

```
In [ ]: 1 #####iris data set#####
```

```
In [1]: 1 #importing libraries
2 import pandas as pd
3 import numpy as np
4 import matplotlib.pyplot as plt
5 %matplotlib inline
```

```
In [2]: 1 import warnings
2 warnings.filterwarnings('ignore')
```

```
In [6]: 1 df=pd.read_csv("C:/Users/KARNARAJ RATHOD/OneDrive/Desktop/oasis/Iris.csv")
```

```
In [8]: 1 df.head()
2
3
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

```
In [9]: 1 df.shape
```

```
In [10]: 1 df.describe()
2
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

```
In [11]: 1 df.info
```

1	df.
2	

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

1	df.info
---	---------

[illegible][illegible][illegible]

```
In [14]: x = df[['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']]
         y = df['Species']
```

```
In [19]: 1 Iris-virginica
2 # Let's create an instance for the LogisticRegression model
3 lr = LogisticRegression()
4 lr.fit(X_train, y_train)
5 lr.predict(X_test)
```

```

In [17]: 1 # Do the train/test split
          2

In [16]: 3 from sklearn.model_selection import train_test_split
          4 X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.20,random_state=42)

In [18]: 1 Iris-setosa
          2 # Training the Linear Regression Model
          3 Iris-setosa
          4 Iris-setosa
          5 from sklearn.linear_model import LogisticRegression
          6 ...

In [19]: 1 Iris-virginica
          2 # Let's create an instance for the LogisticRegression model
          3 lr = LogisticRegression()
          4 Iris-virginica
          5 Iris-virginica
          6 Iris-virginica
          7 # Train the model on our train dataset
          8 lr.fit(X,y)
          9 lr.fit(X_train,y_train)

          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
          36
          37
          38
          39
          40
          41
          42
          43
          44
          45
          46
          47
          48
          49
          50
          51
          52
          53
          54
          55
          56
          57
          58
          59
          60
          61
          62
          63
          64
          65
          66
          67
          68
          69
          70
          71
          72
          73
          74
          75
          76
          77
          78
          79
          80
          81
          82
          83
          84
          85
          86
          87
          88
          89
          90
          91
          92
          93
          94
          95
          96
          97
          98
          99
          100
          101
          102
          103
          104
          105
          106
          107
          108
          109
          110
          111
          112
          113
          114
          115
          116
          117
          118
          119
          120
          121
          122
          123
          124
          125
          126
          127
          128
          129
          130
          131
          132
          133
          134
          135
          136
          137
          138
          139
          140
          141
          142
          143
          144
          145
          146
          147
          148
          149
          150
          151
          152
          153
          154
          155
          156
          157
          158
          159
          160
          161
          162
          163
          164
          165
          166
          167
          168
          169
          170
          171
          172
          173
          174
          175
          176
          177
          178
          179
          180
          181
          182
          183
          184
          185
          186
          187
          188
          189
          190
          191
          192
          193
          194
          195
          196
          197
          198
          199
          200
          201
          202
          203
          204
          205
          206
          207
          208
          209
          210
          211
          212
          213
          214
          215
          216
          217
          218
          219
          220
          221
          222
          223
          224
          225
          226
          227
          228
          229
          230
          231
          232
          233
          234
          235
          236
          237
          238
          239
          240
          241
          242
          243
          244
          245
          246
          247
          248
          249
          250
          251
          252
          253
          254
          255
          256
          257
          258
          259
          260
          261
          262
          263
          264
          265
          266
          267
          268
          269
          270
          271
          272
          273
          274
          275
          276
          277
          278
          279
          280
          281
          282
          283
          284
          285
          286
          287
          288
          289
          290
          291
          292
          293
          294
          295
          296
          297
          298
          299
          300
          301
          302
          303
          304
          305
          306
          307
          308
          309
          310
          311
          312
          313
          314
          315
          316
          317
          318
          319
          320
          321
          322
          323
          324
          325
          326
          327
          328
          329
          330
          331
          332
          333
          334
          335
          336
          337
          338
          339
          340
          341
          342
          343
          344
          345
          346
          347
          348
          349
          350
          351
          352
          353
          354
          355
          356
          357
          358
          359
          360
          361
          362
          363
          364
          365
          366
          367
          368
          369
          370
          371
          372
          373
          374
          375
          376
          377
          378
          379
          380
          381
          382
          383
          384
          385
          386
          387
          388
          389
          390
          391
          392
          393
          394
          395
          396
          397
          398
          399
          400
          401
          402
          403
          404
          405
          406
          407
          408
          409
          410
          411
          412
          413
          414
          415
          416
          417
          418
          419
          420
          421
          422
          423
          424
          425
          426
          427
          428
          429
          430
          431
          432
          433
          434
          435
          436
          437
          438
          439
          440
          441
          442
          443
          444
          445
          446
          447
          448
          449
          450
          451
          452
          453
          454
          455
          456
          457
          458
          459
          460
          461
          462
          463
          464
          465
          466
          467
          468
          469
          470
          471
          472
          473
          474
          475
          476
          477
          478
          479
          480
          481
          482
          483
          484
          485
          486
          487
          488
          489
          490
          491
          492
          493
          494
          495
          496
          497
          498
          499
          500
          501
          502
          503
          504
          505
          506
          507
          508
          509
          510
          511
          512
          513
          514
          515
          516
          517
          518
          519
          520
          521
          522
          523
          524
          525
          526
          527
          528
          529
          530
          531
          532
          533
          534
          535
          536
          537
          538
          539
          540
          541
          542
          543
          544
          545
          546
          547
          548
          549
          550
          551
          552
          553
          554
          555
          556
          557
          558
          559
          560
          561
          562
          563
          564
          565
          566
          567
          568
          569
          570
          571
          572
          573
          574
          575
          576
          577
          578
          579
          580
          581
          582
          583
          584
          585
          586
          587
          588
          589
          590
          591
          592
          593
          594
          595
          596
          597
          598
          599
          600
          601
          602
          603
          604
          605
          606
          607
          608
          609
          610
          611
          612
          613
          614
          615
          616
          617
          618
          619
          620
          621
          622
          623
          624
          625
          626
          627
          628
          629
          630
          631
          632
          633
          634
          635
          636
          637
          638
          639
          640
          641
          642
          643
          644
          645
          646
          647
          648
          649
          650
          651
          652
          653
          654
          655
          656
          657
          658
          659
          660
          661
          662
          663
          664
          665
          666
          667
          668
          669
          670
          671
          672
          673
          674
          675
          676
          677
          678
          679
          680
          681
          682
          683
          684
          685
          686
          687
          688
          689
          690
          691
          692
          693
          694
          695
          696
          697
          698
          699
          700
          701
          702
          703
          704
          705
          706
          707
          708
          709
          710
          711
          712
          713
          714
          715
          716
          717
          718
          719
          720
          721
          722
          723
          724
          725
          726
          727
          728
          729
          730
          731
          732
          733
          734
          735
          736
          737
          738
          739
          740
          741
          742
          743
          744
          745
          746
          747
          748
          749
          750
          751
          752
          753
          754
          755
          756
          757
          758
          759
          760
          761
          762
          763
          764
          765
          766
          767
          768
          769
          770
          771
          772
          773
          774
          775
          776
          777
          778
          779
          780
          781
          782
          783
          784
          785
          786
          787
          788
          789
          790
          791
          792
          793
          794
          795
          796
          797
          798
          799
          800
          801
          802
          803
          804
          805
          806
          807
          808
          809
          810
          811
          812
          813
          814
          815
          816
          817
          818
          819
          820
          821
          822
          823
          824
          825
          826
          827
          828
          829
          830
          831
          832
          833
          834
          835
          836
          837
          838
          839
          840
          841
          842
          843
          844
          845
          846
          847
          848
          849
          850
          851
          852
          853
          854
          855
          856
          857
          858
          859
          860
          861
          862
          863
          864
          865
          866
          867
          868
          869
          870
          871
          872
          873
          874
          875
          876
          877
          878
          879
          880
          881
          882
          883
          884
          885
          886
          887
          888
          889
          890
          891
          892
          893
          894
          895
          896
          897
          898
          899
          900
          901
          902
          903
          904
          905
          906
          907
          908
          909
          910
          911
          912
          913
          914
          915
          916
          917
          918
          919
          920
          921
          922
          923
          924
          925
          926
          927
          928
          929
          930
          931
          932
          933
          934
          935
          936
          937
          938
          939
          940
          941
          942
          943
          944
          945
          946
          947
          948
          949
          950
          951
          952
          953
          954
          955
          956
          957
          958
          959
          960
          961
          962
          963
          964
          965
          966
          967
          968
          969
          970
          971
          972
          973
          974
          975
          976
          977
          978
          979
          980
          981
          982
          983
          984
          985
          986
          987
          988
          989
          990
          991
          992
          993
          994
          995
          996
          997
          998
          999
          1000
          1001
          1002
          1003
          1004
          1005
          1006
          1007
          1008
          1009
          1010
          1011
          1012
          1013
          1014
          1015
          1016
          1017
          1018
          1019
          1020
          1021
          1022
          1023
          1024
          1025
          1026
          1027
          1028
          1029
          1030
          1031
          1032
          1033
          1034
          1035
          1036
          1037
          1038
          1039
          1040
          1041
          1042
          1043
          1044
          1045
          1046
          1047
          1048
          1049
          1050
          1051
          1052
          1053
          1054
          1055
          1056
          1057
          1058
          1059
          1060
          1061
          1062
          1063
          1064
          1065
          1066
          1067
          1068
          1069
          1070
          1071
          1072
          1073
          1074
          1075
          1076
          1077
          1078
          1079
          1080
          1081
          1082
          1083
          1084
          1085
          1086
          1087
          1088
          1089
          1090
          1091
          1092
          1093
          1094
          1095
          1096
          1097
          1098
          1099
          1100
          1101
          1102
          1103
          1104
          1105
          1106
          1107
          1108
          1109
          1110
          1111
          1112
          1113
          1114
          1115
          1116
          1117
          1118
          1119
          1120
          1121
          1122
          1123
          1124
          1125
          1126
          1127
          1128
          1129
          1130
          1131
          1132
          1133
          1134
          1135
          1136
          1137
          1138
          1139
          1140
          1141
          1142
          1143
          1144
          1145
          1146
          1147
          1148
          1149
          1150
          1151
          1152
          1153
          1154
          1155
          1156
          1157
          1158
          1159
          1160
          1161
          1162
          1163
          1164
          1165
          1166
          1167
          1168
          1169
          1170
          1171
          1172
          1173
          1174
          1175
          1176
          1177
          1178
          1179
          1180
          1181
          1182
          1183
          1184
          1185
          1186
          1187
          1188
          1189
          1190
          1191
          1192
          1193
          1194
          1195
          1196
          1197
          1198
          1199
          1200
          1201
          1202
          1203
          1204
          1205
          1206
          1207
          1208
          1209
          1210
          1211
          1212
          1213
          1214
          1215
          1216
          1217
          1218
          1219
          1220
          1221
          1222
          1223
          1224
          1225
          1226
          1227
          1228
          1229
          1230
          1231
          1232
          1233
          1234
          1235
          1236
          1237
          1238
          1239
          1240
          1241
          1242
          1243
          1244
          1245
          1246
          1247
          1248
          1249
          1250
          1251
          1252
          1253
          1254
          1255
          1256
          1257
          1258
          1259
          1260
          1261
          1262
          1263
          1264
          1265
          1266
          1267
          1268
          1269
          1270
          1271
          1272
          1273
          1274
          1275
          1276
          1277
          1278
          1279
          1280
          1281
          1282
          1283
          1284
          1285
          1286
          1287
          1288
          1289
          1290
          1291
          1292
          1293
          1294
          1295
          1296
          1297
          1298
          1299
          1300
          1301
          1302
          1303
          1304
          1305
          1306
          1307
          1308
          1309
          1310
          1311
          1312
          1313
          1314
          1315
          1316
          1317
          1318
          1319
          1320
          1321
          1322
          1323
          1324
          1325
          1326
          1327
          1328
          1329
          1330
          1331
          1332
          1333
          1334
          1335
          1336
          1337
          1338
          1339
          1340
          1341
          1342
          1343
          1344
          1345
          1346
          1347
          1348
          1349
          1350
          1351
          1352
          1353
          1354
          1355
          1356
          1357
          1358
          1359
          1360
          1361
          1362
          1363
          1364
          1365
          1366
          1367
          1368
          1369
          1370
          1371
          1372
          1373
          1374
          1375
          1376
          1377
          1378
          1379
          1380
          1381
          1382
          1383
          1384
          1385
          1386
          1387
          1388
          1389
          1390
          1391
          1392
          1393
          1394
          1395
          1396
          1397
          1398
          1399
          1400
          1401
          1402
          1403
          1404
          1405
          1406
          1407
          1408
          1409
          1410
          1411
          1412
          1413
          1414
          1415
          1416
          1417
          1418
          1419
          1420
          1421
          1422
          1423
          1424
          1425
          1426
          1427
          1428
          1429
          1430
          1431
          1432
          1433
          1434
          1435
          1436
          1437
          1438
          1439
          1440
          1441
          1442
          1443
          1444
          1445
          1446
          1447
          1448
          1449
          1450
          1451
          1452
          1453
          1454
          1455
          1456
          1457
          1458
          1459
          1460
          1461
          1462
          1463
          1464
          1465
          1466
          1467
          1468
          1469
          1470
          1471
          1472
          1473
          1474
          1475
          1476
          1477
          1478
          1479
          1480
          1481
          1482
          1483
          1484
          1485
          1486
          1487
          1488
          1489
          1490
          1491
          1492
          1493
          1494
          1495
          1496
          1497
          1498
          1499
          1500
          1501
          1502
          1503
          1504
          1505
          1506
          1507
          1508
          1509
          1510
          1511
          1512
          1513
          1514
          1515
          1516
          1517
          1518
          1519
          1520
          1521
          1522
          1523
          1524
          1525
          1526
          1527
          1528
          1529
          1530
          1531
          1532
          1533
          1534
          1535
          1536
          1537
          1538
          1539
          1540
          1541
          1542
          1543
          1544
          1545
          1546
          1547
          1548
          1549
          1550
          1551
          1552
          1553
          1554
          1555
          1556
          1557
          1558
          1559
          1560
          1561
          1562
          1563
          1564
          1565
          1566
          1567
          1568
          1569
          1570
          1571
          1572
          1573
          1574
          1575
          1576
          1577
          1578
          1579
          1580
          1581
          1582
          1583
          1584
          1585
          1586
          1587
          1588
          1589
          1590
          1591
          1592
          1593
          1594
          1595
          1596
          1597
          1598
          1599
          1600
          1601
          1602
          1603
          1604
          1605
          1606
          1607
          1608
          1609
          1610
          1611
          1612
          1613
          1614
          1615
          1616
          1617
          1618
          1619
          1620
          1621
          1622
          1623
          1624
          1625
          1626
          1627
          1628
          1629
          1630
          1631
          1632
          1633
          1634
          1635
          1636
          1637
          1638
          1639
          1640
          1641
          1642
          1643
          1644
          1645
          1646
          1647
          1648
          1649
          1650
          1651
          1652
          1653
          1654
          1655
          1656
          1657
          1658
          1659
          1660
          1661
          1662
          1663
          1664
          1665
          1666
          1667
          1668
          1669
          1670
          1671
          1672
          1673
          1674
          1675
          1676
          1677
          1678
          1679
          1680
          1681
          1682
          1683
          1684
          1685
          1686
          1687
          1688
          1689
          1690
          1691
          1692
          1693
          1694
          1695
          1696
          1697
          1698
          1699
          1700
          1701
          1702
          1703
          1704
          1705
          1706
          1707
          1708
          1709
          1710
          1711
          1712
          1713
          1714
          1715
          1716
          1717
          1718
          1719
          1720
          1721
          1722
          1723
          1724
          1725
          1726
          1727
          1728
          1729
          1730
          1731
          1732
          1733
          1734
          1735
          1736
          1737
          1738
          1739
          1740
          1741
          1742
          1743
          1744
          1745
          1746
          1747
          1748
          1749
          1750
          1751
          1752
          1753
          1754
          1755
          1756
          1757
          1758
          1759
          1760
          1761
          1762
          1763
          1764
          1765
          1766
          1767
          1768
          1769
          1770
          1771
          1772
          1773
          1774
          1775
          1776
          1777
          1778
          1779
          1780
          1781
          1782
          1783
          1784
          1785
          1786
          1787
          1788
          1789
          1790
          1791
          1792
          1793
          1794
          1795
          1796
          1797
          1798
          1799
          1800
          1801
          1802
          1803
          1804
          1805
          1806
          1807
          1808
          1809
          1810
          1811
          1812
          1813
          1814
          1815
          1816
          1817
          1818
          1819
          1820
          1821
          1822
          1823
          1824
          1825
          1826
          1827
          1828
          1829
          1830
          1831
          1832
          1833
          1834
          1835
          1836
          1837
          1838
          1839
          1840
          1841
          1842
          1843
          1844
          1845
          1846
          1847
          1848
          1849
          1850
          1851
          1852
          1853
          1854
          1855
          1856
          1857
          1858
          1859
          1860
          1861
          1862
          1863
          1864
          1865
          1866
          1867
          1868
          1869
          1870
          1871
          1872
          1873
          1874
          1875
          1876
          1877
          1878
          1879
          1880
          1881
          1882
          1883
          1884
          1885
         
```

