Exploratory data analysis on GSOEP dataset

TEAM DETAILS

Name	SRN
Sushma Herakal. Tejashwini Hosamani Dolaanand Sallagundla	PES1UG19CS528 PES1UG29CS539 PES1UG19CS423

GERMAN SOCIO-ECONOMIC PANEL 1994-2002

Cross-section data for 675 14-year old children born between 1980 and 1988. The sample is taken from the German Socio-Economic Panel (GSOEP) for the years 1994 to 2002 to investigate the determinants of secondary school choice. It is a data frame containing 675 observations on 12 variables.

DESCRIPTION OF THE VARIABLES

- **School :** Child's secondary school level. There are three types of secondary education in Germany (Categorical variable)

1. Gymnasium - Aimed at students who plan to go for tertiary or university level education, gymnasium schools typically offer rigorous levels of academic education.
2. Realschule - It is the most common form of secondary education. Although not considered as prestigious as the gymnasium schools realschule still offers a highly academic environment with a range of subjects.
3. Hauptschule - It is a vocational school for students between the ages of 10-16. It is intended for students who will enter a trade or apprenticeship and aim to work in industrial sectors.

- **Birth year :** Child's year of birth (Discrete Variable)
- **Gender :** Factor indicating child's gender (Categorical Variable)
- **Kids :** Total number of kids in the household (Discrete Variable)

- **Parity :** Birth order (Discrete Variable)
- **Income :** Household income (Continuous Variable)
- **Size :** Household size (Discrete Variable)
- **State :** Factor indicating German federal state (Categorical Variable)
- **Marital :** Factor indicating mother’s marital status (Categorical Variable)
- **Meducation :** Mother’s educational level in years (Discrete Variable)
- **Memployment :** Factor indicating mother’s employment level (Full-time, Part-time or not working) (Categorical Variable)
- **Year :** Year of data collection (Discrete Variable)

MISSING VALUES

```
print(df.isnull().sum())
```

```
Unnamed: 0      0
school         21
birthyear      23
gender         20
kids           21
parity         19
income         20
size           20
state          21
marital        23
meducation     18
memployment    22
year           24
dtype: int64
```

Number of missing values in each column

```
missing_values = df.isnull().sum()
sum = 0
for value in missing_values:
    sum += value
missing_percent = (sum/(len(df)*12))*100
print(missing_percent)
```

```
3.111111111111111
```

Percentage of missing values

DATA CLEANING

Imputing numerical missing values

For the variables birth year and year, from the description of the dataset we knew that the children under consideration were 14 year olds, in order to calculate the missing birth year, 14 was deducted from the year. If the year was missing then 14 was added to the birth year. In case, both year and birth year were missing, we replaced the values with the median.

For the variable income, the missing values were replaced with the mean.

The rest of the numerical missing values were replaced with the median.

All the categorical missing values were replaced with the mode.

Code Snippet

```
for col in df:
  try:
    if df[col].dtype == int or df[col].dtype == float :
      if col == "birthyear":
        df[col].fillna(df['year'] - 14, inplace = True)
      elif col == "year":
        df[col].fillna(df['birthyear'] + 14, inplace = True)
      elif col == "income":
        df[col].fillna(df[col].mean(), inplace = True)
      else:
        df[col].fillna(round(df[col].median()), inplace = True)
    else:
      df[col].fillna(df[col].mode()[0], inplace = True)
    df[col].fillna(round(df[col].median()),inplace = True)
  except:
    pass
print(df.isnull().sum())
df.to_csv('/content/drive/My Drive/stats_project/GSOEP9402_new.csv', index = False)
```

	Unnamed: 0	school	birthyear	gender	kids	parity	income	size	state	marital	meducation	memployment	year
54	55	Gymnasium	1987.0	female	NaN	1.0	79437.00623	4.0	Niedersachsen	married	12.0	none	2001.0
55	56	Hauptschule	NaN	male	1.0	1.0	67575.01642	3.0	Niedersachsen	married	9.0	fulltime	2002.0
56	57	Hauptschule	1980.0	male	2.0	2.0	68578.18702	4.0	Niedersachsen	married	11.5	parttime	NaN
57	58	Realschule	1983.0	female	2.0	2.0	73933.85018	4.0	Niedersachsen	married	10.5	parttime	1997.0
58	59	Hauptschule	1980.0	female	2.0	1.0	60686.95237	4.0	Niedersachsen	married	10.5	parttime	1994.0
59	60	Realschule	1986.0	female	1.0	1.0	76634.98738	3.0	Niedersachsen	married	9.0	none	2000.0
60	61	NaN	1982.0	male	2.0	1.0	59684.86808	4.0	Niedersachsen	married	11.5	none	1996.0
61	62	Realschule	1985.0	male	2.0	2.0	65546.19029	4.0	Niedersachsen	married	11.5	none	1999.0
62	63	Realschule	1983.0	male	1.0	1.0	61663.97618	3.0	Niedersachsen	married	9.0	none	1997.0
63	64	Realschule	1981.0	female	2.0	1.0	103631.37690	4.0	Niedersachsen	married	11.5	none	1995.0
64	65	Gymnasium	1982.0	female	2.0	2.0	101866.63680	4.0	Niedersachsen	married	11.5	parttime	1996.0
65	66	Gymnasium	1980.0	female	2.0	1.0	86824.98824	4.0	Niedersachsen	married	11.5	parttime	1994.0
66	67	Gymnasium	1984.0	male	2.0	2.0	80931.79466	4.0	Niedersachsen	married	11.5	parttime	1998.0
67	68	Realschule	1983.0	female	2.0	1.0	113734.01220	4.0	NaN	married	11.5	parttime	1997.0
68	69	Realschule	1985.0	female	2.0	2.0	103880.53130	4.0	Niedersachsen	married	11.5	parttime	1999.0
69	70	Gymnasium	NaN	female	1.0	1.0	80802.46293	3.0	Niedersachsen	married	18.0	fulltime	2000.0

