Indian School Education Statistics (Part 2)

Data Analysis on school with boys & girls toilet during 2013 to 2016

Import Libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

In [2]:
```
boys_df = pd.read_csv('schools-with-boys-toilet-2013-2016.csv')
boys_df.head()
```

Out[2]:

	State_UT	year	Primary_Only	Primary_with_U_Primary	Primary_with_U_Primary_Sec_HrSec	U_Prim
0	Andaman & Nicobar Islands	2013-14	91.58	97.37	100.00	
1	Andaman & Nicobar Islands	2014-15	100.00	100.00	100.00	
2	Andaman & Nicobar Islands	2015-16	100.00	100.00	100.00	
3	Andhra Pradesh	2013-14	53.03	62.58	82.05	
4	Andhra Pradesh	2014-15	57.91	76.51	96.00	

In [3]:
```
boys_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110 entries, 0 to 109
Data columns (total 13 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   State_UT                        110 non-null    object
 1   year                            110 non-null    object
 2   Primary_Only                    110 non-null    float64
 3   Primary_with_U_Primary          110 non-null    float64
 4   Primary_with_U_Primary_Sec_HrSec 110 non-null   float64
 5   U_Primary_Only                  110 non-null    float64
 6   U_Primary_With_Sec_HrSec        110 non-null    float64
 7   Primary_with_U_Primary_Sec      110 non-null    float64
 8   U_Primary_With_Sec              110 non-null    float64
 9   Sec_Only                        110 non-null    float64
 10  Sec_with_HrSec.                 110 non-null    float64
 11  HrSec_Only                      110 non-null    float64
 12  All Schools                     110 non-null    float64
dtypes: float64(11), object(2)
memory usage: 11.3+ KB
```

In [4]:
```
boys_df.describe()
```

Out[4]:

	Primary_Only	Primary_with_U_Primary	Primary_with_U_Primary_Sec_HrSec	U_Primary_Only	U_I
count	110.000000	110.000000	110.000000	110.000000	
mean	90.433636	94.101727	95.777182	82.290182	
std	13.312718	9.393651	13.991093	30.076269	
min	38.240000	61.170000	0.000000	0.000000	
25%	86.567500	93.525000	97.932500	83.382500	
50%	96.045000	98.485000	99.620000	95.415000	
75%	99.595000	99.857500	100.000000	100.000000	
max	100.000000	100.000000	100.000000	100.000000	

In [5]:
```
girls_df = pd.read_csv('schools-with-girls-toilet-2013-2016.csv')
girls_df.head()
```

Out[5]:

	State_UT	year	Primary_Only	Primary_with_U_Primary	Primary_with_U_Primary_Sec_HrSec	U_Prim
0	All India	2013-14	88.68	95.98	98.81	
1	All India	2014-15	91.21	96.92	99.48	
2	All India	2015-16	96.95	99.03	99.72	
3	Andaman & Nicobar Islands	2013-14	89.74	97.37	100.00	
4	Andaman & Nicobar Islands	2014-15	100.00	100.00	100.00	

In [6]:
```
girls_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110 entries, 0 to 109
Data columns (total 13 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   State_UT                        110 non-null    object 
 1   year                            110 non-null    object 
 2   Primary_Only                    110 non-null    float64
 3   Primary_with_U_Primary          110 non-null    float64
 4   Primary_with_U_Primary_Sec_HrSec  110 non-null    float64
 5   U_Primary_Only                  110 non-null    float64
 6   U_Primary_With_Sec_HrSec        110 non-null    float64
 7   Primary_with_U_Primary_Sec      110 non-null    float64
 8   U_Primary_With_Sec              110 non-null    float64
 9   Sec_Only                        110 non-null    float64
 10  Sec_with_HrSec.                 110 non-null    float64
 11  HrSec_Only                      110 non-null    float64
 12  All Schools                     110 non-null    float64
dtypes: float64(11), object(2)
memory usage: 11.3+ KB
```

In [7]:
```
girls_df.describe()
```

Out[7]:

	Primary_Only	Primary_with_U_Primary	Primary_with_U_Primary_Sec_HrSec	U_Primary_Only	U_I
count	110.000000	110.000000	110.000000	110.000000	
mean	93.794818	96.927727	97.034909	87.754545	
std	9.886684	6.285276	13.649124	26.539495	
min	50.840000	61.650000	0.000000	0.000000	
25%	91.955000	97.825000	99.522500	91.590000	
50%	97.980000	99.635000	100.000000	98.735000	
75%	99.827500	99.995000	100.000000	100.000000	
max	100.000000	100.000000	100.000000	100.000000	

Data Preprocessing

In [8]:
```
# merge both the dataframe and make a single one named 'df'
df = pd.merge(boys_df,girls_df,how='outer')
```

In [9]:
```
df.head()
```

Out[9]:

	State_UT	year	Primary_Only	Primary_with_U_Primary	Primary_with_U_Primary_Sec_HrSec	U_Prim
0	Andaman & Nicobar Islands	2013-14	91.58	97.37	100.00	
1	Andaman & Nicobar Islands	2014-15	100.00	100.00	100.00	
2	Andaman & Nicobar Islands	2015-16	100.00	100.00	100.00	
3	Andhra Pradesh	2013-14	53.03	62.58	82.05	
4	Andhra Pradesh	2014-15	57.91	76.51	96.00	

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

Out[10]:

	Primary_Only	Primary_with_U_Primary	Primary_with_U_Primary_Sec_HrSec	U_Primary_Only	U_I
count	203.000000	203.000000	203.000000	203.000000	
mean	91.453842	95.139113	96.105074	86.231133	
std	12.074020	8.322953	14.332060	26.126378	
min	38.240000	61.170000	0.000000	0.000000	
25%	88.475000	95.090000	98.325000	87.885000	
50%	96.510000	98.890000	99.830000	97.140000	
75%	99.435000	99.880000	100.000000	100.000000	
max	100.000000	100.000000	100.000000	100.000000	

In [11]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 203 entries, 0 to 202
Data columns (total 13 columns):
 #   Column                           Non-Null Count  Dtype  
---  ------                           --------------  -----  
 0   State_UT                         203 non-null    object 
 1   year                             203 non-null    object 
 2   Primary_Only                     203 non-null    float64
 3   Primary_with_U_Primary           203 non-null    float64
 4   Primary_with_U_Primary_Sec_HrSec 203 non-null    float64
 5   U_Primary_Only                   203 non-null    float64
 6   U_Primary_With_Sec_HrSec         203 non-null    float64
 7   Primary_with_U_Primary_Sec       203 non-null    float64
 8   U_Primary_With_Sec               203 non-null    float64
 9   Sec_Only                         203 non-null    float64
 10  Sec_with_HrSec.                  203 non-null    float64
 11  HrSec_Only                       203 non-null    float64
 12  All Schools                      203 non-null    float64
dtypes: float64(11), object(2)
memory usage: 22.2+ KB
```

In [12]:
```
df.columns
```

Out[12]:
```
Index(['State_UT', 'year', 'Primary_Only', 'Primary_with_U_Primary',
       'Primary_with_U_Primary_Sec_HrSec', 'U_Primary_Only',
       'U_Primary_With_Sec_HrSec', 'Primary_with_U_Primary_Sec',
       'U_Primary_With_Sec', 'Sec_Only', 'Sec_with_HrSec.', 'HrSec_Only',
       'All Schools'],
      dtype='object')
```

In [13]:
```
df.isna().sum()
```

Out[13]:
```
State_UT                          0
year                              0
Primary_Only                      0
Primary_with_U_Primary            0
Primary_with_U_Primary_Sec_HrSec  0
U_Primary_Only                    0
U_Primary_With_Sec_HrSec          0
Primary_with_U_Primary_Sec        0
U_Primary_With_Sec                0
Sec_Only                          0
Sec_with_HrSec.                   0
HrSec_Only                        0
All Schools                       0
dtype: int64
```

In [14]:
```
df['State_UT'].nunique()
```

Out[14]:
```
37
```

In [15]:
```
df['State_UT'].unique()
```

Out[15]:
```
array(['Andaman & Nicobar Islands', 'Andhra Pradesh', 'Arunachal Pradesh',
       'Assam', 'Bihar', 'Chandigarh', 'Chhattisgarh',
       'Dadra & Nagar Haveli', 'Daman & Diu', 'Delhi', 'Goa', 'Gujarat',
       'Haryana', 'Himachal Pradesh', 'Jammu And Kashmir', 'Jharkhand',
       'Karnataka', 'Kerala', 'Lakshadweep', 'Madhya Pradesh',
       'Maharashtra', 'Manipur', 'Meghalaya', 'Mizoram', 'Nagaland',
       'Odisha', 'Puducherry', 'Punjab', 'Rajasthan', 'Sikkim',
       'Tamil Nadu', 'Telangana', 'Tripura', 'Uttar Pradesh',
       'Uttarakhand', 'West Bengal', 'All India'], dtype=object)
```

In [16]:
```
# Make a single column for Primary Class
df['Primary'] = df[['Primary_Only','Primary_with_U_Primary']].mean(axis=1)

