd.merge(left=df_melted_population, right=df_melted_deaths, how='inner', left_on=['fips_code','county_name'], right_on=['fips_code','county_name'])
race_melted_aggregated_merge.drop(['state_y', 'total_deaths_y', 'covid-19_deaths_y'] , axis=1 , inplace=True)
race_melted_aggregated_merge.rename(columns = {'state_x':'state'}, inplace = True)
race_melted_aggregated_merge.rename(columns = {'total_deaths_x':'total_deaths'}, inplace = True)
race_melted_aggregated_merge.rename(columns = {'covid-19_deaths_x':'covid-19_deaths'}, inplace = True)
race_melted_aggregated_merge.rename(columns = {'percentage_x':'percentage_of_distributed_population'}, inplace = True)
race_melted_aggregated_merge.rename(columns = {'percentage_y':'percentage_of_total_deaths'}, inplace = True)
race_melted_aggregated_merge_final = pd.merge(left=race_melted_aggregated_merge, right=df_melted_covid_deaths, how='inner', left_on=['fips_code','county_name'], right_on=['fips_code','county_name'])
race_melted_aggregated_merge_final.drop(['state_y', 'total_deaths_y', 'covid-19_deaths_y'] , axis=1 , inplace=True)
race_melted_aggregated_merge_final.rename(columns = {'state_x':'state'}, inplace = True)
race_melted_aggregated_merge_final.rename(columns = {'total_deaths_x':'total_deaths'}, inplace = True)
race_melted_aggregated_merge_final.rename(columns = {'covid-19_deaths_x':'covid-19_deaths'}, inplace = True)
race_melted_aggregated_merge_final.rename(columns = {'percentage':'percentage_of_covid-19_deaths'}, inplace = True)
race_melted_aggregated_merge_final.head()
```

fips_code	state	county_name	total_deaths	covid-19_deaths	race	percentage_of_distributed_population	percentage_of_total_deaths
0	01015	AL Calhoun County	1274	117	non-hispanic_white	0.716	0.778
1	01055	AL Etowah County	1349	150	non-hispanic_white	0.774	0.840
2	01069	AL Houston County	1669	278	non-hispanic_white	0.660	0.660

```

test = race_melted_aggregated_merge_final.groupby(['state', 'race'])['percentage_of_covid-19_deaths'].mean()
print(test)

      state   race
      AL    hispanic        0.009667
            non-hispanic_american_indian_or_alaska_native 0.000000
            non-hispanic_asian          0.000000
            non-hispanic_black         0.363889
            non-hispanic_native_hawaiian_or_other_pacific_islander 0.000000
            non-hispanic_white          0.604333
            other                      0.000000
      AR    hispanic        0.062000
            non-hispanic_american_indian_or_alaska_native 0.000000
            non-hispanic_asian          0.000000
            non-hispanic_black         0.168000
            non-hispanic_native_hawaiian_or_other_pacific_islander 0.039286
            non-hispanic_white          0.680857
            other                      0.000000
      AZ    hispanic        0.330400
            non-hispanic_american_indian_or_alaska_native 0.217600
            non-hispanic_asian          0.007000
            non-hispanic_black         0.013200
            non-hispanic_native_hawaiian_or_other_pacific_islander 0.000600
            non-hispanic_white          0.387200
            other                      0.009400
      CA    hispanic        0.440750
            non-hispanic_american_indian_or_alaska_native 0.000125
            non-hispanic_asian          0.097958
            non-hispanic_black         0.054042
            non-hispanic_native_hawaiian_or_other_pacific_islander 0.001458
            non-hispanic_white          0.363958
            other                      0.001750
      CO    hispanic        0.263714
            non-hispanic_american_indian_or_alaska_native 0.000000
            non-hispanic_asian          0.025286
            non-hispanic_black         0.058000
            non-hispanic_native_hawaiian_or_other_pacific_islander 0.000000
            non-hispanic_white          0.607857
            other                      0.000000
      CT    hispanic        0.050667
            non-hispanic_american_indian_or_alaska_native 0.000000
            non-hispanic_asian          0.006500
            non-hispanic_black         0.095000
            non-hispanic_native_hawaiian_or_other_pacific_islander 0.000000
            non-hispanic_white          0.798000
            other                      0.000000
      DC    hispanic        0.151000
            non-hispanic_american_indian_or_alaska_native 0.000000
            non-hispanic_asian          0.016000
            non-hispanic_black         0.710000
            non-hispanic_native_hawaiian_or_other_pacific_islander 0.000000
            non-hispanic_white          0.116000
            other                      0.000000
      DE    hispanic        0.038667
            non-hispanic_american_indian_or_alaska_native 0.000000
            non-hispanic_asian          0.000000
            non-hispanic_black         0.261333
            non-hispanic_native_hawaiian_or_other_pacific_islander 0.000000
            non-hispanic_white          0.678333
            other                      0.000000
      FL    hispanic        0.158452
            non-hispanic_american_indian_or_alaska_native 0.000000
            ... -----

```

12. Covid19 Deaths by Race in USA

```

# importing the required library
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
#ax = plt.gca()
race_melted_aggregated_merge_final['distributed_population'] = race_melted_aggregated_merge_final['percentage_of_distributed_population']
df_melted_population['percentage_of_distributed_population'] = df_melted_population['percentage']*100
p = sns.lmplot(x='covid-19_deaths', y ='percentage_of_distributed_population',
                fit_reg = False, hue = 'race', data = df_melted_population)
#ax.set_title("Covid-19 vs Percentage of population on Race")
#p.fig.subtitle("Covid-19 vs Percentage of population on Race")
plt.title("Covid-19 vs Percentage of population on Race", fontsize = 15)
plt.xlabel("Covid-19 deaths". fontsize = 14)

```

```

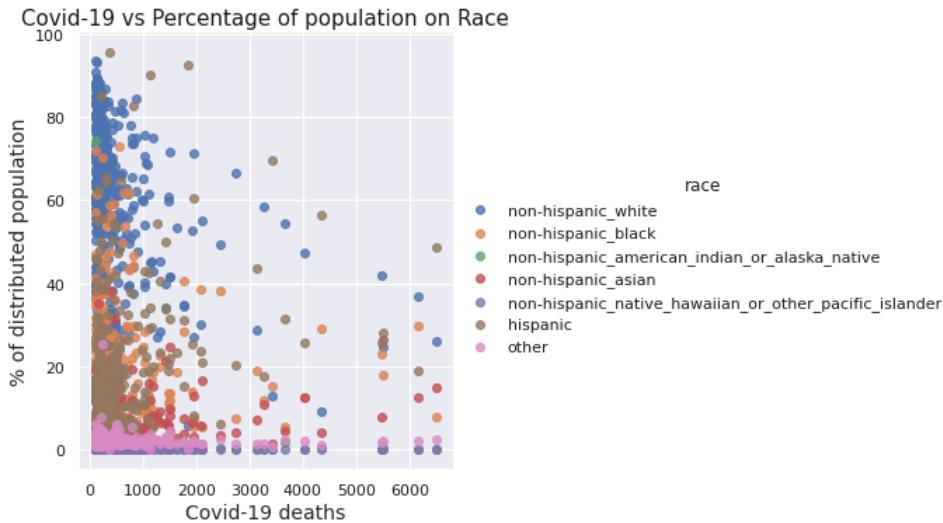
plt.ylabel("% of distributed population", fontsize = 14)
# Show the plot
plt.show()

```

/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:7: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-vers



Though white population is on the higher side, **non-hispanic native hawaiian's** and **other** group are seeing maximum Covid-19 deaths.

13. Total Deaths and Covid-19 Deaths by Race

```

race_melted_aggregated_merge_final['covid-19_deaths_by_race'] = race_melted_aggregated_merge_final['percentage_of_covid-19_deaths']* race_melted_aggregated_merge_final['total_deaths_by_race']
race_melted_aggregated_merge_final['total_deaths_by_race'] = race_melted_aggregated_merge_final['percentage_of_total_deaths']* race_melted_aggregated_merge_final['total_deaths_by_race']
#pd.to_numeric(race_melted_aggregated_merge_final['covid-19_deaths_by_race'], downcast='integer')
#pd.to_numeric(race_melted_aggregated_merge_final['total_deaths_by_race'], downcast='integer')
race_melted_aggregated_merge_final['total_deaths_by_race'] = race_melted_aggregated_merge_final['total_deaths_by_race'].astype(int)
race_melted_aggregated_merge_final['covid-19_deaths_by_race'] = race_melted_aggregated_merge_final['covid-19_deaths_by_race'].astype(int)
covid19_total_death = race_melted_aggregated_merge_final.groupby('race')[['total_deaths_by_race','covid-19_deaths_by_race']].mean()

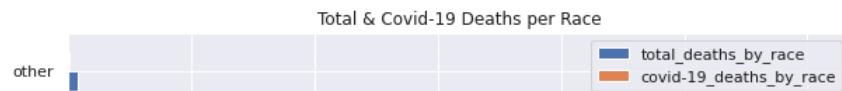
sub = covid19_total_death.plot.barh( title='Total & Covid-19 Deaths per Race')
sub.set_xlabel('Total and Covid-19 Deaths')
sub.set_ylabel('Race')
#race_melted_aggregated_merge_final.head(5)

```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:8: FutureWarning:
```

```
Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.
```

```
Text(0, 0.5, 'Race')
```