Before replacing the missing values

	Unnamed: 0	school	birthyear	gender	kids	parity	income	size	state	marital	meducation	memployment	year
54	55	Gymnasium	1987.0	female	2.0	1.0	79437.00623	4.0	Niedersachsen	married	12.0	none	2001.0
55	56	Hauptschule	1988.0	male	1.0	1.0	67575.01642	3.0	Niedersachsen	married	9.0	fulltime	2002.0
56	57	Hauptschule	1980.0	male	2.0	2.0	68578.18702	4.0	Niedersachsen	married	11.5	parttime	1994.0
57	58	Realschule	1983.0	female	2.0	2.0	73933.85018	4.0	Niedersachsen	married	10.5	parttime	1997.0
58	59	Hauptschule	1980.0	female	2.0	1.0	60686.95237	4.0	Niedersachsen	married	10.5	parttime	1994.0
59	60	Realschule	1986.0	female	1.0	1.0	76634.98738	3.0	Niedersachsen	married	9.0	none	2000.0
60	61	Gymnasium	1982.0	male	2.0	1.0	59684.86808	4.0	Niedersachsen	married	11.5	none	1996.0
61	62	Realschule	1985.0	male	2.0	2.0	65546.19029	4.0	Niedersachsen	married	11.5	none	1999.0
62	63	Realschule	1983.0	male	1.0	1.0	61663.97618	3.0	Niedersachsen	married	9.0	none	1997.0
63	64	Realschule	1981.0	female	2.0	1.0	103631.37690	4.0	Niedersachsen	married	11.5	none	1995.0
64	65	Gymnasium	1982.0	female	2.0	2.0	101866.63680	4.0	Niedersachsen	married	11.5	parttime	1996.0
65	66	Gymnasium	1980.0	female	2.0	1.0	86824.98824	4.0	Niedersachsen	married	11.5	parttime	1994.0
66	67	Gymnasium	1984.0	male	2.0	2.0	80931.79466	4.0	Niedersachsen	married	11.5	parttime	1998.0
67	68	Realschule	1983.0	female	2.0	1.0	113734.01220	4.0	Nordrhein-Westfalen	married	11.5	parttime	1997.0
68	69	Realschule	1985.0	female	2.0	2.0	103880.53130	4.0	Niedersachsen	married	11.5	parttime	1999.0
69	70	Gymnasium	1986.0	female	1.0	1.0	80802.46293	3.0	Niedersachsen	married	18.0	fulltime	2000.0

After replacing the missing values

Fixing inconsistent capitalization

Only the variable ‘state’ had inconsistent capitalisation. Code snippet to fix inconsistent capitalisation

```
df = pd.read_csv('/content/drive/My Drive/stats_project/GSOEP9402 _Cap.csv')
print(df["state"][119:137])
symbols = [" ", "-","/"]
df['state'] = df['state'].str.lower().str.title()
df.to_csv('/content/drive/My Drive/stats_project/cap.csv', index = False)
df1 = pd.read_csv('/content/drive/My Drive/stats_project/cap.csv')
df1["state"][119:137]

```

Before

After


119 Niedersachsen
120 Niedersachsen
121 Niedersachsen
122 Bremen
123 Bremen
124 breMen
125 Bremen
126 Bremen
127 Bremen
128 Schleswig-hoLstein
129 Nordrhein-Westfalen
130 Nordrhein-Westfalen
131 Nordrhein-Westfalen
132 Nordrhein-Westfalen
133 Nordrhein-Westfalen
134 nordrhein-WestfalEn
135 Nordrhein-Westfalen
136 Baden wuertteMberg


119 Niedersachsen
120 Niedersachsen
121 Niedersachsen
122 Bremen
123 Bremen
124 Bremen
125 Bremen
126 Bremen
127 Bremen
128 Schleswig-Holstein
129 Nordrhein-Westfalen
130 Nordrhein-Westfalen
131 Nordrhein-Westfalen
132 Nordrhein-Westfalen
133 Nordrhein-Westfalen
134 Nordrhein-Westfalen
135 Nordrhein-Westfalen
136 Baden Wuerttemberg
Name: state, dtype: object

