Problem Statement:-

1) Analyzing the covid data statewise to get a clear picture of number of tests conducted and positive cases in the year 2020 and 2021

In [52]:

```
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.graph_objects as go
import plotly.express as px
import pandas_profiling
import seaborn as sns
%matplotlib inline

import matplotlib
sns.set_style('darkgrid')
matplotlib.rcParams['font.size'] = 14
matplotlib.rcParams['figure.figsize'] = (9,5)
```

Data Acquisition and Description:-

In [47]:

```
covid_df = pd.read_csv('StatewiseTestingDetails.csv' )
covid_df
```

Out[47]:

	Date	State	TotalSamples	Negative	Positive
0	2020-04-17	Andaman and Nicobar Islands	1403.0	1210	12.0
1	2020-04-24	Andaman and Nicobar Islands	2679.0	NaN	27.0
2	2020-04-27	Andaman and Nicobar Islands	2848.0	NaN	33.0
3	2020-05-01	Andaman and Nicobar Islands	3754.0	NaN	33.0
4	2020-05-16	Andaman and Nicobar Islands	6677.0	NaN	33.0
...	...	...	...	...	...
16331	2021-08-06	West Bengal	15999961.0	NaN	NaN
16332	2021-08-07	West Bengal	16045662.0	NaN	NaN
16333	2021-08-08	West Bengal	16092192.0	NaN	NaN
16334	2021-08-09	West Bengal	16122345.0	NaN	NaN
16335	2021-08-10	West Bengal	16162814.0	NaN	NaN

16336 rows × 5 columns

In [69]:

```
covid_df['Date'] = pd.to_datetime(covid_df['Date'])
covid_df['Month_name'] = covid_df['Date'].dt.month_name()
covid_df
```

Out[69]:

	Date	State	TotalSamples	Negative	Positive	Month_name
0	2020-04-17	Andaman and Nicobar Islands	1403.0	1210.0	12.0	April
1	2020-04-24	Andaman and Nicobar Islands	2679.0	NaN	27.0	April

	Date	State	TotalSamples	Negative	Positive	Month_name
2	2020-04-27	Andaman and Nicobar Islands	2848.0	NaN	33.0	April
3	2020-05-01	Andaman and Nicobar Islands	3754.0	NaN	33.0	May
4	2020-05-16	Andaman and Nicobar Islands	6677.0	NaN	33.0	May
...	...	...	...	...	...	...
16331	2021-08-06	West Bengal	15999961.0	NaN	NaN	August
16332	2021-08-07	West Bengal	16045662.0	NaN	NaN	August
16333	2021-08-08	West Bengal	16092192.0	NaN	NaN	August
16334	2021-08-09	West Bengal	16122345.0	NaN	NaN	August
16335	2021-08-10	West Bengal	16162814.0	NaN	NaN	August

16336 rows × 6 columns

In [20]:

```
covid_df.shape
```

Out[20]:

```
(16336, 5)
```

Data Description:-

In [43]:

```
covid_df.describe()
```

Out[43]:

	TotalSamples	Negative	Positive
count	1.633500e+04	1.633400e+04	1.633500e+04
mean	5.376795e+06	7.972548e+05	1.959308e+04
std	8.780506e+06	2.464614e+06	1.021048e+05
min	5.800000e+01	0.000000e+00	0.000000e+00
25%	1.729730e+05	0.000000e+00	0.000000e+00
50%	9.311430e+05	0.000000e+00	0.000000e+00
75%	7.285036e+06	2.954118e+05	7.460000e+02
max	6.789786e+07	8.356103e+07	1.638961e+06

Data Information:- Column Negative contains string values which needs to be converted to float values

In [62]:

```
covid_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16336 entries, 0 to 16335
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Date          16336 non-null  object 
 1   State         16336 non-null  object 
 2   TotalSamples  16336 non-null  float64
 3   Negative      6969 non-null   object 
 4   Positive      5662 non-null   float64
dtypes: float64(2), object(3)
memory usage: 638.2+ KB
```

Data Pre-profiling:- 1) There are 20041 missing cells and 1 duplicate row. 2) 3 Categorical values and 2 numeric values

In [4]:

```
covid_Profile=pandas_profiling.ProfileReport(covid_df)
covid_Profile.to_file("Covidata_Before_Processing.html")
covid_Profile
```

Overview

Dataset statistics

Number of variables	5
Number of observations	16336
Missing cells	20041
Missing cells (%)	24.5%
Duplicate rows	1
Duplicate rows (%)	< 0.1%
Total size in memory	638.2 KiB
Average record size in memory	40.0 B