Of the total deaths maximum Covid-19 deaths are recorded by **non-hispanic blacks** followed by **hispanics**.

```
#Reading the dataset
Airlines_Monthly_Data = pd.read_csv("https://storage.googleapis.com/tgundapaneni/Airlines_Monthly_Data_1.csv")
#Displaying top 5 rows
Airlines_Monthly_Data.head()
#Change all the column names to lowercase
Airlines_Monthly_Data.columns = Airlines_Monthly_Data.columns.str.lower()
#Replacing space with -
Airlines_Monthly_Data.columns = Airlines_Monthly_Data.columns.str.strip().str.lower().str.replace(' ', '_').str.replace('(', '').str.replace(
#Replacing ,
Airlines_Monthly_Data = Airlines_Monthly_Data.replace(',', '', regex=True)
#Displaying top 5 rows
Airlines_Monthly_Data.head()
#Dispalying information on columns
Airlines_Monthly_Data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19 entries, 0 to 18
Data columns (total 35 columns):
 #   Column           Non-Null Count  Dtype  
 ---  -- 
 0   year              19 non-null      int64  
 1   month             19 non-null      object 
 2   total_scheduled Domestic flights 19 non-null      object 
 3   total Domestic flights operated 19 non-null      object 
 4   total Domestic flight cancellations 19 non-null      object 
 5   alaska_scheduled df          19 non-null      object 
 6   alaska_operated df         19 non-null      object 
 7   alaska_cancelled df        19 non-null      object 
 8   allegiant_scheduled df    19 non-null      object 
 9   allegiant_operated df     19 non-null      object 
 10  allegiant_cancelled df   19 non-null      object 
 11  american_scheduled df   19 non-null      object 
 12  american_operated df    19 non-null      object 
 13  american_cancelled df   19 non-null      object 
 14  delta_scheduled df      19 non-null      object 
 15  delta_operated df       19 non-null      object 
 16  delta_cancelled df      19 non-null      object 
 17  frontier_scheduled df  19 non-null      object 
 18  frontier_operated df   19 non-null      object 
 19  frontier_cancelled df  19 non-null      object 
 20  hawaiian_scheduled df  19 non-null      object 
 21  hawaiian_operated df   19 non-null      object 
 22  hawaiian_cancelled df  19 non-null      int64  
 23  jetblue_scheduled df   19 non-null      object 
 24  jetblue_operated df    19 non-null      object 
 25  jetblue_cancelled df   19 non-null      object 
 26  southwest_scheduled df 19 non-null      object 
 27  southwest_operated df  19 non-null      object 
 28  southwest_cancelled df 19 non-null      object 
 29  spirit_scheduled df    19 non-null      object 
 30  spirit_operated df     19 non-null      object 
 31  spirit_cancelled df    19 non-null      object 
 32  united_scheduled df   19 non-null      object 
 33  united_operated df    19 non-null      object 
 34  united_cancelled df   19 non-null      object 
dtypes: int64(2), object(33)
memory usage: 5.3+ KB
```

```
#Changing the datatype of columns to int
```

```
Airlines_Monthly_Data["total_scheduled Domestic flights"] = Airlines_Monthly_Data["total_scheduled Domestic flights"].astype(int)
Airlines_Monthly_Data["total Domestic flights operated"] = Airlines_Monthly_Data["total Domestic flights operated"].astype(int)
Airlines_Monthly_Data["total Domestic flight cancellations"] = Airlines_Monthly_Data["total Domestic flight cancellations"].astype(int)
Airlines_Monthly_Data["total Domestic flights operated"] = Airlines_Monthly_Data["total Domestic flights operated"].astype(int)
Airlines_Monthly_Data["alaska_scheduled df"] = Airlines_Monthly_Data["alaska_scheduled df"].astype(int)
Airlines_Monthly_Data["alaska_operated df"] = Airlines_Monthly_Data["alaska_operated df"].astype(int)
Airlines_Monthly_Data["frontier_scheduled df"] = Airlines_Monthly_Data["frontier_scheduled df"].astype(int)
Airlines_Monthly_Data["frontier_operated df"] = Airlines_Monthly_Data["frontier_operated df"].astype(int)
Airlines_Monthly_Data["frontier_cancelled df"] = Airlines_Monthly_Data["frontier_cancelled df"].astype(int)
Airlines_Monthly_Data["hawaiian_scheduled df"] = Airlines_Monthly_Data["hawaiian_scheduled df"].astype(int)
Airlines_Monthly_Data["hawaiian_operated df"] = Airlines_Monthly_Data["hawaiian_operated df"].astype(int)
Airlines_Monthly_Data["hawaiian_cancelled df"] = Airlines_Monthly_Data["hawaiian_cancelled df"].astype(int)
Airlines_Monthly_Data["jetblue_scheduled df"] = Airlines_Monthly_Data["jetblue_scheduled df"].astype(int)
Airlines_Monthly_Data["jetblue_operated df"] = Airlines_Monthly_Data["jetblue_operated df"].astype(int)
Airlines_Monthly_Data["jetblue_cancelled df"] = Airlines_Monthly_Data["jetblue_cancelled df"].astype(int)
Airlines_Monthly_Data["southwest_scheduled df"] = Airlines_Monthly_Data["southwest_scheduled df"].astype(int)
Airlines_Monthly_Data["southwest_operated df"] = Airlines_Monthly_Data["southwest_operated df"].astype(int)
Airlines_Monthly_Data["southwest_cancelled df"] = Airlines_Monthly_Data["southwest_cancelled df"].astype(int)
Airlines_Monthly_Data["spirit_scheduled df"] = Airlines_Monthly_Data["spirit_scheduled df"].astype(int)
Airlines_Monthly_Data["spirit_operated df"] = Airlines_Monthly_Data["spirit_operated df"].astype(int)
Airlines_Monthly_Data["spirit_cancelled df"] = Airlines_Monthly_Data["spirit_cancelled df"].astype(int)
Airlines_Monthly_Data["united_scheduled df"] = Airlines_Monthly_Data["united_scheduled df"].astype(int)
Airlines_Monthly_Data["united_operated df"] = Airlines_Monthly_Data["united_operated df"].astype(int)
Airlines_Monthly_Data["united_cancelled df"] = Airlines_Monthly_Data["united_cancelled df"].astype(int)
```

```

Airlines_Monthly_Data['alaska_cancelled_at'] = Airlines_Monthly_Data['alaska_cancelled_at'].astype(int)
Airlines_Monthly_Data["allegiant_scheduled_df"] = Airlines_Monthly_Data["allegiant_scheduled_df"].astype(int)
Airlines_Monthly_Data["allegiant_operated_df"] = Airlines_Monthly_Data["allegiant_operated_df"].astype(int)
Airlines_Monthly_Data["allegiant_cancelled_df"] = Airlines_Monthly_Data["allegiant_cancelled_df"].astype(int)
Airlines_Monthly_Data["american_scheduled_df"] = Airlines_Monthly_Data["american_scheduled_df"].astype(int)
Airlines_Monthly_Data["american_operated_df"] = Airlines_Monthly_Data["american_operated_df"].astype(int)
Airlines_Monthly_Data["american_cancelled_df"] = Airlines_Monthly_Data["american_cancelled_df"].astype(int)
Airlines_Monthly_Data["delta_scheduled_df"] = Airlines_Monthly_Data["delta_scheduled_df"].astype(int)
Airlines_Monthly_Data["delta_operated_df"] = Airlines_Monthly_Data["delta_operated_df"].astype(int)
Airlines_Monthly_Data["delta_cancelled_df"] = Airlines_Monthly_Data["delta_cancelled_df"].astype(int)
Airlines_Monthly_Data["frontier_scheduled_df"] = Airlines_Monthly_Data["frontier_scheduled_df"].astype(int)
Airlines_Monthly_Data["frontier_operated_df"] = Airlines_Monthly_Data["frontier_operated_df"].astype(int)
Airlines_Monthly_Data["frontier_cancelled_df"] = Airlines_Monthly_Data["frontier_cancelled_df"].astype(int)
Airlines_Monthly_Data["hawaiian_scheduled_df"] = Airlines_Monthly_Data["hawaiian_scheduled_df"].astype(int)
Airlines_Monthly_Data["hawaiian_operated_df"] = Airlines_Monthly_Data["hawaiian_operated_df"].astype(int)
Airlines_Monthly_Data["hawaiian_cancelled_df"] = Airlines_Monthly_Data["hawaiian_cancelled_df"].astype(int)
Airlines_Monthly_Data["jetblue_scheduled_df"] = Airlines_Monthly_Data["jetblue_scheduled_df"].astype(int)
Airlines_Monthly_Data["jetblue_operated_df"] = Airlines_Monthly_Data["jetblue_operated_df"].astype(int)
Airlines_Monthly_Data["jetblue_cancelled_df"] = Airlines_Monthly_Data["jetblue_cancelled_df"].astype(int)
Airlines_Monthly_Data["southwest_scheduled_df"] = Airlines_Monthly_Data["southwest_scheduled_df"].astype(int)
Airlines_Monthly_Data["southwest_operated_df"] = Airlines_Monthly_Data["southwest_operated_df"].astype(int)
Airlines_Monthly_Data["southwest_cancelled_df"] = Airlines_Monthly_Data["southwest_cancelled_df"].astype(int)
Airlines_Monthly_Data["spirit_scheduled_df"] = Airlines_Monthly_Data["spirit_scheduled_df"].astype(int)
Airlines_Monthly_Data["spirit_operated_df"] = Airlines_Monthly_Data["spirit_operated_df"].astype(int)
Airlines_Monthly_Data["spirit_cancelled_df"] = Airlines_Monthly_Data["spirit_cancelled_df"].astype(int)
Airlines_Monthly_Data["united_scheduled_df"] = Airlines_Monthly_Data["united_scheduled_df"].astype(int)
Airlines_Monthly_Data["united_operated_df"] = Airlines_Monthly_Data["united_operated_df"].astype(int)
Airlines_Monthly_Data["united_cancelled_df"] = Airlines_Monthly_Data["united_cancelled_df"].astype(int)