GRAPH VISUALIZATION

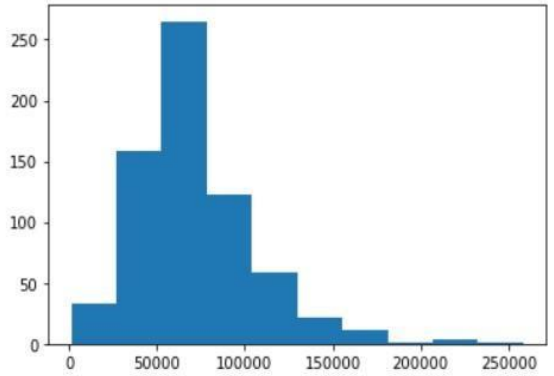
The three types of graphs we plotted are

- Histogram
- Barchart
- Boxplot

There were a few outliers in the income column that were handled using the interquartile range method.

Handling the outliers

In [5]:
```
plt.hist(df['income'])
plt.show()
```

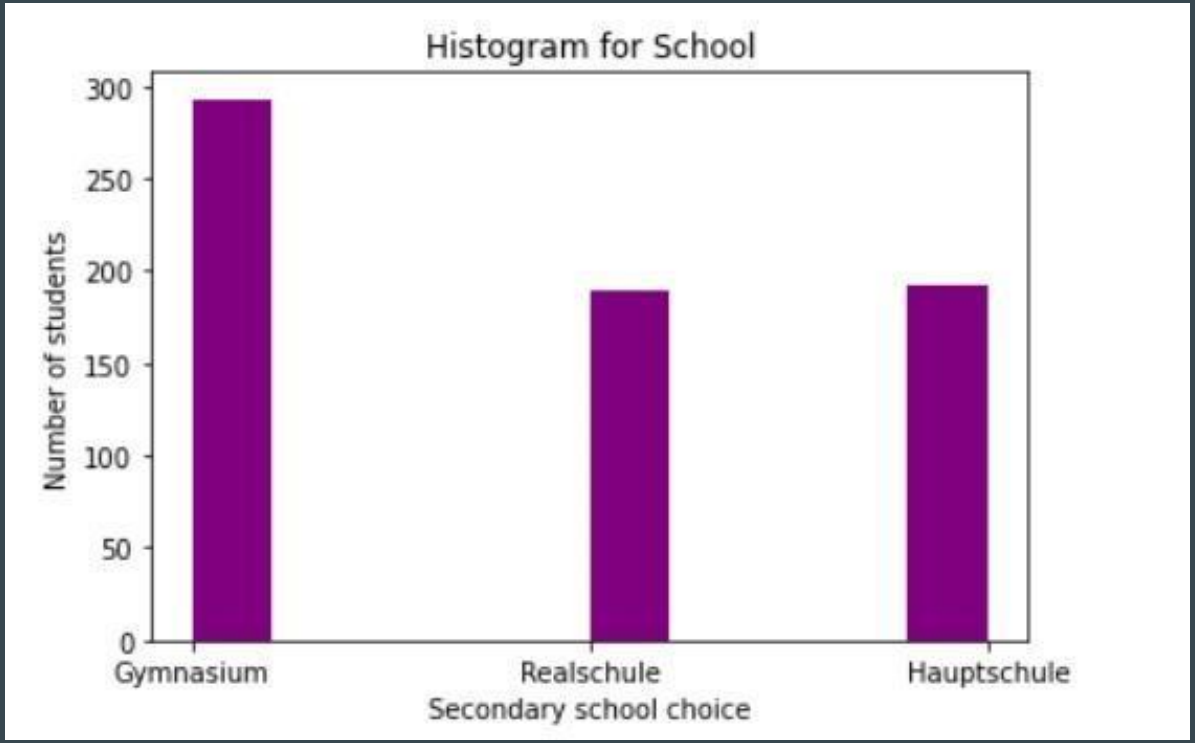


In [6]:
```
lower_bound = 0.1
upper_bound = 0.95
outliers = df.income.quantile([lower_bound, upper_bound])
print(outliers)
```

```
0.10     36329.095606
0.95    131283.151540
Name: income, dtype: float64
```

In [15]:
```
accepted_income = (df.income.values > outliers.loc[lower_bound]) & (df.income.values < outliers.loc[upper_bound])
median = np.median(df.income[accepted_income])
rejected_income = ~(accepted_income)
df.income[rejected_income] = median     #replace outliers with median of accepted income
median
```