# Make a single column for Upper Primary Class
df['Upper_Primary'] = df[['U_Primary_Only','U_Primary_With_Sec','Primary_with_U_Primar

# Make a single column for Secondary Class
df['Secondary'] = df[['Sec_Only','U_Primary_With_Sec_HrSec']].mean(axis=1)

# Make a single column for Hr Secondary Class
df['Hr_Secondary'] = df[['HrSec_Only','Sec_with_HrSec.']].mean(axis=1)

# Make a single column for All school class
df['All_Schools'] = df[['All Schools','Primary_with_U_Primary_Sec_HrSec']].mean(axis=1
```

In [17]:
```
df.drop(['Primary_Only', 'Primary_with_U_Primary','Primary_with_U_Primary_Sec_HrSec',
       'U_Primary_With_Sec_HrSec', 'Primary_with_U_Primary_Sec','U_Primary_With_Sec',
         'Sec_with_HrSec.', 'HrSec_Only','All Schools'],axis=1,inplace=True)
```

In [18]:
```
df.head()
```

Out[18]:

	State_UT	year	Primary	Upper_Primary	Secondary	Hr_Secondary	All_Schools
0	Andaman & Nicobar Islands	2013-14	94.475	33.333333	50.000	50.000	97.260
1	Andaman & Nicobar Islands	2014-15	100.000	66.666667	50.000	50.000	100.000
2	Andaman & Nicobar Islands	2015-16	100.000	33.333333	50.000	50.000	100.000
3	Andhra Pradesh	2013-14	57.805	60.773333	61.685	79.575	69.465
4	Andhra Pradesh	2014-15	67.210	81.473333	86.990	73.270	80.670

In [19]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 203 entries, 0 to 202
Data columns (total 7 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   State_UT       203 non-null    object 
 1   year           203 non-null    object 
 2   Primary        203 non-null    float64
 3   Upper_Primary  203 non-null    float64
 4   Secondary      203 non-null    float64
 5   Hr_Secondary   203 non-null    float64
 6   All_Schools    203 non-null    float64
dtypes: float64(5), object(2)
memory usage: 12.7+ KB
```

In [20]:
```
df.describe()
```

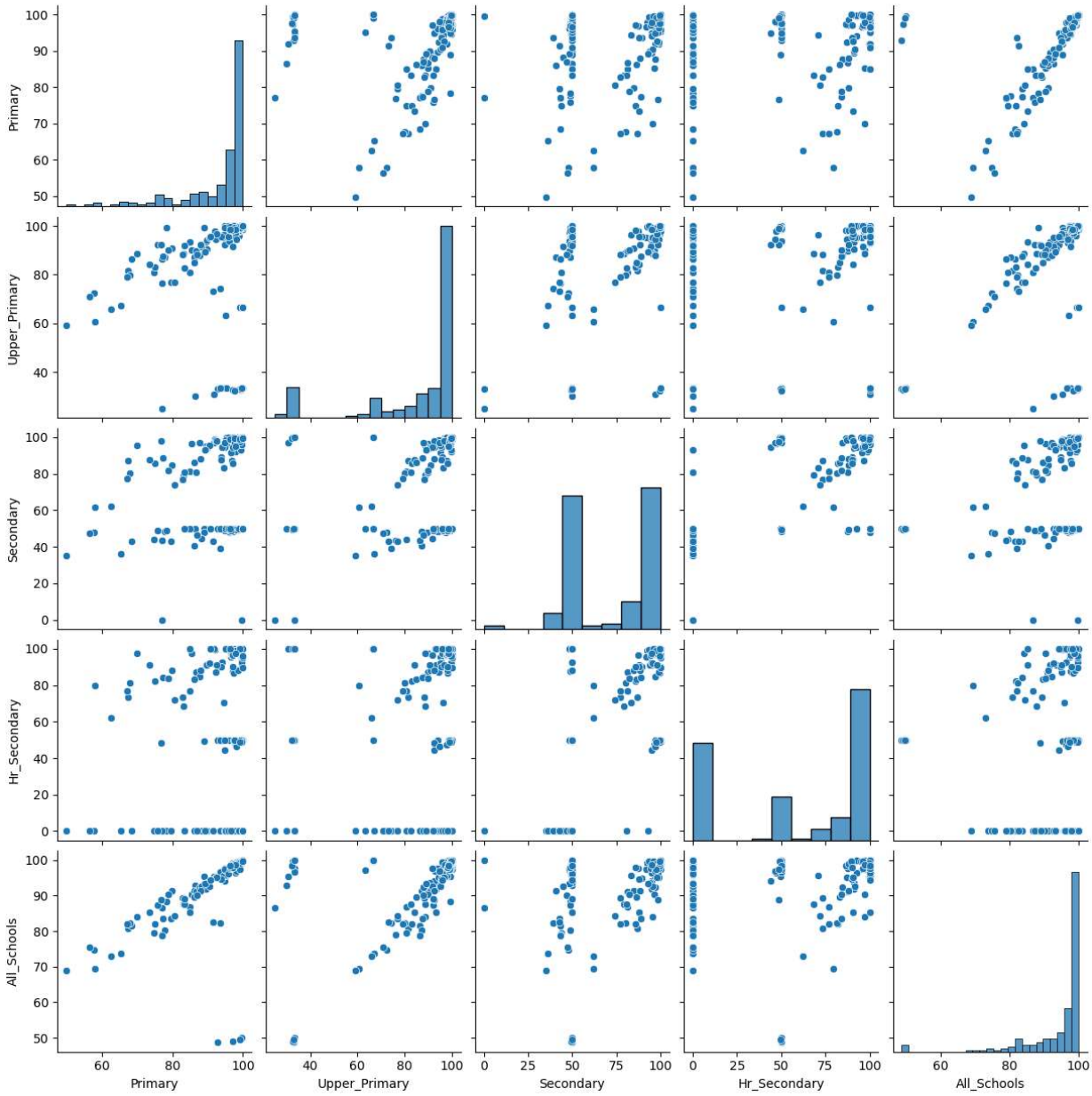
Out[20]:

	Primary	Upper_Primary	Secondary	Hr_Secondary	All_Schools
count	203.000000	203.000000	203.000000	203.000000	203.000000
mean	93.296478	86.575419	72.913177	60.392241	94.484581
std	9.900067	20.872246	24.787215	42.477615	9.224605
min	49.705000	25.130000	0.000000	0.000000	48.670000
25%	91.842500	84.531667	50.000000	0.000000	93.450000
50%	97.545000	97.193333	80.875000	84.235000	98.320000
75%	99.600000	99.465000	98.717500	99.592500	99.640000
max	100.000000	100.000000	100.000000	100.000000	100.000000