```

```

#Displaying info about columns
Airlines_Monthly_Data.info()
#Displaying data
Airlines_Monthly_Data.head(5)
#Defining the new dataset idx which shows us the maximum date in the month
idx = pd.date_range(start='2019-01', freq='M', periods=19)
idx
#Inserting idx as a column into our dataset
Airlines_Monthly_Data.insert(2,"daterange",idx,True)
#Displaying our data
Airlines_Monthly_Data.head(5)

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19 entries, 0 to 18
Data columns (total 35 columns):
 #   Column           Non-Null Count Dtype  
 --- 
 0   year            19 non-null    int64   
 1   month           19 non-null    object  
 2   total_scheduled Domestic flights 19 non-null    int64   
 3   total Domestic flights operated 19 non-null    int64   
 4   total Domestic flight cancellations 19 non-null    int64   
 5   alaska_scheduled_df 19 non-null    int64   
 6   alaska_operated_df 19 non-null    int64   
 7   alaska_cancelled_df 19 non-null    int64   
 8   allegiant_scheduled_df 19 non-null    int64   
 9   allegiant_operated_df 19 non-null    int64   
 10  allegiant_cancelled_df 19 non-null    int64   
 11  american_scheduled_df 19 non-null    int64   
 12  american_operated_df 19 non-null    int64   
 13  american_cancelled_df 19 non-null    int64   
 14  delta_scheduled_df 19 non-null    int64   
 15  delta_operated_df 19 non-null    int64   
 16  delta_cancelled_df 19 non-null    int64

```

Airlines_Monthly_Data.info()

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19 entries, 0 to 18
Data columns (total 36 columns):
 #   Column           Non-Null Count Dtype  
 --- 
 0   year            19 non-null    int64   
 1   month           19 non-null    object  
 2   daterange        19 non-null    datetime64[ns] 
 3   total_scheduled Domestic flights 19 non-null    int64   
 4   total Domestic flights operated 19 non-null    int64   
 5   total Domestic flight cancellations 19 non-null    int64   
 6   alaska_scheduled_df 19 non-null    int64   
 7   alaska_operated_df 19 non-null    int64   
 8   alaska_cancelled_df 19 non-null    int64   
 9   allegiant_scheduled_df 19 non-null    int64   
 10  allegiant_operated_df 19 non-null    int64   
 11  allegiant_cancelled_df 19 non-null    int64   
 12  american_scheduled_df 19 non-null    int64   
 13  american_operated_df 19 non-null    int64   
 14  american_cancelled_df 19 non-null    int64   
 15  delta_scheduled_df 19 non-null    int64   
 16  delta_operated_df 19 non-null    int64   
 17  delta_cancelled_df 19 non-null    int64   
 18  frontier_scheduled_df 19 non-null    int64   
 19  frontier_operated_df 19 non-null    int64   
 20  frontier_cancelled_df 19 non-null    int64   
 21  hawaiian_scheduled_df 19 non-null    int64   
 22  hawaiian_operated_df 19 non-null    int64   
 23  hawaiian_cancelled_df 19 non-null    int64   
 24  jetblue_scheduled_df 19 non-null    int64   
 25  jetblue_operated_df 19 non-null    int64   
 26  jetblue_cancelled_df 19 non-null    int64   
 27  southwest_scheduled_df 19 non-null    int64   
 28  southwest_operated_df 19 non-null    int64   
 29  southwest_cancelled_df 19 non-null    int64   
 30  spirit_scheduled_df 19 non-null    int64   
 31  spirit_operated_df 19 non-null    int64   
 32  spirit_cancelled_df 19 non-null    int64   
 33  united_scheduled_df 19 non-null    int64   
 34  united_operated_df 19 non-null    int64   
 35  united_cancelled_df 19 non-null    int64
dtypes: datetime64[ns](1), int64(34), object(1)
memory usage: 5.5+ KB

```

```

#Dividing the datasets based on year. Using grouping functions
grouped = Airlines_Monthly_Data.groupby(Airlines_Monthly_Data.year)
grouped
Airlines_Monthly_Data_2019 = grouped.get_group(2019)
Airlines_Monthly_Data_2019.head(5)
#Year 2020 data
Airlines_Monthly_Data_2020 = grouped.get_group(2020)
Airlines_Monthly_Data_2020
#Forming new dataset from Airlines_Monthly_Data_2019
Airlines_Monthly_Data_2019_OA = pd.DataFrame(Airlines_Monthly_Data_2019, columns = ['daterange', 'total_scheduled Domestic flights', ...
Airlines_Monthly_Data_2019_OA

```

```

#Forming new dataset from Airlines_Monthly_Data_2020
Airlines_Monthly_Data_2020_OA = pd.DataFrame(Airlines_Monthly_Data_2020, columns = ['daterange', 'total_scheduled Domestic flights', 'total Domestic flights operated', 'total Domestic flight cancellations'])
# convert the datetime column to a datetime type and assign it back to the column
Airlines_Monthly_Data_2019_OA.daterange = pd.to_datetime(Airlines_Monthly_Data_2019_OA.daterange)
display(Airlines_Monthly_Data_2019_OA.head())

```

	daterange	total_scheduled_domestic_flights	total_domestic_flights_operated	total_domestic_flight_cancellations
0	2019-01-31	638649	619099	19550
1	2019-02-28	582966	564614	18352
2	2019-03-31	689221	675659	13562
3	2019-04-30	668259	652533	15726
4	2019-05-31	694311	680165	14146

14. How the Flight Schedule Pattern changed from 2019 to 2020?

```

#TimeSeries Plot
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure

fig, (ax1, ax2) = plt.subplots(1, 2)

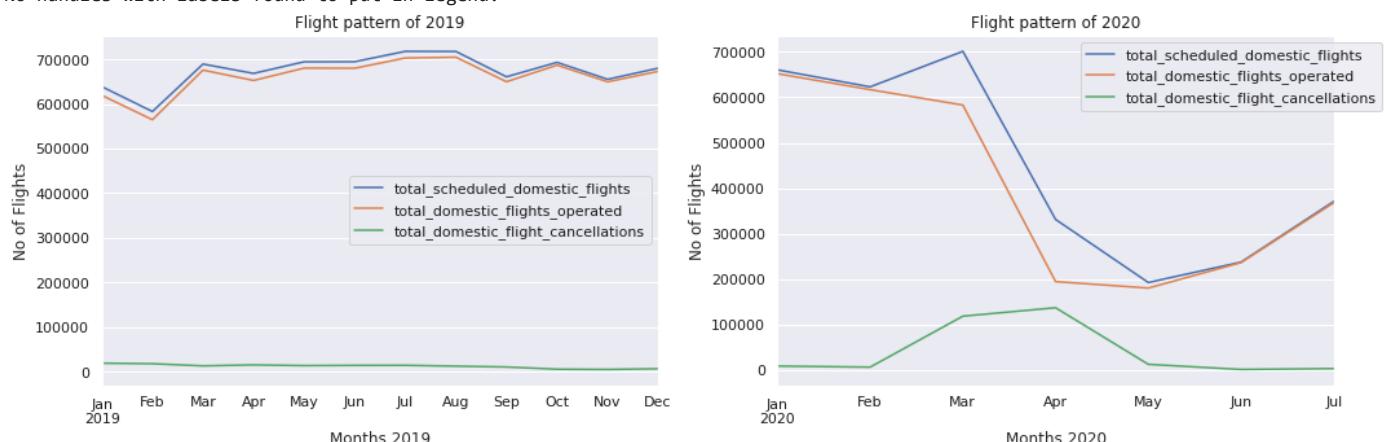
plt.title("Flight pattern of 2020")
plt.ylabel('No of Flights', fontsize='medium')

Airlines_Monthly_Data_2019_OA.plot(x='daterange', ax=ax1, figsize=(15,5), xlabel='Months 2019', ylabel='No of Flights', title='Flight |')
plt.legend(bbox_to_anchor=(1.1, 1.05))

Airlines_Monthly_Data_2020_OA.plot(x='daterange', ax=ax2, xlabel='Months 2020')
plt.legend(bbox_to_anchor=(1.1, 1.00))
plt.tight_layout()

```

No handles with labels found to put in legend.



There is a deep spike coming down in the flight schedule pattern for the year 2020. From which we can say Air travel is another measure which is affected in 2020