Variable types

Categorical	3
Numeric	2

Alerts

Dataset has 1 (< 0.1%) duplicate rows	**Duplicates**
`Date` has a high cardinality: 497 distinct values	**High cardinality**
`Negative` has a high cardinality: 6898 distinct values	**High cardinality**
`TotalSamples` is highly correlated with `Positive`	**High correlation**

Out[4]:

In [21]:

```
#drop duplicate value
covid_df.drop_duplicates(inplace=True)
covid_df.shape
```

Out[21]:

```
(16335, 5)
```

In [22]:

```

#Tofind NAN values

covid_df.isna().any()
```

Out[22]:

```
Date            False
State           False
TotalSamples    False
Negative         True
Positive         True
dtype: bool
```

In [45]:

```

#handle missing values
covid_df['Negative'] = covid_df['Negative'].fillna(0)
covid_df['Positive'] = covid_df['Positive'].fillna(0)
covid_df['TotalSamples'] = covid_df['TotalSamples'].fillna(0)
covid_df
```

Out[45]:

	Date	State	TotalSamples	Negative	Positive
0	2020-04-17	Andaman and Nicobar Islands	1403.0	1210.0	12.0
1	2020-04-24	Andaman and Nicobar Islands	2679.0	0.0	27.0
2	2020-04-27	Andaman and Nicobar Islands	2848.0	0.0	33.0
3	2020-05-01	Andaman and Nicobar Islands	3754.0	0.0	33.0
4	2020-05-16	Andaman and Nicobar Islands	6677.0	0.0	33.0
...	...	...	...	...	...
16331	2021-08-06	West Bengal	15999961.0	0.0	0.0
16332	2021-08-07	West Bengal	16045662.0	0.0	0.0
16333	2021-08-08	West Bengal	16092192.0	0.0	0.0
16334	2021-08-09	West Bengal	16122345.0	0.0	0.0
16335	2021-08-10	West Bengal	16162814.0	0.0	0.0

16335 rows × 5 columns

In [63]:

```

#convert string to float
covid_df['Negative'] = pd.to_numeric(covid_df['Negative'],
                                     errors = 'coerce')
```

Data Post Profiling:-

In [20]:

```

covid_Profile=pandas_profiling.ProfileReport(covid_df)
covid_Profile.to_file("Covidata_Post_Processing.html")
covid_Profile
```

Overview

Dataset statistics

Number of variables	5
Number of observations	16336
Missing cells	20042
Missing cells (%)	24.5%
Duplicate rows	1
Duplicate rows (%)	< 0.1%
Total size in memory	638.2 KiB
Average record size in memory	40.0 B

Variable types

Categorical	2
Numeric	3

Alerts

Dataset has 1 (< 0.1%) duplicate rows	**Duplicates**
`Date` has a high cardinality: 497 distinct values	**High cardinality**
`TotalSamples` is highly correlated with `Negative` and 1 other fields (Negative, Positive)	**High correlation**
`Negative` is highly correlated with `TotalSamples` and 1 other fields (TotalSamples	

Out[20]:

Which State Conducted tests maximum number of days?