0 iris-setosa iris-setosa

iris-setosa iris-setosa

```
3 import calendar
```

	Region	Date	Frequency	Estimated Unemployment Rate (%)	Estimated Employed	Estimated Labour Participation Rate (%)	Region.1	longitude	latitude
0	Andhra Pradesh	31-01-2020	M	5.48	16635535	41.02	South	15.9129	79.74
1	Andhra Pradesh	29-02-2020	M	5.83	16545652	40.90	South	15.9129	79.74
2	Andhra Pradesh	31-03-2020	M	5.79	15881197	39.18	South	15.9129	79.74
3	Andhra Pradesh	30-04-2020	M	20.51	11336911	33.10	South	15.9129	79.74
4	Andhra Pradesh	31-05-2020	M	17.43	12988845	36.46	South	15.9129	79.74

data column	non-null count	dtype
Region	267 non-null	object
Estimated Unemployment Rate (%)	267 non-null	float64
Estimated Employed	267 non-null	int64
Estimated Labour Participation Rate (%)	267 non-null	float64
Region	267 non-null	object
Estimated Unemployment Rate (%)	267 non-null	float64
Estimated Employed	267 non-null	int64
Estimated Labour Participation Rate (%)	267 non-null	float64
Region	267 non-null	object
longitude	267 non-null	float64
latitude	267 non-null	float64

```
In [29]: 1 pd.info(df)
2 # Renaming columns for better clarity
3 df.columns = ['States', 'Date', 'Frequency', 'Estimated Unemployment Rate', 'Estimated Employed',
4               'Estimated Labour Participation Rate', 'Region', 'longitude', 'latitude']
```

```
In [28]: 1 df.isnull().sum()
          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
          36
          37
          38
          39
          40
          41
          42
          43
          44
          45
          46
          47
          48
          49
          50
          51
          52
          53
          54
          55
          56
          57
          58
          59
          60
          61
          62
          63
          64
          65
          66
          67
          68
          69
          70
          71
          72
          73
          74
          75
          76
          77
          78
          79
          80
          81
          82
          83
          84
          85
          86
          87
          88
          89
          90
          91
          92
          93
          94
          95
          96
          97
          98
          99
          100
          101
          102
          103
          104
          105
          106
          107
          108
          109
          110
          111
          112
          113
          114
          115
          116
          117
          118
          119
          120
          121
          122
          123
          124
          125
          126
          127
          128
          129
          130
          131
          132
          133
          134
          135
          136
          137
          138
          139
          140
          141
          142
          143
          144
          145
          146
          147
          148
          149
          150
          151
          152
          153
          154
          155
          156
          157
          158
          159
          160
          161
          162
          163
          164
          165
          166
          167
          168
          169
          170
          171
          172
          173
          174
          175
          176
          177
          178
          179
          180
          181
          182
          183
          184
          185
          186
          187
          188
          189
          190
          191
          192
          193
          194
          195
          196
          197
          198
          199
          200
          201
          202
          203
          204
          205
          206
          207
          208
          209
          210
          211
          212
          213
          214
          215
          216
          217
          218
          219
          220
          221
          222
          223
          224
          225
          226
          227
          228
          229
          230
          231
          232
          233
          234
          235
          236
          237
          238
          239
          240
          241
          242
          243
          244
          245
          246
          247
          248
          249
          250
          251
          252
          253
          254
          255
          256
          257
          258
          259
          260
          261
          262
          263
          264
          265
          266
          267
          268
          269
          270
          271
          272
          273
          274
          275
          276
          277
          278
          279
          280
          281
          282
          283
          284
          285
          286
          287
          288
          289
          290
          291
          292
          293
          294
          295
          296
          297
          298
          299
          300
          301
          302
          303
          304
          305
          306
          307
          308
          309
          310
          311
          312
          313
          314
          315
          316
          317
          318
          319
          320
          321
          322
          323
          324
          325
          326
          327
          328
          329
          330
          331
          332
          333
          334
          335
          336
          337
          338
          339
          340
          341
          342
          343
          344
          345
          346
          347
          348
          349
          350
          351
          352
          353
          354
          355
          356
          357
          358
          359
          360
          361
          362
          363
          364
          365
          366
          367
          368
          369
          370
          371
          372
          373
          374
          375
          376
          377
          378
          379
          380
          381
          382
          383
          384
          385
          386
          387
          388
          389
          390
          391
          392
          393
          394
          395
          396
          397
          398
          399
          400
          401
          402
          403
          404
          405
          406
          407
          408
          409
          410
          411
          412
          413
          414
          415
          416
          417
          418
          419
          420
          421
          422
          423
          424
          425
          426
          427
          428
          429
          430
          431
          432
          433
          434
          435
          436
          437
          438
          439
          440
          441
          442
          443
          444
          445
          446
          447
          448
          449
          450
          451
          452
          453
          454
          455
          456
          457
          458
          459
          460
          461
          462
          463
          464
          465
          466
          467
          468
          469
          470
          471
          472
          473
          474
          475
          476
          477
          478
          479
          480
          481
          482
          483
          484
          485
          486
          487
          488
          489
          490
          491
          492
          493
          494
          495
          496
          497
          498
          499
          500
          501
          502
          503
          504
          505
          506
          507
          508
          509
          510
          511
          512
          513
          514
          515
          516
          517
          518
          519
          520
          521
          522
          523
          524
          525
          526
          527
          528
          529
          530
          531
          532
          533
          534
          535
          536
          537
          538
          539
          540
          541
          542
          543
          544
          545
          546
          547
          548
          549
          550
          551
          552
          553
          554
          555
          556
          557
          558
          559
          560
          561
          562
          563
          564
          565
          566
          567
          568
          569
          570
          571
          572
          573
          574
          575
          576
          577
          578
          579
          580
          581
          582
          583
          584
          585
          586
          587
          588
          589
          590
          591
          592
          593
          594
          595
          596
          597
          598
          599
          600
          601
          602
          603
          604
          605
          606
          607
          608
          609
          610
          611
          612
          613
          614
          615
          616
          617
          618
          619
          620
          621
          622
          623
          624
          625
          626
          627
          628
          629
          630
          631
          632
          633
          634
          635
          636
          637
          638
          639
          640
          641
          642
          643
          644
          645
          646
          647
          648
          649
          650
          651
          652
          653
          654
          655
          656
          657
          658
          659
          660
          661
          662
          663
          664
          665
          666
          667
          668
          669
          670
          671
          672
          673
          674
          675
          676
          677
          678
          679
          680
          681
          682
          683
          684
          685
          686
          687
          688
          689
          690
          691
          692
          693
          694
          695
          696
          697
          698
          699
          700
          701
          702
          703
          704
          705
          706
          707
          708
          709
          710
          711
          712
          713
          714
          715
          716
          717
          718
          719
          720
          721
          722
          723
          724
          725
          726
          727
          728
          729
          730
          731
          732
          733
          734
          735
          736
          737
          738
          739
          740
          741
          742
          743
          744
          745
          746
          747
          748
          749
          750
          751
          752
          753
          754
          755
          756
          757
          758
          759
          760
          761
          762
          763
          764
          765
          766
          767
          768
          769
          770
          771
          772
          773
          774
          775
          776
          777
          778
          779
          780
          781
          782
          783
          784
          785
          786
          787
          788
          789
          790
          791
          792
          793
          794
          795
          796
          797
          798
          799
          800
          801
          802
          803
          804
          805
          806
          807
          808
          809
          810
          811
          812
          813
          814
          815
          816
          817
          818
          819
          820
          821
          822
          823
          824
          825
          826
          827
          828
          829
          830
          831
          832
          833
          834
          835
          836
          837
          838
          839
          840
          841
          842
          843
          844
          845
          846
          847
          848
          849
          850
          851
          852
          853
          854
          855
          856
          857
          858
          859
          860
          861
          862
          863
          864
          865
          866
          867
          868
          869
          870
          871
          872
          873
          874
          875
          876
          877
          878
          879
          880
          881
          882
          883
          884
          885
          886
          887
          888
          889
          890
          891
          892
          893
          894
          895
          896
          897
          898
          899
          900
          901
          902
          903
          904
          905
          906
          907
          908
          909
          910
          911
          912
          913
          914
          915
          916
          917
          918
          919
          920
          921
          922
          923
          924
          925
          926
          927
          928
          929
          930
          931
          932
          933
          934
          935
          936
          937
          938
          939
          940
          941
          942
          943
          944
          945
          946
          947
          948
          949
          950
          951
          952
          953
          954
          955
          956