```

df2_grp_state1 = df2_grp_state.drop(columns=['death_covid', 'death_percentage'])
df2_grp_state1.head(5)
df2_grp_state_transposed = df2_grp_state1.T
df2_grp_state_transposed["sum"] = df2_grp_state_transposed.sum(axis=1)
df2_grp_state_transposed
df2_grp_state_transposed_wp=df2_grp_state_transposed.drop(df2_grp_state_transposed.index[0])
df2_grp_state_transposed_wp
idx1 = pd.date_range(start='2020-01', freq='M', periods=11)
idx1
df2_grp_state_transposed_wp.insert(2,"daterange",idx1,True)
df2_grp_state_transposed_wp_sum = df2_grp_state_transposed_wp[['daterange', 'sum']]

```

```

df2_grp_state_transposed_wp_sum
#Creating a new dataset by merging two datasets
tm_ds = pd.DataFrame(Airlines_Monthly_Data_2020_OA, columns = ['daterange', 'total_scheduled Domestic flights', 'total Domestic flight cancellations'])
tm_ds
df2_grp_state_transposed_wp_sum_dropped = df2_grp_state_transposed_wp_sum
df2_grp_state_transposed_wp_sum_dropped
df2_grp_state_transposed_wp_sum_dropped = df2_grp_state_transposed_wp_sum_dropped.drop(['aug_d', 'sep_d', 'oct_d', 'nov_d'])
df2_grp_state_transposed_wp_sum_dropped
df2_grp_state_transposed_wp_sum_dropped.columns = df2_grp_state_transposed_wp_sum_dropped.columns.str.replace('sum', 'total_number_of')
tm_ds = pd.merge(tm_ds, df2_grp_state_transposed_wp_sum_dropped)
tm_ds

```

	daterange	total_scheduled_domestic_flights	total_domestic_flights_operated	total_domestic_flight_cancellations	total_number_of
0	2020-01-31	660556	652140	8416	
1	2020-02-29	623103	617008	6095	
2	2020-03-31	701274	582998	118276	
3	2020-04-30	331238	194390	136848	
	2020-05-				

15-17 Impact of Covid-19 on Domestic Flight Schedules

```

import matplotlib.pyplot as plt
from matplotlib.pyplot import figure

fig, (ax1, ax2, ax3) = plt.subplots(1, 3)

plt.title("Domestic Air Travel pattern of 2020")
plt.ylabel('No of Flights', fontsize='medium')

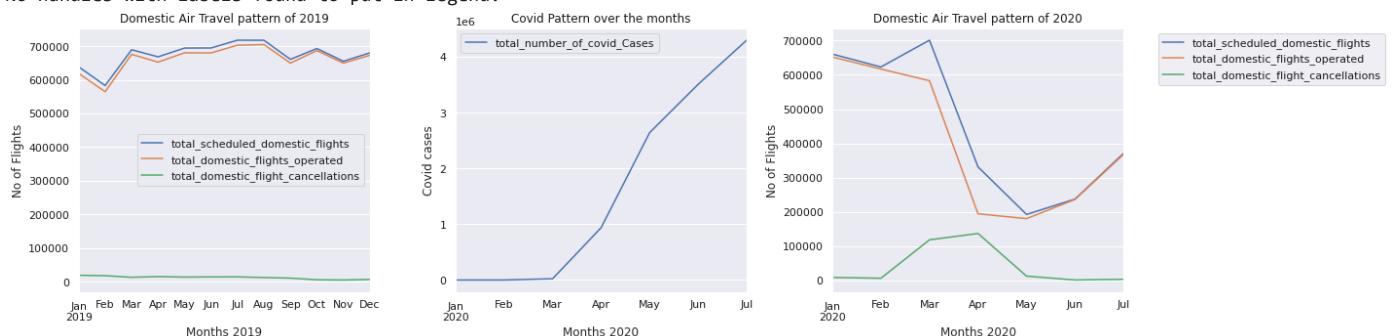
Airlines_Monthly_Data_2019_OA.plot(x='daterange', ax=ax1, figsize=(20,5), xlabel='Months 2019', ylabel='No of Flights', title='Domestic Air Travel pattern of 2019')
plt.legend(bbox_to_anchor=(1.1, 1.05))

tm_ds.plot(x='daterange', y='total_number_of_covid_Cases', ax=ax2, xlabel='Months 2020', ylabel='Covid cases', title='Covid Pattern over the months')
plt.legend(bbox_to_anchor=(1.1, 1.05))

tm_ds.plot(x='daterange', y=['total_scheduled_domestic_flights', "total_domestic_flights_operated", "total_domestic_flight_cancellations"], ax=ax3, xlabel='Months 2020', ylabel='No of Flights', title='Domestic Air Travel pattern of 2020')
plt.legend(bbox_to_anchor=(1.1, 1.05))
plt.tight_layout()

```

No handles with labels found to put in legend.
 No handles with labels found to put in legend.



From the above plots, it can be observed that in

2019, flight cancellations are close to zero and scheduled vs operated flights are following the same trend throughout the year.

2020 as Covid-19(pandemic) trend is increasing (center Plot), No of Domestic flight cancellations started to spike up rapidly while the no. of operated and scheduled flights are slashed like never before.

```
top_5_carriers = Airlines_Monthly_Data_2020
top_5_carriers
```

	year	month	daterange	total_scheduled_domestic_flights	total_domestic_flights_operated	total_domestic_flight_cancellat
12	2020	January	2020-01-31	660556	652140	
13	2020	February	2020-02-29	623103	617008	
14	2020	March	2020-03-31	701274	582998	11
15	2020	April	2020-04-30	331238	194390	13
16	2020	May	2020-05-31	192412	180151	1
17	2020	June	2020-06-30	237264	236234	
18	2020	July	2020-07-31	370859	367933	

18. Number of airline cancellations by month 2020

```
#top_5_carriers.drop("year","daterange","total_scheduled_domestic_flights", "total_domestic_flights_operated", "total_domestic_flight_cancellat")
top_5_carriers.drop(top_5_carriers.columns.difference(['month','alaska_cancelled_df', 'allegiant_cancelled_df', 'american_cancelled_df']),
top_5_carriers
df20 = top_5_carriers
df20
top_5_carriers['sum_column'] = top_5_carriers['alaska_cancelled_df'] +top_5_carriers['allegiant_cancelled_df']+ top_5_carriers['american_cancelled_df']
top_5_carriers
top_5_carriers['sum_column']
top_5_carriers['month'] = top_5_carriers['month'].str.strip()
import numpy as np
import pandas as pd
#pos = np.arange(len(top_5_carriers['month']))
plt.bar(top_5_carriers['month'],top_5_carriers['sum_column'],color='blue',edgecolor='black')
plt.xticks(rotation=90)
plt.xlabel('Month 2020', fontsize=16)
plt.ylabel('Number of flights cancelled', fontsize=16)
plt.title('Count of flights cancelled in 2020',fontsize=20)
plt.show()
```

```
/usr/local/lib/python3.6/dist-packages/pandas/core/frame.py:4174: SettingWithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-vers

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:6: SettingWithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-vers

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:9: SettingWithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-vers

Count of flights cancelled in 2020



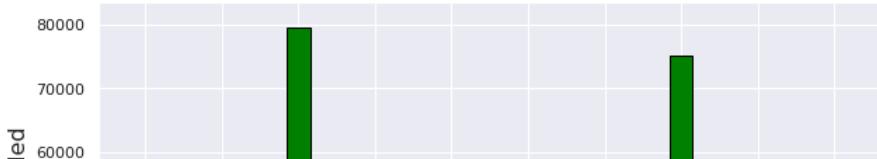
April has the more number of flight cancellations than March. So employees of Air travel are impacted because of Pandemic.



19. Top Domestic Carriers with maximum flight cancellations in USA.

```
Σ
df20
df21 = df20.T
df21
list(df21.columns)
df21['sum'] = df21[12] +df21[13]+df21[14] + df21[15] +df21[16]+df21[17]+df21[18]
df21
df22 = df21.drop(df21.index[0])
df23 = df22.drop(df22.index[10])
df23
df23.insert(0, "airline_carrier", ["Alaska", "Allegiant Air", "American", "Delta","Frontier", "Hawaiian", "JetBlue", "Southwest","Spirit"])
import numpy as np
import pandas as pd
#pos = np.arange(len(top_5_carriers['month']))
plt.bar(df23['airline_carrier'],df23['sum'],color='green',edgecolor='black',width=0.3)
plt.xticks(rotation=90)
#plt.xticks(pos, top_5_carriers['sum_column'])
plt.xlabel('Airline Carrier', fontsize=16)
plt.ylabel('Count of flights cancelled', fontsize=16)
plt.title('Domestic flights cancellation in 2020',fontsize=20)
plt.show()
```