Histogram

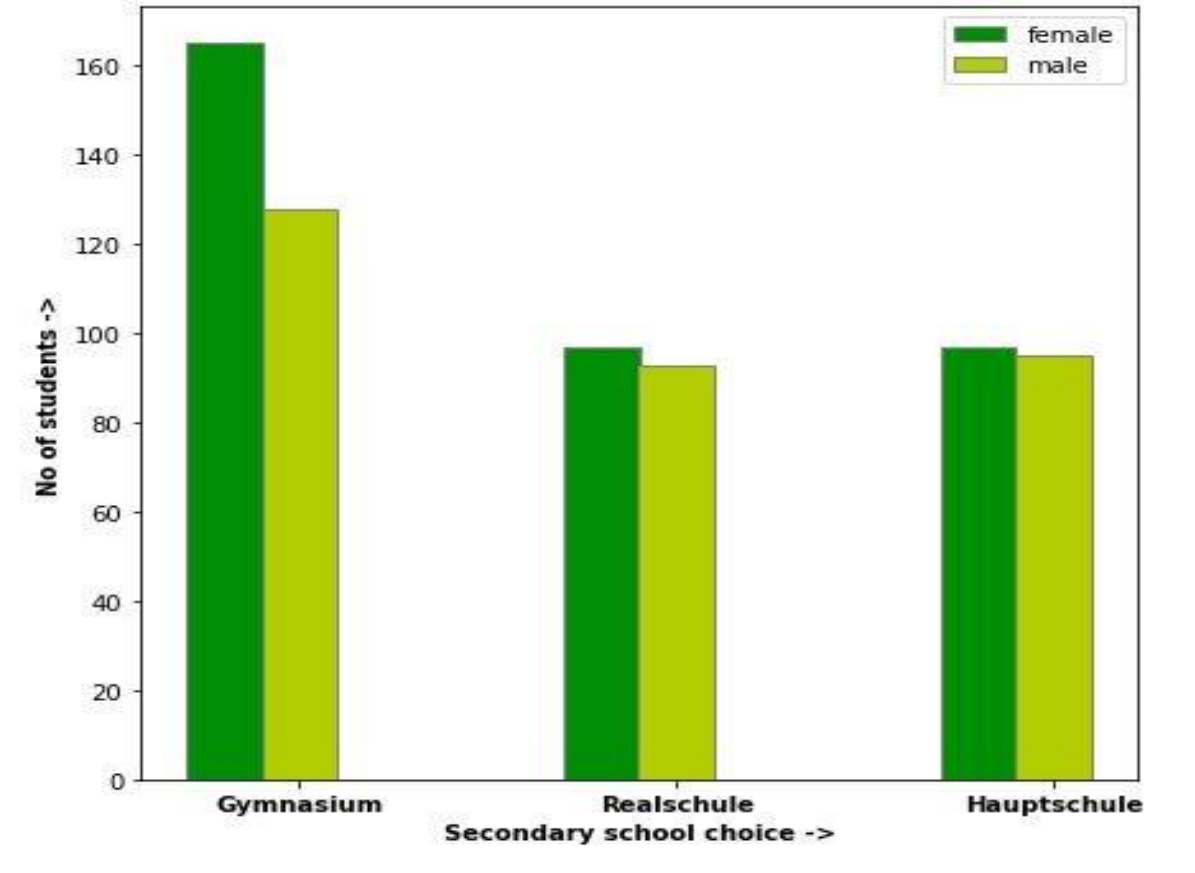


Inference drawn from the histogram:

Most of the students, (approx 300 of them) opted for Gymnasium as their secondary school choice.

Realschule and Hauptschule is equally popular among the students.

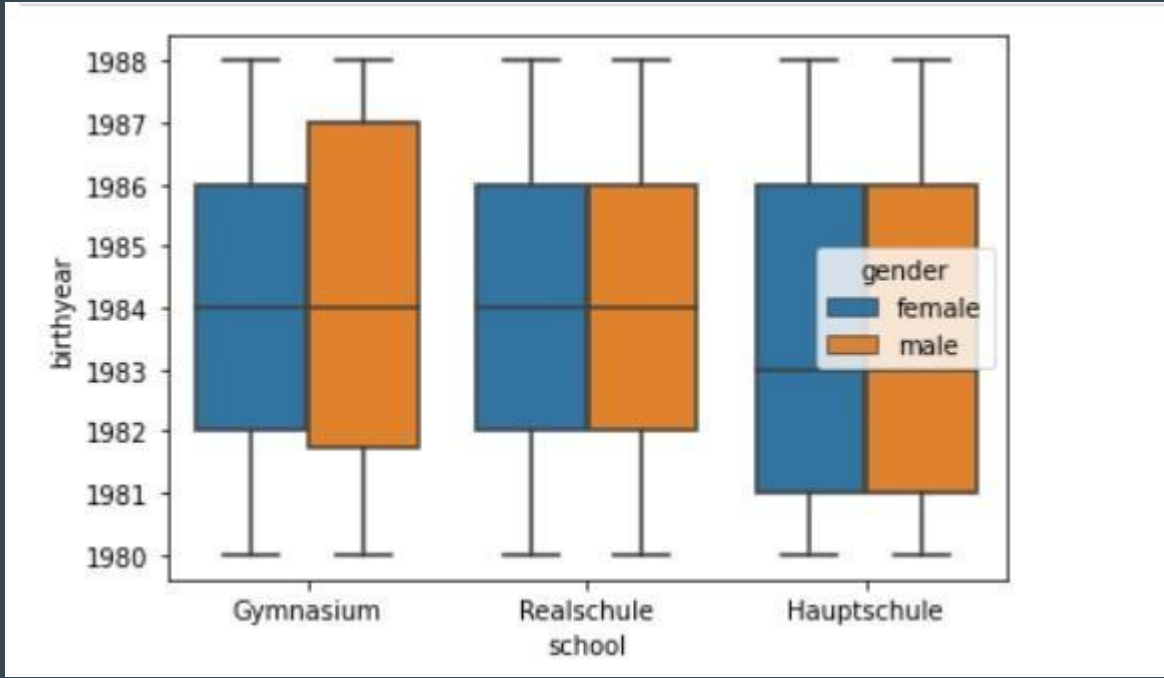
Bar Chart



Inference drawn from barchart:

The number of female students in gymnasium(around 165) is more than the male students(around 125). While they are almost equally distributed in Realschule and Hauptschule.