          957
          958
          959
          960
          961
          962
          963
          964
          965
          966
          967
          968
          969
          970
          971
          972
          973
          974
          975
          976
          977
          978
          979
          980
          981
          982
          983
          984
          985
          986
          987
          988
          989
          990
          991
          992
          993
          994
          995
          996
          997
          998
          999
          1000
          1001
          1002
          1003
          1004
          1005
          1006
          1007
          1008
          1009
          1010
          1011
          1012
          1013
          1014
          1015
          1016
          1017
          1018
          1019
          1020
          1021
          1022
          1023
          1024
          1025
          1026
          1027
          1028
          1029
          1030
          1031
          1032
          1033
          1034
          1035
          1036
          1037
          1038
          1039
          1040
          1041
          1042
          1043
          1044
          1045
          1046
          1047
          1048
          1049
          1050
          1051
          1052
          1053
          1054
          1055
          1056
          1057
          1058
          1059
          1060
          1061
          1062
          1063
          1064
          1065
          1066
          1067
          1068
          1069
          1070
          1071
          1072
          1073
          1074
          1075
          1076
          1077
          1078
          1079
          1080
          1081
          1082
          1083
          1084
          1085
          1086
          1087
          1088
          1089
          1090
          1091
          1092
          1093
          1094
          1095
          1096
          1097
          1098
          1099
          1100
          1101
          1102
          1103
          1104
          1105
          1106
          1107
          1108
          1109
          1110
          1111
          1112
          1113
          1114
          1115
          1116
          1117
          1118
          1119
          1120
          1121
          1122
          1123
          1124
          1125
          1126
          1127
          1128
          1129
          1130
          1131
          1132
          1133
          1134
          1135
          1136
          1137
          1138
          1139
          1140
          1141
          1142
          1143
          1144
          1145
          1146
          1147
          1148
          1149
          1150
          1151
          1152
          1153
          1154
          1155
          1156
          1157
          1158
          1159
          1160
          1161
          1162
          1163
          1164
          1165
          1166
          1167
          1168
          1169
          1170
          1171
          1172
          1173
          1174
          1175
          1176
          1177
          1178
          1179
          1180
          1181
          1182
          1183
          1184
          1185
          1186
          1187
          1188
          1189
          1190
          1191
          1192
          1193
          1194
          1195
          1196
          1197
          1198
          1199
          1200
          1201
          1202
          1203
          1204
          1205
          1206
          1207
          1208
          1209
          1210
          1211
          1212
          1213
          1214
          1215
          1216
          1217
          1218
          1219
          1220
          1221
          1222
          1223
          1224
          1225
          1226
          1227
          1228
          1229
          1230
          1231
          1232
          1233
          1234
          1235
          1236
          1237
          1238
          1239
          1240
          1241
          1242
          1243
          1244
          1245
          1246
          1247
          1248
          1249
          1250
          1251
          1252
          1253
          1254
          1255
          1256
          1257
          1258
          1259
          1260
          1261
          1262
          1263
          1264
          1265
          1266
          1267
          1268
          1269
          1270
          1271
          1272
          1273
          1274
          1275
          1276
          1277
          1278
          1279
          1280
          1281
          1282
          1283
          1284
          1285
          1286
          1287
          1288
          1289
          1290
          1291
          1292
          1293
          1294
          1295
          1296
          1297
          1298
          1299
          1300
          1301
          1302
          1303
          1304
          1305
          1306
          1307
          1308
          1309
          1310
          1311
          1312
          1313
          1314
          1315
          1316
          1317
          1318
          1319
          1320
          1321
          1322
          1323
          1324
          1325
          1326
          1327
          1328
          1329
          1330
          1331
          1332
          1333
          1334
          1335
          1336
          1337
          1338
          1339
          1340
          1341
          1342
          1343
          1344
          1345
          1346
          1347
          1348
          1349
          1350
          1351
          1352
          1353
          1354
          1355
          1356
          1357
          1358
          1359
          1360
          1361
          1362
          1363
          1364
          1365
          1366
          1367
          1368
          1369
          1370
          1371
          1372
          1373
          1374
          1375
          1376
          1377
          1378
          1379
          1380
          1381
          1382
          1383
          1384
          1385
          1386
          1387
          1388
          1389
          1390
          1391
          1392
          1393
          1394
          1395
          1396
          1397
          1398
          1399
          1400
          1401
          1402
          1403
          1404
          1405
          1406
          1407
          1408
          1409
          1410
          1411
          1412
          1413
          1414
          1415
          1416
          1417
          1418
          1419
          1420
          1421
          1422
          1423
          1424
          1425
          1426
          1427
          1428
          1429
          1430
          1431
          1432
          1433
          1434
          1435
          1436
          1437
          1438
          1439
          1440
          1441
          1442
          1443
          1444
          1445
          1446
          1447
          1448
          1449
          1450
          1451
          1452
          1453
          1454
          1455
          1456
          1457
          1458
          1459
          1460
          1461
          1462
          1463
          1464
          1465
          1466
          1467
          1468
          1469
          1470
          1471
          1472
          1473
          1474
          1475
          1476
          1477
          1478
          1479
          1480
          1481
          1482
          1483
          1484
          1485
          1486
          1487
          1488
          1489
          1490
          1491
          1492
          1493
          1494
          1495
          1496
          1497
          1498
          1499
          1500
          1501
          1502
          1503
          1504
          1505
          1506
          1507
          1508
          1509
          1510
          1511
          1512
          1513
          1514
          1515
          1516
          1517
          1518
          1519
          1520
          1521
          1522
          1523
          1524
          1525
          1526
          1527
          1528
          1529
          1530
          1531
          1532
          1533
          1534
          1535
          1536
          1537
          1538
          1539
          1540
          1541
          1542
          1543
          1544
          1545
          1546
          1547
          1548
          1549
          1550
          1551
          1552
          1553
          1554
          1555
          1556
          1557
          1558
          1559
          1560
          1561
          1562
          1563
          1564
          1565
          1566
          1567
          1568
          1569
          1570
          1571
          1572
          1573
          1574
          1575
          1576
          1577
          1578
          1579
          1580
          1581
          1582
          1583
          1584
          1585
          1586
          1587
          1588
          1589
          1590
          1591
          1592
          1593
          1594
          1595
          1596
          1597
          1598
          1599
          1600
          1601
          1602
          1603
          1604
          1605
          1606
          1607
          1608
          1609
          1610
          1611
          1612
          1613
          1614
          1615
          1616
          1617
          1618
          1619
          1620
          1621
          1622
          1623
          1624
          1625
          1626
          1627
          1628
          1629
          1630
          1631
          1632
          1633
          1634
          1635
          1636
          1637
          1638
          1639
          1640
          1641
          1642
          1643
          1644
          1645
          1646
          1647
          1648
          1649
          1650
          1651
          1652
          1653
          1654
          1655
          1656
          1657
          1658
          1659
          1660
          1661
          1662
          1663
          1664
          1665
          1666
          1667
          1668
          1669
          1670
          1671
          1672
          1673
          1674
          1675
          1676
          1677
          1678
          1679
          1680
          1681
          1682
          1683
          1684
          1685
          1686
          1687
          1688
          1689
          1690
          1691
          1692
          1693
          1694
          1695
          1696
          1697
          1698
          1699
          1700
          1701
          1702
          1703
          1704
          1705
          1706
          1707
          1708
          1709
          1710
          1711
          1712
          1713
          1714
          1715
          1716
          1717
          1718
          1719
          1720
          1721
          1722
          1723
          1724
          1725
          1726
          1727
          1728
          1729
          1730
          1731
          1732
          1733
          1734
          1735
          1736
          1737
          1738
          1739
          1740
          1741
          1742
          1743
          1744
          1745
          1746
          1747
          1748
          1749
          1750
          1751
          1752
          1753
          1754
          1755
          1756
          1757
          1758
          1759
          1760
          1761
          1762
          1763
          1764
          1765
          1766
          1767
          1768
          1769
          1770
          1771
          1772
          1773
          1774
          1775
          1776
          1777
          1778
          1779
          1780
          1781
          1782
          1783
          1784
          1785
          1786
          1787
          1788
          1789
          1790
          1791
          1792
          1793
          1794
          1795
          1796
          1797
          1798
          1799
          1800
          1801
          1802
          1803
          1804
          1805
          1806
          1807
          1808
          1809
          1810
          1811
          1812
          1813
          1814
          1815
          1816
          1817
          1818
          1819
          1820
          1821
          1822
          1823
          1824
          1825
          1826
          1827
          1828
          1829
          1830
          1831
          1832
          1833
          1834
          1835
          1836
          1837
          1838
          1839
          1840
          1841
          1842
          1843
          1844
          1845
          1846
          1847
          1848
          1849
          1850
          1851
          1852
          1853
          1854
          1855
          1856
          1857
          1858
          1859
          1860
          1861
          1862
          1863
          1864
          1865
          1866
          1867
          1868
          1869
          1870
          1871
          1872
          1873
          1874
          1875
          1876
          1877
          1878
          1879
          1880
          1881
          1882
          1883
          1884
          1885
          1886
          1887
          1888
          1889
          1890
          1891
          1892
          1893
          1894
          1895
          1896
          1897
          1898
          1899
          1900
          1901
          1902
          1903
          1904
          1905
          1906
          1907
          1908
          1909