Domestic flights cancellation in 2020



American Airlines have cancelled most of its scheduled flights from January 2020 to July 2020 regardless of the number of flights scheduled.

```

Airlines_employment = pd.read_excel('https://storage.googleapis.com/tgundapaneni/Airlines_Employment.xlsx', parse_dates=['Month'])
Airlines_employment.head(5)
Airlines_employment.columns = Airlines_employment.columns.str.lower()
Airlines_employment.columns = Airlines_employment.columns.str.strip().str.lower().str.replace(' ', '_').str.replace('(', '').str.replace(
Airlines_employment.head(5)
Airlines_employment.dtypes
from datetime import datetime
Airlines_employment['month'] = pd.to_datetime(Airlines_employment['month'], format='%m/%d/%Y') #Change the date values to date format
#Airlines_employment = datetime.strptime(Airlines_employment.month, '%m-%d-%Y').date()
#print(type(date_object))
#print(date_object)
Airlines_employment.head(5)
df10 = Airlines_employment.copy(deep=True) #Copy the data frame to another dataframe to sum all the date values
df10.drop(['full_time','part_time'] , axis=1 , inplace=True) #To the Copied data frame drop all columns other than date values
df11 = df10.groupby(['carrier_name', 'month']).sum('total').reset_index() #Group by carrier_name and month to sum the total for each c
df11.head(5)

```

	carrier_name	month	total
0	Alaska	2019-01-31	17502
1	Alaska	2019-02-28	17567
2	Alaska	2019-03-31	17590
3	Alaska	2019-04-30	17620
4	Alaska	2019-05-31	17685

```

df_alaska = df11[df11['carrier_name'] == "Alaska"]
df_allegiant = df11[df11['carrier_name'] == "Allegiant Air"]
df_americana = df11[df11['carrier_name'] == "American"]
df_delta = df11[df11['carrier_name'] == "Delta"]
df_frontier = df11[df11['carrier_name'] == "Frontier"]
df_hawaiian = df11[df11['carrier_name'] == "Hawaiian"]
df_jetblue = df11[df11['carrier_name'] == "JetBlue"]
df_southwest = df11[df11['carrier_name'] == "Southwest"]
df_spirit = df11[df11['carrier_name'] == "Spirit"]
df_united = df11[df11['carrier_name'] == "United"]
df_alaska.head(5)

```

	carrier_name	month	total
0	Alaska	2019-01-31	17502
1	Alaska	2019-02-28	17567
2	Alaska	2019-03-31	17590
3	Alaska	2019-04-30	17620
4	Alaska	2019-05-31	17685

20. Number of Employees reduced in US Domestic Airline Carriers in 2020.

```

sns.set(font_scale=1.4)
df_alaska.set_index('month')['total'].plot(figsize=(12, 10), linewidth=2.5, color='maroon')
df_allegiant.set_index('month')['total'].plot(figsize=(12, 10), linewidth=2.5, color='blue')
df_americana.set_index('month')['total'].plot(figsize=(12, 10), linewidth=2.5, color='red')
df_delta.set_index('month')['total'].plot(figsize=(12, 10), linewidth=2.5, color='green')
df_frontier.set_index('month')['total'].plot(figsize=(12, 10), linewidth=2.5, color='pink')
df_hawaiian.set_index('month')['total'].plot(figsize=(12, 10), linewidth=2.5, color='purple')

```

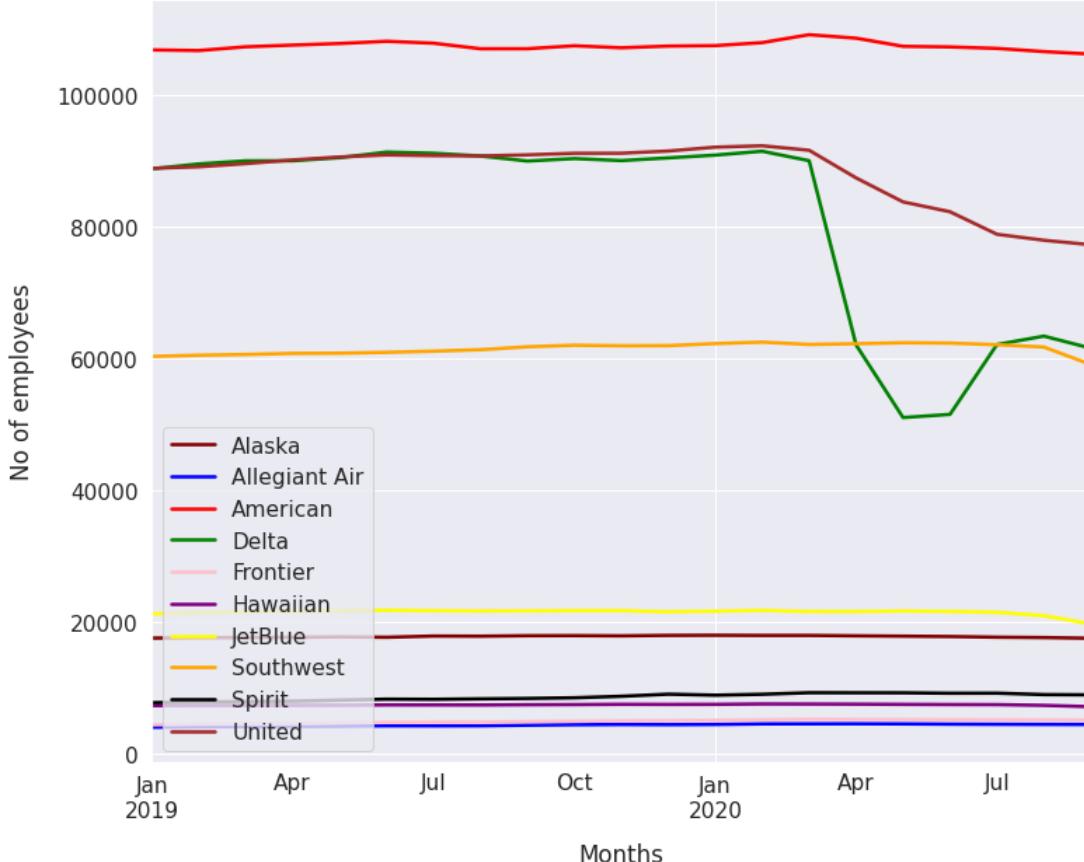
```

df_jetblue.set_index('month')['total'].plot(figsize=(12, 10), linewidth=2.5, color='yellow')
df_southwest.set_index('month')['total'].plot(figsize=(12, 10), linewidth=2.5, color='orange')
df_spirit.set_index('month')['total'].plot(figsize=(12, 10), linewidth=2.5, color='black')
df_united.set_index('month')['total'].plot(figsize=(12, 10), linewidth=2.5, color='brown')
plt.xlabel("Months", labelpad=15)
plt.ylabel("No of employees", labelpad=15)
plt.title("Employees trend in airline carriers 2020", y=1.02, fontsize=22);
plt.legend(['Alaska', 'Allegiant Air', 'American', 'Delta', 'Frontier', 'Hawaiian', 'JetBlue', 'Southwest', 'Spirit', 'United'], loc=3

```

<matplotlib.legend.Legend at 0x7f75aad50898>

Employees trend in airline carriers 2020



Maximum number of employees are laid off in Delta carrier followed by Alaska.

```
unemploy = pd.read_excel("https://storage.googleapis.com/tgundapaneni/County%20Unemployment%20Data.xlsx")
```

```
unemploy.head(5)
```

	LAUS_Area_Code	State	County	Area_Title	Period	Civilian_Labor_Force	Employed	Unemployment_Level	Unemployment_Rate
0	CN0100100000000	1	1	Autauga County, AL	Aug-19	26079	25368	711	2.7
1	CN0100300000000	1	3	Baldwin County, AL	Aug-19	97939	95367	2572	2.6
2	CN0100500000000	1	5	Barbour County, AL	Aug-19	8652	8322	330	3.8

```
unemploy.columns = unemploy.columns.str.lower()
unemploy.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45066 entries, 0 to 45065
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype  
---  -- 
 0   laus_area_code    45066 non-null   object 

```

```

1 state          45066 non-null int64
2 county         45066 non-null int64
3 area_title    45066 non-null object
4 period         45066 non-null object
5 civilian_labor_force 45066 non-null object
6 employed       45066 non-null object
7 unemployment_level 45066 non-null object
8 unemployment_rate   45066 non-null object
dtypes: int64(2), object(7)
memory usage: 3.1+ MB

```

```

unemploy['county'], unemploy['state'] = unemploy['area_title'].str.split(',', 1).str
unemploy.head()

```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:1: FutureWarning:
```

Columnar iteration over characters will be deprecated in future releases.

	laus_area_code	state	county	area_title	period	civilian_labor_force	employed	unemployment_level	unemployment_rate
0	CN0100100000000	AL	Autauga County	Autauga County, AL	Aug-19	26079	25368	711	2.7
1	CN0100300000000	AL	Baldwin County	Baldwin County, AL	Aug-19	97939	95367	2572	2.6
2	CN0100500000000	AL	Barbour County	Barbour County, AL	Aug-19	8652	8322	330	3.8

```

unemploy['county1'] = unemploy['county'].str.replace(' County', '')
unemploy['county2'] = unemploy['county1'].str.strip()
unemploy.head()

```

	laus_area_code	state	county	area_title	period	civilian_labor_force	employed	unemployment_level	unemployment_rate	county2
0	CN0100100000000	AL	Autauga County	Autauga County, AL	Aug-19	26079	25368	711	2.7	A
1	CN0100300000000	AL	Baldwin County	Baldwin County, AL	Aug-19	97939	95367	2572	2.6	E
2	CN0100500000000	AL	Barbour County	Barbour County, AL	Aug-19	8652	8322	330	3.8	E