Boxplot



Inference drawn from the boxplot:

Realschule and Gymnasium have most of their students born in 1984, whereas majority of Hauptschule students were born in 1983.

Gymnasium is more popular among male students than female but Realschule and Hauptschule have a better male to female gender ratio.

NORMALIZATION AND STANDARDIZATION

What is Normalization?

Normalisation of ratings means adjusting values measured on different scales to a notionally common scale, often prior to averaging. In more complicated cases, normalisation may refer to more sophisticated adjustments where the intention is to bring the entire probability distributions of adjusted values into alignment.

Why is it required?

The aim of Normalization is to change the values in the dataset to a common scale, without changing the differences in the range of values.

How does it affect the dataset?

Normalization only changes the numeric values of mean, median and mode without affecting the overall behaviour of the dataset.

NORMALIZATION AND STANDARDIZATION

There are many Normalization techniques.

- The maximum absolute scaling
- The min-max feature scaling
- Z-score method

We have used the Z-score method in which each numerical value is replaced by Z score.

$$Z = (x - \mu) / \sigma$$

```
columns=['income','kids','parity','meducation','size','year','birthyear']
df_std = df.copy()

for cols in columns:
    df_std[cols] = (df_std[cols] - df_std[cols].mean())/(df_std[cols].std()) # z score

print(df_std)
```

Mean and Variance before Normalisation

Mean and Variance after Normalisation

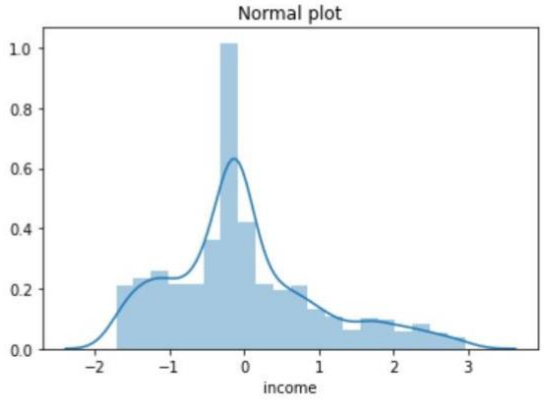
```
mean of birthyear : 1983.962962962963
variance of birthyear : 6.917023848774609
mean of kids : 2.4962962962962965
variance of kids : 1.1375975381910217
mean of parity : 1.7614814814814814
variance of parity : 0.7130541817782183
mean of income : 71295.24029567941
variance of income : 1074506152.1675084
mean of size : 4.2444444444444445
variance of size : 1.2561819980217583
mean of meducation : 11.42962962962963
variance of meducation : 4.676420485767675
mean of year : 1997.962962962963
variance of year : 6.917023848774609
```

```
In [13]: abs(round(df_std.mean()))

Out[13]: Unnamed: 0      337.0
         Unnamed: 0.1    338.0
         birthyear         0.0
         kids              0.0
         parity            0.0
         income            0.0
         size              0.0
         meducation        0.0
         year              0.0
         dtype: float64

In [14]: df_std.var()

Out[14]: Unnamed: 0      38025.0
         Unnamed: 0.1    38025.0
         birthyear           1.0
         kids                1.0
         parity              1.0
         income              1.0
         size                1.0
         meducation          1.0
         year                1.0
         dtype: float64
```

NORMALIZED GRAPHS

- INCOME

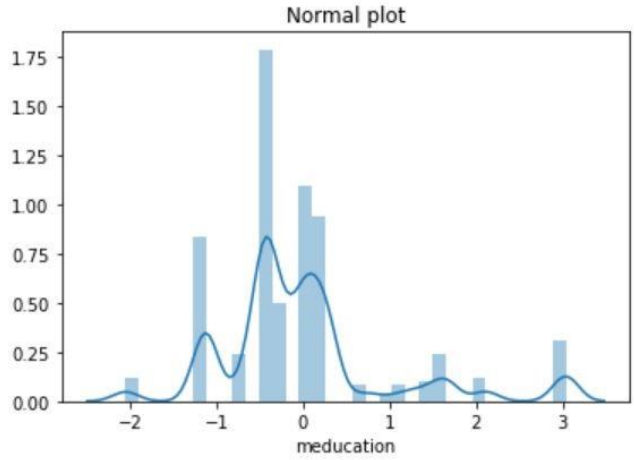
```
In [7]: sns.distplot(df_std['income'])
        plt.title("Normal plot")
        plt.figure()

Out[7]: <Figure size 432x288 with 0 Axes>
```