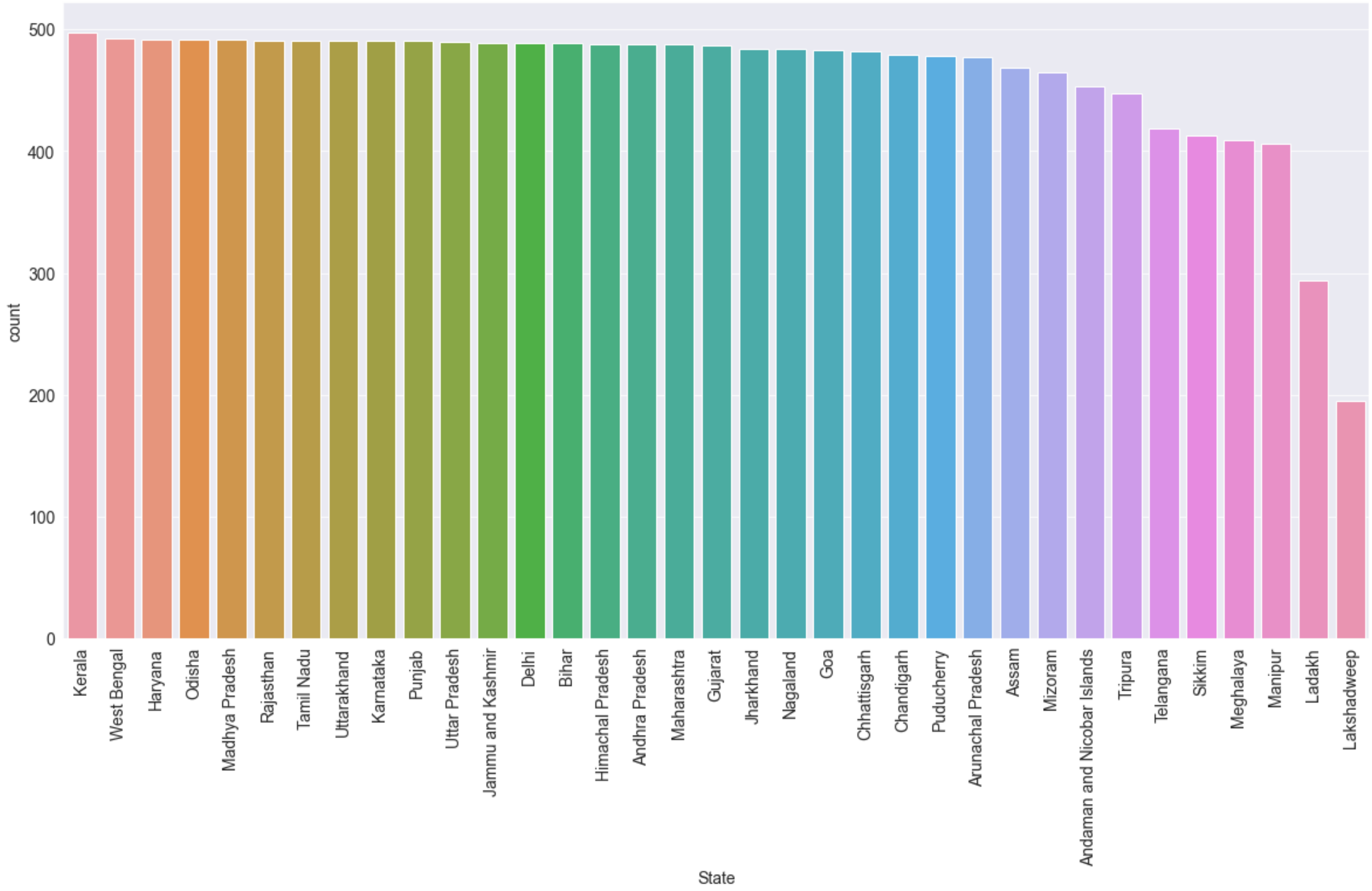
In [39]:

```
#Kerala conducted tests maximum number of days

plt.figure(figsize=(20,10))
State = sns.countplot(x='State',data=covid_df, order=covid_df['State'].value_counts().inde
plt.xticks(rotation=90)
```

Out[39]:

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
        34]),
[Text(0, 0, 'Kerala'),
 Text(1, 0, 'West Bengal'),
 Text(2, 0, 'Haryana'),
 Text(3, 0, 'Odisha'),
 Text(4, 0, 'Madhya Pradesh'),
 Text(5, 0, 'Rajasthan'),
 Text(6, 0, 'Tamil Nadu'),
 Text(7, 0, 'Uttarakhand'),
 Text(8, 0, 'Karnataka'),
 Text(9, 0, 'Punjab'),
 Text(10, 0, 'Uttar Pradesh'),
 Text(11, 0, 'Jammu and Kashmir'),
 Text(12, 0, 'Delhi'),
 Text(13, 0, 'Bihar'),
 Text(14, 0, 'Himachal Pradesh'),
 Text(15, 0, 'Andhra Pradesh'),
 Text(16, 0, 'Maharashtra'),
 Text(17, 0, 'Gujarat'),
 Text(18, 0, 'Jharkhand'),
 Text(19, 0, 'Nagaland'),
 Text(20, 0, 'Goa'),
 Text(21, 0, 'Chhattisgarh'),
 Text(22, 0, 'Chandigarh'),
 Text(23, 0, 'Puducherry'),
 Text(24, 0, 'Arunachal Pradesh'),
 Text(25, 0, 'Assam'),
 Text(26, 0, 'Mizoram'),
 Text(27, 0, 'Andaman and Nicobar Islands'),
 Text(28, 0, 'Tripura'),
 Text(29, 0, 'Telangana'),
 Text(30, 0, 'Sikkim'),
 Text(31, 0, 'Meghalaya'),
 Text(32, 0, 'Manipur'),
 Text(33, 0, 'Ladakh'),
 Text(34, 0, 'Lakshadweep')])
```



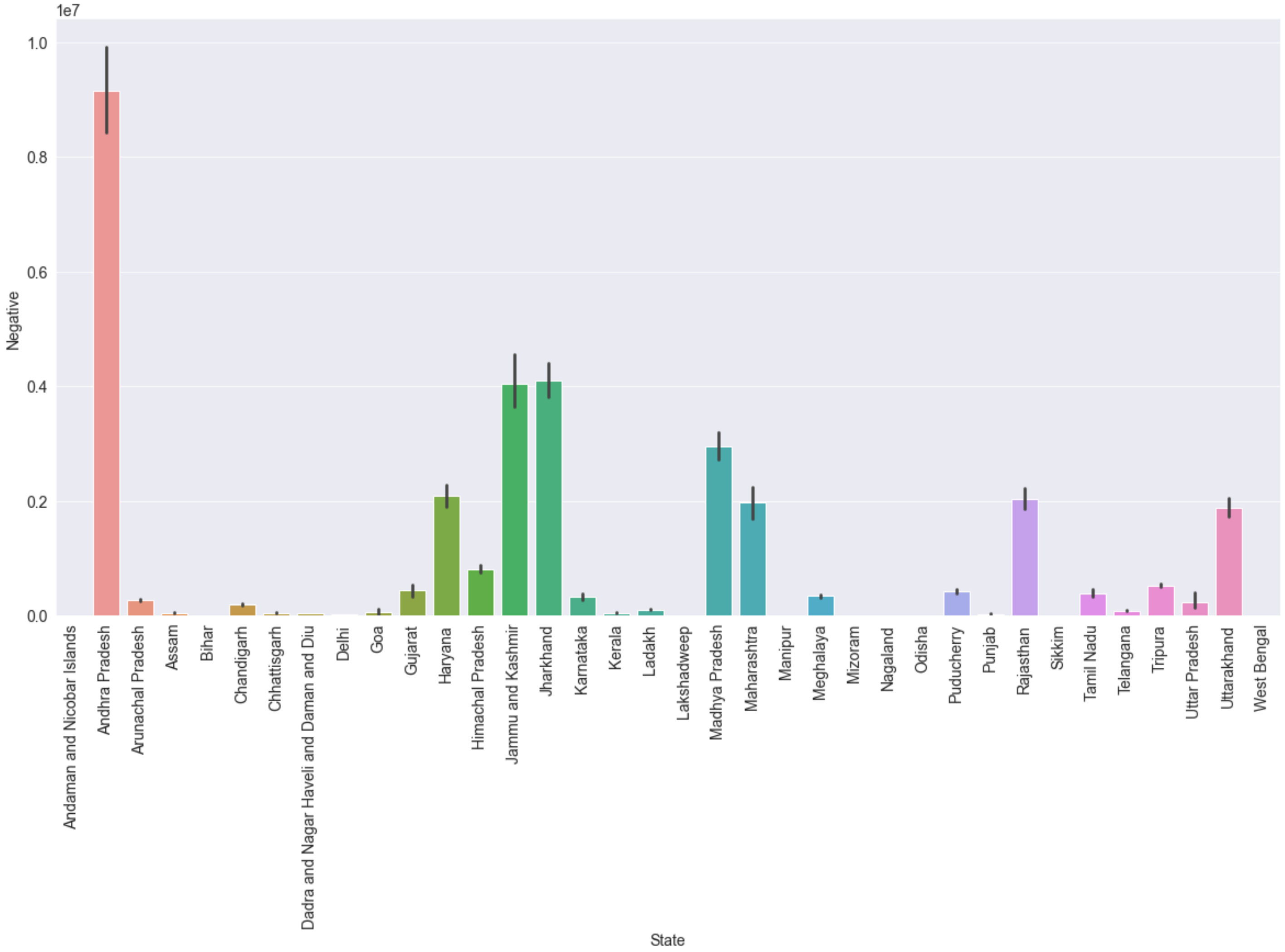
Which state recorded max negative tests?

In [14]:

```python
# Andhra had max number of negative cases
plt.figure(figsize=(20,10))
sns.barplot(x='State', y="Negative", data=covid_df)
plt.xticks(rotation=90)
```

Out[14]:

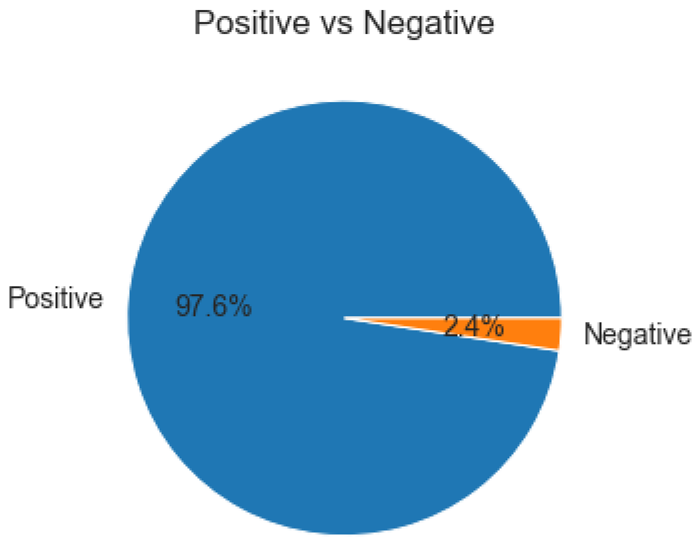
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
        34, 35]),
 [Text(0, 0, 'Andaman and Nicobar Islands'),
  Text(1, 0, 'Andhra Pradesh'),
  Text(2, 0, 'Arunachal Pradesh'),
  Text(3, 0, 'Assam'),
  Text(4, 0, 'Bihar'),
  Text(5, 0, 'Chandigarh'),
  Text(6, 0, 'Chhattisgarh'),
  Text(7, 0, 'Dadra and Nagar Haveli and Daman and Diu'),
  Text(8, 0, 'Delhi'),
  Text(9, 0, 'Goa'),
  Text(10, 0, 'Gujarat'),
  Text(11, 0, 'Haryana'),
  Text(12, 0, 'Himachal Pradesh'),
  Text(13, 0, 'Jammu and Kashmir'),
  Text(14, 0, 'Jharkhand'),
  Text(15, 0, 'Karnataka'),
  Text(16, 0, 'Kerala'),
  Text(17, 0, 'Ladakh'),
  Text(18, 0, 'Lakshadweep'),
  Text(19, 0, 'Madhya Pradesh'),
  Text(20, 0, 'Maharashtra'),
  Text(21, 0, 'Manipur'),
  Text(22, 0, 'Meghalaya'),
  Text(23, 0, 'Mizoram'),
  Text(24, 0, 'Nagaland'),
  Text(25, 0, 'Odisha'),
  Text(26, 0, 'Puducherry'),
  Text(27, 0, 'Punjab'),
  Text(28, 0, 'Rajasthan'),
  Text(29, 0, 'Sikkim'),
  Text(30, 0, 'Tamil Nadu'),
  Text(31, 0, 'Telangana'),
  Text(32, 0, 'Tripura'),
  Text(33, 0, 'Uttar Pradesh'),
  Text(34, 0, 'Uttarakhand'),
  Text(35, 0, 'West Bengal')])
```



Number of positive cases as compared to negative?

In [65]:

```
#Positive vs negative
y=covid_df['Positive'].sum()
x=covid_df['Negative'].sum()
data=[x,y]
plt.title('Positive vs Negative')
mylabels = ["Positive", "Negative"]
plt.pie(data, labels = mylabels, autopct='%1.1f%%')
plt.show()
```



Which state collected max number of swab samples?