```

```

In [30]: 19 # Mapping integer month values to abbreviated month names
20 df.head()
21
22 # Dropping the original 'Month' column
23 df.drop(columns='Month', inplace=True)

```

	States	Date	Frequency	Unemployment Rate	Estimated Employed	Estimated Labour Participation Rate	Region	longitude	latitude	Month_int	Month_name
0	Andhra Pradesh	2020-01-31	M	5.48	16635535	41.02	South	15.9129	79.74	1	Jan
1	Andhra Pradesh	2020-02-29	M	5.83	16545652	40.90	South	15.9129	79.74	2	Feb
2	Andhra Pradesh	2020-03-31	M	5.79	15881197	39.18	South	15.9129	79.74	3	Mar
3	Andhra Pradesh	2020-04-30	M	20.51	11336911	33.10	South	15.9129	79.74	4	Apr
4	Andhra Pradesh	2020-05-31	M	17.43	12988845	36.46	South	15.9129	79.74	5	May

```

In [31]: 1 df_stat = df[['Estimated Unemployment Rate', 'Estimated Employed', 'Estimated Labour Participation Rate']
2         print(round(df_stat.describe().T, 2))

```

	count	mean	std
Estimated Unemployment Rate	267.0	12.24	10.80
Estimated Employed	267.0	13962105.72	13366318.36
Estimated Labour Participation Rate	267.0	41.68	7.85

	min	25%	50%
Estimated Unemployment Rate	0.50	4.84	9.65
Estimated Employed	117542.00	2838930.50	9732417.00
Estimated Labour Participation Rate	16.77	37.26	40.39

	75%	max
Estimated Unemployment Rate	16.76	75.85
Estimated Employed	21878686.00	59433759.00
Estimated Labour Participation Rate	44.06	69.69

```

In [32]: 1 region_stats = df.groupby(['Region'])[['Estimated Unemployment Rate', 'Estimated Employed',
In [33]: 1         'Estimated Labour Participation Rate']].mean().reset_index()
2         import matplotlib.pyplot as plt
3         import seaborn as sns
4         print(round(region_stats, 2))

```

```

In [34]: 1 #Boxplot of Unemployment rate per States
2         hm = df[['Estimated Unemployment Rate', 'Estimated Employed', 'Estimated Labour Participation Rate', 'Region']]
3         hm = hm.corr()
4         plt.figure(figsize=(6,4))
5         sns.set_context('notebook', font_scale=1)
6         sns.boxplot(data=hm, y='Estimated Unemployment Rate', x='Region', hue='Region', palette='cubehelix', cmap=sns.cubehelix_palette(as_cmap=True))
7         plt.show()
8         print(hm)
9         print(hm.corr())
10        print(hm)
11        print(hm.corr())
12        print(hm)
13        print(hm)
14        print(hm)
15        print(hm)
16        print(hm)
17        print(hm)
18        print(hm)
19        print(hm)
20        print(hm)
21        print(hm)
22        print(hm)
23        print(hm)
24        print(hm)
25        print(hm)
26        print(hm)
27        print(hm)
28        print(hm)
29        print(hm)
30        print(hm)
31        print(hm)
32        print(hm)
33        print(hm)
34        print(hm)
35        print(hm)
36        print(hm)
37        print(hm)
38        print(hm)
39        print(hm)
40        print(hm)
41        print(hm)
42        print(hm)
43        print(hm)
44        print(hm)
45        print(hm)
46        print(hm)
47        print(hm)
48        print(hm)
49        print(hm)
50        print(hm)
51        print(hm)
52        print(hm)
53        print(hm)
54        print(hm)
55        print(hm)
56        print(hm)
57        print(hm)
58        print(hm)
59        print(hm)
60        print(hm)
61        print(hm)
62        print(hm)
63        print(hm)
64        print(hm)
65        print(hm)
66        print(hm)
67        print(hm)
68        print(hm)
69        print(hm)
70        print(hm)
71        print(hm)
72        print(hm)
73        print(hm)
74        print(hm)
75        print(hm)
76        print(hm)
77        print(hm)
78        print(hm)
79        print(hm)
80        print(hm)
81        print(hm)
82        print(hm)
83        print(hm)
84        print(hm)
85        print(hm)
86        print(hm)
87        print(hm)
88        print(hm)
89        print(hm)
90        print(hm)
91        print(hm)
92        print(hm)
93        print(hm)
94        print(hm)
95        print(hm)
96        print(hm)
97        print(hm)
98        print(hm)
99        print(hm)
100       print(hm)

```

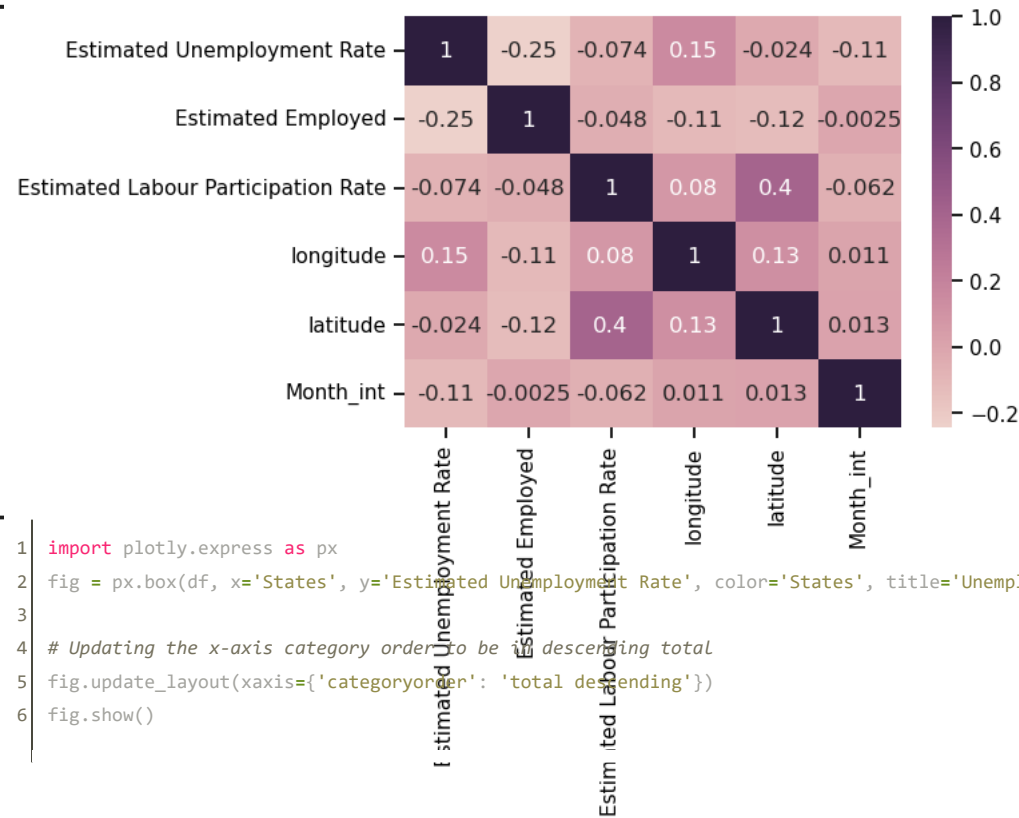
```

In [33]: 1 region_stats = df.groupby(['Region'])[['Estimated Unemployment Rate', 'Estimated Employed',
2         'Estimated Labour Participation Rate']].mean().reset_index()
3 print(round(region_stats, 2))

In [34]: 1 #Region of Unemployment Rate per States
2 hm = smf[['Estimated Unemployment Rate', 'Estimated Employed', 'Estimated Labour Participation Rate', '1
3 hm = hm.corr()
4 plt.figure(figsize=(6,4))
5 sns.set_context('notebook', font_scale=1)
6 sns.heatmap(hm, annot=True, cmap=sns.cubehelix_palette(as_cmap=True))

~
2 Axes: >
3
4

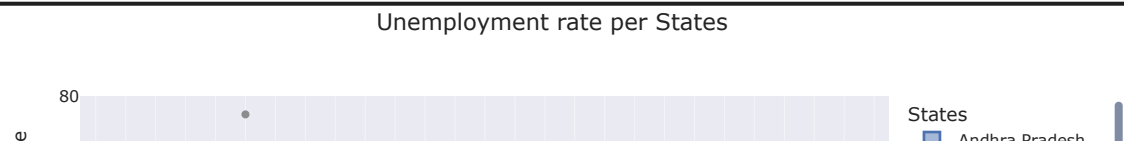
```



```

In [35]: 1 import plotly.express as px
2 fig = px.box(df, x='States', y='Estimated Unemployment Rate', color='States', title='Unemployment rate
3
4 # Updating the x-axis category order to be in descending total
5 fig.update_layout(xaxis={'categoryorder': 'total descending'})
6 fig.show()

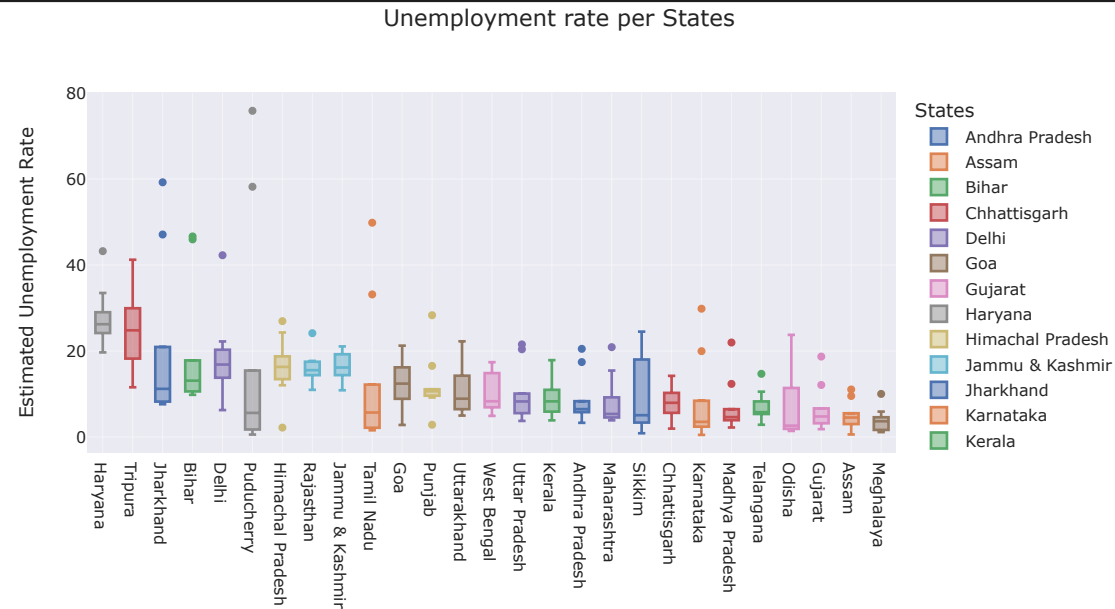
```




```

In [35]: 1 import plotly.express as px
2 fig = px.box(df, x='States', y='Estimated Unemployment Rate', color='States', title='Unemployment rate
3
4 # Updating the x-axis category order to be in descending total
5 fig.update_layout(xaxis={'categoryorder': 'total descending'})
6 fig.show()