NORMALIZED GRAPHS

- MEDUCATION

```
In [6]: sns.distplot(df_std['meducation'])
        plt.title("Normal plot")
        plt.figure()

Out[6]: <Figure size 432x288 with 0 Axes>
```

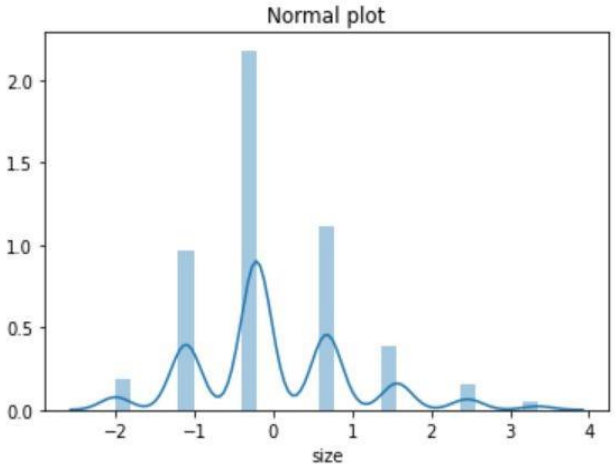


NORMALIZED GRAPHS

- SIZE

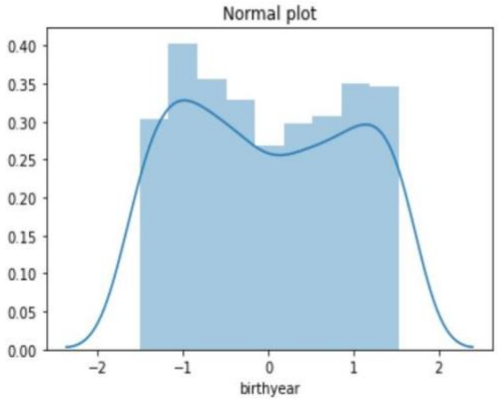
NORMALIZED GRAPHS

```
In [10]: sns.distplot(df_std['size'])
         plt.title("Normal plot")
         plt.figure()

Out[10]: <Figure size 432x288 with 0 Axes>
```



- BIRTH YEAR

```
In [9]: sns.distplot(df_std['birthyear'])
        plt.title("Normal plot")
        plt.figure()

Out[9]: <Figure size 432x288 with 0 Axes>
```




```
<Figure size 432x288 with 0 Axes>
```

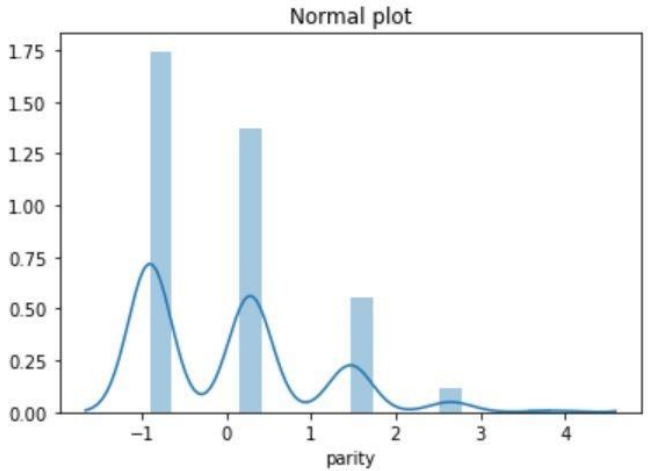
NORMALIZED GRAPHS

NORMALIZED GRAPHS

- PARITY

In [9]:
```
sns.distplot(df_std['parity'])
plt.title("Normal plot")
plt.figure()
```

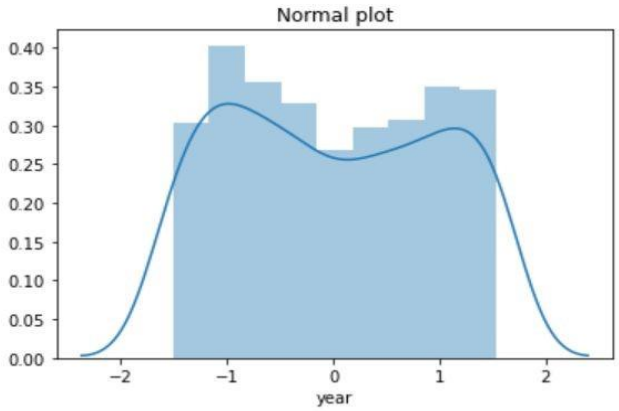
Out[9]: <Figure size 432x288 with 0 Axes>



NORMALIZED GRAPHS

- YEAR

```
In [11]: sns.distplot(df_std['year'])
         plt.title("Normal plot")
         plt.figure()

Out[11]: <Figure size 432x288 with 0 Axes>
```




```
<Figure size 432x288 with 0 Axes>
```