Exploratory Data Analysis

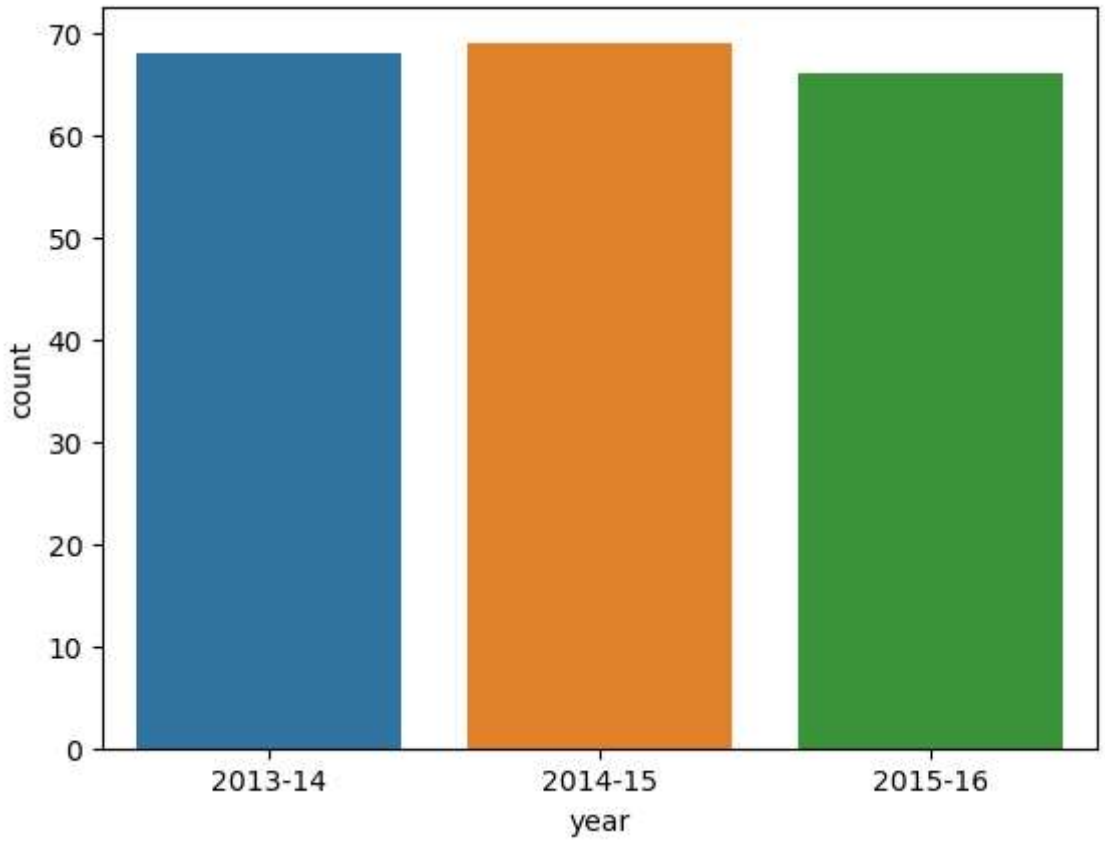
In [21]:
```
plt.figure(figsize=(20,20))
sns.pairplot(data=df)
```

Out[21]:
```
<seaborn.axisgrid.PairGrid at 0x1de0f965850>

<Figure size 2000x2000 with 0 Axes>
```



In [22]:
```
sns.countplot(data=df,x='year')
```

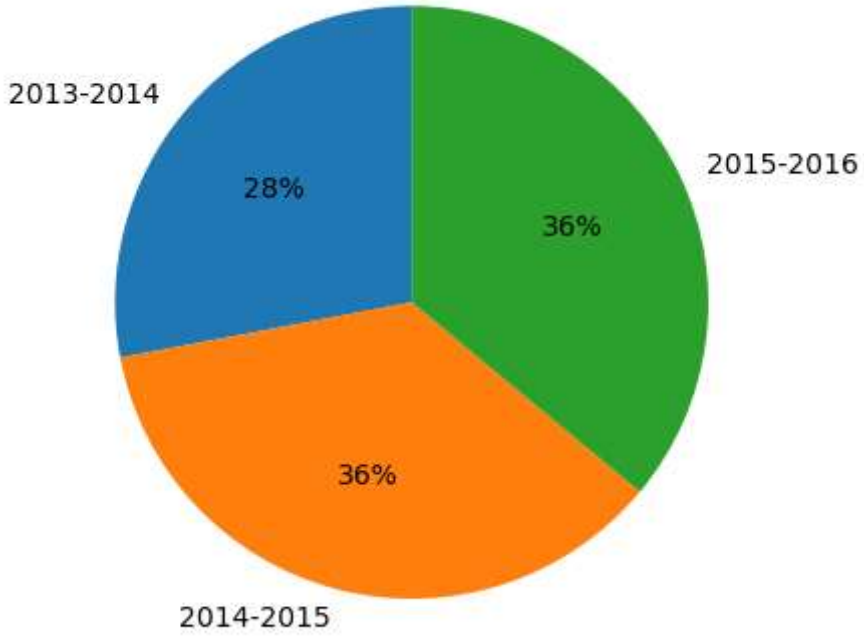
Out[22]:
```
<AxesSubplot:xlabel='year', ylabel='count'>
```



In [23]:
```
# Distribution of Years in Percentage

year_dr1 = df[['Primary','Upper_Primary','Secondary','Hr_Secondary','All_Schools']][df
year_dr2 = df[['Primary','Upper_Primary','Secondary','Hr_Secondary','All_Schools']][df
year_dr3 = df[['Primary','Upper_Primary','Secondary','Hr_Secondary','All_Schools']][df
```

In [24]:
```
years_duration_data = [year_dr1.sum(), year_dr2.sum(), year_dr3.sum()]
years_duration = ['2013-2014', '2014-2015', '2015-2016']

plt.pie(years_duration_data, labels=years_duration, autopct='%.0f%%',startangle=90)
```

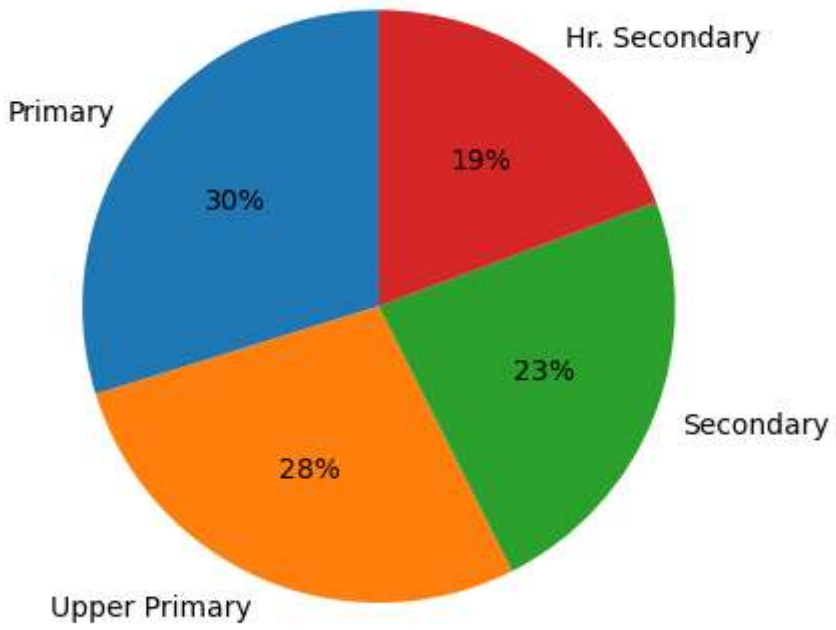
Out[24]:
```
([<matplotlib.patches.Wedge at 0x1de1722be50>,
  <matplotlib.patches.Wedge at 0x1de172395b0>,
  <matplotlib.patches.Wedge at 0x1de17239cd0>],
 [Text(-0.8474067305762509, 0.7013571365389174, '2013-2014'),
  Text(-0.27505754960578316, -1.0650555593042375, '2014-2015'),
  Text(0.9958626117105921, 0.4671805417576364, '2015-2016')],
 [Text(-0.46222185304159136, 0.3825584381121367, '28%'),
  Text(-0.15003139069406354, -0.5809393959841295, '36%'),
  Text(0.5431977882057775, 0.2548257500496198, '36%')])
```