```



```

In [37]: 1 fig = px.bar(df, x='Region', y='Estimated Unemployment Rate', animation_frame='Month_name', color='State
2 title='Unemployment Rate across Regions from Jan. 2020 to Oct. 2020', height=700, template
3
4 # Updating the x-axis category order to be in descending total
5 fig.update_layout(xaxis={'categoryorder': 'total descending'})
6
7 # Adjusting the animation frame duration
8 fig.layout.updatemenus[0].buttons[0].args[1]["frame"]["duration"] = 1000
9 fig.show()

```

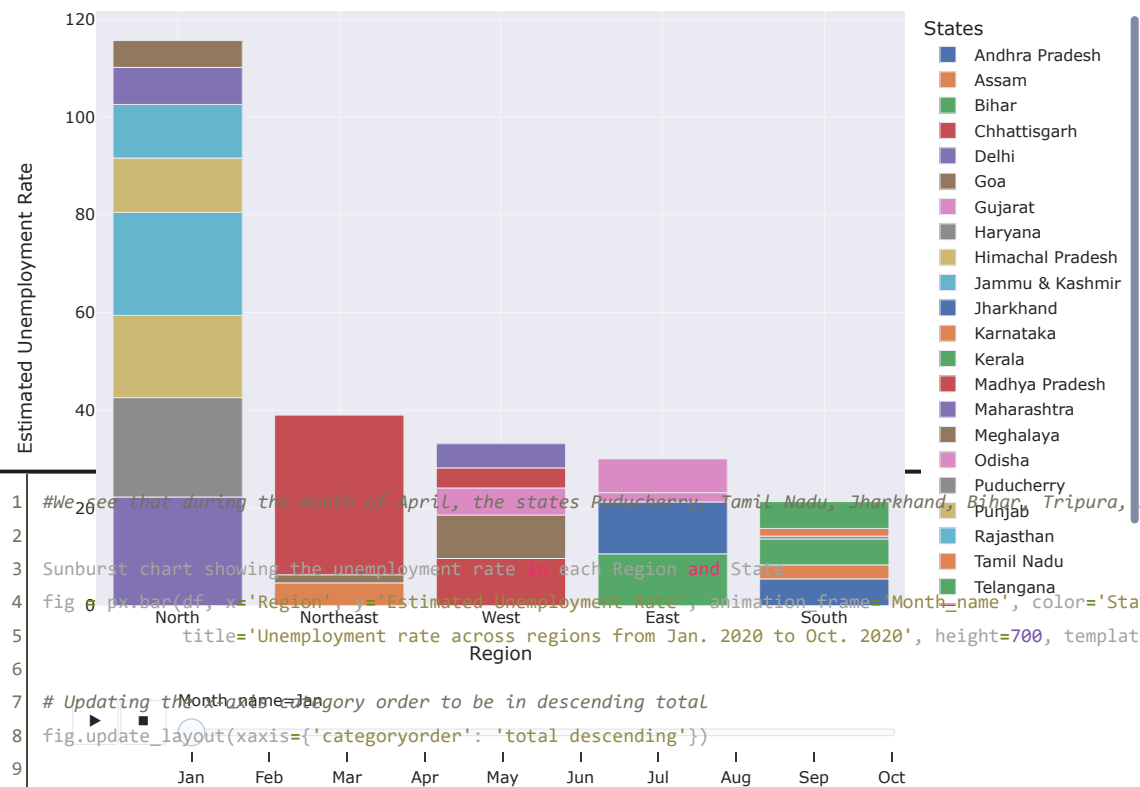
Unemployment rate across regions from Jan. 2020 to Oct. 2020

```

In [37]: 1 fig = px.bar(df, x='Region', y='Estimated Unemployment Rate', animation_frame='Month_name', color='State',
2             title='Unemployment rate across Regions from Jan. 2020 to Oct. 2020', height=700, template
3
4             # Updating the x-axis category order to be in descending total
5             fig.update_layout(xaxis={'categoryorder': 'total descending'})
6
7             # Adjusting the animation frame duration
8             fig.layout.updatemenus[0].buttons[0].args[1]["frame"]["duration"] = 1000
9             fig.show()

```

Unemployment rate across regions from Jan. 2020 to Oct. 2020



```

In [39]: 1 # We see that during the month of April, the states Puducherry, Tamil Nadu, Jharkhand, Bihar, Tripura, h
2
3             Sunburst chart showing the unemployment rate in each Region and State
4             fig = px.bar(df, x='Region', y='Estimated Unemployment Rate', animation_frame='Month_name', color='State',
5             title='Unemployment rate across regions from Jan. 2020 to Oct. 2020', height=700, template
6
7             # Updating the x-axis category order to be in descending total
8             fig.update_layout(xaxis={'categoryorder': 'total descending'})
9
10            # Adjusting the animation frame duration
11            fig.layout.updatemenus[0].buttons[0].args[1]["frame"]["duration"] = 1000
12            fig.show()

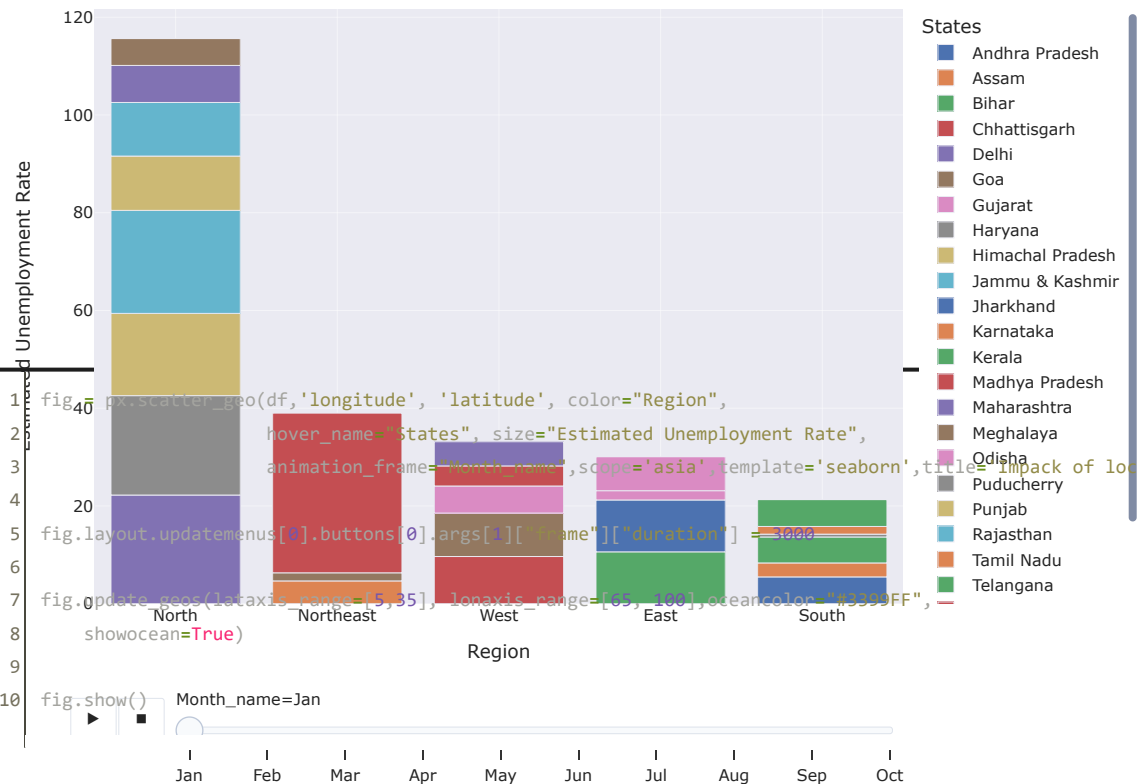
```

```

In [39]: 1 #We see that during the month of April, the states Puducherry, Tamil Nadu, Jharkhand, Bihar, Tripura, h
2
3 Sunburst chart showing the unemployment rate in each Region and State
4 fig = px.sunburst(df, x='Region', y='Estimated Unemployment Rate', animation_frame='Month_name', color='Stat
5 title='Unemployment rate across regions from Jan. 2020 to Oct. 2020', height=700, template
6
7 # Updating the Month name category order to be in descending total
8 fig.update_layout(xaxis={'categoryorder': 'total descending'})
9
10 # Adjusting the animation frame duration
11 fig.layout.update(menu=[0].buttons[0].args[1]["frame"]["duration"] = 1000)
12 fig.show()