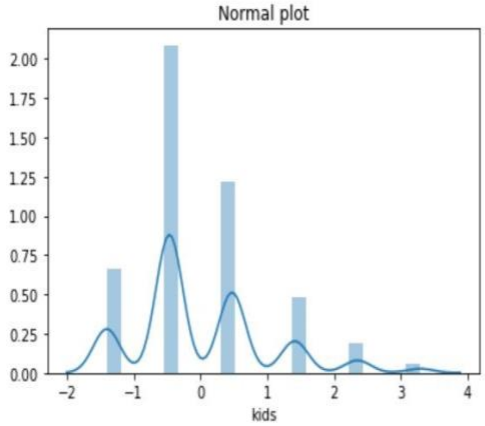
NORMALIZED GRAPHS

- KIDS

NORMALIZED GRAPHS

```
In [5]: sns.distplot(df_std['kids'])
        plt.title("Normal plot")
        plt.figure()
```

```
Out[5]: <Figure size 432x288 with 0 Axes>
```



HYPOTHESIS TESTING

Q. At 5% level of significance, can it be concluded that the mean population income is less than 70000?

Research Hypothesis: Population mean of income is less than 70000.

Null Hypothesis :

$H_0 : \mu \geq 70000$

Alternate Hypothesis :

$H_1 : \mu < 70000$ (Left tailed test)

HYPOTHESIS TESTING

```
import scipy.stats as st

sampdata=df['income'].sample(40)
N=40
meansampdata=sampdata.mean()

hypmean=70000
standpopulation=np.std(df['income'])
```

```
z=(meansampdata-hypmean)/(standpopulation/math.sqrt(N))
pval = st.norm.cdf(z)

if pval<=0.05:
    print("reject Null hypothesis")
else:
    print("H0 is plausible or Fail to reject Null hypothesis")
print('Standard deviation of the population is ',standpopulation)
print('Mean of the sample is ',meansampdata)
print('Z score is ',z)
print('P value is ',pval)
```

```
H0 is plausible or Fail to reject Null hypothesis
Standard deviation of the population is  20198.570549869466
Mean of the sample is  72981.44702400001
Z score is  0.9335475790915961
P value is  0.8247313416990405
```

CORRELATION

```
corr = df.corr()
corr
```

	birthyear	kids	parity	income	size	meducation	year
birthyear	1.000000	0.023488	0.046121	0.183256	0.003579	0.081063	1.000000
kids	0.023488	1.000000	0.543465	0.084092	0.801912	-0.078431	0.023488
parity	0.046121	0.543465	1.000000	0.133153	0.309387	-0.066893	0.046121
income	0.183256	0.084092	0.133153	1.000000	0.159274	0.249289	0.183256
size	0.003579	0.801912	0.309387	0.159274	1.000000	-0.012175	0.003579
meducation	0.081063	-0.078431	-0.066893	0.249289	-0.012175	1.000000	0.081063
year	1.000000	0.023488	0.046121	0.183256	0.003579	0.081063	1.000000

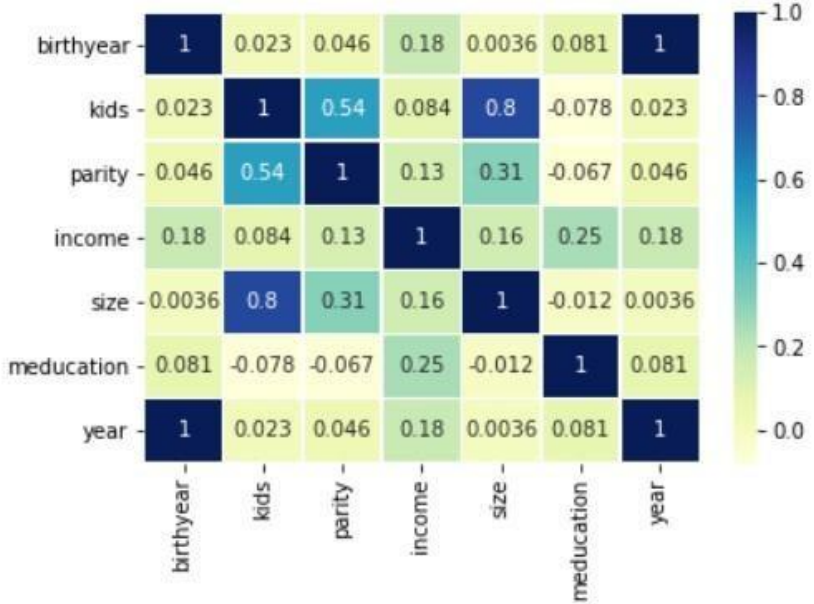
To find the correlation between the variables:

- A matrix showing the correlation coefficients for all variables was constructed.
- A heatmap was plotted for the same followed by scatterplots.

Heatmap

```
sns.heatmap(corr, annot = True, cmap = "YlGnBu", linewidths = 0.5)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x20a5c6a3ac0>
```



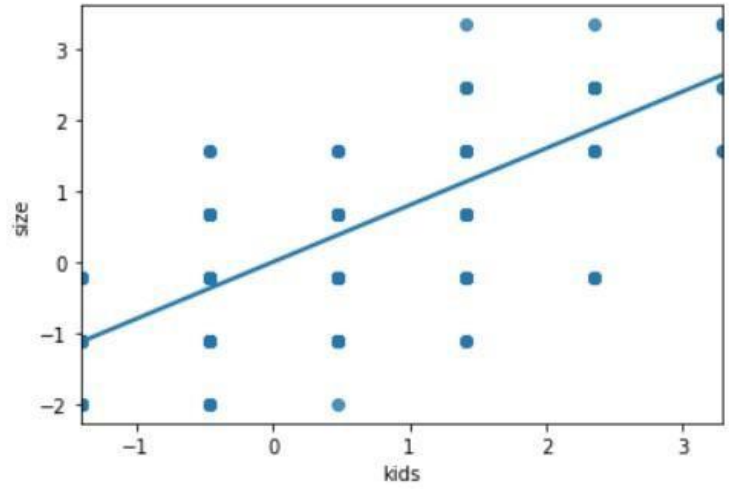
From the heatmap, we can observe that,

- There is a strong positive correlation between number of kids and size of family.
- There is a low negative correlation between mother's education and size of the family.

Inferences

```
sns.regplot(df['kids'], df['size'], ci = None)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1da6a4c3880>
```



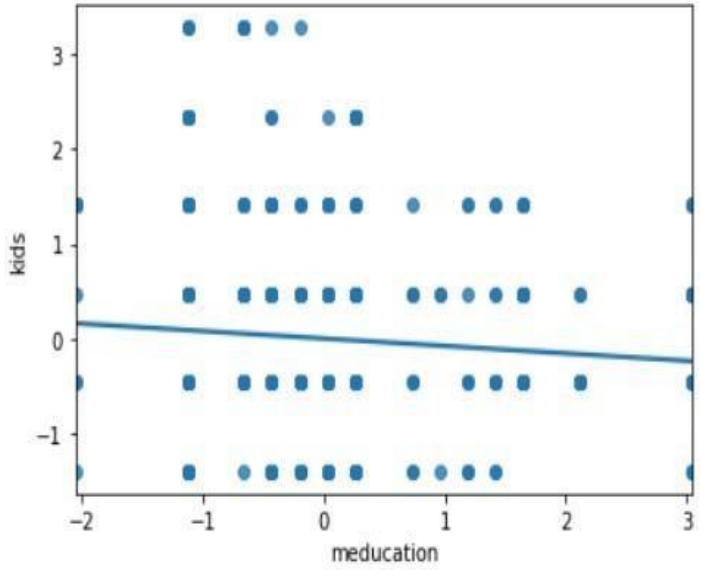
The variables kids and size have the correlation coefficient of 0.801912, which means they have a strong positive correlation.

As the number of kids increases, size of the family increases linearly.

Size of the family depends on the number of kids.

```
sns.regplot(df['meducation'], df['kids'], ci = None)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1da6a474190>
```

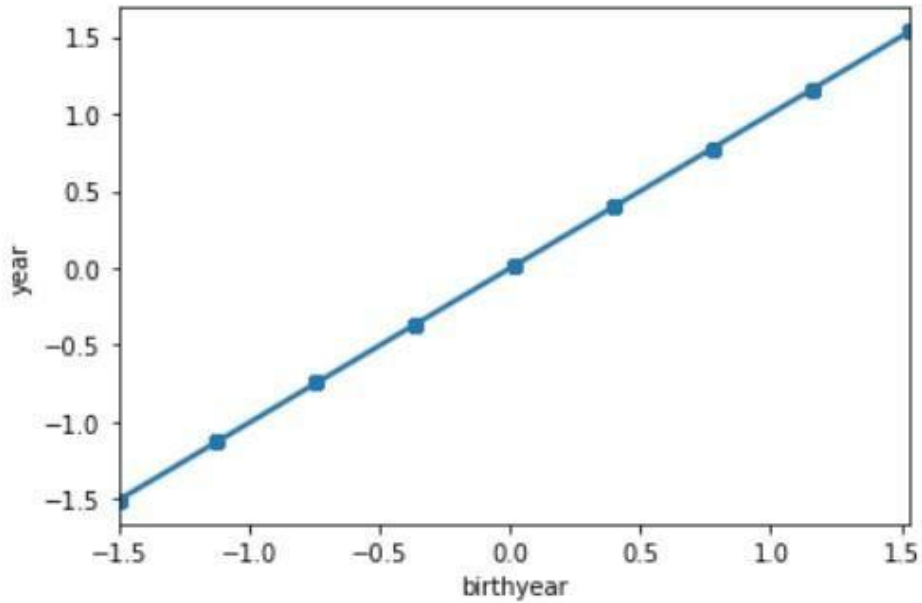


The variable ‘meducation’ is negatively correlated to the variable kids with a correlation factor of -0.078431.

As the mother’s education increases the number of kids in the family is observed to decrease but since the value of ρ is very close to zero and the graph is almost parallel to x-axis we may conclude that the variables may be independent.

```
sns.regplot(df['birthyear'], df['year'], ci = None)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x20a5c53d370>
```



From the scatterplot, it is very clear that the variables year and birth year have a strong positive correlation with the value of ρ being 1.

It is true since all the students under consideration were 14 year olds.

THANK YOU