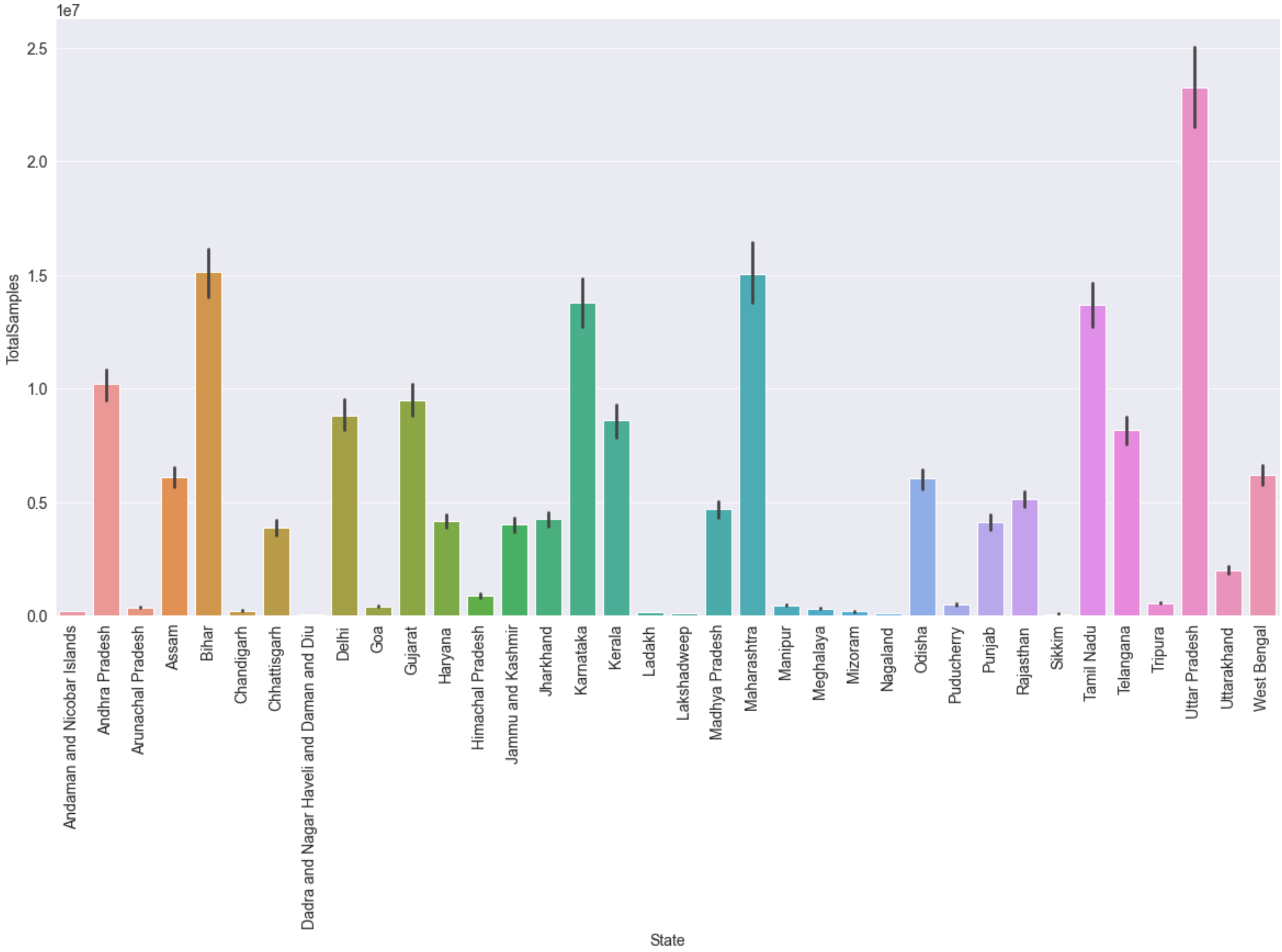
In [5]:

```
#uttar pradesh collected max number of samples

plt.figure(figsize=(20,10))
sns.barplot(x='State', y="TotalSamples", data=covid_df)
plt.xticks(rotation=90)
```

Out[5]:

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
        34, 35]),
 [Text(0, 0, 'Andaman and Nicobar Islands'),
  Text(1, 0, 'Andhra Pradesh'),
  Text(2, 0, 'Arunachal Pradesh'),
  Text(3, 0, 'Assam'),
  Text(4, 0, 'Bihar'),
  Text(5, 0, 'Chandigarh'),
  Text(6, 0, 'Chhattisgarh'),
  Text(7, 0, 'Dadra and Nagar Haveli and Daman and Diu'),
  Text(8, 0, 'Delhi'),
  Text(9, 0, 'Goa'),
  Text(10, 0, 'Gujarat'),
  Text(11, 0, 'Haryana'),
  Text(12, 0, 'Himachal Pradesh'),
  Text(13, 0, 'Jammu and Kashmir'),
  Text(14, 0, 'Jharkhand'),
  Text(15, 0, 'Karnataka'),
  Text(16, 0, 'Kerala'),
  Text(17, 0, 'Ladakh'),
  Text(18, 0, 'Lakshadweep'),
  Text(19, 0, 'Madhya Pradesh'),
  Text(20, 0, 'Maharashtra'),
  Text(21, 0, 'Manipur'),
  Text(22, 0, 'Meghalaya'),
  Text(23, 0, 'Mizoram'),
  Text(24, 0, 'Nagaland'),
  Text(25, 0, 'Odisha'),
  Text(26, 0, 'Puducherry'),
  Text(27, 0, 'Punjab'),
  Text(28, 0, 'Rajasthan'),
  Text(29, 0, 'Sikkim'),
  Text(30, 0, 'Tamil Nadu'),
  Text(31, 0, 'Telangana'),
  Text(32, 0, 'Tripura'),
  Text(33, 0, 'Uttar Pradesh'),
  Text(34, 0, 'Uttarakhand'),
  Text(35, 0, 'West Bengal')])
```



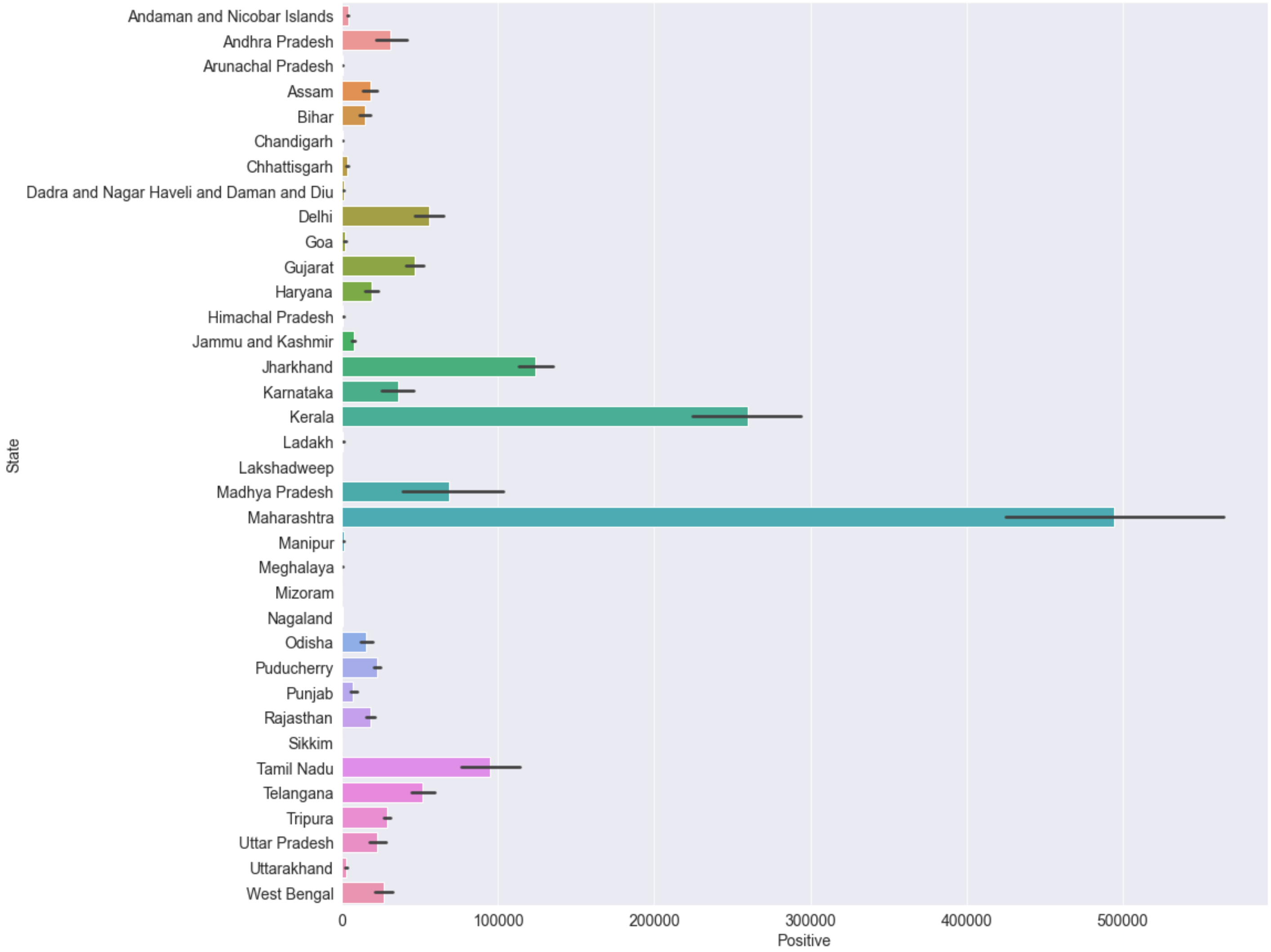
Which State had maximum number of positive cases?

In [21]:

```
#Maharashtra had max number of positive cases
plt.figure(figsize=(15,15))
sns.barplot(x='Positive', y="State", data=covid_df)
plt.xticks(rotation=360)
```

Out[21]:

```
(array([     0., 100000., 200000., 300000., 400000., 500000., 600000.]),
 [Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, '')])
```



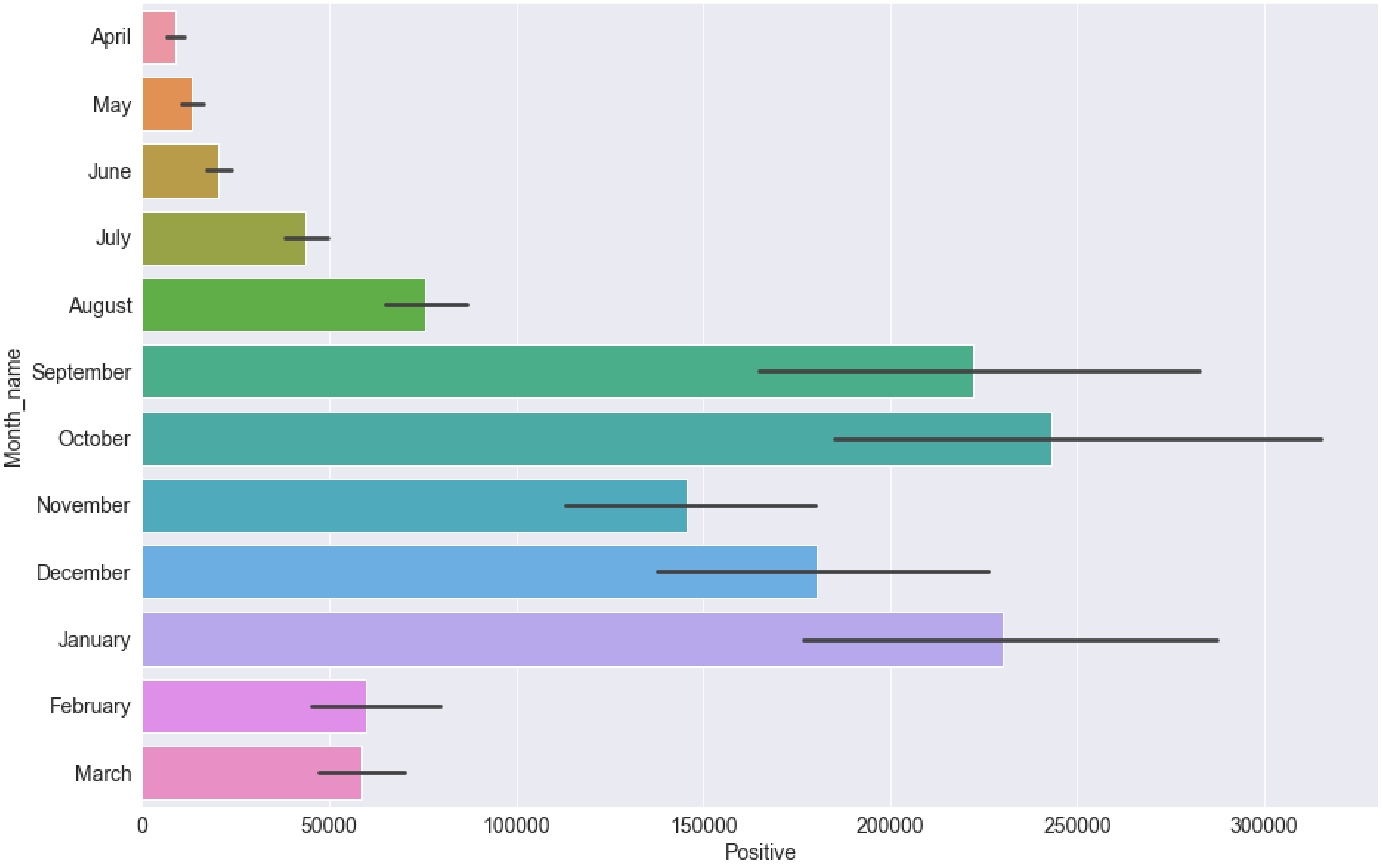
In which months peak of covid arrived?

In [70]:

```
# October, January and September months recorded max positive cases in both the years
plt.figure(figsize=(15,10))
sns.barplot(x='Positive', y='Month_name', data=covid_df)
plt.xticks(rotation=360)
```

Out[70]:

```
(array([     0.,  50000., 100000., 150000., 200000., 250000., 300000.,
        350000.]),
 [Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, '')])
```


April
May
June
July
August
September
October
November
December
January
February
March
Month_name
0
50000
100000
150000
200000
250000
300000
Positive

```
In [ ]:
```