```
unemploy['county2'].str.len()
```

```

0      7
1      7
2      7
3      4
4      6
..
45061  19
45062  17
45063  18
45064  17
45065  15
Name: county2, Length: 45066, dtype: int64

```

```

unemploy.drop(['laus_area_code','county','area_title','county1'] , axis=1 , inplace=True)
unemploy.head()

```

	state	period	civilian_labor_force	employed	unemployment_level	unemployment_rate	county2
0	AL	Aug-19	26079	25368	711	2.7	Autauga
1	AL	Aug-19	97939	95367	2572	2.6	Baldwin
2	AL	Aug-19	8652	8322	330	3.8	Barbour
3	AL	Aug-19	8670	8403	267	3.1	Bibb
4	AL	Aug-19	25309	24641	668	2.6	Blount

```

unemploy['county2'] = unemploy['county2'].str.replace('Borough', '')
unemploy['county2'] = unemploy['county2'].str.strip()
unemploy.head()

```

	state	period	civilian_labor_force	employed	unemployment_level	unemployment_rate	county2
0	AL	Aug-19	26079	25368	711	2.7	Autauga
1	AL	Aug-19	97939	95367	2572	2.6	Baldwin
2	AL	Aug-19	8652	8322	330	3.8	Barbour
3	AL	Aug-19	8670	8403	267	3.1	Bibb
4	AL	Aug-19	25309	24641	668	2.6	Blount

```
unemploy['state'] = unemploy['state'].str.strip()
#unemploy[unemploy.state == 'AK']
```

```
unemploy['county2'] = unemploy['county2'].str.replace('Census Area', '')
unemploy['county2'] = unemploy['county2'].str.replace('/city', '')
unemploy['county2'] = unemploy['county2'].str.replace('/municipality', '')
unemploy['county2'] = unemploy['county2'].str.replace('Municipality', '')
unemploy['county2'] = unemploy['county2'].str.strip()
#unemploy[unemploy.state == 'AK']
```

```
unemploy['civilian_labor_force'] = unemploy.civilian_labor_force.astype(str).str.replace('-', '0')
unemploy['employed'] = unemploy.employed.astype(str).str.replace('-', '0')
unemploy['unemployment_level'] = unemploy.unemployment_level.astype(str).str.replace('-', '0')
unemploy['unemployment_rate'] = unemploy.unemployment_rate.astype(str).str.replace('-', '0')
#unemploy[unemploy.state == 'PR']
```

```
unemploy["civilian_labor_force"] = unemploy["civilian_labor_force"].astype(int)
unemploy["employed"] = unemploy["employed"].astype(int)
unemploy["unemployment_level"] = unemploy["unemployment_level"].astype(int)
unemploy["unemployment_rate"] = unemploy["unemployment_rate"].astype(float)
unemploy['period'] = unemploy['period'].str.strip()
unemploy_mi = unemploy[(unemploy.state == 'MI')]
unemploy_mi_apr = unemploy_mi[(unemploy_mi.period == 'Apr-20')]
#unemploy_mi_apr
```

```
d = {'civilian_labor_force': 'sum', 'employed': 'sum', 'unemployment_level': 'sum', 'unemployment_rate': 'mean'}
unemploy1 = unemploy.groupby(['state', 'period'], as_index=False).aggregate(d).reindex(columns=unemploy.columns)
#unemploy1[unemploy1.state=='PR']
```

```
unemploy1 = unemploy1.drop(columns='county2')
#unemploy1[unemploy1.state=='PR']
```

```
unemploy1[unemploy1.civilian_labor_force== 0]
```

	state	period	civilian_labor_force	employed	unemployment_level	unemployment_rate
532	PR	Apr-20	0	0	0	0.0
540	PR	Mar-20	0	0	0	0.0

```
unemploy1['civilian_labor_force'].replace(to_replace=0, method='bfill', inplace=True)
unemploy1['employed'].replace(to_replace=0, method='bfill', inplace=True)
unemploy1['unemployment_level'].replace(to_replace=0, method='bfill', inplace=True)
unemploy1['unemployment_rate'].replace(to_replace=0, method='bfill', inplace=True)
unemploy1[unemploy1.state=='PR']
```

	state	period	civilian_labor_force	employed	unemployment_level	unemployment_rate
532	PR	Apr-20	1078606	1002850	75756	8.411538
533	PR	Aug-19	1078606	1002850	75756	8.411538
534	PR	Aug-20	1048936	960456	88480	9.023077
535	PR	Dec-19	1048334	952351	95983	11.053846
536	PR	Feb-20	1038061	957322	80739	9.747436
537	PR	Jan-20	1032481	933739	98742	11.747436
538	PR	Jul-20	1042135	966377	75758	7.814103

```
unemploy1.columns = ['postal_code','period','civilian_labor_force', 'employed', 'unemployed', 'unemployment_rate']
unemploy1.head(5)
```

	postal_code	period	civilian_labor_force	employed	unemployed	unemployment_rate
0	AK	Apr-20	334391	288432	45959	14.524138
1	AK	Aug-19	353858	335052	18806	6.462069
2	AK	Aug-20	335367	314014	21353	6.755172
3	AK	Dec-19	341668	320803	20865	9.706897
4	AK	Feb-20	338950	319054	19896	10.003448

```
unemploy1['period'] = unemploy1['period'].str.replace(r"\(.?\)", "")
unemploy1['period'] = unemploy1['period'].str.strip()
unemploy19 = unemploy1[unemploy1['period'].str.contains('19')]
unemploy19['period'] = unemploy19['period'].str.replace(r"\(.?\)", "")
unemploy19['period'] = unemploy19['period'].str.strip()
unemploy2 = unemploy1[unemploy1['period'].str.contains('20')]
unemploy2['period'] = unemploy2['period'].str.replace(r"\(.?\)", "")
unemploy2['period'] = unemploy2['period'].str.strip()
unemploy2.head(5)
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel launcher.py:4: SettingWithCopyWarning:
```

```
unemploy_mi = unemploy2[(unemploy2.postal_code == 'MI')]  
unemploy_mi_apr = unemploy_mi[(unemploy_mi.period == 'Apr-20')]  
unemploy_mi_apr
```

postal_code	period	civilian_labor_force	employed	unemployed	unemployment_rate
294	MI Apr-20	4553125	3476660	1076465	25.539759

```
Final_pay = pd.read_excel("https://storage.googleapis.com/tgundapaneni/Final_Pay.xlsx")  
Final_pay.head()
```

	State	2020-03-01 00:00:00	2020-04-01 00:00:00	2020-05-01 00:00:00	2020-06-01 00:00:00	2020-07-01 00:00:00	2020-08-01 00:00:00	2020-09-01 00:00:00
0	Alabama	1467	4812	7455	36832	50345	18055	18281
1	Alaska	975	1259	1306	680	2856	3789	4748
2	Arizona	1604	2002	2237	3720	6734	4267	4889
3	Arkansas	1196	1341	2403	2564	26293	14167	5366
..

```
State_code = pd.read_excel ("https://storage.googleapis.com/tgundapaneni/US_StateCodes.xlsx")  
State_code.head()
```

	State/District	Postal Code
0	Alabama	AL
1	Alaska	AK
2	Arizona	AZ
3	Arkansas	AR
4	California	CA

```
State_code.columns = ['State', 'postal_code']  
State_code.head()
```

	State	postal_code
0	Alabama	AL
1	Alaska	AK
2	Arizona	AZ
3	Arkansas	AR
4	California	CA

```
Final_state = pd.merge(State_code,Final_pay)  
Final_state.head()
```

	State	postal_code	2020-03-01 00:00:00	2020-04-01 00:00:00	2020-05-01 00:00:00	2020-06-01 00:00:00	2020-07-01 00:00:00	2020-08-01 00:00:00	2020-09-01 00:00:00
0	Alabama	AL	1467	4812	7455	36832	50345	18055	18281
1	Alaska	AK	975	1259	1306	680	2856	3789	4748
2	Arizona	AZ	1604	2002	2237	3720	6734	4267	4889
3	Arkansas	AR	1196	1341	2403	2564	26293	14167	5366
..