```

Unemployment rate across regions from Jan. 2020 to Oct. 2020

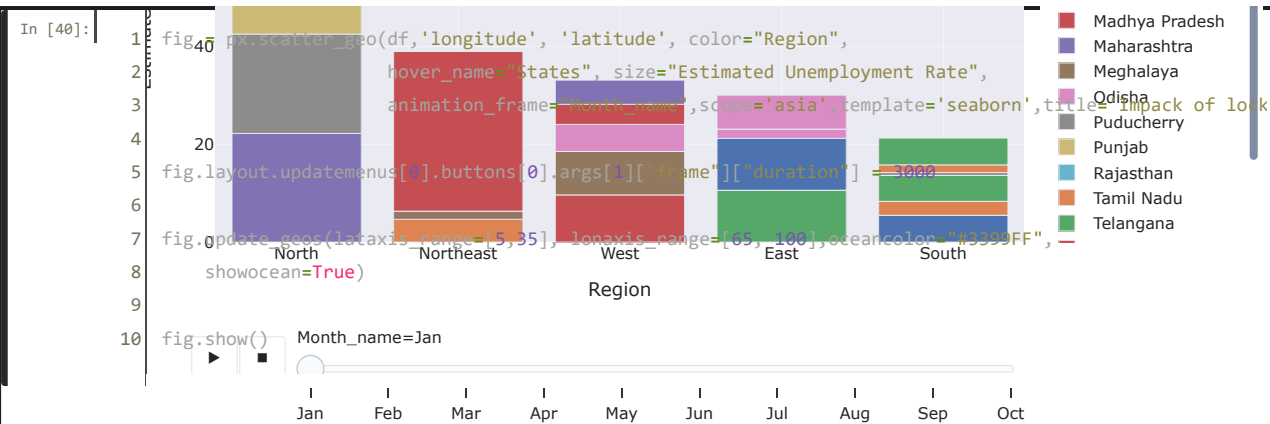


```

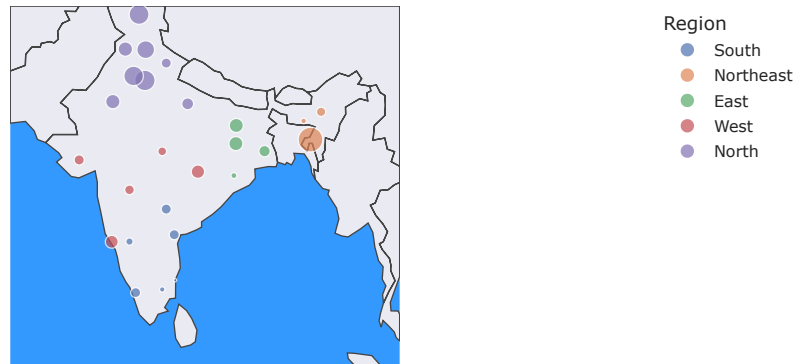
In [40]: 1 fig = px.scatter_geo(df, 'longitude', 'latitude', color="Region",
2
3 hover_name="States", size="Estimated Unemployment Rate",
4 animation_frame="Month_name", scope='asia', template='seaborn', title='Impact of lockdown on Employment across regions'
5 fig.layout.update(menu=[0].buttons[0].args[1]["frame"]["duration"] = 1000)
6
7 fig.update_geos(lataxis_range=[5, 35], lonaxis_range=[0, 100], oceancolor='rgb(99, 99, 99)',
8 showocean=True)
9
10 fig.show()

```

Impact of lockdown on Employment across regions



Impact of lockdown on Employment across regions



In [42]:

```

1 # Filtering data for the period before the Lockdown (January to April)
2 bf_lockdown = df[(df['Month_int'] >= 1) & (df['Month_int'] <= 4)]
3
4 # Filtering data for the Lockdown period (April to July)
5 lockdown = df[(df['Month_int'] >= 4) & (df['Month_int'] <= 7)]
6
7 # Calculating the mean unemployment rate before Lockdown by state
8 m_bf_lock = bf_lockdown.groupby('States')['Estimated Unemployment Rate'].mean().reset_index()
9
10 # Calculating the mean unemployment rate after Lockdown by state
11 m_lock = lockdown.groupby('States')['Estimated Unemployment Rate'].mean().reset_index()
12
13 # Combining the mean unemployment rates before and after Lockdown by state
14 m_lock['Unemployment Rate before lockdown'] = m_bf_lock['Estimated Unemployment Rate']

```

```

In [42]: 1 # Filtering data for the period before the lockdown (January to April)
2 bf_lockdown = df[(df['Month_int'] >= 1) & (df['Month_int'] <= 4)]
3
4 # Filtering data for the lockdown period (April to July)
5 lockdown = df[(df['Month_int'] >= 4) & (df['Month_int'] <= 7)]
6
7 # Calculating the mean unemployment rate before lockdown by state
8 m_bf_lock = bf_lockdown.groupby('States')['Estimated Unemployment Rate'].mean().reset_index()
9
10 # Calculating the mean unemployment rate after lockdown by state
11 m_lock = lockdown.groupby('States')['Estimated Unemployment Rate'].mean().reset_index()
12
13 # Combining the mean unemployment rates before and after lockdown by state
14 m_lock['Unemployment Rate before lockdown'] = m_bf_lock['Estimated Unemployment Rate']
15
16 m_lock.columns = ['States', 'Unemployment Rate before lockdown', 'Unemployment Rate after lockdown']
17 m_lock.head()

```

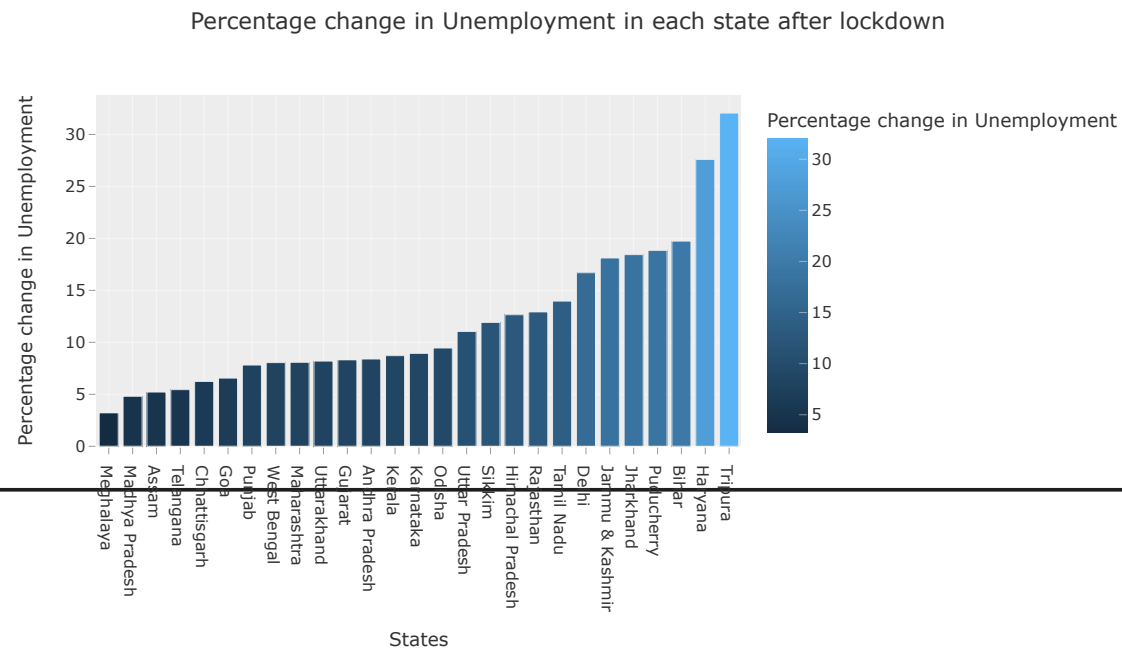
	States	Unemployment Rate before lockdown	Unemployment Rate after lockdown
0	Andhra Pradesh	12.3975	9.4025
1	Assam	6.2450	6.2250
2	Bihar	30.8025	20.7425
3	Chhattisgarh	9.6025	7.2450
4	Delhi	24.3600	17.6975

```

In [44]: 1 # percentage change in unemployment rate
2
3 m_lock['Percentage change in Unemployment'] = round(m_lock['Unemployment Rate after lockdown'] - m_lock
4 plot_per = m_lock.sort_values('Percentage change in Unemployment')
5
6
7 # percentage change in unemployment after Lockdown
8
9 fig = px.bar(plot_per, x='States', y='Percentage change in Unemployment', color='Percentage change in Une
10 title='Percentage change in Unemployment in each state after lockdown', template='ggplot2')
11 fig.show()

```

```
In [44]: 1 # percentage change in unemployment rate
2
3 m_lock['Percentage change in Unemployment'] = round(m_lock['Unemployment Rate after lockdown'] - m_lock
4 plot_per = m_lock.sort_values('Percentage change in Unemployment')
5
6
7 # percentage change in unemployment after Lockdown
8
9 fig = px.bar(plot_per, x='States',y='Percentage change in Unemployment',color='Percentage change in Une
10             title='Percentage change in Unemployment in each state after lockdown',template='ggplot2')
11 fig.show()
```



States

Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
Goa
Gujarat
Haryana
Himachal Pradesh
Jharkhand
Karnataka
Kerala
Madhya Pradesh
Maharashtra
Manipur
Meghalaya
Mizoram
Nagaland
Odisha
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttar Pradesh
West Bengal

```
In [46]: 1 # Analyzing COVID-19 Impact on Unemployment in India
          2
          3 # Objective:
          4
          5 # The primary aim of this analysis is to assess the repercussions of the COVID-19 pandemic on India's j
          6
          7 # Dataset Details:
          8
          9 # The dataset provides insights into the unemployment scenario across different Indian states:
         10
         11 # States: The states within India.
         12 # Date: The date when the unemployment rate was recorded.
         13 # Measuring Frequency: The frequency at which measurements were taken (Monthly).
         14 # Estimated Unemployment Rate (%): The percentage of individuals unemployed in each state of India.
```

```

In [46]: 1 # Analyzing COVID-19 Impact on Unemployment in India
2
3 # Objective:
4
5 # The primary aim of this analysis is to assess the repercussions of the COVID-19 pandemic on India's j
6
7 # Dataset Details:
8
9 # The dataset provides insights into the unemployment scenario across different Indian states:
10
11 # States: The states within India.
12 # Date: The date when the unemployment rate was recorded.
13 # Measuring Frequency: The frequency at which measurements were taken (Monthly).
14 # Estimated Unemployment Rate (%): The percentage of individuals unemployed in each state of India.
15 # Estimated Employed Individuals: The count of people currently employed.
16 # Estimated Labour Participation Rate (%): The proportion of the working population (age group: 16-64 y
17 # This dataset aids in comprehending the unemployment dynamics across India's states during the COVID-1
18
19 # Importing necessary libraries
20
21 # import pandas as pd
22 # import numpy as np
23 # import calendar
24 # Loading the dataset into pandas dataframe
25
26 # df = pd.read_csv('/kaggle/input/unemployment-in-india/Unemployment_Rate_upto_11_2020.csv')
27 # df.head()
28 # Region      Date      Frequency      Estimated Unemployment Rate (%) Estimated Employed      Estimated Labour Pa
29 # 0 Andhra Pradesh 31-01-2020 M      5.48      16635535      41.02      South      15.9129      79.74
30 # 1 Andhra Pradesh 29-02-2020 M      5.83      16545652      40.90      South      15.9129      79.74
31 # 2 Andhra Pradesh 31-03-2020 M      5.79      15881197      39.18      South      15.9129      79.74
32 # 3 Andhra Pradesh 30-04-2020 M      20.51      11336911      33.10      South      15.9129      79.74
33 # 4 Andhra Pradesh 31-05-2020 M      17.43      12988845      36.46      South      15.9129      79.74
34 # Basic information about the dataset
35 # 6 Region.1
36 # 7 Longitude
37 # 8 Latitude
38 # dtypes: float64(4), int64(1), object(4)
39 # RangeIndex: 267 entries, 0 to 266
40 # memory usage: 18.9+ KB
41 # Data columns (total 9 columns):
42 # Checking for null values
43 # # Column Non-Null Count Dtype
44 # ---
45 # df.isnull().sum()
46 # 0 Region 0 267 non-null object
47 # Date

```



```

43 # 0 Andhra Pradesh 31-01-2020 M 5.48 16635535 41.02 South 15.9129 79.74
44 # 1 Andhra Pradesh 29-02-2020 M 5.83 16545652 40.90 South 15.9129 79.74
45 # 2 Andhra Pradesh 31-03-2020 M 5.79 15881197 39.18 South 15.9129 79.74
46 # 3 Andhra Pradesh 30-04-2020 M 20.51 11336911 33.10 South 15.9129 79.74
47 # 4 Andhra Pradesh 31-05-2020 M 17.43 12988845 36.46 South 15.9129 79.74
48 # Basic information about the dataset
49 # 6 Region.1 267 non-null object
50 # 7 Longitude 267 non-null float64
51 # 8 Latitude 267 non-null float64
52 # df.info()
53 # <class 'pandas.core.frame.DataFrame'>
54 # dtypes: float64(4), int64(1), object(4)
55 # RangeIndex: 267 entries, 0 to 266
56 # memory usage: 18.9+ KB
57 # Data columns (total 9 columns):
58 # Checking for null values
59 # # Column Non-Null Count Dtype
60 # ---
61 # 0 Region 267 non-null object
62 # 1 Date 0
63 # 2 Frequency 0
64 # 3 Estimated Unemployment Rate (%) 0
65 # 4 Estimated Employed 0
66 # 5 Estimated Labour Participation Rate (%) 0
67 # 6 Region.1 0
68 # 7 Longitude 0
69 # 8 Latitude 0
70 # dtype: int64
71 # Formatting the columns and their datatypes
72 #
73 # import datetime as dt
74 # # Renaming columns for better clarity
75 # df.columns = ['States', 'Date', 'Frequency', 'Estimated Unemployment Rate', 'Estimated Employed',
76 #               'Estimated Labour Participation Rate', 'Region', 'Longitude', 'Latitude']
77 # # Mapping integer month values to abbreviated month names
78 # df['Month'] = df['Month'].apply(lambda x: calendar.month_abbr[x])
79 # df['Date'] = pd.to_datetime(df['Date'], dayfirst=True)
80 # # Dropping the original 'Month' column
81 # df.drop('Month', axis=1, inplace=True)
82 # # Converting 'Frequency' and 'Region' columns to categorical data type
83 # df['Frequency'] = df['Frequency'].astype('category')
84 # df['Region'] = df['Region'].astype('category')
85 # # Creating a new column 'Month' from 'Date' and adding it to the DataFrame
86 # df['Month'] = df['Date'].dt.month
87 # # Dropping the original 'Date' column
88 # df.drop('Date', axis=1, inplace=True)
89 # # Converting 'Month' column to integer format
90 # df['Month'] = df['Month'].apply(lambda x: int(x))
91 # # Basic statistics

```

[illegible]

```

129 # Estimated Unemployment Rate      40.11      16.76      75.85
130 # Estimated Employed      38.701878686.00  59433759.00
131 # Estimated Labour Participation Rate 52.06      44.06      69.69
132 # region_stats = df.groupby(['Region'])[['Estimated Unemployment Rate', 'Estimated Employed',
133 # 4      41.26 'Estimated Labour Participation Rate']].mean().reset_index()
134 # plt.figure(figsize=(10,5))
135 # import seaborn as sns
136 # sns.set(font_scale=1.4)
137 # plt.figure(figsize=(10,5))
138 # sns.heatmap(df[['Estimated Unemployment Rate', 'Estimated Employed', 'Estimated Labour Participation Rate']],
139 # 3      South      10.45      14040589.33
140 # hm = df[['Estimated Unemployment Rate', 'Estimated Employed', 'Estimated Labour Participation Rate'],
141 # 1      North      10.95      3617105.53
142 # hm = hm.corr()
143 # plt.figure(figsize=(10,5))
144 # sns.set_context('notebook', font_scale=1)
145 # sns.heatmap(data=hm, annot=True, cmap=sns.cubehelix_palette(as_cmap=True))
146 # <Axes: >
147 # Boxplot of Unemployment rate per States
148
149 # import plotly.express as px
150 # fig = px.box(df, x='States', y='Estimated Unemployment Rate', color='States', title='Unemployment rat
151
152 # # Updating the x-axis category order to be in descending total
153 # fig.update_layout(xaxis={'categoryorder': 'total descending'})
154 # fig.show()
155 # Scatter matrix cosidering the employed and unemployed rates
156
157 # fig = px.scatter_matrix(df, template='seaborn', dimensions=['Estimated Unemployment Rate', 'Estimated E
158 # Bar chart showing the unemployment rate across regions from Jan. 2020 to Oct. 2020
159 # fig.show()
160 # fig = px.line(df, x='Month_name', y='Estimated Unemployment Rate', color='States', animation_frame='Month_name', color='St
161 # title='Unemployment rate across regions from Jan. 2020 to Oct. 2020', height=700, templa
162 # plot_unemp = df[['Estimated Unemployment Rate', 'States']]
163 # df_unemployed = df[df['Estimated Unemployment Rate'] > 40]
164 # fig.update_layout(xaxis={'categoryorder': 'total descending'})
165 # df_unemployed = df_unemployed.sort_values('Estimated Unemployment Rate')
166 # # Adjusting the animation frame duration
167 # fig.layout.update(unemp=fig.layout.unemp, args=[fig.layout.unemp['duration']])
168 # fig.show()
169 # We see that during the month of April, the states Puducherry, Tamil Nadu, Jharkhand, Bihar, Tripura,
170 # Haryana and Jharkhand have Long been the most unemployed.
171 # Sunburst chart showing the unemployment rate in each Region and State

```

```

138 # Bar chart showing the unemployment rate across regions from Estimated Unemployment Rate', color=
139 # fig.show()
140 # Bar chart showing the Unemployment Rate by Region and State, animation_frame='Month_name', color='St
141 # title='Unemployment rate across regions from Jan. 2020 to Oct. 2020', height=700, template=
142 # plot_unemp = df[['Estimated Unemployment Rate', 'States']]
143 # df_unemployed = px.scatter(df_unemployed, x='States', y='Estimated Unemployment Rate', color='St
144 # fig.update_layout(xaxis={'categoryorder': 'total descending'})
145 # df_unemployed = df_unemployed.sort_values('Estimated Unemployment Rate')
146 # Adjusting the animation frame duration
147 # fig.layout.update(frames=[0], buttons=[0], args=[1], frame=[1], duration=[1], color=[1], title=[1])
148 # fig.show() template='seaborn')
149 # Fig shows that during the month of April, the states Puducherry, Tamil Nadu, Jharkhand, Bihar, Tripura,
150 # Haryana and Jharkhand have long been the most unemployed.
151 # Sunburst chart showing the unemployment rate in each Region and State
152
153
154
155
156
157 ## Creating a DataFrame with relevant columns
158 # unemployed_df = df[['States', 'Region', 'Estimated Unemployment Rate', 'Estimated Employed', 'Estimat
159
160
161 # unemployed = unemployed_df.groupby(['Region', 'States'])['Estimated Unemployment Rate'].mean().reset_
162
163
164 ## Creating a Sunburst chart
165 # fig = px.sunburst(unemployed, path=['Region', 'States'], values='Estimated Unemployment Rate', color_
166 # title='Unemployment rate in each Region and State', height=550, template='presentat
167
168
169 # fig.show()
170 # Impact of Lockdown on States Estimated Employed
171
172
173 # fig = px.scatter_geo(df, 'Longitude', 'Latitude', color="Region",
174 # hover_name="States", size="Estimated Unemployment Rate",
175 # Lockdown = df[(df['Month_in_Year'] == 'April', scope='Asia', template='seaborn', title='Impact of Lo
176
177
178 # fig.layout.update(frames=[0], buttons=[0], args=[1], frame=[1], duration=[1], color=[1], title=[1])
179 # m_lock = bf_lockdown.groupby('States')['Estimated Unemployment Rate'].mean().reset_index()
180 # fig.update_geos(lataxis_range=[5, 35], lonaxis_range=[65, 100], oceancolor="#3399FF",
181 # Call the mean unemployment rate after lockdown by state
182 # m_lock = Lockdown.groupby('States')['Estimated Unemployment Rate'].mean().reset_index()
183 # fig.show()
184 # The combination of the regions unemployment rate before and after lockdown by state
185 # m_lock['Unemployment Rate before Lockdown'] = m_lock['Estimated Unemployment Rate']
186
187
188 ## Filtering data for the period before the Lockdown (January to April)
189 # m_lock = m_lock[m_lock['Month_in_Year'] != 'April']
190 # m_lock.head()
191
192
193 # State Unemployment Rate before Lockdown, Unemployment Rate after Lockdown
194 # m_lock.head()

```

```

201 # Lockdown = df[(df['Month_int'] > 4) & (df['Month_int'] <= 7)], template='seaborn', title='Impact of Lo
202
203 # fig.set_xlabel('Date') # Date of the month before lockdown by state
204 # m_bf_lock = bf_lockdown.groupby('States')['Estimated Unemployment Rate'].mean().reset_index()
205
206 # fig.update_geos(lataxis_range=[5,35], lonaxis_range=[65, 100], oceancolor="#3399FF",
207 # # Call to the mean unemployment rate after lockdown by state
208 # m_lock = lockdown.groupby('States')['Estimated Unemployment Rate'].mean().reset_index()
209
210 # fig.show()
211 # The combination of the states before and after lockdown by state
212 # m_lock['Unemployment Rate before Lockdown'] = m_bf_lock['Estimated Unemployment Rate']
213
214 # # Filtering data for the period before the lockdown (January to April)
215 # before_lockdown = df[(df['Month_int'] < 4) & (df['Month_int'] <= 4)]
216 # m_lock.head()
217
218 # # Filtering data for the period after lockdown (April to July)
219 # 0 Andhra Pradesh 12.3975 9.4025
220
221 # 1 Assam 6.2450 6.2250
222
223 # 2 Bihar 30.8025 20.7425
224
225 # 3 Chhattisgarh 9.6025 7.2450
226
227 # 4 Delhi 24.3600 17.6975
228
229 # # percentage change in unemployment rate
230
231
232
233
234
235
236 # m_lock['Percentage change in Unemployment'] = round(m_lock['Unemployment Rate after Lockdown'] - m_Lo
237 # plot_per = m_lock.sort_values('Percentage change in Unemployment')
238
239
240 # # percentage change in unemployment after lockdown
241
242
243 # fig = px.bar(plot_per, x='States', y='Percentage change in Unemployment', color='Percentage change in U
244 # title='Percentage change in Unemployment in each state after Lockdown', template='ggplot2'

```

In [50]:

```

244 # fig.show()
245 # The most affected states/territories in India during the Lockdown in case of unemployment were:

```

In [51]:

```

246 import pandas as pd
247 # Tripura
248 import matplotlib.pyplot as plt
249 # Haryana
250 import seaborn as sns
251 # Bihar
252 from sklearn.model_selection import train_test_split
253 # Puducherry
254 from sklearn.linear_model import LinearRegression
255 # Jharkhand
256 from sklearn.linear_model import Lasso
257 # Jammu & Kashmir
258 from sklearn import metrics
259 # Delhi

```

In [53]:

1

In [47]:

```

1 #####car_data = pd.read_csv('C:/Users/KARNARAJ RATHOD/OneDrive/Desktop/oasis/car data.csv')