In [25]:
```
# Distribution of Classes in Percentage

pri_class = df['Primary'].sum()
upp_pri_class = df['Upper_Primary'].sum()
secondary_class = df['Secondary'].sum()
hr_sec_class = df['Hr_Secondary'].sum()
```

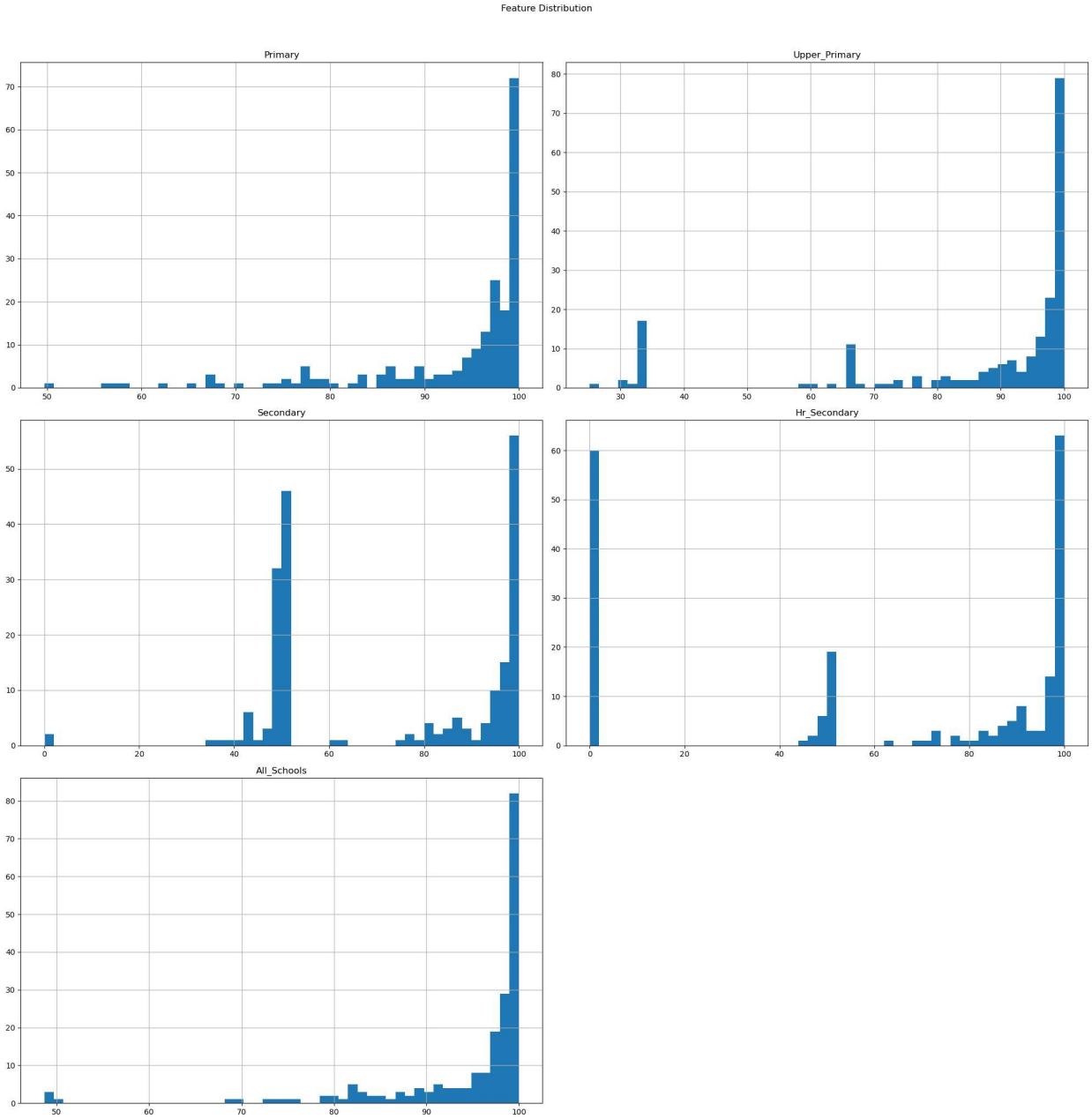
In [26]:
```
class_data = [pri_class, upp_pri_class, secondary_class, hr_sec_class]
class_label = ['Primary', 'Upper Primary', 'Secondary', 'Hr. Secondary']

plt.pie(class_data, labels=class_label, autopct='%.0f%%',startangle=90)
```

Out[26]:
```
([<matplotlib.patches.Wedge at 0x1de1727fcd0>,
  <matplotlib.patches.Wedge at 0x1de1728c430>,
  <matplotlib.patches.Wedge at 0x1de1728cb50>,
  <matplotlib.patches.Wedge at 0x1de172992b0>],
 [Text(-0.8856400276916334, 0.6524122480076252, 'Primary'),
  Text(-0.4297213110820474, -1.0125905365950871, 'Upper Primary'),
  Text(1.0246610573991346, -0.4000871373209685, 'Secondary'),
  Text(0.6263763997073603, 0.9042414533130216, 'Hr. Secondary')],
 [Text(-0.4830763787408909, 0.3558612261859773, '30%'),
  Text(-0.23439344240838947, -0.5523221108700475, '28%'),
  Text(0.5589060313086188, -0.21822934762961915, '23%'),
  Text(0.3416598543858328, 0.4932226108980117, '19%')])
```

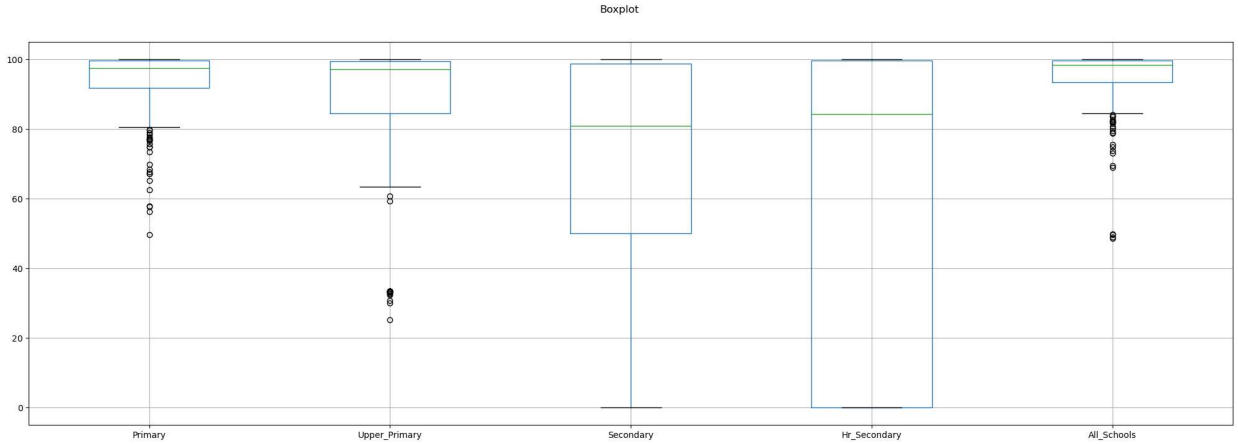


Univariant Data Analysis

In [27]:
```
df.hist(bins=50,figsize=(20,20))
plt.suptitle('Feature Distribution',fontsize='large',x=0.5,y=1.02,ha='center')
plt.tight_layout()
```