```
Final_state.columns = ['state','postal_code','Mar-20','Apr-20','May-20','Jun-20','Jul-20','Aug-20','Sep-20']  
Final_state.head()
```

	state	postal_code	Mar-20	Apr-20	May-20	Jun-20	Jul-20	Aug-20	Sep-20
0	Alabama	AL	1467	4812	7455	36832	50345	18055	18281
1	Alaska	AK	975	1259	1306	680	2856	3789	4748

```
Final_state_merge= Final_state.melt(id_vars=['postal_code','state'], value_vars=('Mar-20','Apr-20','May-20','Jun-20','Jul-20','Aug-20')
Final_state_merge.head(5)
```

	postal_code	state	period	claims_count
0	AL	Alabama	Mar-20	1467
1	AK	Alaska	Mar-20	975
2	AZ	Arizona	Mar-20	1604
3	AR	Arkansas	Mar-20	1196
4	CA	California	Mar-20	35608

```
unemploy_claim = pd.merge(Final_state_merge,unemploy2, how='inner', left_on=['postal_code','period'], right_on = ['postal_code','period'])
unemploy_claim.head()
```

	postal_code	state	period	claims_count	civilian_labor_force	employed	unemployed	unemployment_rate
0	AL	Alabama	Mar-20	1467	2227107	2160065	67042	3.483582
1	AK	Alaska	Mar-20	975	340654	320701	19953	9.086207
2	AZ	Arizona	Mar-20	1604	3588452	3368002	220450	8.353333
3	AR	Arkansas	Mar-20	1196	1378024	1311081	66943	5.764000
4	CA	California	Mar-20	35608	19192117	18085711	1106406	7.598276

21. Unemployment Rate by State in USA in 2020

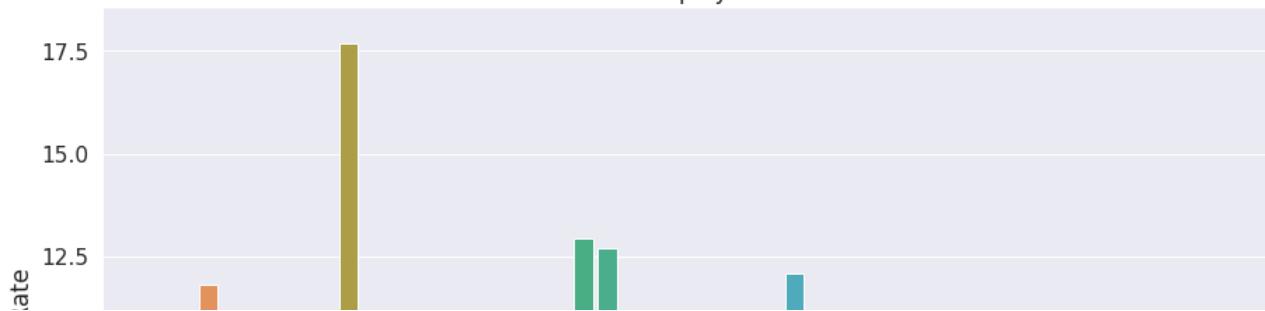
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

unemploy_claim_grp = unemploy_claim.groupby(['state']).mean('unemployment_rate').reset_index()

plt.figure(figsize=(15,10))
ax = sns.barplot(x="state", y="unemployment_rate", data=unemploy_claim_grp)
ax.set_xticklabels(ax.get_xticklabels(), rotation=90, fontsize = 10)
ax.set_xlabel('USA states', ylabel='Unemployment Rate')
ax.set_title("Statewise - Unemployment Rate USA")
```

Text(0.5, 1.0, 'Statewise - Unemployment Rate USA')

Statewise - Unemployment Rate USA



The maximum unemployment rate is recorded in **Hawaii** followed by **California, Massachusetts, Michigan, New Jersey**.

Thus proving our Hypothesis Wrong. Texas Currently is recording Maximum Covid-Cases with New Jersey having highest mortality rate and Hawaii with maximum unemployment rate



22. Trend of Unemployment Rate in USA from Aug-19 to Sep-20.



```
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import dates
import seaborn as sns

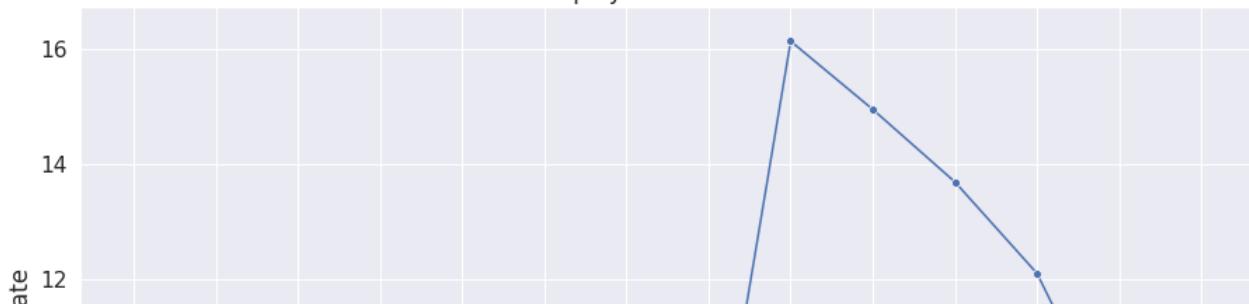
unemploy_claim_cali_19 = unemploy1[(unemploy1.postal_code == 'CA')].copy(deep=True).reset_index()

unemploy_claim_cali_19_il = pd.DataFrame()
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[1])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[12])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[11])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[10])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[3])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[5])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[4])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[8])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[0])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[9])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[7])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[6])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[2])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[13])

plt.figure(figsize=(15,10))
ax = sns.lineplot(data=unemploy_claim_cali_19_il, x='period', y='unemployment_rate', marker="o")
ax.set(xlabel='Unemployment Period', ylabel='Unemployment Rate')
ax.set(title="Unemployment Rate in USA")
```

```
[Text(0.5, 1.0, 'Unemployment Rate in USA')]
```

Unemployment Rate in USA



A steep increase in unemployment rate is seen from Feb-20 until April-20 and then is gradually coming down.



23. Unemployment Rate across USA in April 2020



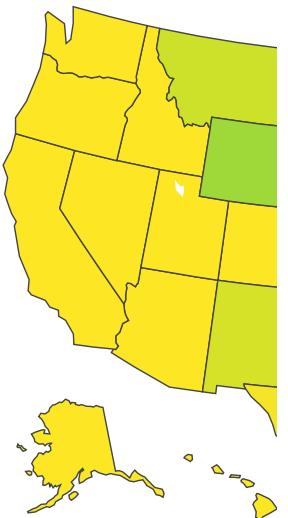
```
import plotly.express as px # Be sure to import express

unemploy_claim_apr = unemploy_claim[(unemploy_claim.period == 'Apr-20')].copy(deep=True)

fig = px.choropleth(unemploy_claim_apr, # Input Pandas DataFrame
                     locations="postal_code", # DataFrame column with locations
                     color="unemployment_rate", # DataFrame column with color values
                     hover_name="postal_code", # DataFrame column hover info
                     locationmode = 'USA-states',
                     color_continuous_scale="Viridis",
                     range_color=(0, 10)) # Set to plot as US States

fig.update_layout(
    title_text = 'Apr 2020 - Unemployment Rate in USA', # Create a Title
    geo_scope='usa', # Plot only the USA instead of globe
)
fig.show()
```

Apr 2020 - Unemployment Rate in USA



Central America is has recorded least unemployment rate when compared to other States.

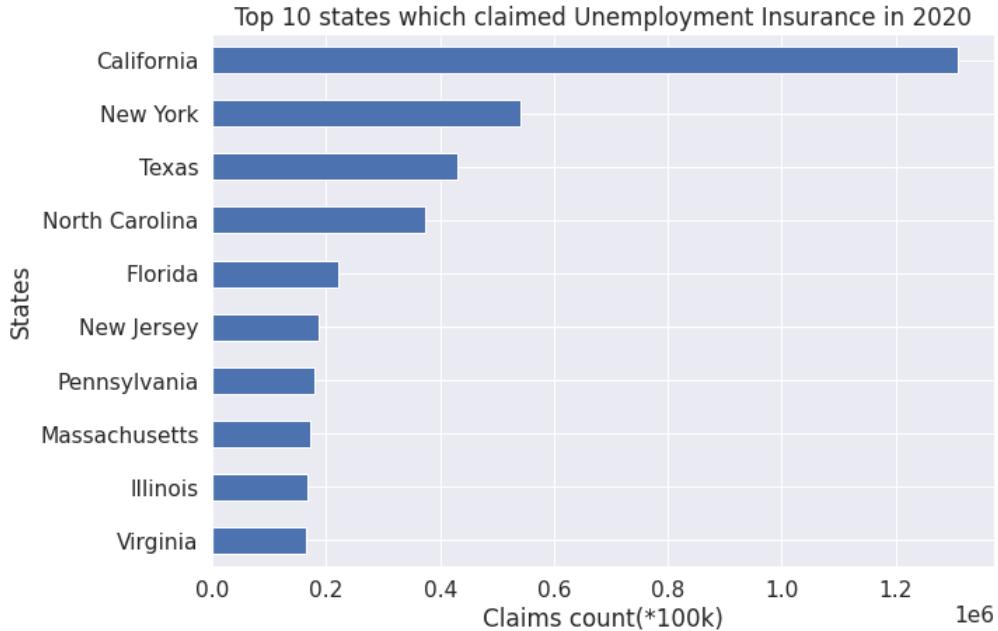
24. Top 10 states claimed Unemployment Insurance in 2020

```

state_group = unemploy_claim.groupby('state')
Top10_claims= state_group['claims_count'].sum()
Top10_claims= (Top10_claims.nlargest(10))
Top10_claims= Top10_claims.sort_values()
Top10_claims.plot.barh()
plt.title('Top 10 states which claimed Unemployment Insurance in 2020')
plt.ylabel('States')
plt.xlabel('Claims count(*100k)')

