

In [1]:

```
import pandas as pd
```

In [2]:

```
df = pd.read_csv('internet.csv')
```

In [3]:

```
df.head()
```

Out[3]:

	Unnamed: 0	Entity	Code	Year	Cellular Subscription	Internet Users(%)	No. of Internet Users	Broadband Subscription
0	0	Afghanistan	AFG	1980	0.0	0.0	0	0.0
1	1	Afghanistan	AFG	1981	0.0	0.0	0	0.0
2	2	Afghanistan	AFG	1982	0.0	0.0	0	0.0
3	3	Afghanistan	AFG	1983	0.0	0.0	0	0.0
4	4	Afghanistan	AFG	1984	0.0	0.0	0	0.0

In [4]:

```
df.tail()
```

Out[4]:

	Unnamed: 0	Entity	Code	Year	Cellular Subscription	Internet Users(%)	No. of Internet Users	Broadband Subscription
8862	8862	Zimbabwe	ZWE	2016	91.793457	23.119989	3341464	1.217633
8863	8863	Zimbabwe	ZWE	2017	98.985077	24.400000	3599269	1.315694
8864	8864	Zimbabwe	ZWE	2018	89.404869	25.000000	3763048	1.406322
8865	8865	Zimbabwe	ZWE	2019	90.102287	25.100000	3854006	1.395818
8866	8866	Zimbabwe	ZWE	2020	88.755806	29.299999	4591211	1.368916

In [5]:

```
df.shape
```

Out[5]:

(8867, 8)

In [6]:

df.columns

Out[6]:

```
Index(['Unnamed: 0', 'Entity', 'Code', 'Year', 'Cellular Subscription',
      'Internet Users(%)', 'No. of Internet Users', 'Broadband Subscripti
on'],
      dtype='object')
```

In [7]:

df.duplicated().sum()

Out[7]:

0

In [8]:

df.isnull().sum()

Out[8]:

```
Unnamed: 0      0
Entity          0
Code            0
Year            0
Cellular Subscription  0
Internet Users(%)  0
No. of Internet Users  0
Broadband Subscription  0
dtype: int64
```

In [9]:

df = df.drop('Unnamed: 0', axis = 1)

In [10]:

df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8867 entries, 0 to 8866
Data columns (total 7 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Entity                                8867 non-null   object
1   Code                                  8867 non-null   object
2   Year                                  8867 non-null   int64
3   Cellular Subscription                 8867 non-null   float64
4   Internet Users(%)                    8867 non-null   float64
5   No. of Internet Users                 8867 non-null   int64
6   Broadband Subscription                8867 non-null   float64
dtypes: float64(3), int64(2), object(2)
memory usage: 485.0+ KB
```

In [11]:

```
df.describe()
```

Out[11]:

	Year	Cellular Subscription	Internet Users(%)	No. of Internet Users	Broadband Subscription
count	8867.000000	8867.000000	8867.000000	8.867000e+03	8867.000000
mean	2000.151799	39.989614	17.043606	1.089138e+07	4.440695
std	11.812151	51.981410	26.883498	1.248841e+08	9.755705
min	1980.000000	0.000000	0.000000	0.000000e+00	0.000000
25%	1990.000000	0.000000	0.000000	0.000000e+00	0.000000
50%	2000.000000	5.501357	0.855662	1.004700e+04	0.000000
75%	2010.000000	82.231594	25.449939	8.664195e+05	2.007603
max	2020.000000	436.103027	100.000000	4.699886e+09	78.524361

In [12]:

```
df.nunique()
```

Out[12]:

```
Entity          229
Code            216
Year            41
Cellular Subscription    6344
Internet Users(%)      4702
No. of Internet Users   5058
Broadband Subscription  3858
dtype: int64
```

In [13]:

```
obj_cols = df.select_dtypes(include=['object']).columns
```

In [14]:

```
num_cols = df.select_dtypes(include=['int64', 'float64']).columns
```

In [15]:

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

In [16]:

```
import numpy as np
```

In [17]:

```
import warnings  
warnings.filterwarnings('ignore')
```

In [18]:

```
df['Entity'].unique()
```

Out[18]:

```
array(['Afghanistan', 'Albania', 'Algeria', 'American Samoa', 'Andorra',
      'Angola', 'Antigua and Barbuda', 'Argentina', 'Armenia', 'Aruba',
      'Australia', 'Austria', 'Azerbaijan', 'Bahamas', 'Bahrain',
      'Bangladesh', 'Barbados', 'Belarus', 'Belgium', 'Belize', 'Benin',
      'Bermuda', 'Bhutan', 'Bolivia', 'Bosnia and Herzegovina',
      'Botswana', 'Brazil', 'British Virgin Islands', 'Brunei',
      'Bulgaria', 'Burkina Faso', 'Burundi', 'Cambodia', 'Cameroon',
      'Canada', 'Cape Verde', 'Cayman Islands',
      'Central African Republic', 'Chad', 'Chile', 'China', 'Colombia',
      'Comoros', 'Congo', 'Costa Rica', 'Cote d'Ivoire', 'Croatia',
      'Cuba', 'Curacao', 'Cyprus', 'Czechia',
      'Democratic Republic of Congo', 'Denmark', 'Djibouti', 'Dominica',
      'Dominican Republic', 'East Asia and Pacific', 'Ecuador', 'Egypt',
      'El Salvador', 'Equatorial Guinea', 'Eritrea', 'Estonia',
      'Eswatini', 'Ethiopia', 'Europe and Central Asia',
      'European Union', 'Faeroe Islands', 'Fiji', 'Finland', 'France',
      'French Polynesia', 'Gabon', 'Gambia', 'Georgia', 'Germany',
      'Ghana', 'Gibraltar', 'Greece', 'Greenland', 'Grenada', 'Guam',
      'Guatemala', 'Guinea', 'Guinea-Bissau', 'Guyana', 'Haiti',
      'High income', 'Honduras', 'Hong Kong', 'Hungary', 'Iceland',
      'India', 'Indonesia', 'Iran', 'Iraq', 'Ireland', 'Israel', 'Italy',
      'Jamaica', 'Japan', 'Jordan', 'Kazakhstan', 'Kenya', 'Kiribati',
      'Kosovo', 'Kuwait', 'Kyrgyzstan', 'Laos',
      'Latin America and Caribbean', 'Latvia', 'Lebanon', 'Lesotho',
      'Liberia', 'Libya', 'Liechtenstein', 'Lithuania',
      'Low and middle income', 'Low income', 'Lower middle income',
      'Luxembourg', 'Macao', 'Madagascar', 'Malawi', 'Malaysia',
      'Maldives', 'Mali', 'Malta', 'Marshall Islands', 'Mauritania',
      'Mauritius', 'Mexico', 'Micronesia (country)',
      'Middle East and North Africa', 'Middle income', 'Moldova',
      'Monaco', 'Mongolia', 'Montenegro', 'Morocco', 'Mozambique',
      'Myanmar', 'Namibia', 'Nauru', 'Nepal', 'Netherlands',
      'New Caledonia', 'New Zealand', 'Nicaragua', 'Niger', 'Nigeria',
      'North America', 'North Korea', 'North Macedonia',
      'Northern Mariana Islands', 'Norway', 'Oman', 'Pakistan', 'Palau',
      'Palestine', 'Panama', 'Papua New Guinea', 'Paraguay', 'Peru',
      'Philippines', 'Poland', 'Portugal', 'Puerto Rico', 'Qatar',
      'Romania', 'Russia', 'Rwanda', 'Saint Kitts and Nevis',
      'Saint Lucia', 'Saint Vincent and the Grenadines', 'Samoa',
      'San Marino', 'Sao Tome and Principe', 'Saudi Arabia', 'Senegal',
      'Serbia', 'Seychelles', 'Sierra Leone', 'Singapore',
      'Sint Maarten (Dutch part)', 'Slovakia', 'Slovenia',
      'Solomon Islands', 'Somalia', 'South Africa', 'South Asia',
      'South Korea', 'South Sudan', 'Spain', 'Sri Lanka',
      'Sub-Saharan Africa', 'Sudan', 'Suriname', 'Sweden', 'Switzerland',
      'Syria', 'Tajikistan', 'Tanzania', 'Thailand', 'Timor', 'Togo',
      'Tonga', 'Trinidad and Tobago', 'Tunisia', 'Turkey',
      'Turkmenistan', 'Turks and Caicos Islands', 'Tuvalu', 'Uganda',
      'Ukraine', 'United Arab Emirates', 'United Kingdom',
      'United States', 'United States Virgin Islands',
      'Upper middle income', 'Uruguay', 'Uzbekistan', 'Vanuatu',
      'Venezuela', 'Vietnam', 'World', 'Yemen', 'Zambia', 'Zimbabwe'],
      dtype=object)
```

In [19]:

```
df['Entity'].value_counts()
```

Out[19]:

```
Afghanistan      41
Mongolia          41
Mali              41
Malta             41
Mauritania        41
..
Palau             16
Curacao          12
South Sudan       11
Kosovo            6
Sint Maarten (Dutch part)  1
Name: Entity, Length: 229, dtype: int64
```

In [20]:

```
df['Code'].unique()
```

Out[20]:

```
array(['AFG', 'ALB', 'DZA', 'ASM', 'AND', 'AGO', 'ATG', 'ARG', 'ARM',
      'ABW', 'AUS', 'AUT', 'AZE', 'BHS', 'BHR', 'BGD', 'BRB', 'BLR',
      'BEL', 'BLZ', 'BEN', 'BMU', 'BTN', 'BOL', 'BIH', 'BWA', 'BRA',
      'VGB', 'BRN', 'BGR', 'BFA', 'BDI', 'KHM', 'CMR', 'CAN', 'CPV',
      'CYM', 'CAF', 'TCD', 'CHL', 'CHN', 'COL', 'COM', 'COG', 'CRI',
      'CIV', 'HRV', 'CUB', 'CUW', 'CYP', 'CZE', 'COD', 'DNK', 'DJI',
      'DMA', 'DOM', 'Region', 'ECU', 'EGY', 'SLV', 'GNQ', 'ERI', 'EST',
      'SWZ', 'ETH', 'FRO', 'FJI', 'FIN', 'FRA', 'PYF', 'GAB', 'GMB',
      'GEO', 'DEU', 'GHA', 'GIB', 'GRC', 'GRL', 'GRD', 'GUM', 'GTM',
      'GIN', 'GNB', 'GUY', 'HTI', 'HND', 'HKG', 'HUN', 'ISL', 'IND',
      'IDN', 'IRN', 'IRQ', 'IRL', 'ISR', 'ITA', 'JAM', 'JPN', 'JOR',
      'KAZ', 'KEN', 'KIR', 'OWID_KOS', 'KWT', 'KGZ', 'LAO', 'LVA', 'LBN',
      'LSO', 'LBR', 'LBY', 'LIE', 'LTU', 'LUX', 'MAC', 'MDG', 'MWI',
      'MYS', 'MDV', 'MLI', 'MLT', 'MHL', 'MRT', 'MUS', 'MEX', 'FSM',
      'MDA', 'MCO', 'MNG', 'MNE', 'MAR', 'MOZ', 'MMR', 'NAM', 'NRU',
      'NPL', 'NLD', 'NCL', 'NZL', 'NIC', 'NER', 'NGA', 'PRK', 'MKD',
      'MNP', 'NOR', 'OMN', 'PAK', 'PLW', 'PSE', 'PAN', 'PNG', 'PRY',
      'PER', 'PHL', 'POL', 'PRT', 'PRI', 'QAT', 'ROU', 'RUS', 'RWA',
      'KNA', 'LCA', 'VCT', 'WSM', 'SMR', 'STP', 'SAU', 'SEN', 'SRB',
      'SYC', 'SLE', 'SGP', 'SXM', 'SVK', 'SVN', 'SLB', 'SOM', 'ZAF',
      'KOR', 'SSD', 'ESP', 'LKA', 'SDN', 'SUR', 'SWE', 'CHE', 'SYR',
      'TJK', 'TZA', 'THA', 'TLS', 'TGO', 'TON', 'TTO', 'TUN', 'TUR',
      'TKM', 'TCA', 'TUV', 'UGA', 'UKR', 'ARE', 'GBR', 'USA', 'VIR',
      'URY', 'UZB', 'VUT', 'VEN', 'VNM', 'OWID_WRL', 'YEM', 'ZMB', 'ZW
      E'],
      dtype=object)
```

In [21]:

```
df['Entity'].value_counts()
```

Out[21]:

```
Afghanistan      41
Mongolia          41
Mali              41
Malta             41
Mauritania        41
..
Palau             16
Curacao          12
South Sudan       11
Kosovo            6
Sint Maarten (Dutch part)  1
Name: Entity, Length: 229, dtype: int64
```

In [22]:

```
df['Year'].unique()
```

Out[22]:

```
array([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], dtype=int64)
```

In [23]:

```
df['Year'].value_counts()
```

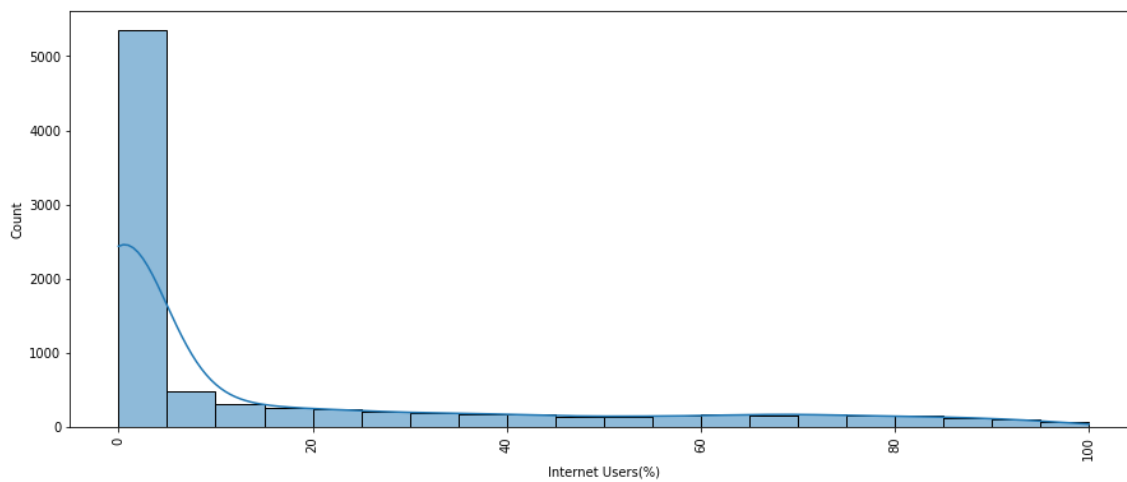
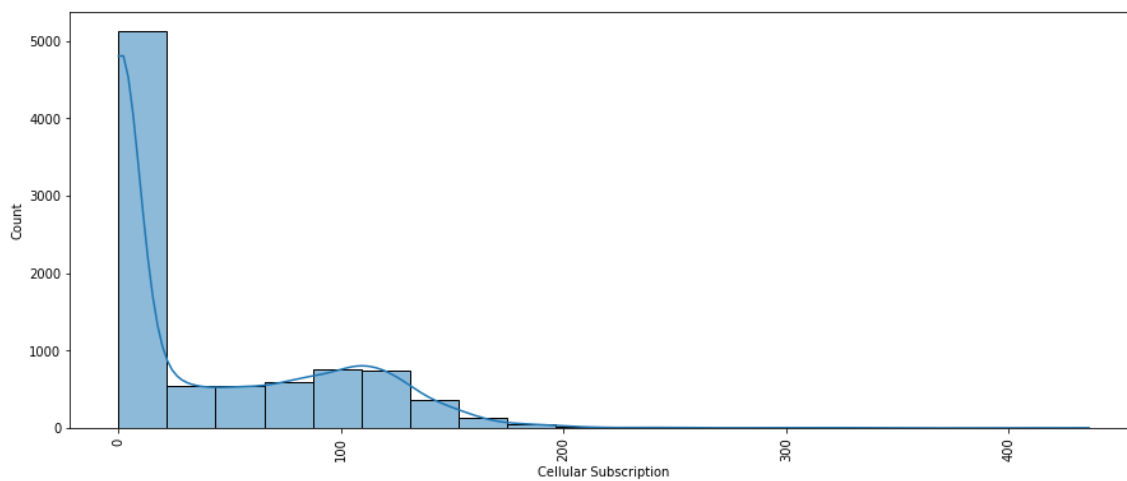
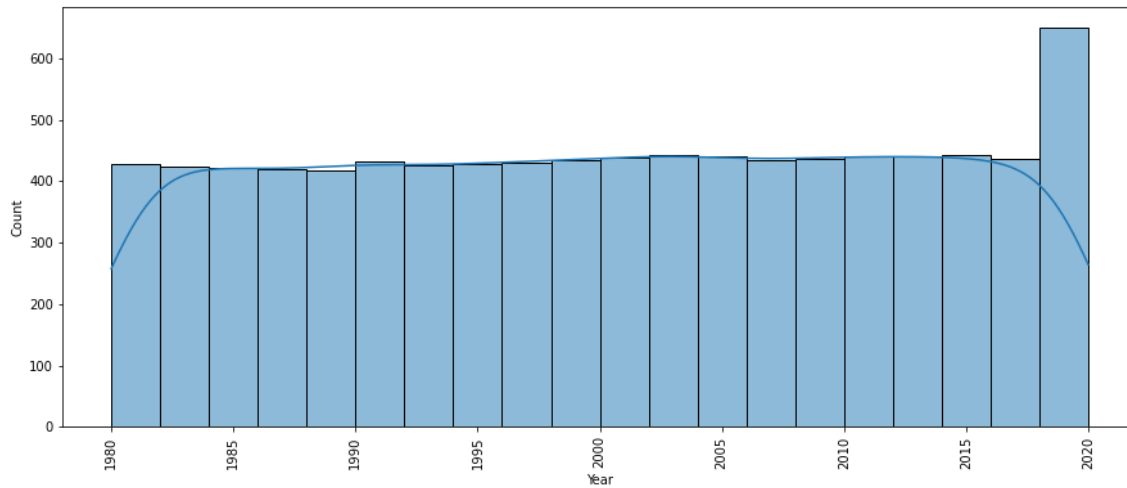
Out[23]:

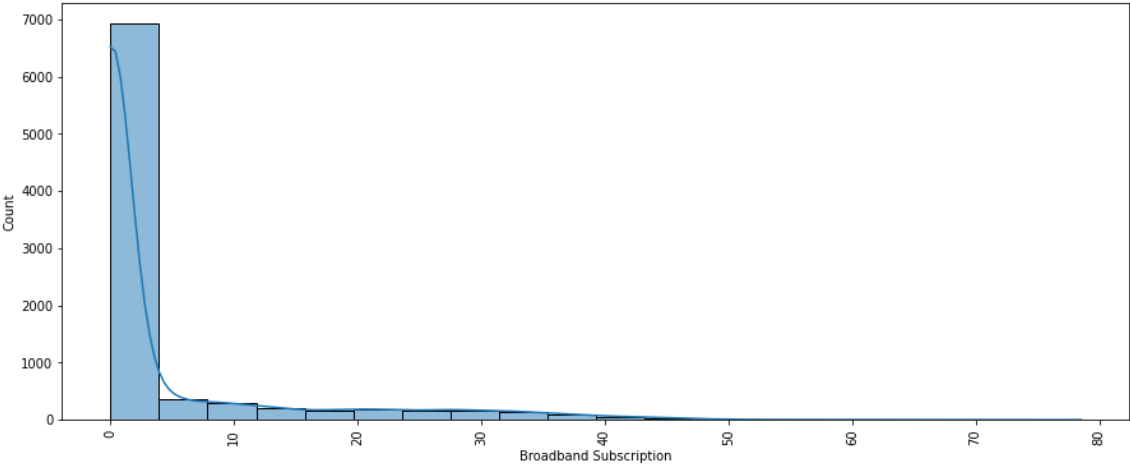
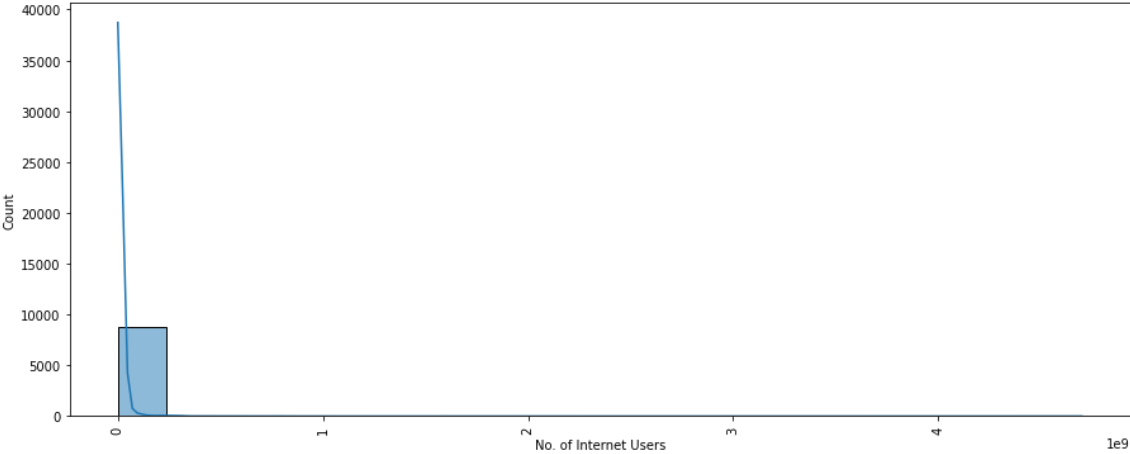
2020	223
2019	223
2015	222
2012	222
2010	222
1990	222
2004	222
2003	221
2002	221
2014	220
2001	219
2017	219
2013	219
2009	219
2007	219
2005	219
2000	219
1999	218
2011	218
2016	217
1998	217
2008	217
2006	216
1996	216
1997	215
1980	215
1995	215
1992	213
1993	213
1994	213
1981	213
1982	213
1983	212
1991	211
1986	211
1984	211
1988	210
1985	210
1987	209
1989	208
2018	205

Name: Year, dtype: int64

In [24]:

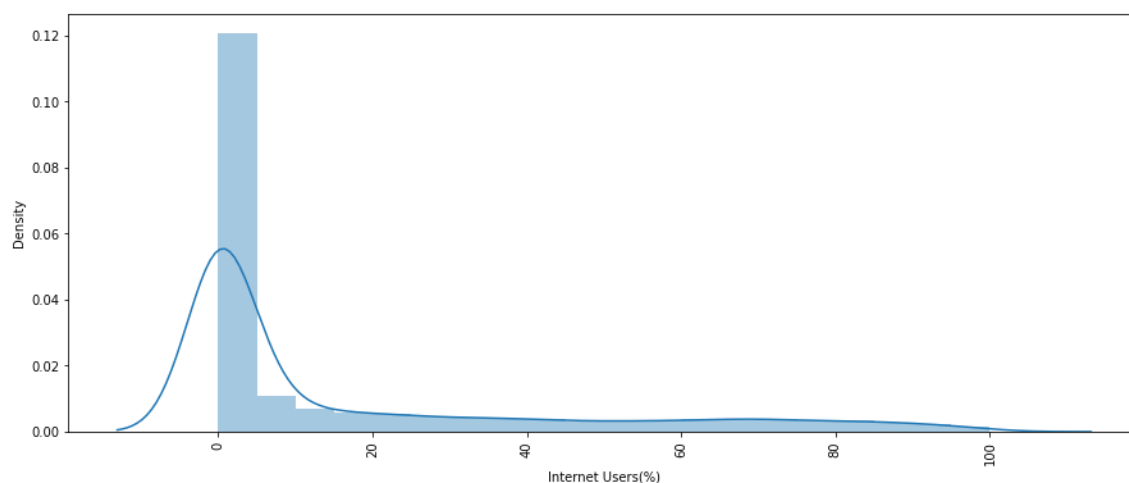
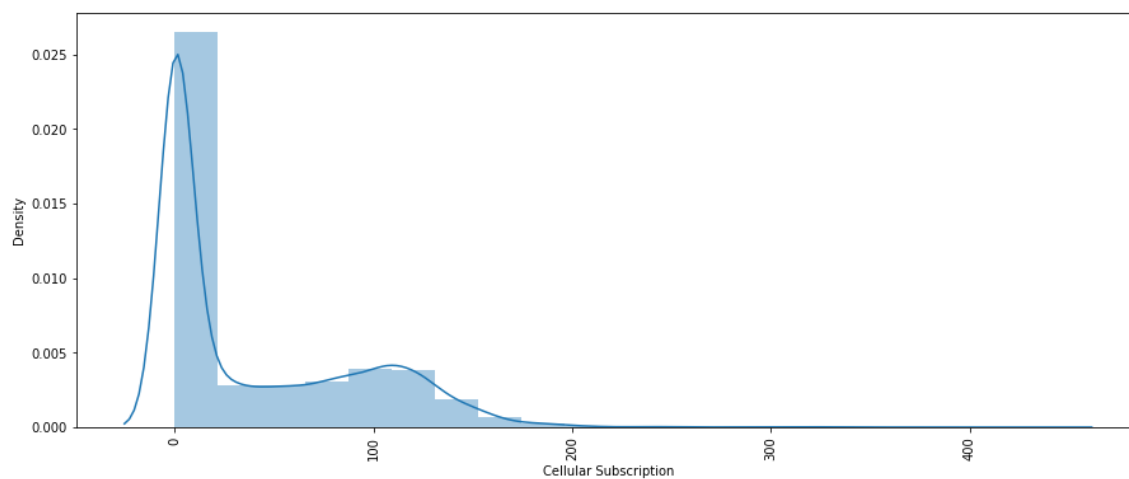
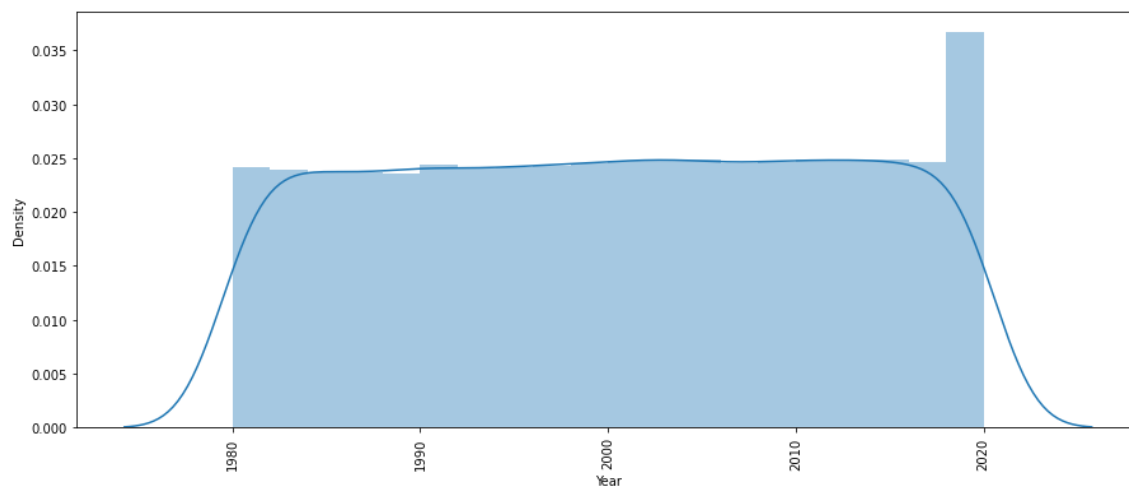
```
for i in num_cols:  
    plt.figure(figsize=(15,6))  
    sns.histplot(df[i], kde = True, bins = 20, palette = 'hls')  