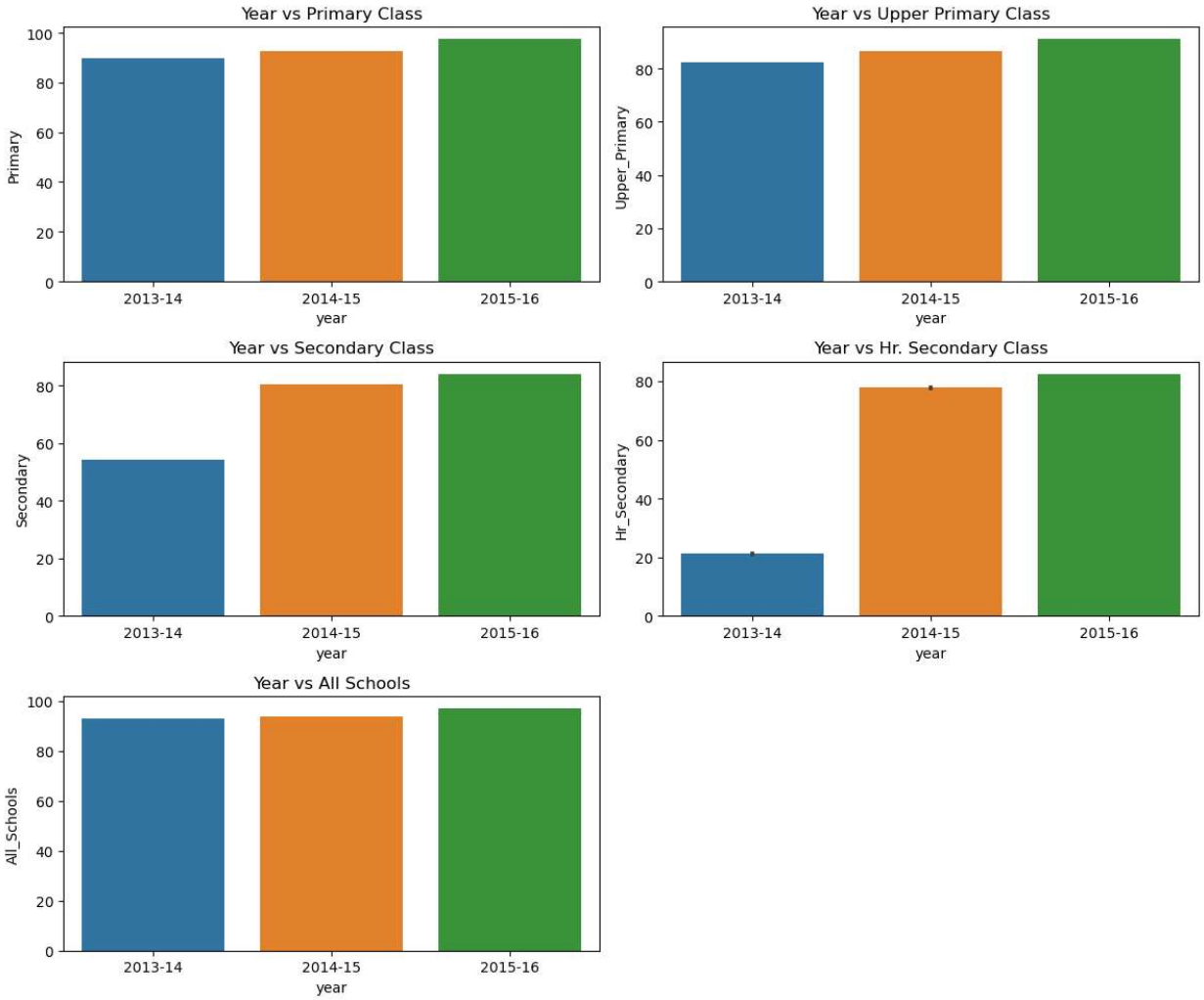


In [28]:
```
df.boxplot(figsize=(20,7))
plt.suptitle('Boxplot',fontsize='large',x=0.5,y=1.02,ha='center')
plt.tight_layout()
```