```

Text(0.5, 0, 'Claims count(*100k)')



California is seeing approximately 1.3 million of unemployment claims and stays at top in giving/receiving unemployment benefits

25-28. Trend of Unemployment Rate and number of Unemployment claims received in California

```

import matplotlib.pyplot as plt
from matplotlib.pyplot import figure

unemploy_claim_cali_19 = unemploy1[(unemploy1.postal_code == 'CA')].copy(deep=True).reset_index()

unemploy_claim_cali_19_il = pd.DataFrame()
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[1])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[12])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[11])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[10])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[3])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[5])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[4])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[8])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[0])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[9])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[7])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[6])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[2])
unemploy_claim_cali_19_il = unemploy_claim_cali_19_il.append(unemploy_claim_cali_19.loc[13])

unemploy_claim_cali = unemploy_claim[(unemploy_claim.state == 'California')].copy(deep=True)

df2_cali = df2[(df2.province_state == 'California')].copy(deep=True)
df2_cali.drop(["fips", "admin2", "lat", "long", "combined_key", "population", "death_percentage", "death_covid"] , axis=1 , inplace=True)
cols=df2_cali.filter(regex=':*/20').columns #Fetch all the date column names
df2_cali_melt = df2_cali.melt(id_vars=['province_state'], value_vars=(cols), value_name='count', var_name='date')

fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2,2)

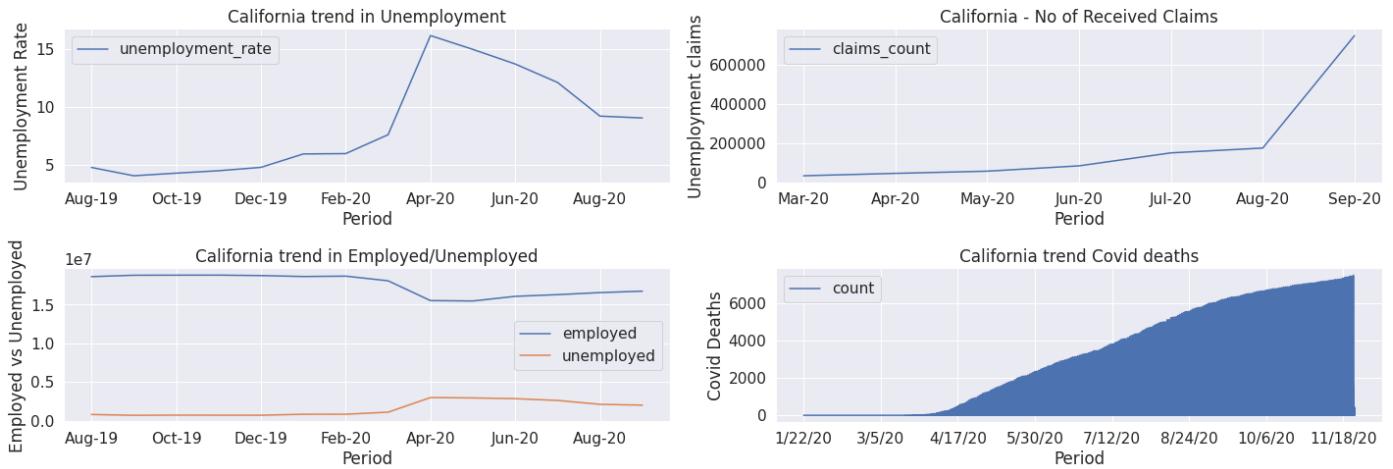
unemploy_claim_cali_19_il.plot(x='period', y='unemployment_rate', ax=ax1, figsize=(20,7), xlabel='Period', ylabel='Unemployment Rate',

```

```

unemploy_claim_cali.plot(x='period', y='claims_count', ax=ax2, xlabel='Period', ylabel='Unemployment claims', title='California - No of Received Claims')
unemploy_claim_cali_19_il.plot(x='period', y=['employed','unemployed'], ax=ax3, xlabel='Period', ylabel='Employed vs Unemployed', title='Employed vs Unemployed')
df2_cali_melt.plot(x='date', y='count', ax=ax4, xlabel='Period', ylabel='Covid Deaths', title='California trend Covid deaths')
plt.tight_layout()

```



From the above plots, it can be observed that

The unemployment trend started to go up in Dec-19 and spiked up sharply starting Mar-2020 (top-left Plot)

No of Unemployment Received Claims gradually increased by Mar-20 and a spike occurred in Aug-2020 - tells us that more people started using unemployment benefits.(Top-right plot)

Starting Mar-2020 employment started decreasing and unemployment rate increasing, but beginning June-2020 employment is showing slight increase. (bottom-left plot)

(bottom-right plot) is showing the Covid-Trend during these timelines

29. Covid Trend in Mar-20,April-20, Sep-20 in USA

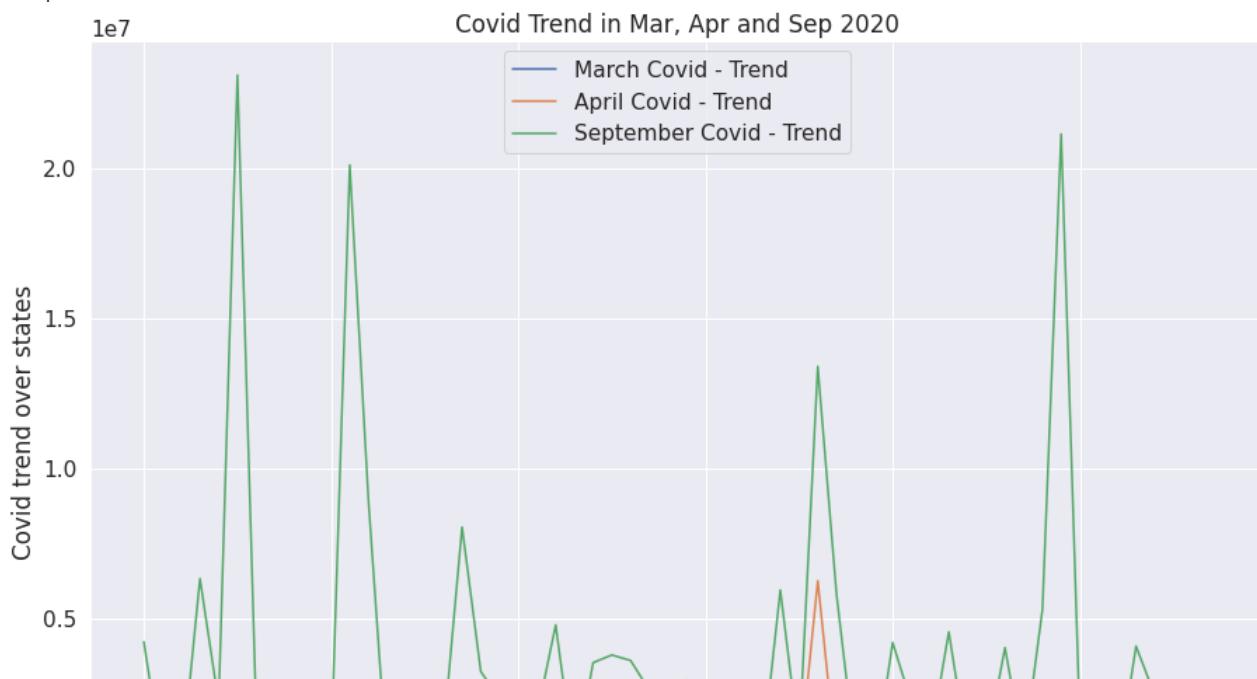
```

df1_grp_state_cpy = df1_grp_state.copy(deep=True)

df1_grp_state_cpy.plot(x='province_state', y=['mar_c','apr_c', 'sep_c'], kind='line', figsize=(15,10))
plt.title("Covid Trend in Mar, Apr and Sep 2020")
plt.xlabel('US States', fontsize=10)
plt.ylabel('Covid trend over states', fontsize='medium')
plt.legend(bbox_to_anchor=(1.1, 1.05))
plt.legend(['March Covid - Trend', "April Covid - Trend", 'September Covid - Trend'])
plt.Formatter()

```

<matplotlib.ticker.Formatter at 0x7f75a9dab828>



From the above plot, COVID-19 trend is alarmingly increasing in Sep-20 but from the above analysis unemployment rate is showing a downward trend in Sep-20, Hoping a Better Market for all of us.

...

CONCLUSION

1. Texas is recording the highest number of Covid Cases, while New Jersey is showing the highest Mortality Rate and it is in Red Zone.
2. Analysis Shows that 45% of the total Covid-Deaths are recorded Metro Areas and maximum are in the age-group of 50 and above. Male are most vulnerable to Covid-19 deaths.
3. Of the Total Covid-Deaths recorded Non-Hispanic Native Hawaiian's have highest mortality Rate.
4. With Covid-19 Increasing, the number of Airline cancellations are spiked up and also scheduled and operated flights have been slashed down like never before. Delta carrier has fired maximum number of employees.
5. With Covid-19 increasing, unemployment rate spiked up in March-20 and with Covid-19 still showing increasing trend in Sep-20, unemployment claims started rapidly increasing (California on Top) and unemployment rate is coming down gradually which means Market is opening up and is a Good Sign for all of us.