    plt.xticks(rotation = 90)  
    plt.show()
```

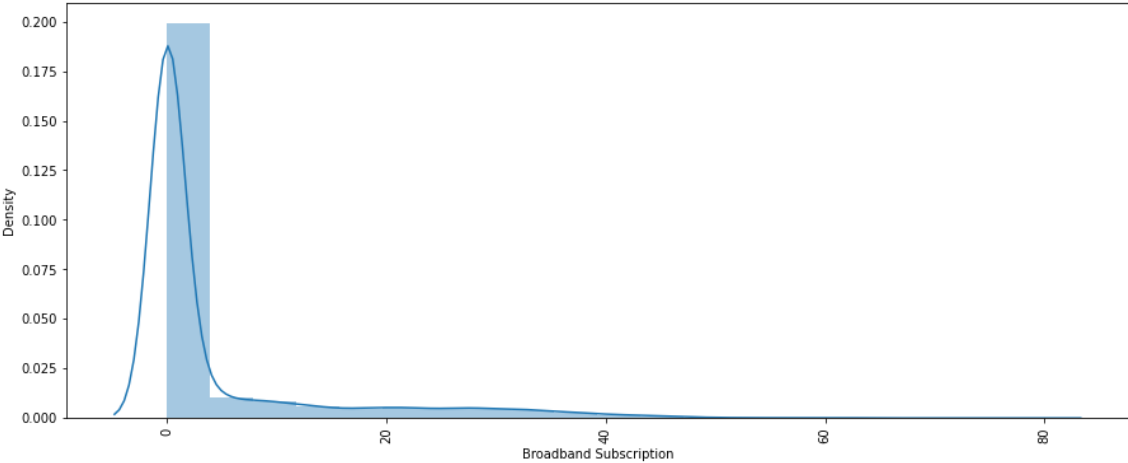
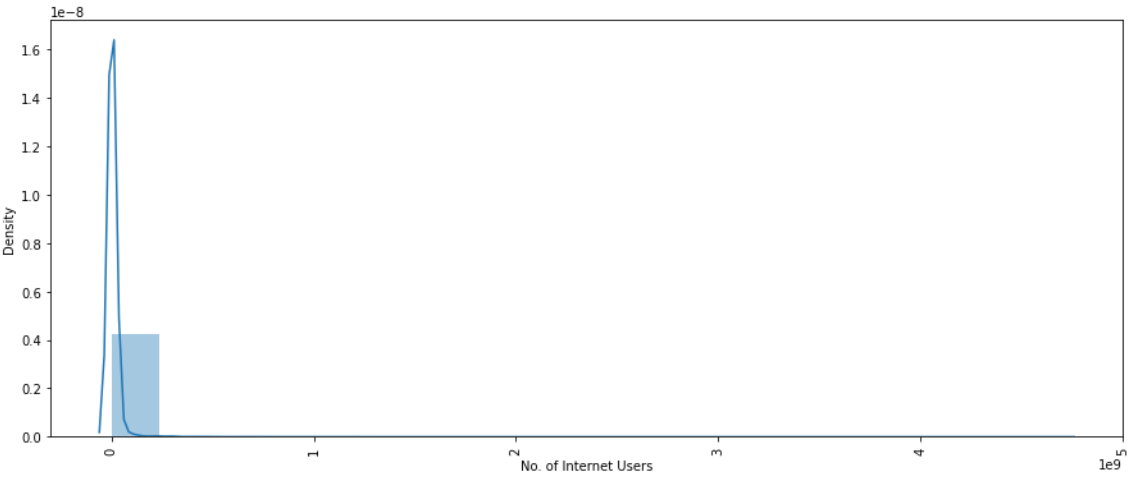




In [25]:

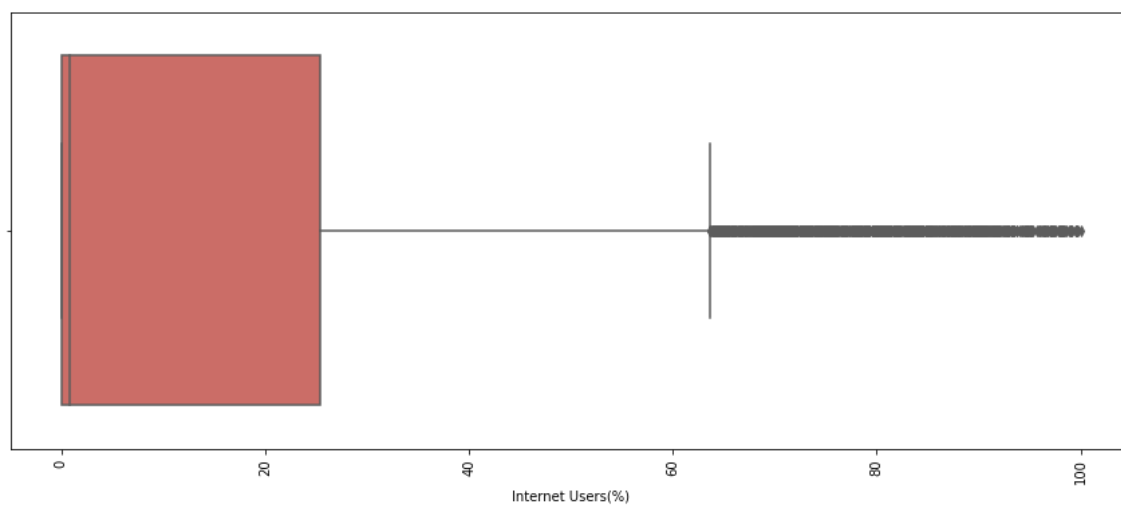
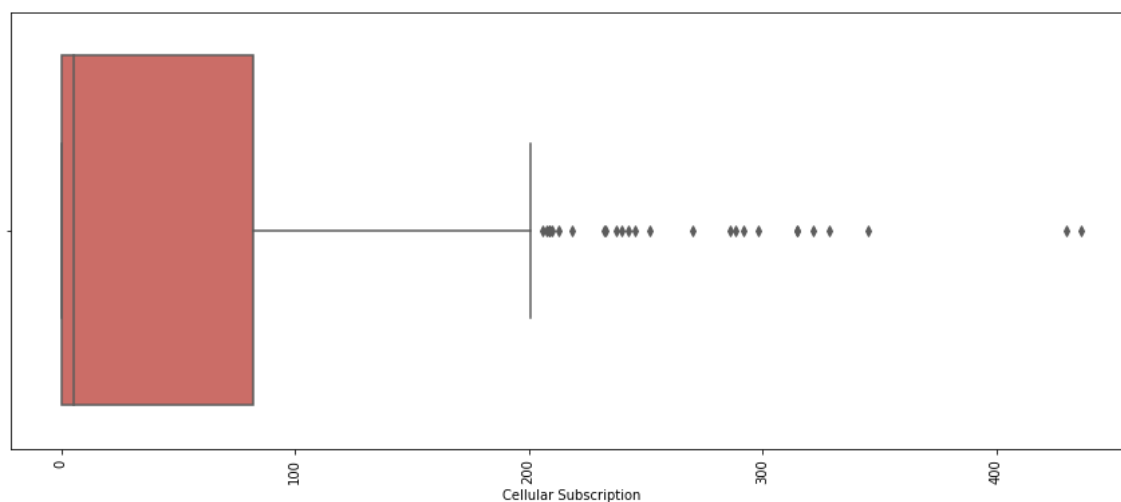
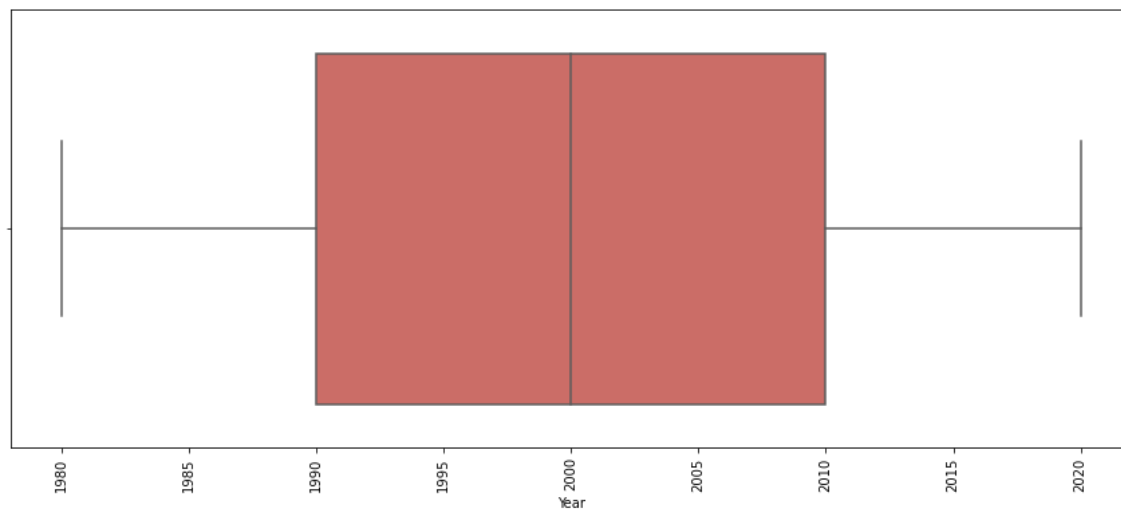
```
for i in num_cols:  
    plt.figure(figsize=(15,6))  
    sns.distplot(df[i], kde = True, bins = 20)  
    plt.xticks(rotation = 90)  
    plt.show()
```

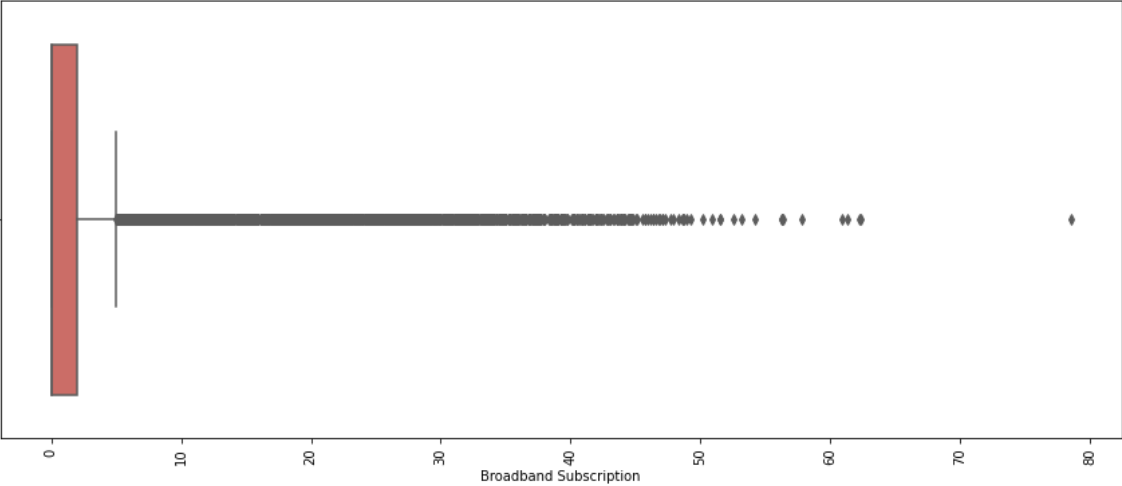
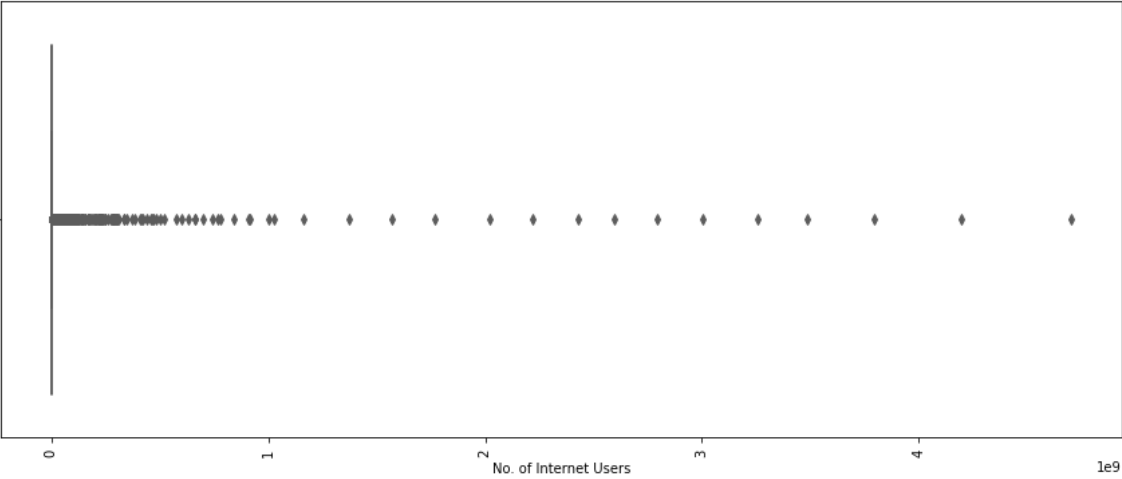




In [26]:

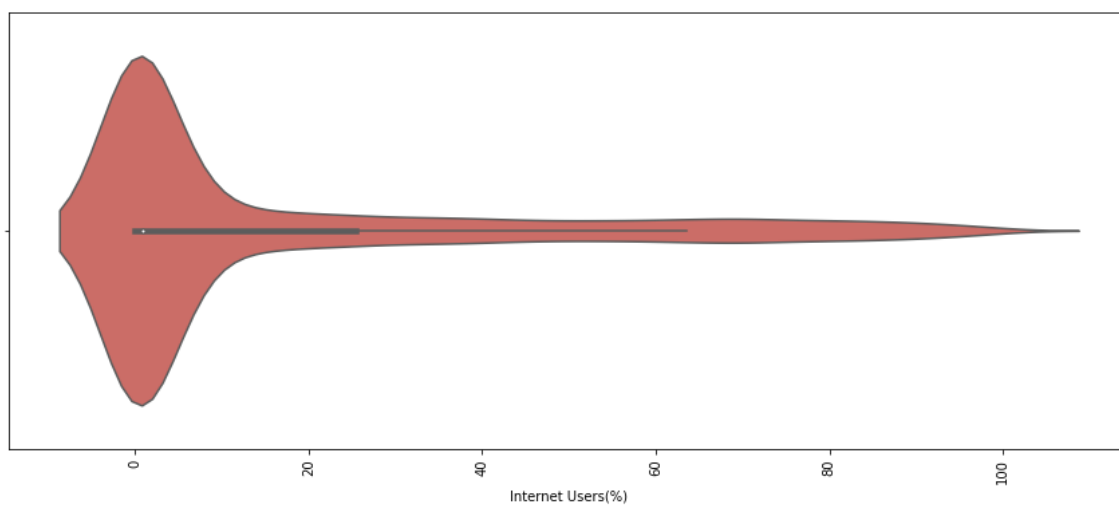
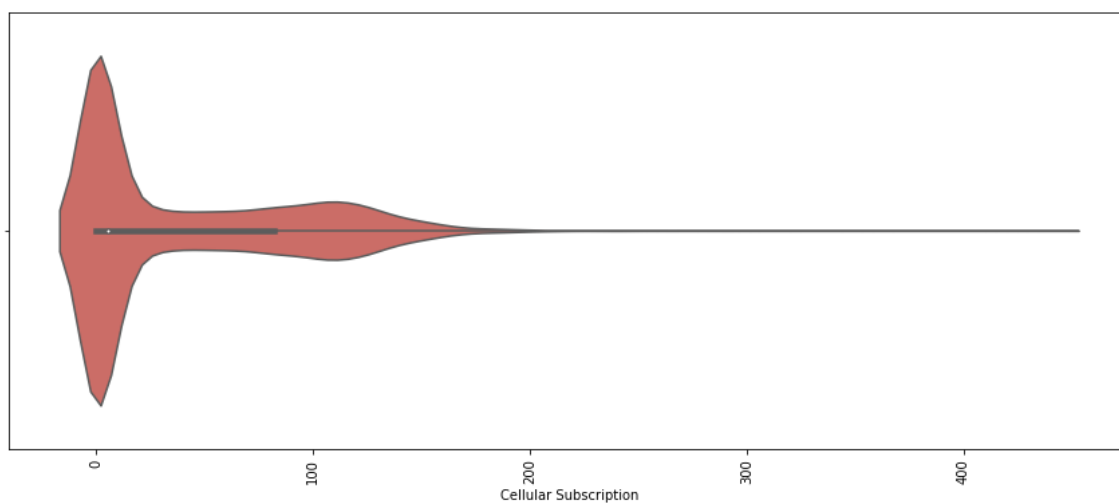
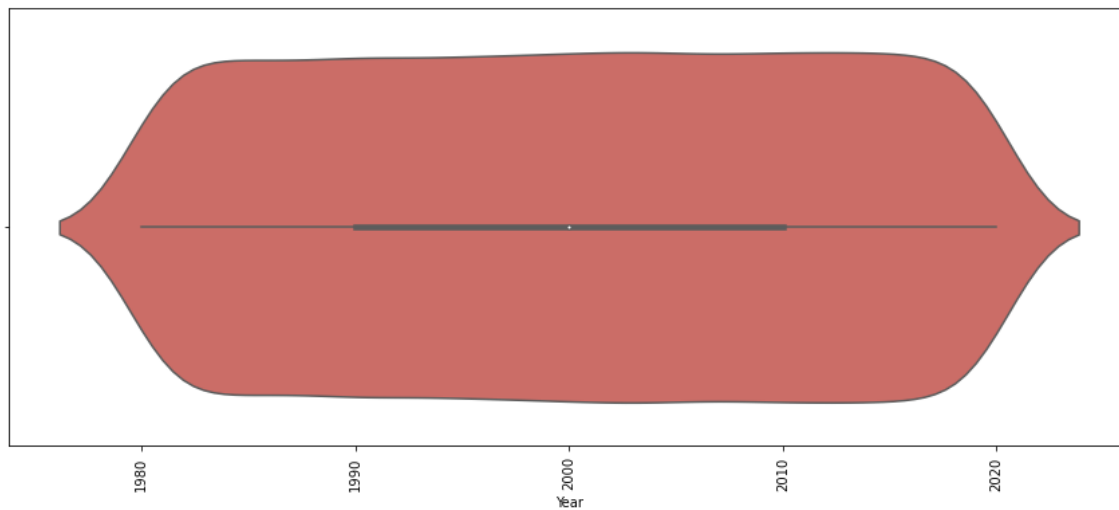
```
for i in num_cols:  
    plt.figure(figsize=(15,6))  
    sns.boxplot(df[i], data = df, palette = 'hls')  
    plt.xticks(rotation = 90)  
    plt.show()
```

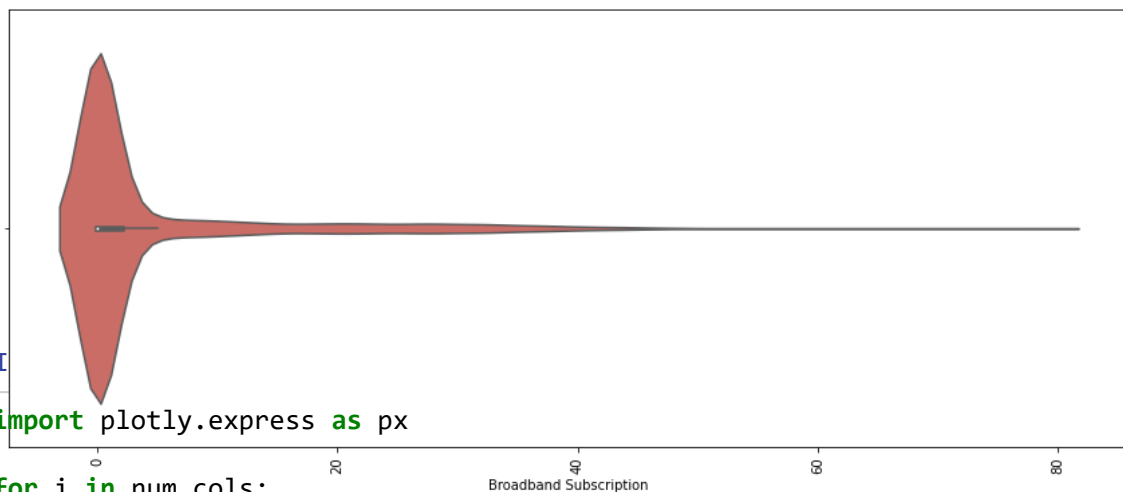
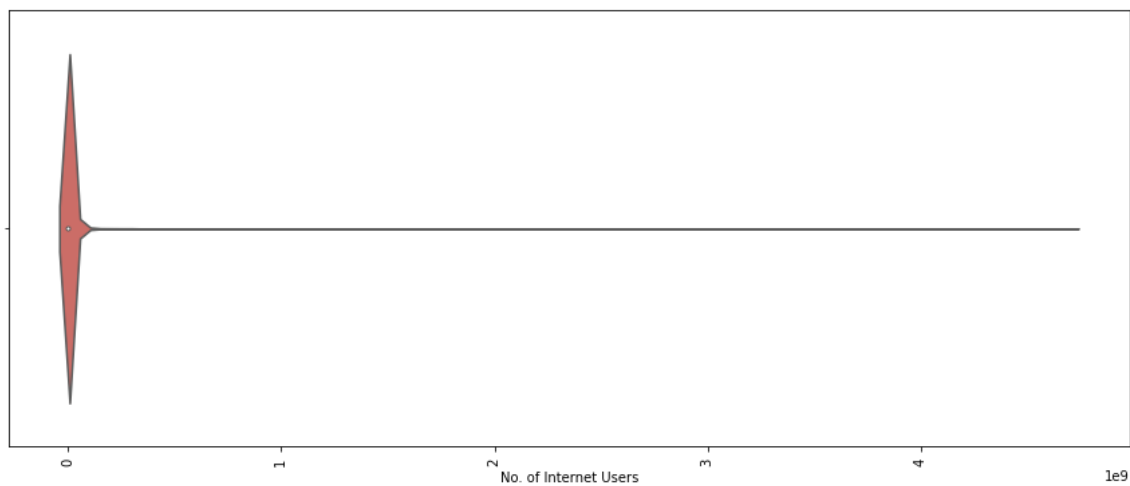




In [27]:

```
for i in num_cols:  
    plt.figure(figsize=(15,6))  
    sns.violinplot(df[i], data = df, palette = 'hls')  
    plt.xticks(rotation = 90)  
    plt.show()
```





```
import plotly.express as px
```

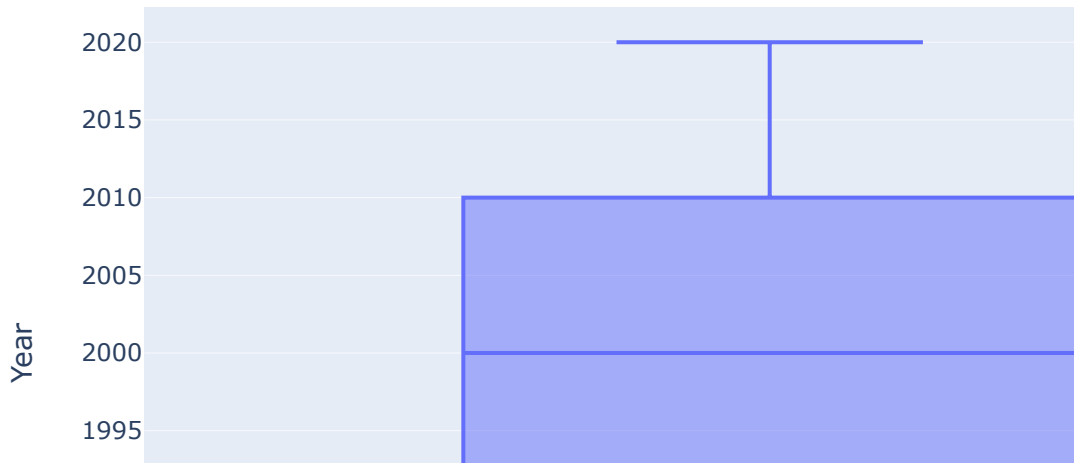
```
for i in num_cols:  
    fig = px.histogram(df, x=i, nbins=20)  
    fig.show()
```



In [29]:

```
for col in num_cols:  
    fig = px.box(df, y=col, title=f"{col} Boxplot")  
    fig.show()
```

Year Boxplot



In [30]:

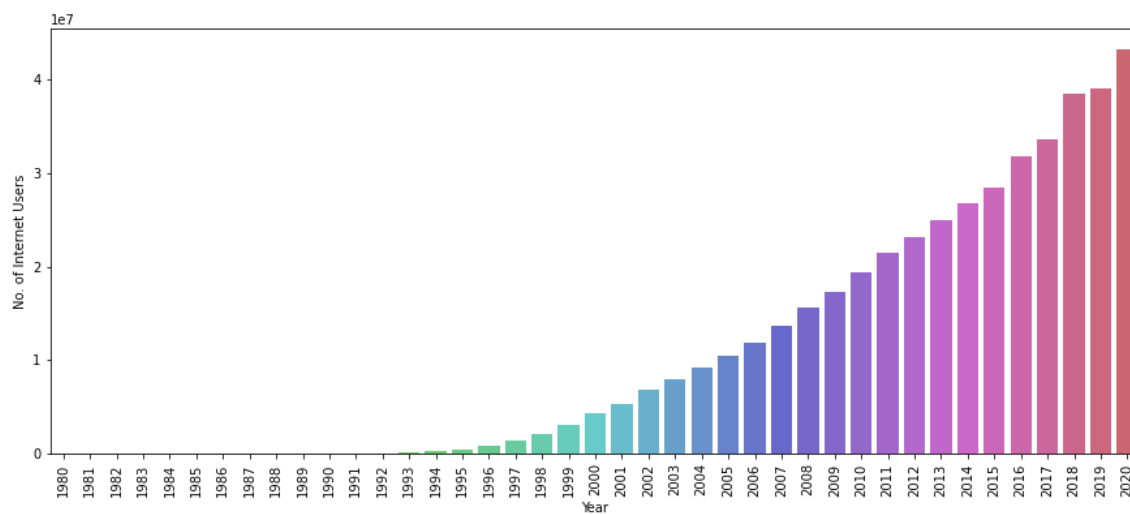
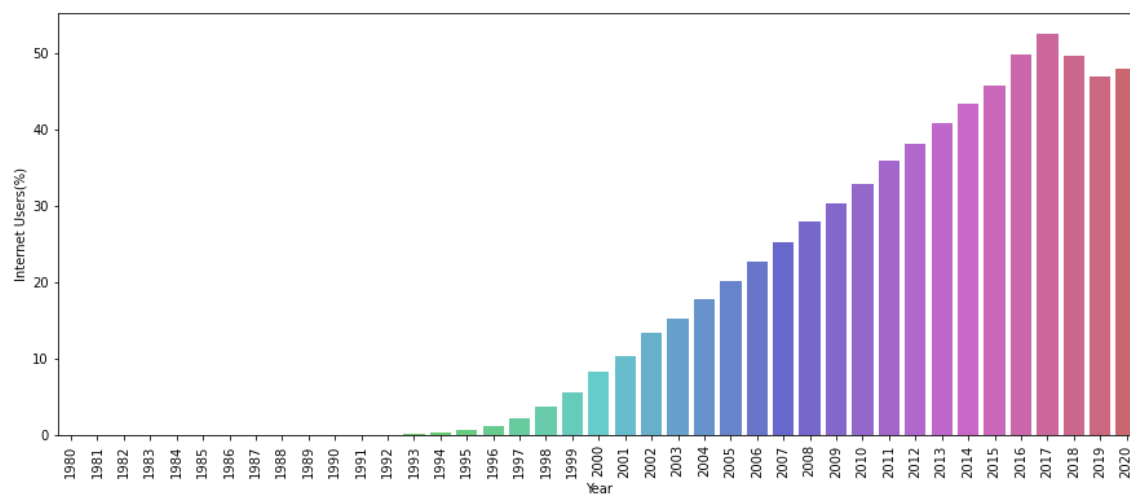
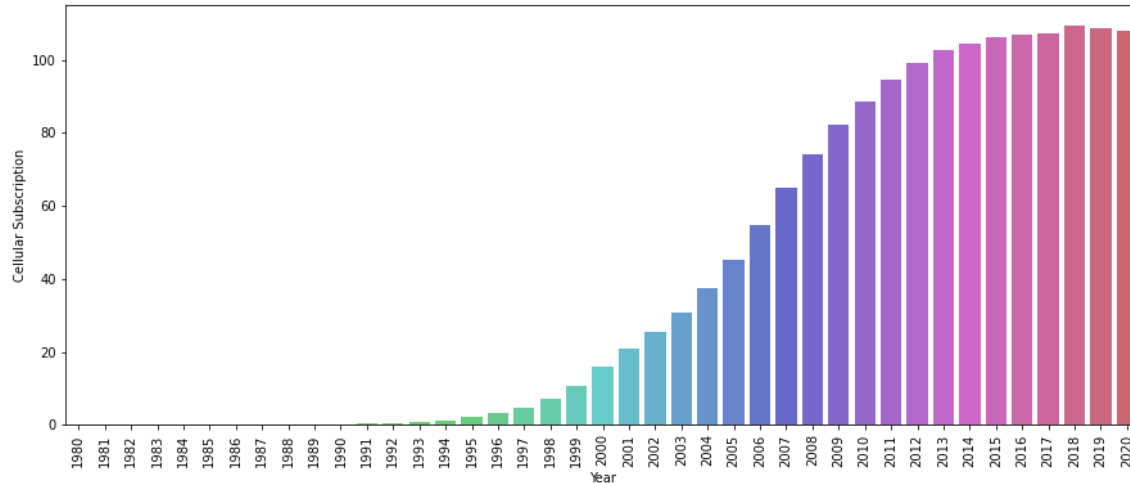
```
for col in num_cols:  
    fig = px.violin(df, y=col, title=f"{col} Boxplot")  
    fig.show()
```

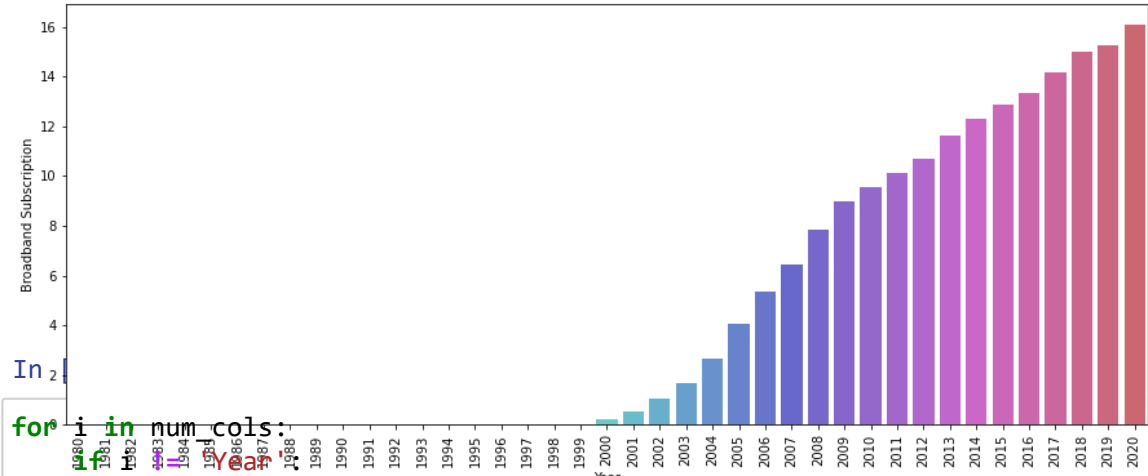
Year Boxplot



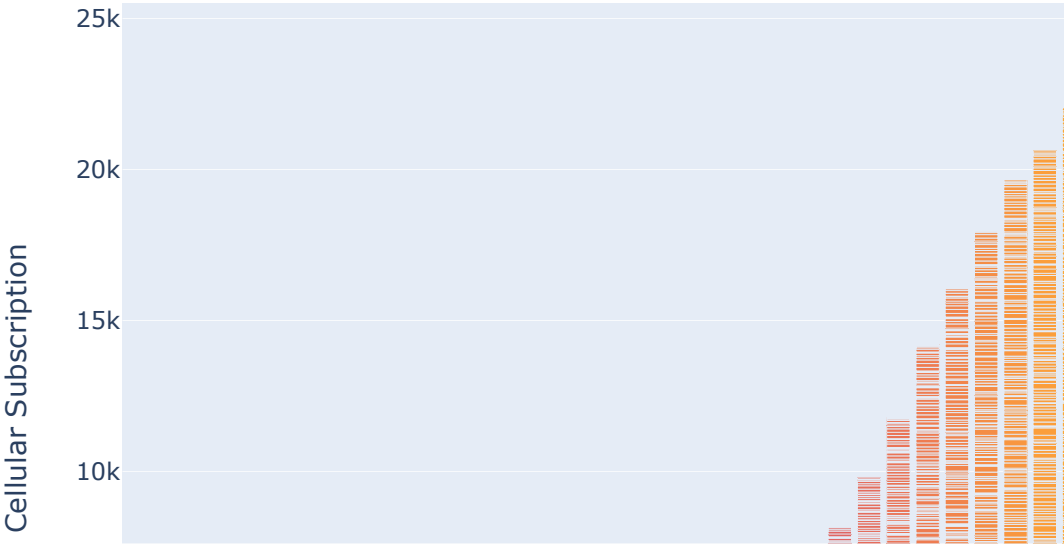
In [31]:

```
for i in num_cols:  
    if i != 'Year':  
        plt.figure(figsize=(15,6))  
        sns.barplot(x = df['Year'], y = df[i], data = df, ci = None, palette = 'hls')  
        plt.xticks(rotation = 90)  
        plt.show()
```





```
for i in num_cols:
    if i == 'Year':
        fig = px.bar(df, x='Year', y=i, color='Year')
    fig.show()
```

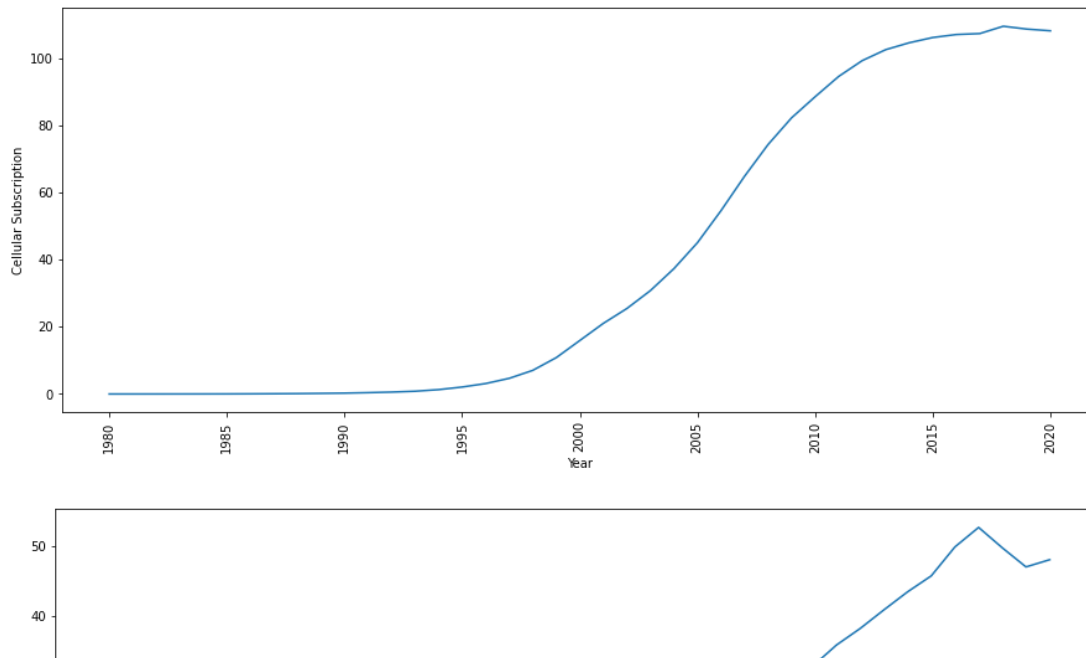


In [33]:

```

for i in num_cols:
    for j in num_cols:
        if i != j:
            plt.figure(figsize=(15,6))
            sns.lineplot(x = df[i], y = df[j], data = df, ci = None, palette = 'hls')
            plt.xticks(rotation = 90)
            plt.show()

```



In [34]:

```
temp=df[~(df['Code']=='Region')]
```

In [35]:

```
temp1= temp.groupby(['Entity','Year']).sum().reset_index()
```

In [36]:

```
df1=temp1[(temp1['Entity']=='India')|(temp1['Entity']=='China')|(temp1['Entity']=='Unite
```

In [37]:

```
df1
```

Out[37]:

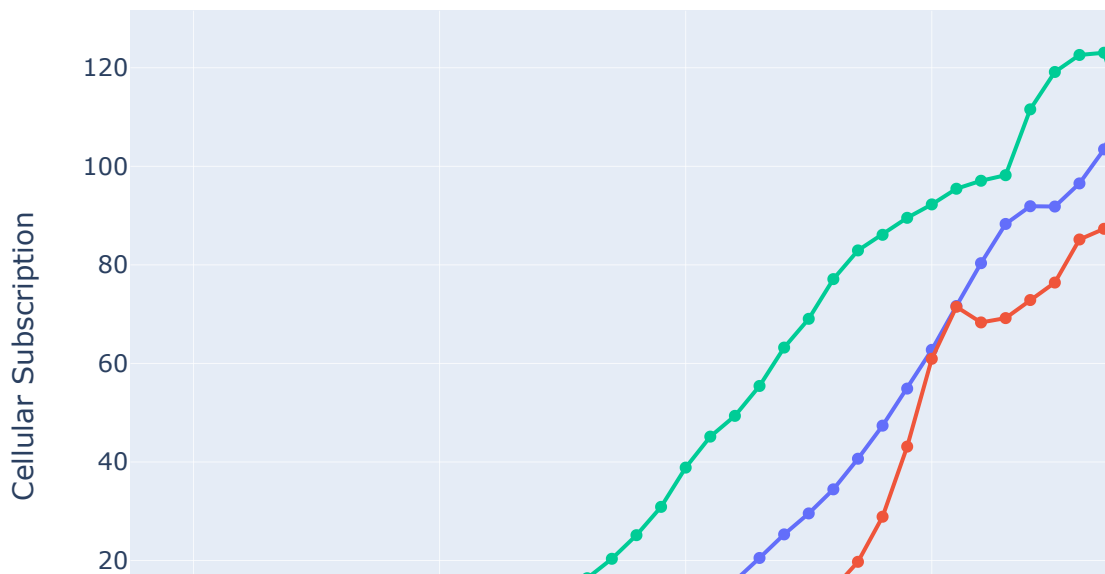
	Entity	Year	Cellular Subscription	Internet Users(%)	No. of Internet Users	Broadband Subscription
1579	China	1980	0.000000	0.000000	0	0.000000
1580	China	1981	0.000000	0.000000	0	0.000000
1581	China	1982	0.000000	0.000000	0	0.000000
1582	China	1983	0.000000	0.000000	0	0.000000
1583	China	1984	0.000000	0.000000	0	0.000000
...
7900	United States	2016	122.594551	85.544418	279910069	32.727173
7901	United States	2017	123.044838	87.274887	287824925	33.283627
7902	United States	2018	106.464684	88.498901	293940279	33.860367
7903	United States	2019	108.092651	89.430283	298983035	34.725369
7904	United States	2020	106.185554	90.900002	305371298	36.608768

120 rows × 6 columns

In [38]:

```
fig = px.line(df1, x='Year', y='Cellular Subscription', color='Entity', markers=True,  
              title='Mobile phone subscriptions per 100 people 1980 to 2020')  
fig.update_layout(hovermode='x unified')  
fig.show()
```

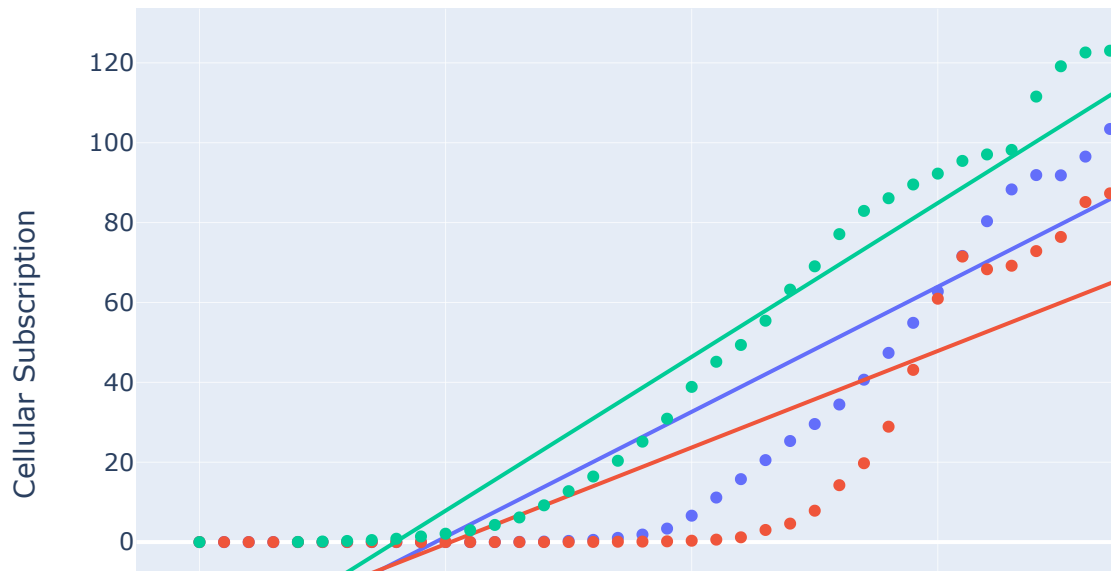
Mobile phone subscriptions per 100 people 1980 to 2020



In [39]:

```
fig = px.scatter(df1, x="Year", y="Cellular Subscription", color="Entity", trendline="ol")  
fig.show()
```

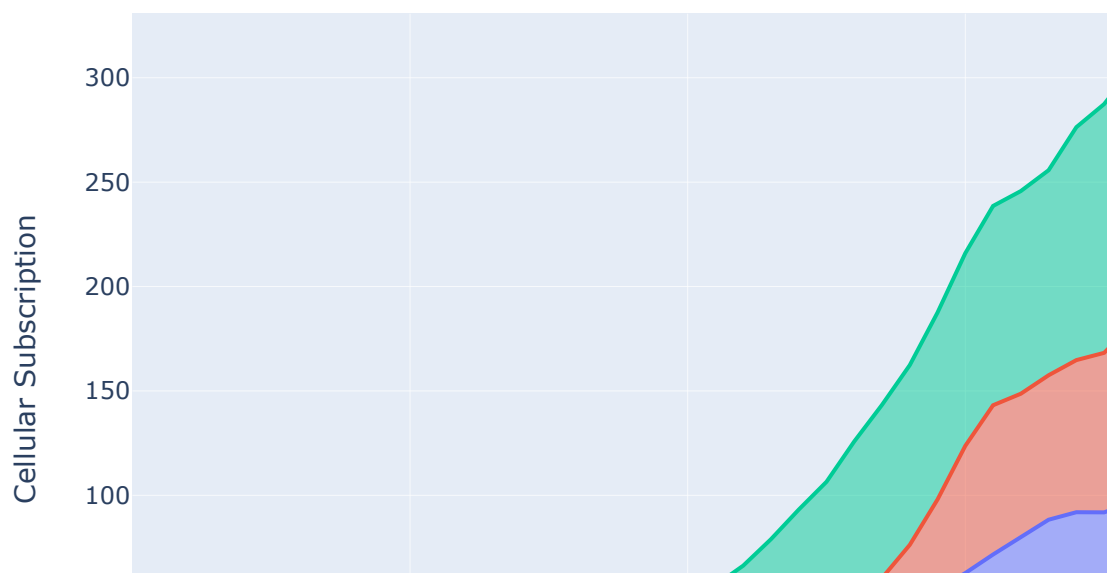
Mobile phone subscriptions per 100 people 1980 to 2020



In [40]:

```
fig = px.area(df1, x="Year", y="Cellular Subscription", color="Entity", title="Mobile ph  
fig.show()
```

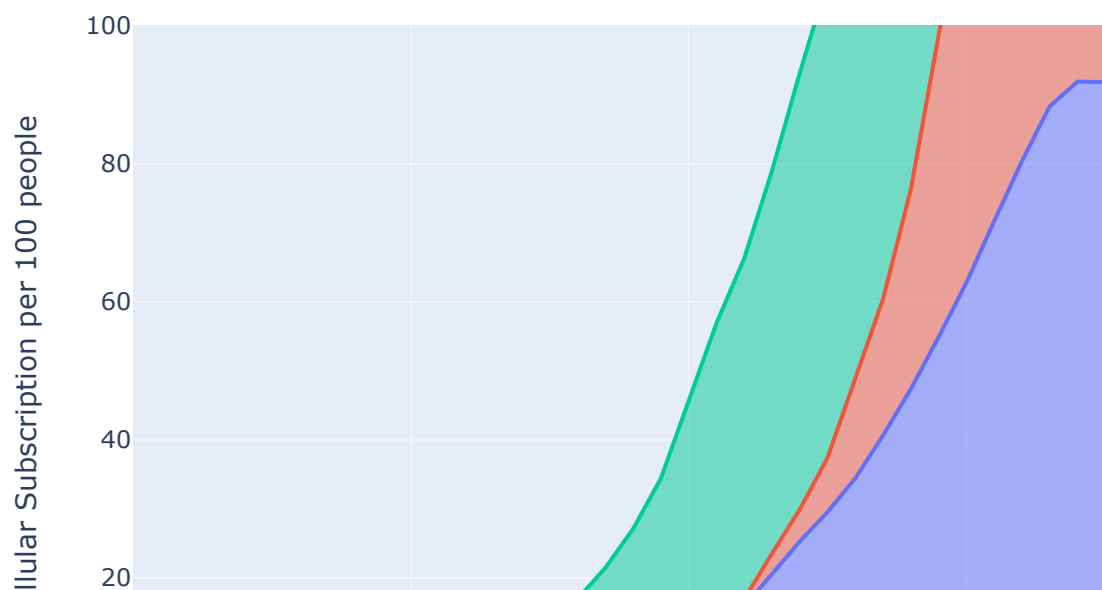
Mobile phone subscriptions per 100 people 1980 to 2020



In [41]:

```
fig = px.area(df1, x="Year", y="Cellular Subscription", color="Entity", title="Mobile ph  
            labels={"Year": "Year", "Cellular Subscription": "Cellular Subscription per 1  
fig.update_layout(yaxis_range=[0,100])  
fig.show()
```

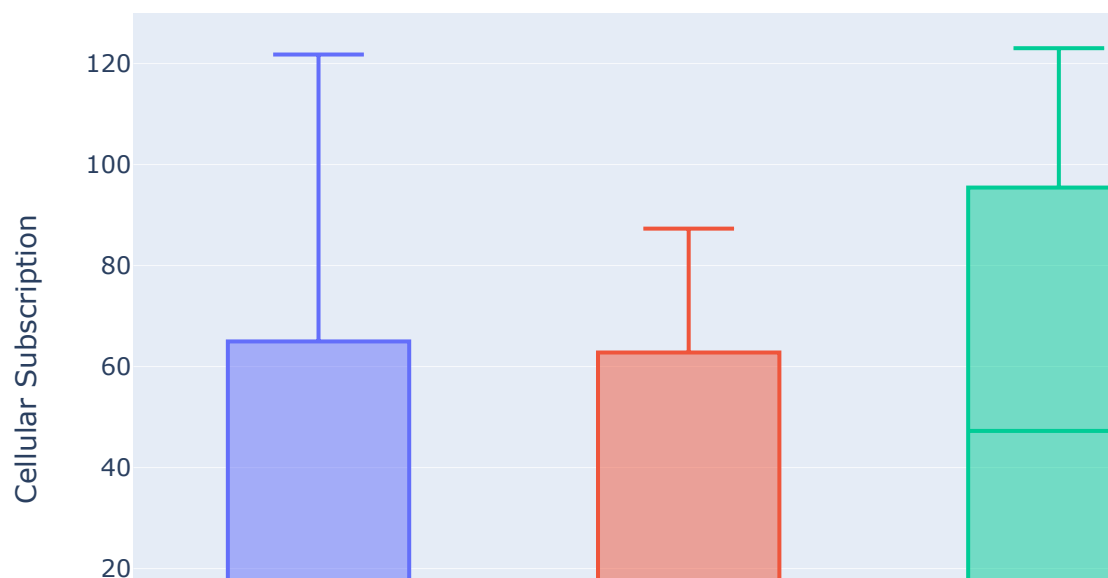
Mobile phone subscriptions per 100 people 1980 to 2020



In [42]:

```
fig = px.box(df1, x="Entity", y="Cellular Subscription", color="Entity", title="Mobile p  
fig.show()
```

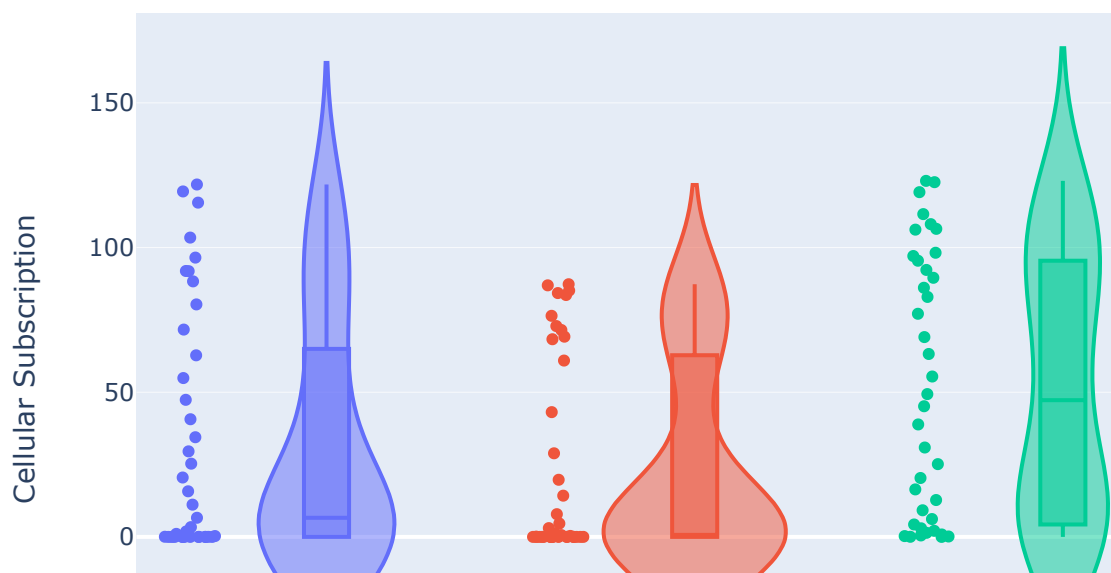
Mobile phone subscriptions per 100 people 1980 to 2020



In [43]:

```
fig = px.violin(df1, x="Entity", y="Cellular Subscription", color="Entity", box=True, po  
fig.show()
```

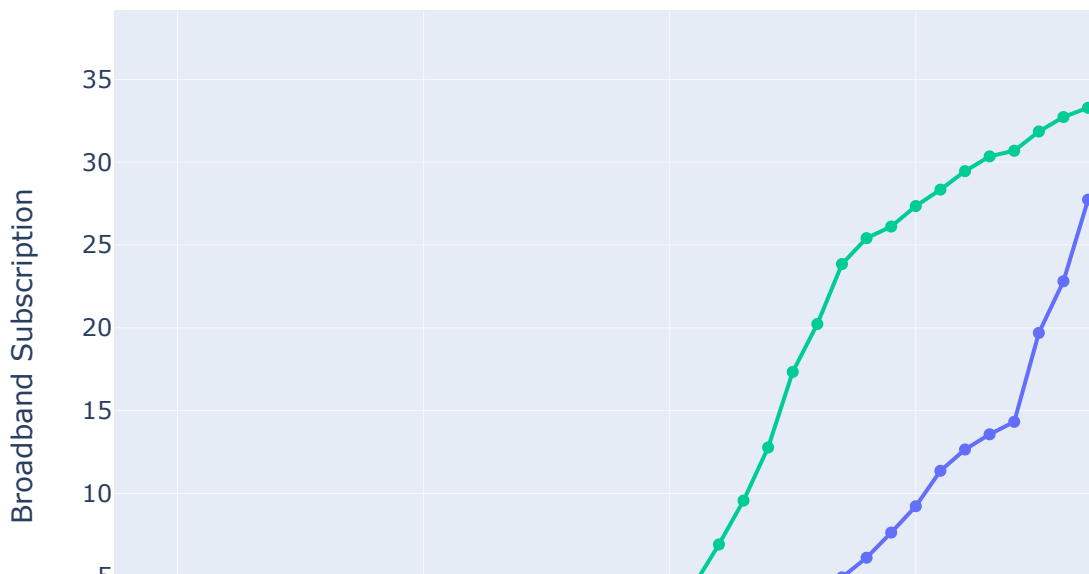
Mobile phone subscriptions per 100 people 1980 to 2020



In [44]:

```
fig = px.line(df1, x='Year', y='Broadband Subscription', color='Entity', markers=True,  
              title='Landline Internet subscriptions per 100 people, 1998 to 2020')  
fig.update_layout(hovermode='x unified')  
fig.show()
```

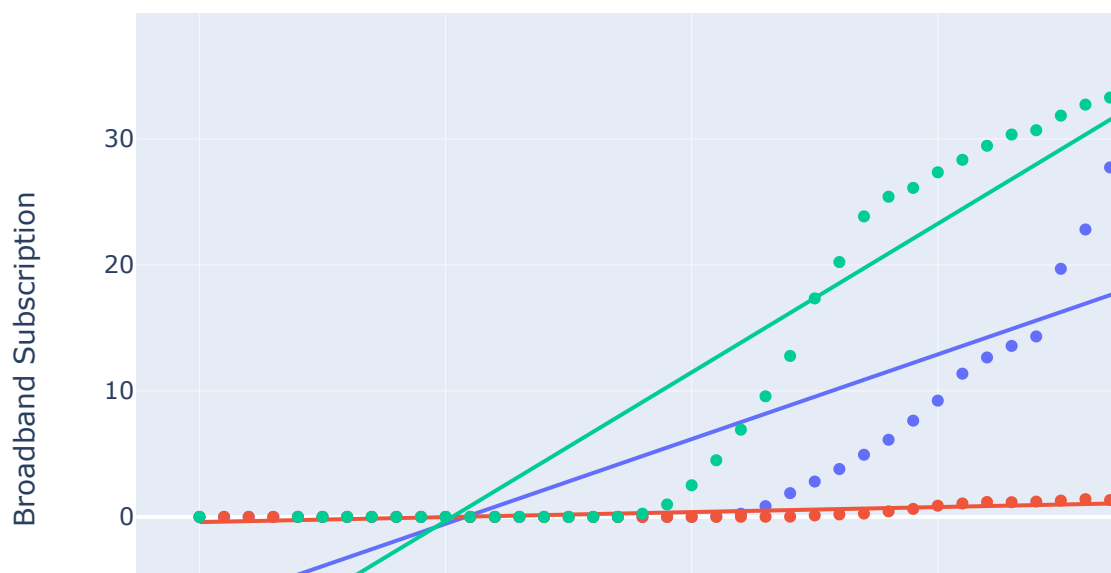
Landline Internet subscriptions per 100 people, 1998 to 2020



In [45]:

```
fig = px.scatter(df1, x="Year", y="Broadband Subscription", color="Entity", trendline="o")  
fig.show()
```

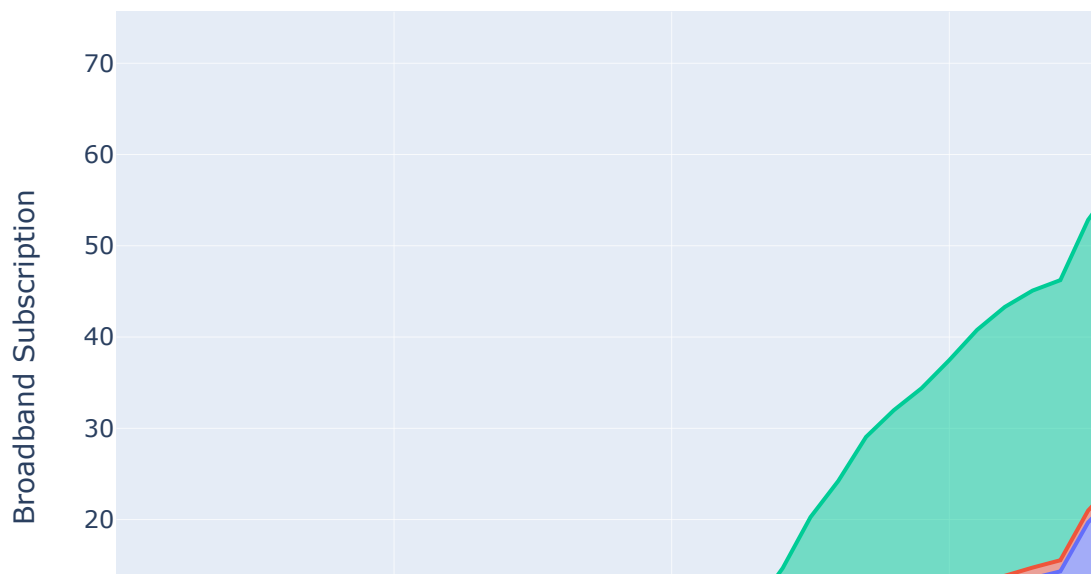
Broadband subscriptions per 100 people 1980 to 2020



In [46]:

```
fig = px.area(df1, x="Year", y="Broadband Subscription", color="Entity", title="Broadban  
fig.show()
```

Broadband subscriptions per 100 people 1980 to 2020



In [47]:

```
fig = px.area(df1, x="Year", y="Broadband Subscription", color="Entity", title="Broadband  
          labels={"Year": "Year", "Broadband Subscription": "Broadband Subscription per  
fig.update_layout(yaxis_range=[0,100])  
fig.show()
```

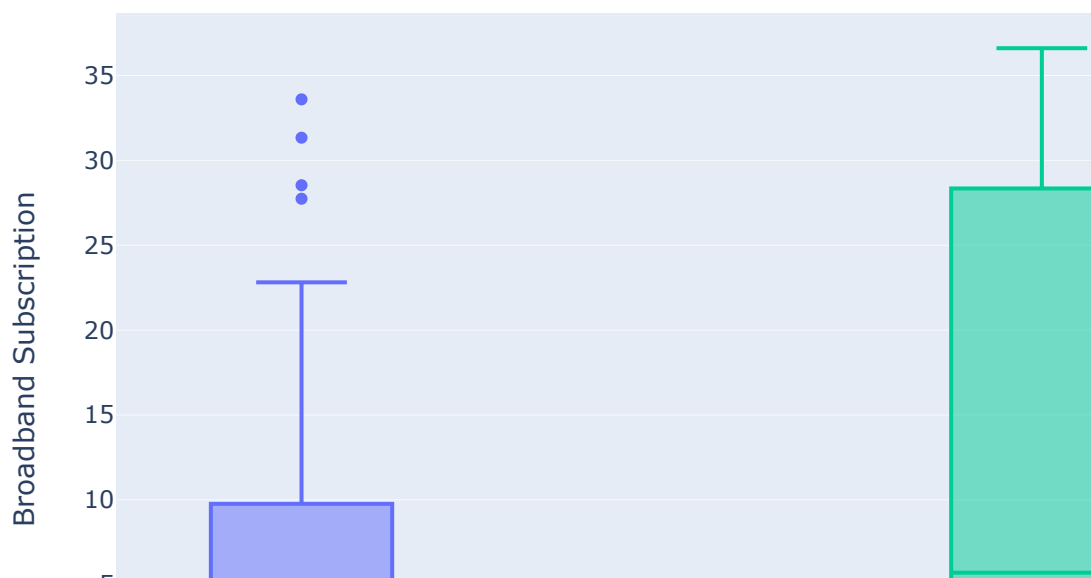
Broadband subscriptions per 100 people 1980 to 2020



In [48]:

```
fig = px.box(df1, x="Entity", y="Broadband Subscription", color="Entity", title="Broadba  
fig.show()
```

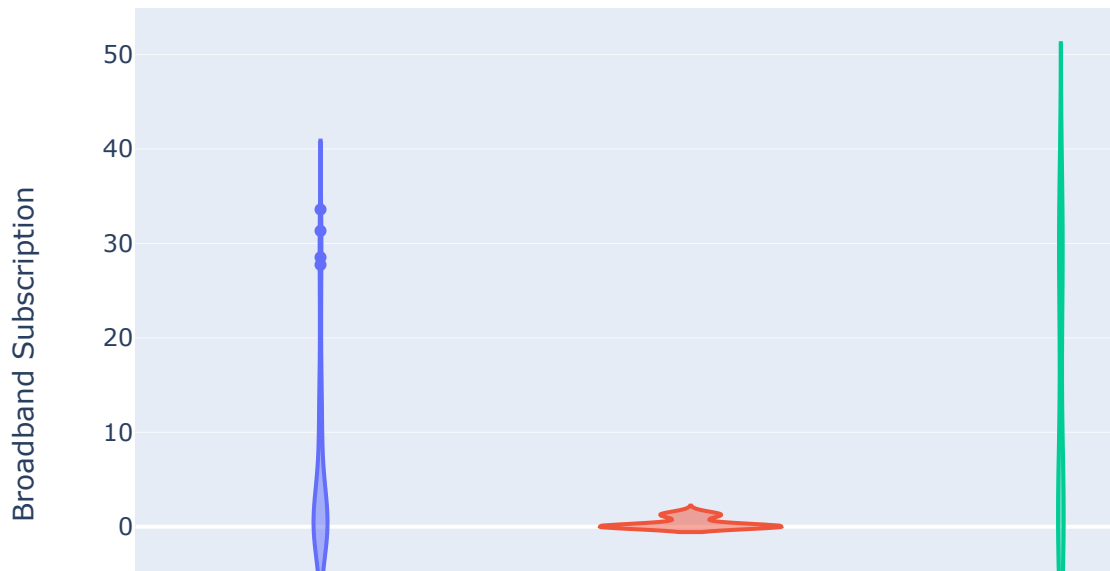
Broadband subscriptions per 100 people 1980 to 2020



In [49]:

```
fig = px.violin(df1, x="Entity", y="Broadband Subscription", color="Entity", title="Broa  
fig.show()
```

Broadband subscriptions per 100 people 1980 to 2020



In [50]:

```
df2 = df[df['Code']=='Region']
```

In [51]:

```
df2['Entity'].unique()
```

Out[51]:

```
array(['East Asia and Pacific', 'Europe and Central Asia',  
      'European Union', 'High income', 'Latin America and Caribbean',  
      'Low and middle income', 'Low income', 'Lower middle income',  
      'Middle East and North Africa', 'Middle income', 'North America',  
      'South Asia', 'Sub-Saharan Africa', 'Upper middle income'],  

```

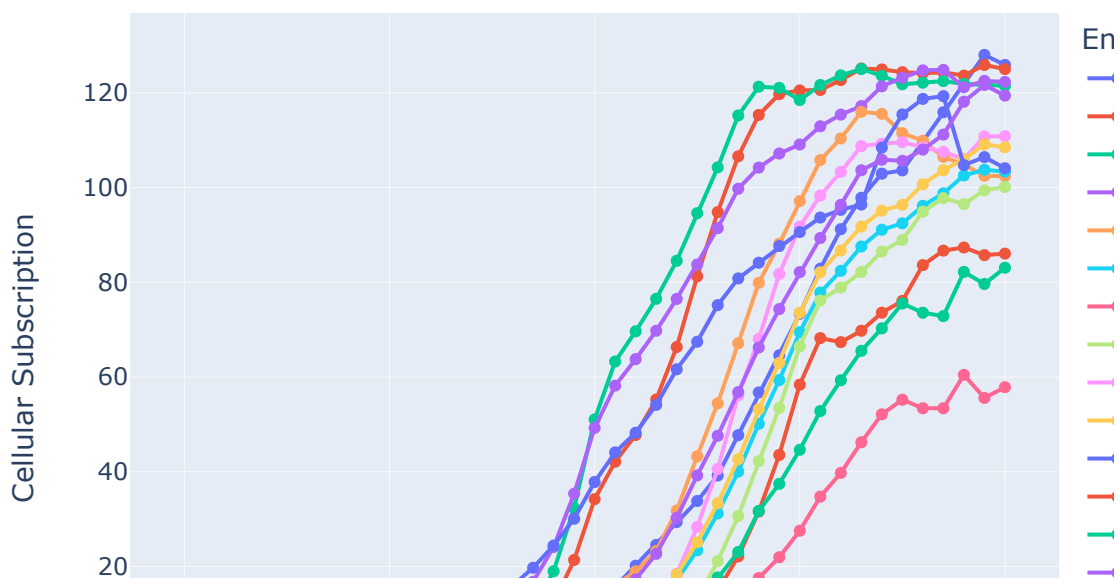
In [52]:

```
temp=df2[(df2['Entity']=='East Asia and Pacific')|(df2['Entity']=='Europe and Central As
```

In [53]:

```
fig = px.line(df2, x='Year', y='Cellular Subscription', color='Entity', markers=True,  
              title='Mobile Phone subscriptions per 100 people, 1998 to 2020')  
fig.update_layout(hovermode='x unified')  
fig.show()
```

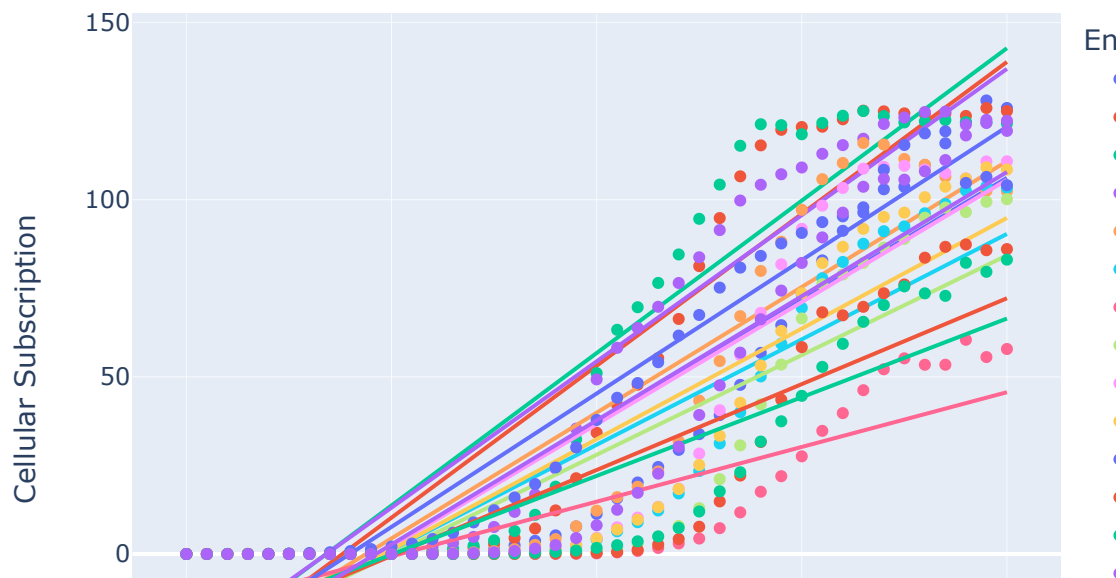
Mobile Phone subscriptions per 100 people, 1998 to 2020



In [54]:

```
fig = px.scatter(df2, x="Year", y="Cellular Subscription", color="Entity", trendline="ol")  
fig.show()
```

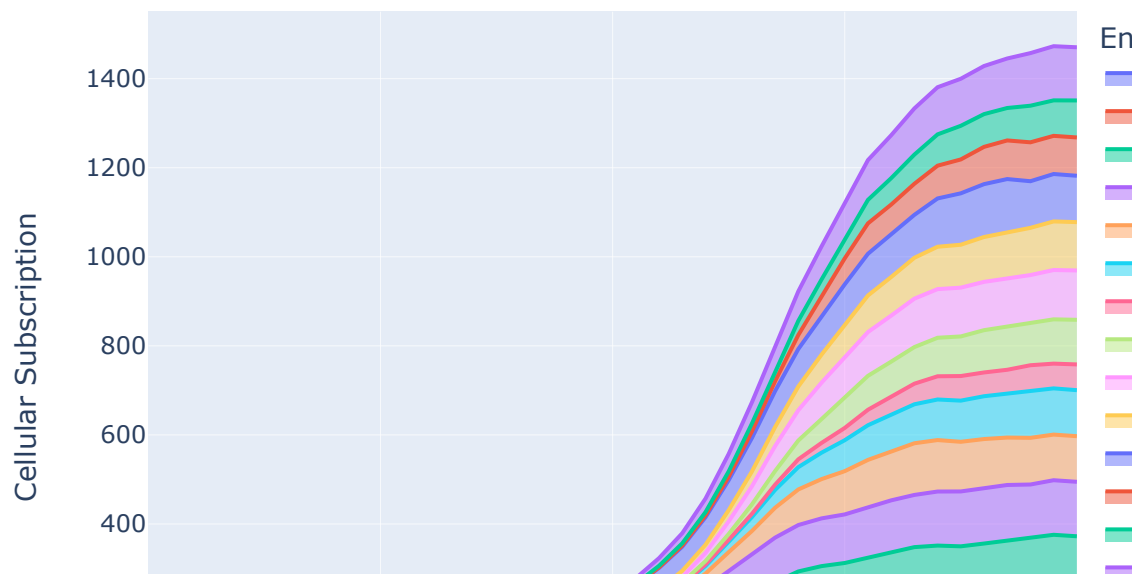
Mobile phone subscriptions per 100 people 1980 to 2020



In [55]:

```
fig = px.area(df2, x="Year", y="Cellular Subscription", color="Entity", title="Mobile ph  
fig.show()
```

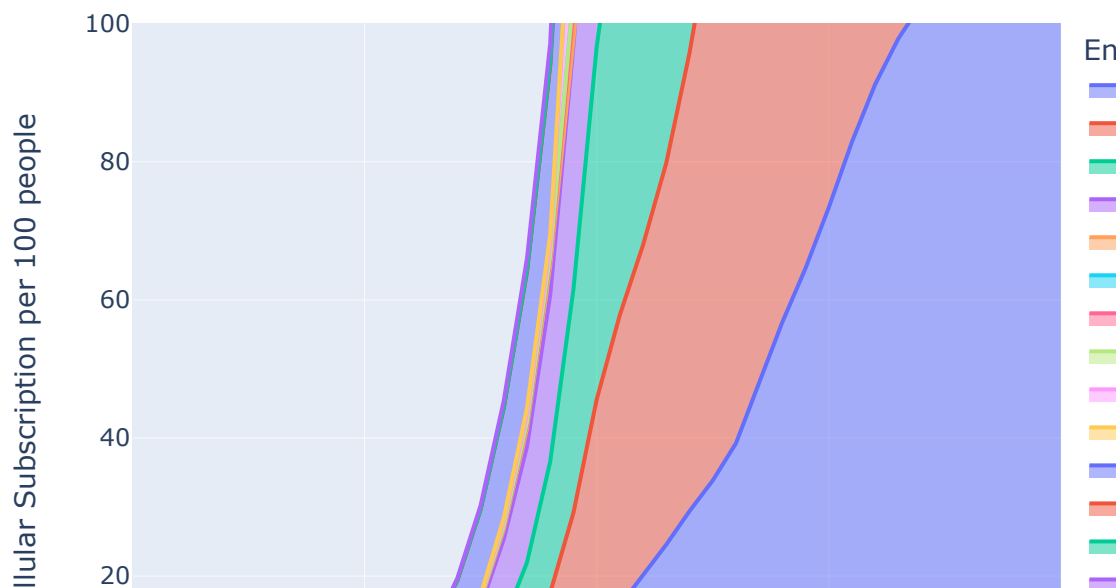
Mobile phone subscriptions per 100 people 1980 to 2020



In [56]:

```
fig = px.area(df2, x="Year", y="Cellular Subscription", color="Entity", title="Mobile ph  
labels={"Year": "Year", "Cellular Subscription": "Cellular Subscription per 1  
fig.update_layout(yaxis_range=[0,100])  
fig.show()
```

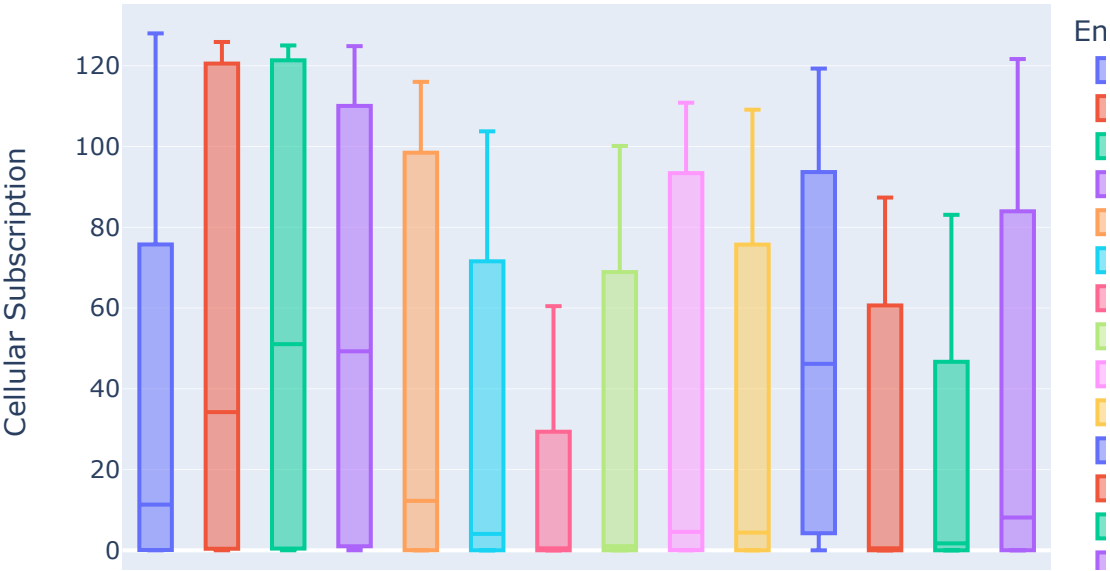
Mobile phone subscriptions per 100 people 1980 to 2020



In [57]:

```
fig = px.box(df2, x="Entity", y="Cellular Subscription", color="Entity", title="Mobile p
fig.show()
```

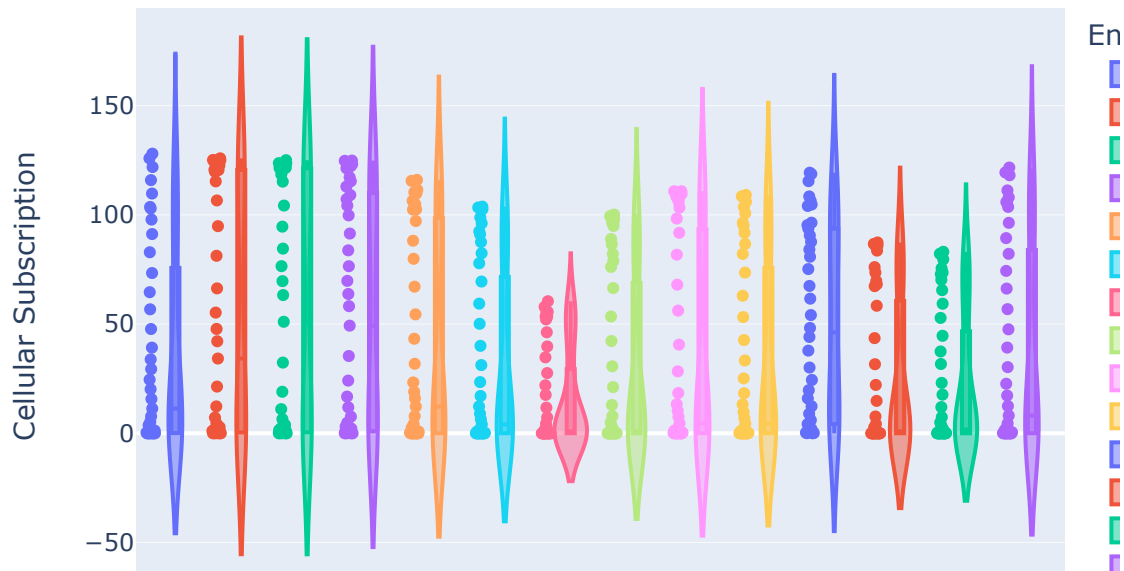
Mobile phone subscriptions per 100 people 1980 to 2020



In [58]:

```
fig = px.violin(df2, x="Entity", y="Cellular Subscription", color="Entity", box=True, po  
fig.show()
```

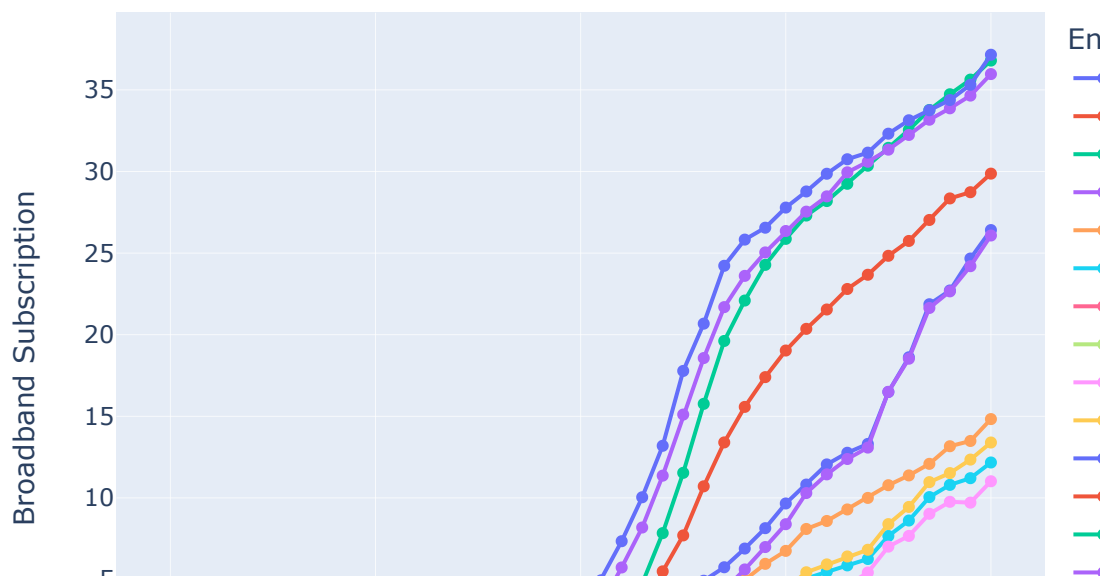
Mobile phone subscriptions per 100 people 1980 to 2020



In [59]:

```
fig = px.line(df2, x='Year', y='Broadband Subscription', color='Entity', markers=True,  
              title='Landline Internet subscriptions per 100 people, 1998 to 2020')  
fig.update_layout(hovermode='x unified')  
fig.show()
```

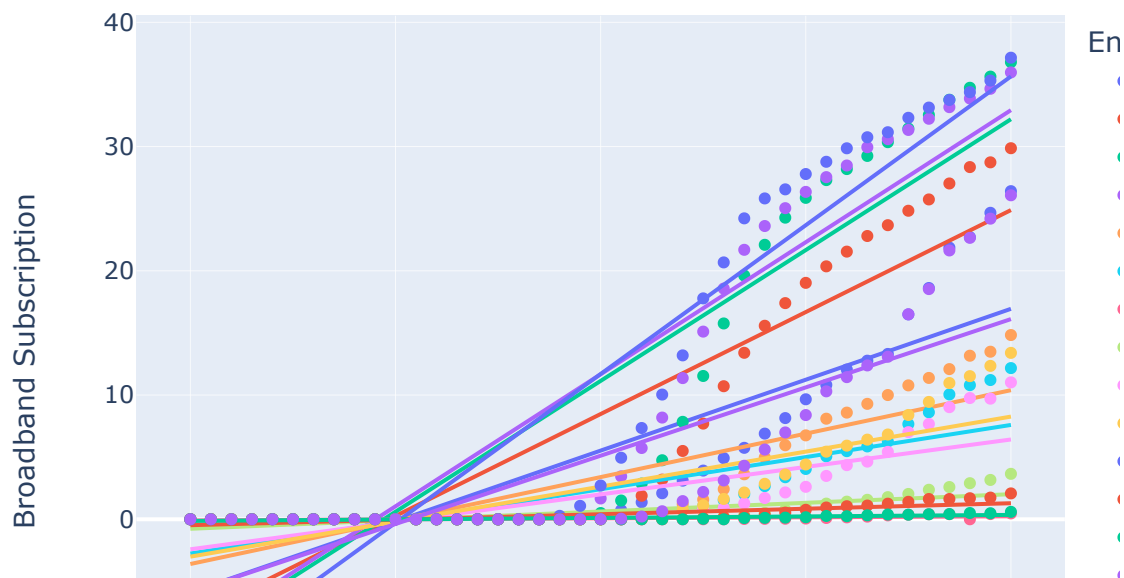
Landline Internet subscriptions per 100 people, 1998 to 2020



In [60]:

```
fig = px.scatter(df2, x="Year", y="Broadband Subscription", color="Entity", trendline="o")  
fig.show()
```

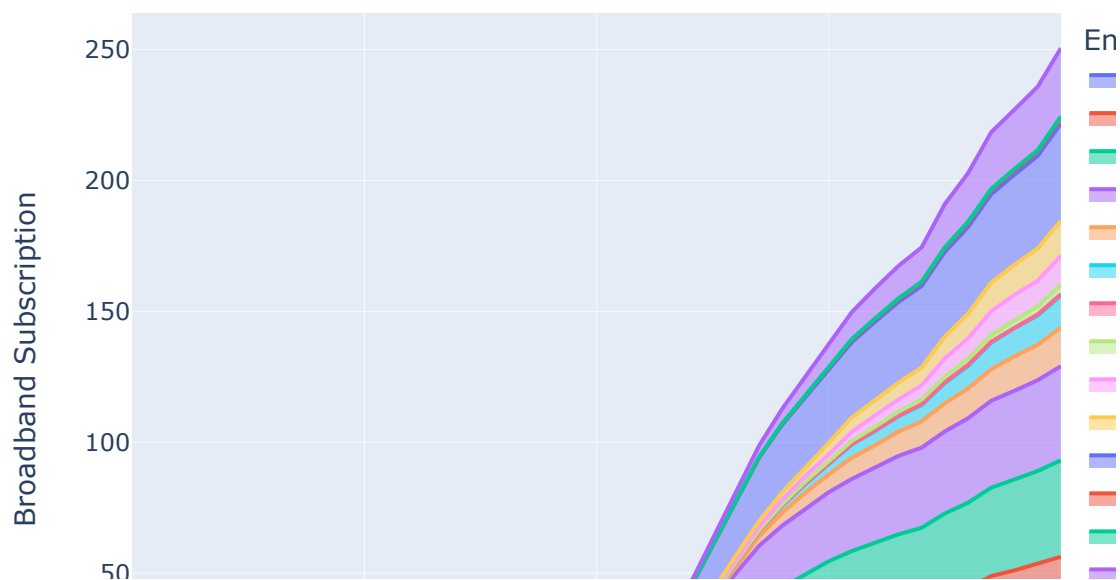
Landline Internet subscriptions per 100 people 1980 to 2020



In [61]:

```
fig = px.area(df2, x="Year", y="Broadband Subscription", color="Entity", title="Landline  
fig.show()
```

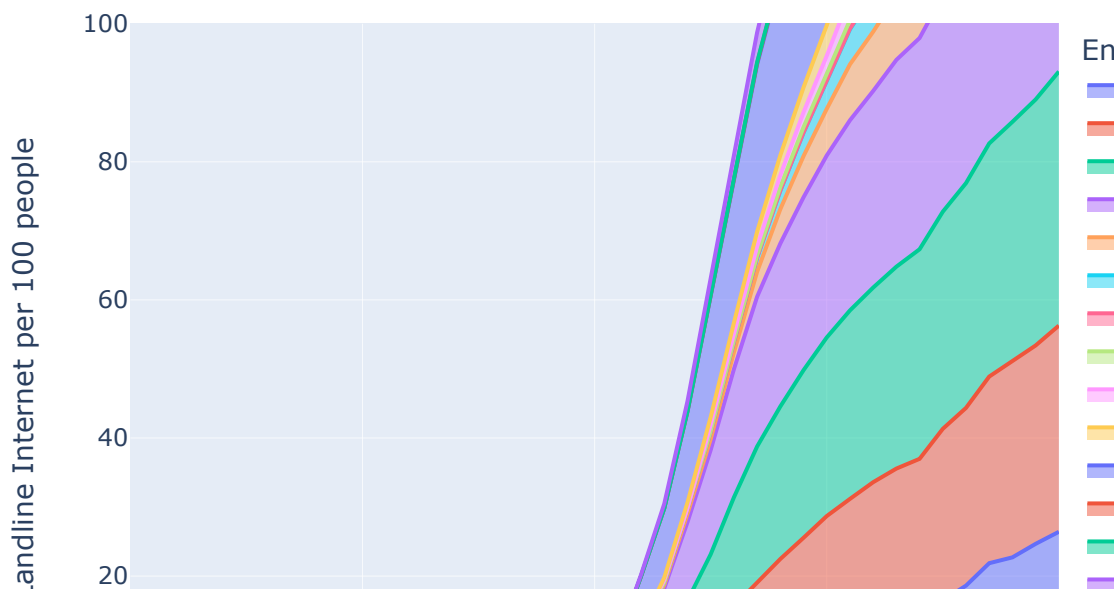
Landline Internet subscriptions per 100 people 1980 to 2020



In [62]:

```
fig = px.area(df2, x="Year", y="Broadband Subscription", color="Entity", title="Landline  
labels={"Year": "Year", "Broadband Subscription": "Landline Internet per 100  
fig.update_layout(yaxis_range=[0,100])  
fig.show()
```

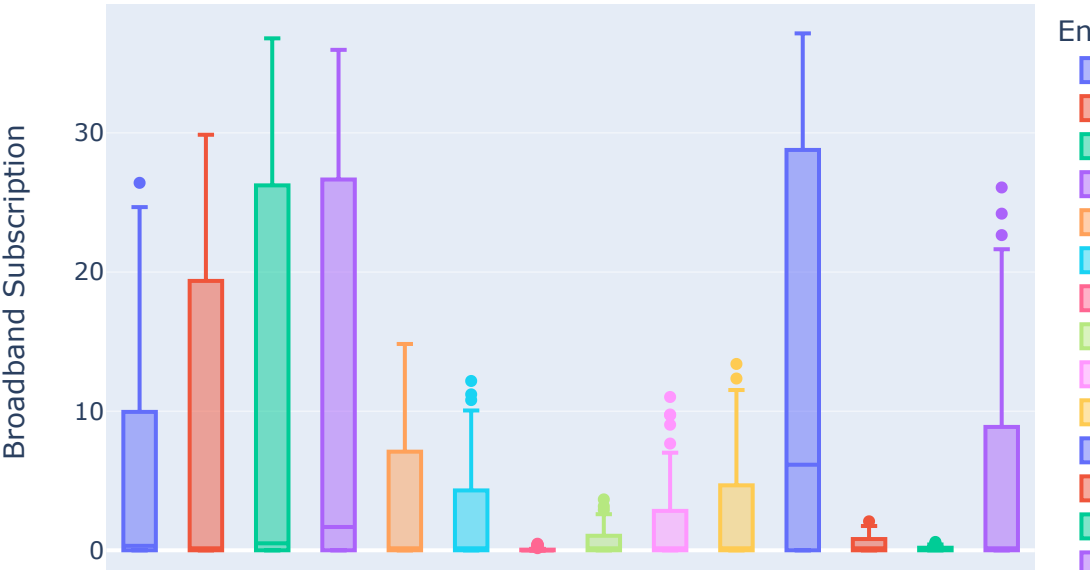
Landline Internet subscriptions per 100 people 1980 to 2020



In [63]:

```
fig = px.box(df2, x="Entity", y="Broadband Subscription", color="Entity", title="Landlin  
fig.show()
```

Landline Internet subscriptions per 100 people 1980 to 2020



In [64]:

```
fig = px.violin(df2, x="Entity", y="Broadband Subscription", color="Entity", title="Land  
fig.show()
```

Landline Internet subscriptions per 100 people 1980 to 2020