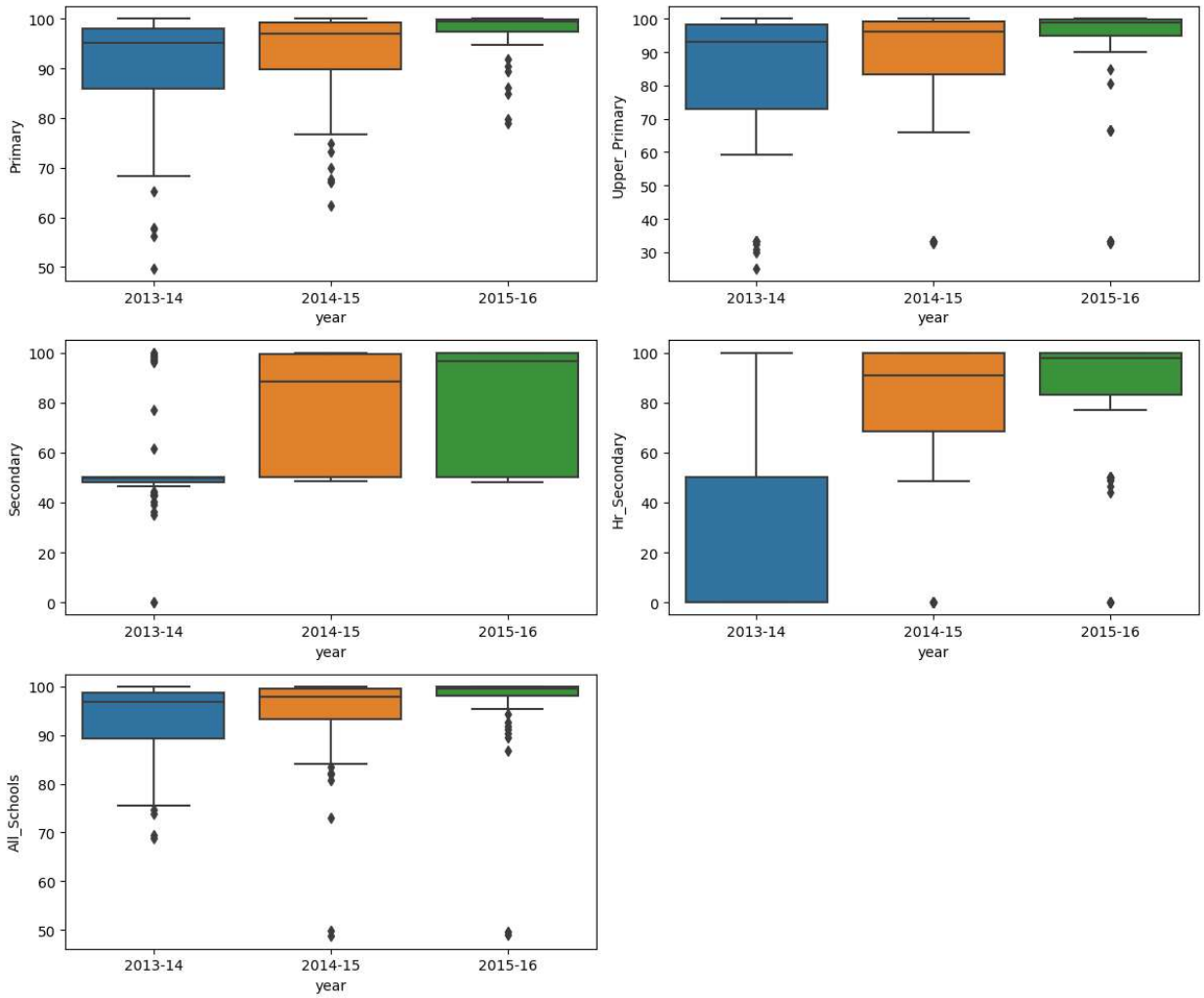


Bivariant Data Analysis

In [29]:
```
plt.figure(figsize=(12,10))

plt.subplot(3,2,1)
sns.barplot(df['year'],df['Primary'],ci=True)
plt.title('Year vs Primary Class')

plt.subplot(3,2,2)
sns.barplot(df['year'],df['Upper_Primary'],ci=True)
plt.title('Year vs Upper Primary Class')

plt.subplot(3,2,3)
sns.barplot(df['year'],df['Secondary'],ci=True)
plt.title('Year vs Secondary Class')

plt.subplot(3,2,4)
sns.barplot(df['year'],df['Hr_Secondary'],ci=True)
plt.title('Year vs Hr. Secondary Class')

plt.subplot(3,2,5)
sns.barplot(df['year'],df['All_Schools'],ci=True)
plt.title('Year vs All Schools')

plt.tight_layout()
```



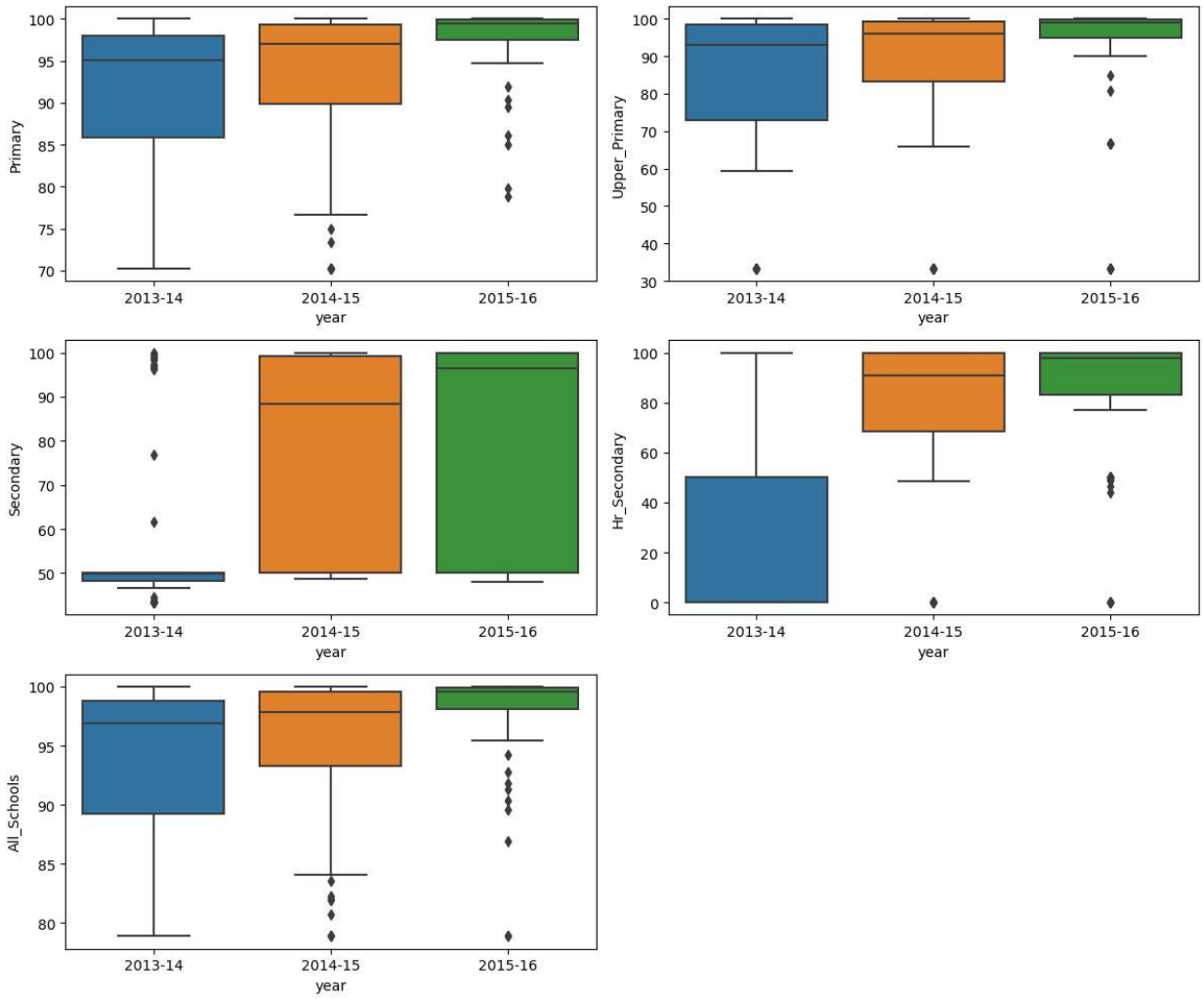
In [30]:
```
plt.figure(figsize=(12,10))

plt.subplot(3,2,1)
sns.boxplot(x='year',y='Primary',data=df)

plt.subplot(3,2,2)
sns.boxplot(x='year',y='Upper_Primary',data=df)

plt.subplot(3,2,3)
sns.boxplot(x='year',y='Secondary',data=df)

plt.subplot(3,2,4)
sns.boxplot(x='year',y='Hr_Secondary',data=df)

plt.subplot(3,2,5)
sns.boxplot(x='year',y='All_Schools',data=df)

plt.tight_layout()
```



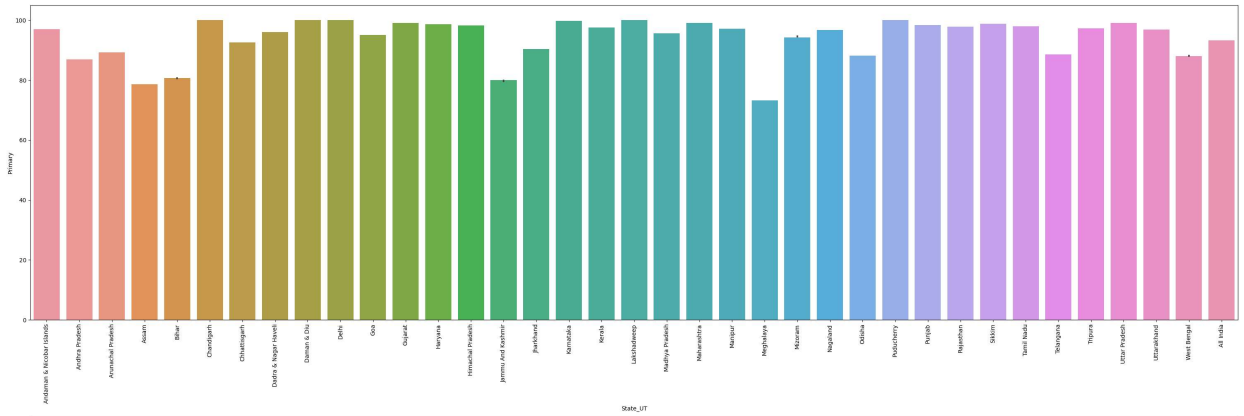
In [31]:
```
df['Primary']=df['Primary'].clip(lower=df['Primary'].quantile(0.05), upper=df['Primary

df['Upper_Primary']=df['Upper_Primary'].clip(lower=df['Upper_Primary'].quantile(0.05),

df['Secondary']=df['Secondary'].clip(lower=df['Secondary'].quantile(0.05), upper=df['S

df['Hr_Secondary']=df['Hr_Secondary'].clip(lower=df['Hr_Secondary'].quantile(0.05), up

df['All_Schools']=df['All_Schools'].clip(lower=df['All_Schools'].quantile(0.05), upper
```