```

In [54]:

```

1 # inspecting the first 5 rows of the dataframe

```

```

In [50]: 244 # fig.show()
          245 # The most affected states/territories in India during the Lockdown in case of unemployment were:

In [51]: 246
          1 import pandas as pd
          2 # Tripura
          3 import matplotlib.pyplot as plt
          4 # Haryana
          5 import seaborn as sns
          6 # Bihar
          7 from sklearn.model_selection import train_test_split
          8 # Puducherry
          9 from sklearn.linear_model import LinearRegression
         10 # Jharkhand
         11 from sklearn.linear_model import Lasso
         12 # Jammu & Kashmir
         13 from sklearn import metrics
         14 # Delhi

In [53]: 1

In [47]: 2 #####head price providers/KARNARAJ RATHOD/OneDrive/Desktop/oasis/car_data.csv")

In [54]: 1 # inspecting the first 5 rows of the dataframe
          2 car_dataset.head()

          Car_Name  Year  Selling_Price  Present_Price  Driven_kms  Fuel_Type  Selling_type  Transmission  Owner
0  ritz           2014    3.35           5.59           27000      Petrol      Dealer        Manual        0
1  sx4            2013    4.75           9.54           43000      Diesel      Dealer        Manual        0
2  ciaz           2017    7.25           9.85           6900       Petrol      Dealer        Manual        0
3  wagon r        2011    2.85           4.15           5200       Petrol      Dealer        Manual        0
4  swift          2014    4.60           6.87           42450      Diesel      Dealer        Manual        0

In [55]: 1 # checking the number of rows and columns
          2 car_dataset.shape

          (301, 9)

In [56]: 1 # checking the number of missing values
In [57]: 2 # getting some information about the dataset
          3 car_dataset.isnull().sum()
          4 car_dataset.info()

Car Name      0
<class 'pandas.core.frame.DataFrame'>
Year          0
RangeIndex: 301 entries, 0 to 300
Selling_Price 0
Data columns (total 9 columns):
# Column      Non-Null Count  Dtype
---  -
Present_Price  0
Driven_kms     0
Fuel_Type      0
0 - Car Name   301 non-null  object
1 - Year       301 non-null  int64
2 - Transmission  301 non-null  float64
3 - Selling_Price  301 non-null  float64
4 - Owner      301 non-null  float64
dtype: int64
5 - Driven_kms  301 non-null  int64
6 - Fuel_Type   301 non-null  object
7 - Selling_type 301 non-null  object
8 - Transmission 301 non-null  object
9 - Owner       301 non-null  int64
dtypes: float64(2), int64(3), object(4)
memory usage: 21.3+ KB

```

```
In [56]: 1 # checking the number of missing values
In [57]: 2 # getting some information about the dataset
3 car_dataset.isnull().sum()
4 car_dataset.info()
```

```
Car_Name      0
<class 'pandas.core.frame.DataFrame'>
Year          0
RangeIndex: 301 entries, 0 to 300
Selling_Price 0
Data columns (total 9 columns):
 #   Column              Non-Null Count  Dtype
---  -
0   Car_Name            301 non-null   object
1   Year                301 non-null   int64
2   Selling_Price        301 non-null   float64
3   Present_Price        301 non-null   float64
4   Driven_kms           301 non-null   int64
5   Fuel_Type            301 non-null   object
6   Selling_type         301 non-null   object
7   Transmission         301 non-null   object
8   Owner               301 non-null   int64
dtypes: float64(2), int64(3), object(4)
memory usage: 21.3+ KB
```

```
In [58]: 1 # checking the distribution of categorical data
2 print(car_dataset.Fuel_Type.value_counts())
3 print(car_dataset.Selling_type.value_counts())
4 print(car_dataset.Transmission.value_counts())
```

```
Fuel_Type
Petrol    239
Diesel     60
CNG        2
Name: count, dtype: int64

Selling_type
Dealer     195
Individual 106
Name: count, dtype: int64

Transmission
Manual     261
Automatic   40
Name: count, dtype: int64
```

```
In [60]: 1 e: count, dtype: int64
2 car_dataset.head()
```

```
In [59]: 1 # encoding "Fuel_Type" Column
2 car_dataset.replace({'Fuel_Type':{'Petrol':0,'Diesel':1,'CNG':2}},inplace=True)
3
4 # encoding "Seller_Type" Column
5 car_dataset.replace({'Selling_type':{'Dealer':0,'Individual':1}},inplace=True)
6
7 # encoding "Transmission" Column
```

```
In [61]: 8 car_dataset.replace({'Transmission':{'Manual':0,'Automatic':1}},inplace=True)
1 X = car_dataset.drop(['Car_Name','Selling_Price'],axis=1)
2 Y = car_dataset['Selling_Price']
```

```
In [62]: 1 print(X)
```

In [60]:	1	e: count, dtype: int64 car_dataset.head()
In [59]:	1	# encoding "Fuel_Type" Column car_dataset.replace({'Fuel_Type':{'Petrol':0,'Diesel':1,'CNG':2}},inplace=True)
	2	car_dataset.replace({'Selling_type':{'Dealer':0,'Individual':1}},inplace=True)
	3	# encoding "Seller_Type" Column
	4	car_dataset.replace({'Selling_type':{'Dealer':0,'Individual':1}},inplace=True)
	5	car_dataset.replace({'Selling_type':{'Dealer':0,'Individual':1}},inplace=True)
	6	# encoding "Transmission" Column
In [61]:	8	car_dataset.replace({'Transmission':{'Manual':0,'Automatic':1}},inplace=True)
	1	X = car_dataset.drop(['Car_Name','Selling_Price'],axis=1)
	2	Y = car_dataset['Selling_Price']
In [62]:	1	print(X)
		<pre> Year Present_Price Driven_kms Fuel_Type Selling_type Transmission \ 0 2014 5.59 27000 0 0 0 1 2013 9.54 43000 1 0 0 2 2017 9.85 6900 0 0 0 3 2011 4.15 5200 0 0 0 4 2014 6.87 42450 1 0 0 296 2016 11.60 33988 1 0 0 297 2015 5.90 60000 0 0 0 298 2009 11.00 87934 0 0 0 299 2017 12.50 9000 1 0 0 300 2016 5.90 5464 0 0 0 Owner 0 0 1 0 2 0 3 0 4 0 296 0 297 0 298 0 299 0 300 0 </pre>
In [63]:	1	print(Y)
		<pre> 0 0 1 0 299 0 000 3035 1 4.75 301 row(s) * 257 columns] 3 2.85 4 4.60 296 9.50 297 4.00 298 3.35 299 11.50 300 5.30 Name: Selling_Price, Length: 301, dtype: float64 </pre>
In [64]:	1	X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.1, random_state=2)
In [65]:		


```

0
1 print(Y)
0
299 0
000 3035
1 4.75
[301 rows x 2 columns]
3 2.85
4 4.60
...
296 9.50
297 4.00
298 3.35
299 11.50
300 5.30
Name: Selling_Price, dtype: float64
```

```
1 X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.1, random_state=2)
```

```
1 # Loading the linear regression model
2 lin_reg_model = LinearRegression()
```

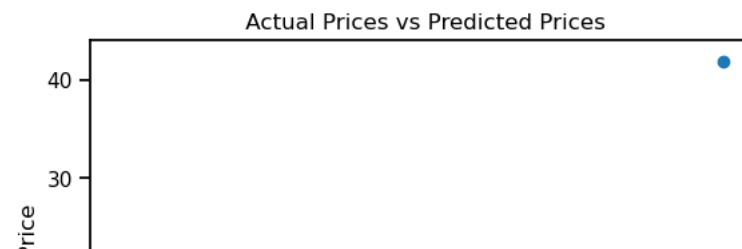
```
1 lin_reg_model.fit(X_train,Y_train)
```

```
LinearRegression
LinearRegressi
on()
```

```
1 # prediction on Training data
2 training_data_prediction = lin_reg_model.predict(X_train)
```

```
1 # R squared Error
```

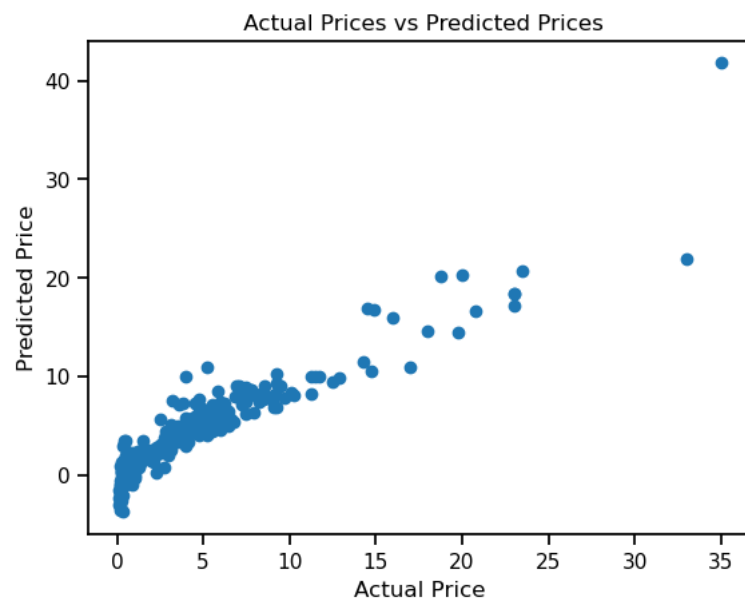
```
2 error_score = metrics.r2_score(Y_train, training_data_prediction)
3 plt.scatter(Y_train, training_data_prediction)
4 print("R squared Error :", error_score)
5 plt.xlabel("Actual Price")
6
7 plt.ylabel("Predicted Price")
8 squared Error : 0.87964838009370217
9 plt.title(" Actual Prices vs Predicted Prices")
10
11 plt.show()
```



```

In [69]: 2 error_score = metrics.r2_score(Y_train, training_data_prediction)
          3 plt.scatter(Y_train, training_data_prediction)
          4 print("R squared Error :", error_score)
          5 plt.xlabel("Actual Price")
          6
          7 plt.ylabel("Predicted Price")
          8 squared Error : 0.8796483009370217
          9 plt.title(" Actual Prices vs Predicted Prices")
         10
         11 plt.show()

```



```

In [71]: 1 #prediction on Training data
          2 test_data_prediction = lin_reg_model.predict(X_test)

```

```

In [73]: 1 plt.scatter(Y_test, test_data_prediction)
In [72]: 2 # R squared Error
          3 plt.xlabel("Actual Price")
          4 error_score = metrics.r2_score(Y_test, test_data_prediction)
          5 plt.ylabel("Predicted Price")
          6 print("R squared Error :", error_score)
          7 plt.title(" Actual Prices vs Predicted Prices")
          8
          9 plt.show()
         10 squared Error : 0.8365861023209662

```



```
In [73]:  
In [72]:  
1 plt.scatter(Y_test, test_data_prediction)  
2 # R Squared Error  
3 plt.xlabel("Actual Price")  
4 error_score = metrics.r2_score(Y_test, test_data_prediction)  
5 plt.ylabel("Predicted Price")  
6 print("R Squared Error : ", error_score)  
7 plt.title(" Actual Prices vs Predicted Prices")  
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  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212  
2213  
2214  
2215  
2216  
22
```