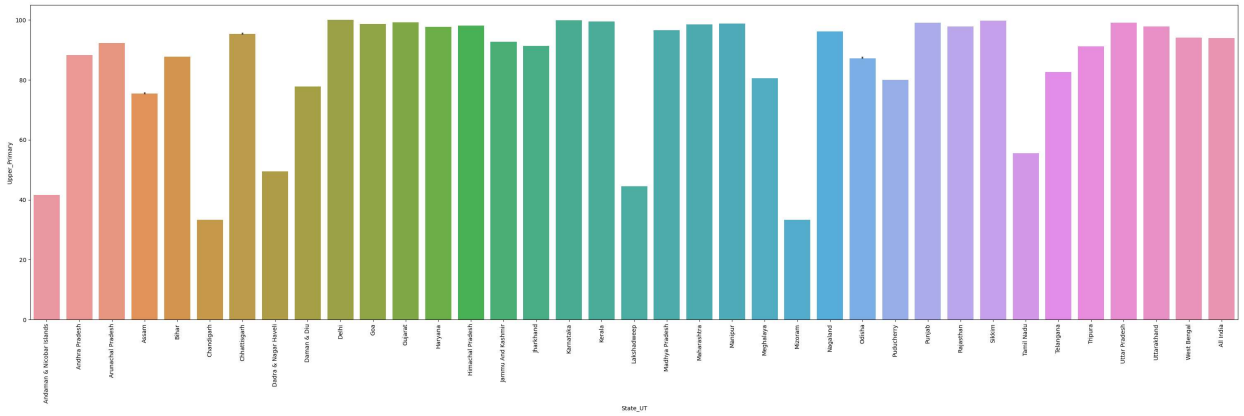
In [32]:
```
plt.figure(figsize=(12,10))

plt.subplot(3,2,1)
sns.boxplot(x='year',y='Primary',data=df)

plt.subplot(3,2,2)
sns.boxplot(x='year',y='Upper_Primary',data=df)

plt.subplot(3,2,3)
sns.boxplot(x='year',y='Secondary',data=df)

plt.subplot(3,2,4)
sns.boxplot(x='year',y='Hr_Secondary',data=df)

plt.subplot(3,2,5)
sns.boxplot(x='year',y='All_Schools',data=df)
```

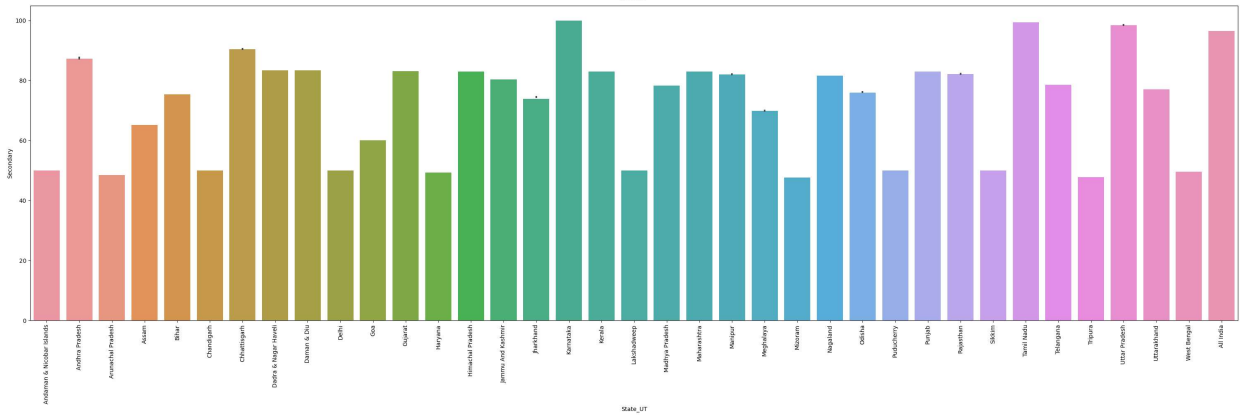
```
plt.tight_layout()
```

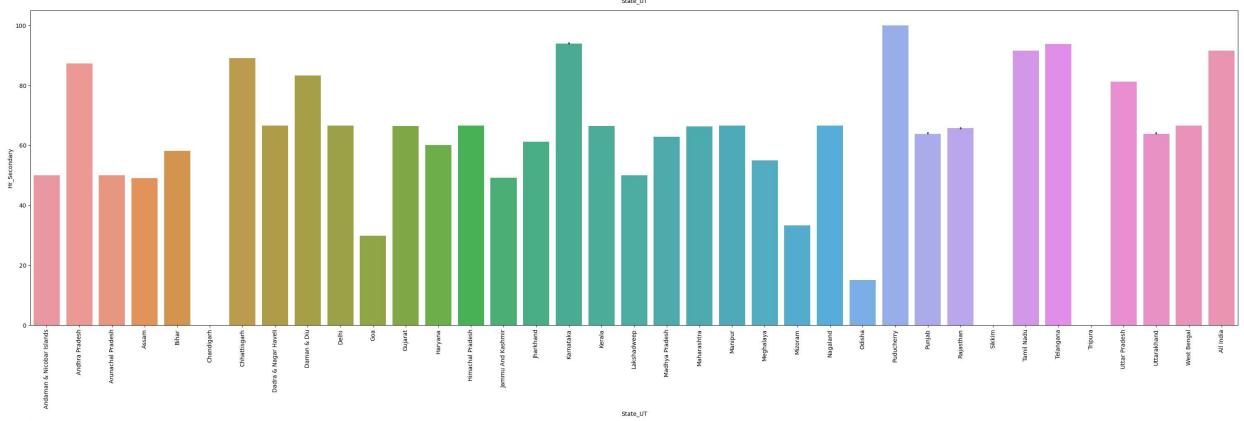


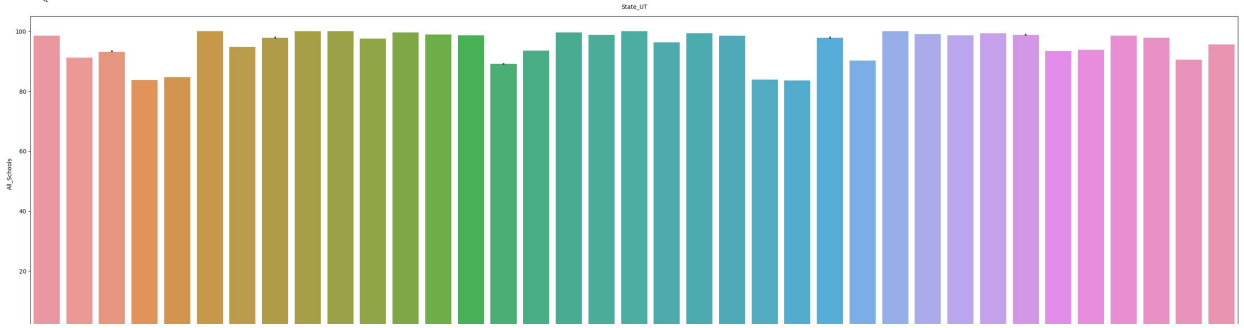
In [33]:
```
plt.figure(figsize=(30,50))
plt.subplot(5,1,1)
sns.barplot(x='State_UT',y='Primary',data=df,ci=True)
plt.xticks(rotation = 90)

plt.subplot(5,1,2)
sns.barplot(x='State_UT',y='Upper_Primary',data=df,ci=True)
plt.xticks(rotation = 90)

plt.subplot(5,1,3)
sns.barplot(x='State_UT',y='Secondary',data=df,ci=True)
plt.xticks(rotation = 90)

plt.subplot(5,1,4)
sns.barplot(x='State_UT',y='Hr_Secondary',data=df,ci=True)
plt.xticks(rotation = 90)

plt.subplot(5,1,5)
sns.barplot(x='State_UT',y='All_Schools',data=df,ci=True)
plt.xticks(rotation = 90)

plt.xticks(rotation = 90)
plt.tight_layout()
```

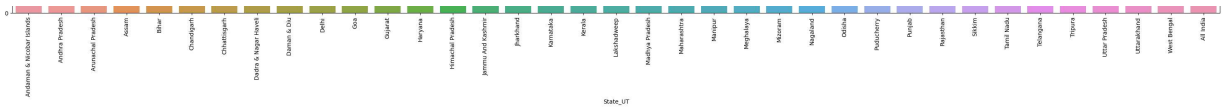

Primary
State_UT


Upper_Primary
State_UT

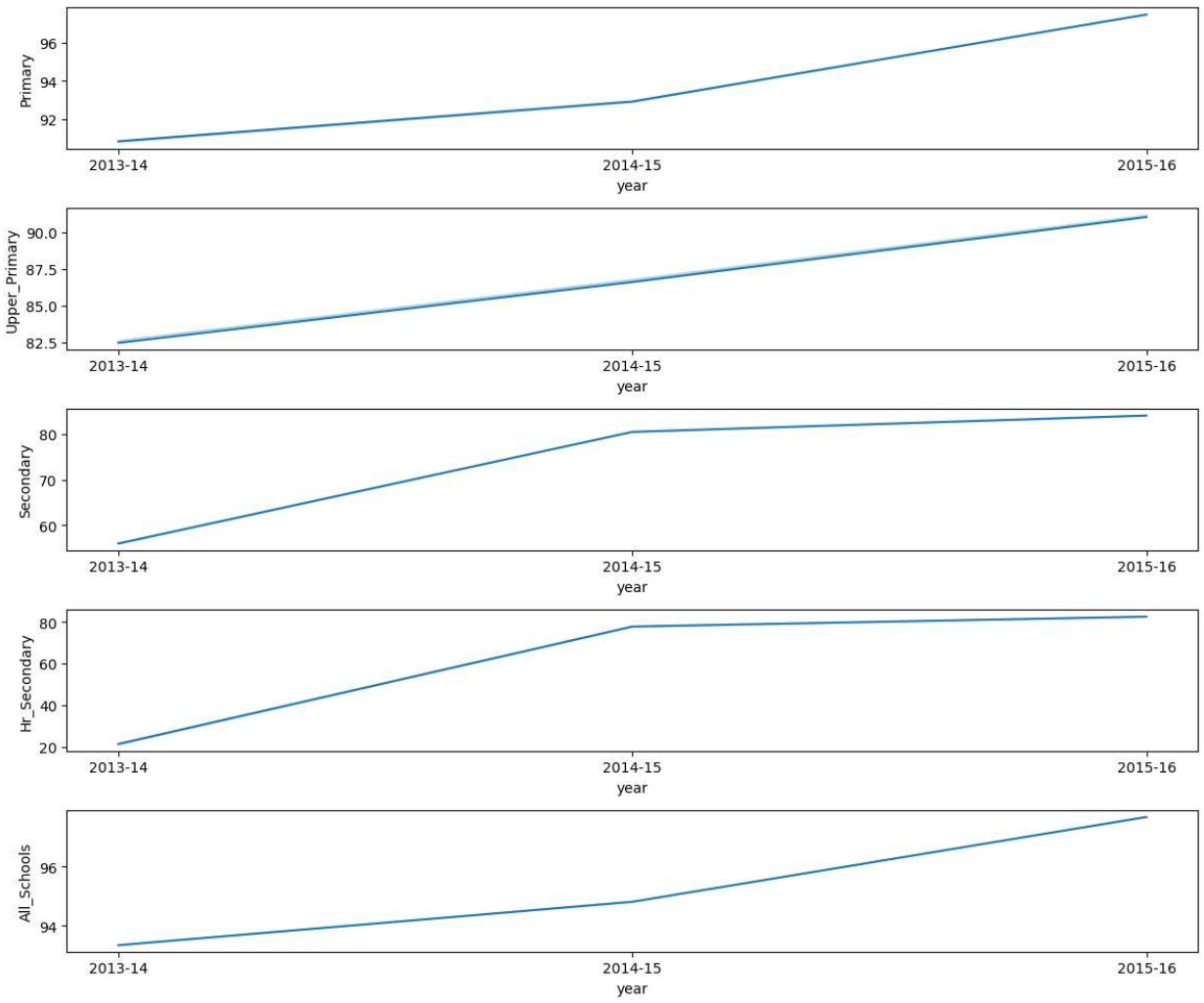

Secondary
State_UT


Hr_Secondary
State_UT


All_Schools



In [37]:
```
plt.figure(figsize=(12,10))

plt.subplot(5,1,1)
sns.lineplot(x='year',y='Primary',data=df,ci=True)

plt.subplot(5,1,2)
sns.lineplot(x='year',y='Upper_Primary',data=df,ci=True)

plt.subplot(5,1,3)
sns.lineplot(x='year',y='Secondary',data=df,ci=True)

plt.subplot(5,1,4)
sns.lineplot(x='year',y='Hr_Secondary',data=df,ci=True)

plt.subplot(5,1,5)
sns.lineplot(x='year',y='All_Schools',data=df,ci=True)

plt.tight_layout()
```



Summary of Analysis

In this Data Analysis Project, we have to analyze the data between 2013 to 2016 which show the availability of toilets in schools for both male and female students. To make an insights we had two datasets, Firstone is indicate for the boys toilets and Secondone indicate for the grils toilets. We check both the datasets one by one and merge both the datasets. Now, we have one single dataset which is easy to process. Our dataset is much good because they not contain any null values, missing values and etc. But, thier is one problem that the number of columns is more and they labeled in some confusing manner. So, we need to make a single column for a particular class like:- Primary, Upper Primary, Secondary, Hr. Secondary and All Schools (those who not categories in any class). After all Data Preprocessing, we need to do some exploratory data analysis (EDA) to make some usefull insights from our data. We done some Univariate Data Analysis and Bivariate Data Analysis for better understanding of data.

Conclusion

1. Data has high variation in Upper Primary, Seconday and Hr. Secondary Classes/Schools. This means data is spreadout in those classes.
2. Availability of toilet is increased in year 2014 to 2016 as compare to 2013-2014. In year 2013-2014 availability is 28% and In year 2014-2016 & 2015-2016 availability is 36%.
3. If we check the toilets avaialability Class/School Wise then, Primary Classes/Schools have more toilets available for students and Hr. Secondary Classes/Schools have less toilets available for students in comaparison to other classes/schools.
4. According to above barplots, the availability of the toilets is increasing continously in every year from 2013 to 2016.
5. According to barplots for States and Union Territories in India, All Indian States and Union Territories have good ratio of toilets availability for students in Primary and Upper Primary Classes/Schools. But, these ratio is decreased for the Secondary and Hr. Secondary Classes/Schools as compared to other two. All Schools has the combination of all the unlabeled classes/schools that's why it show the best ratio of toilet availability.
6. According to lineplot of year vs classes/schools for toilet availability, Primary and All Schools show increasing line after year 2014-2015 as compared to previous year duration, Upper Primary Classes/Schools is constant in all years duration and Secondary and Hr. Secondary Classes/Schools are showing decreasing line after year duration 2014-2015 as compared to previous year duration.

In [ ]: