CS 4821 - Laura Brown

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sb
%matplotlib inline
```

Census Data

(a) Variable Definitions

age is the age of an individual as reported on the 1994 census; The value is an integer that represents years.

workclass is an individuals type of employment as reported on the 1994 census; The value is a string that enumerates employment types.

fnlwgt is the final weight of a row of data in the 1994 census; The value is an integer that represents the number of people that the row represents.

education is the level of education that an individual has received as reported on the 1994 census; The value is a string that enumerates education level.

education-num is the level of education that an individual has received as reported on the 1994 census; The value is an integer that represents education level with larger values being more educated.

marital-status is the individuals marital status as reported on the 1994 census; The value is a string that enumerates marital statuses.

occupation is the area of work that an individual occupies as reported on the 1994 census; The value is a string that enumerates fields of work.

relationship is an individuals place in their family as reported on the 1994 census; The value is a string that enumerates types of family members.

race is an individuals race as reported on the 1994 census; The value is a string that enumerates races.

sex is an individuals sex as reported on the 1994 census; The value is a string that represents the boolean options male or female.

capital-gain is an individuals capital gains as reported on the 1994 census; The value is an integer that represents dollars.

capital-loss is an individuals capital loss as reported on the 1994 census; The value is an integer that represents dollars.

hours-per-week is the amount of hours an individual works each week as reported on the 1994 census; The value is an integer that represents hours.

native-country is an individuals native country as reported on the 1994 census; The value is a string that enumerates countries.

(b) Missing Data

(i) Missing data is represented by a ? in the place where the data would normally be.

(ii)

In [2]:

```
file1 = open('adult.data.txt', 'r')
lines = file1.readlines()
total = 0
cols = ["age", "workclass", "fnlwgt", "education", "education-num", "marital-sta
missing = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
for line in lines:
    line = line.rstrip()
    if line:
        total += 1
        exploded = line.split(", ")
        index = 0
        while index < len(exploded):
            if exploded[index] == '?':
                missing[index] += 1
            index += 1
col = 0
while col < len(cols):
    print("{0}:\t{1}% missing".format(cols[col],round(missing[col]/total,4)))
    col += 1
```

```
age:    0.0% missing
workclass:      0.0564% missing
fnlwgt: 0.0% missing
education:      0.0% missing
education-num:  0.0% missing
marital-status: 0.0% missing
occupation:     0.0566% missing
relationship:   0.0% missing
race:   0.0% missing
sex:    0.0% missing
capital-gain:   0.0% missing
capital-loss:   0.0% missing
hours-per-week: 0.0% missing
native-country: 0.0179% missing
```

(c) Variable Types

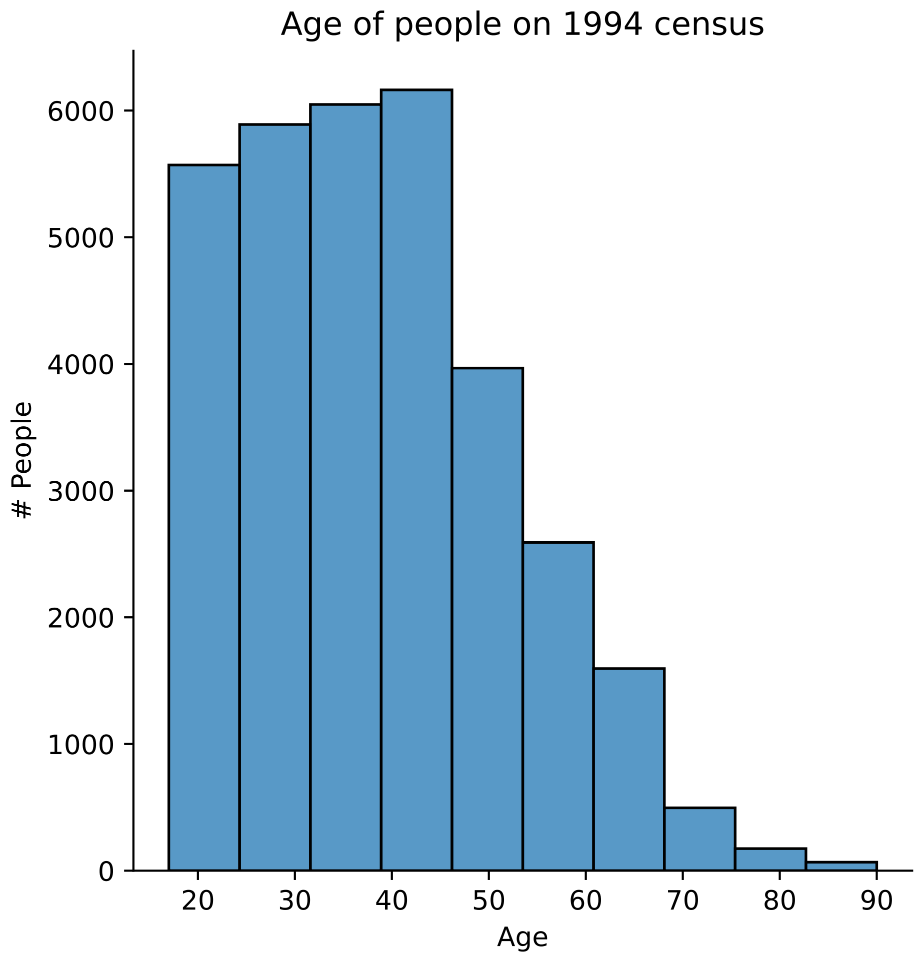
Numerical: age, fnlwgt, education-num, capital-gain, capital-loss, hours-per-week
Categorical: workclass, education, marital-status, occupation, relationship, race, sex, native-country

(d) Numeric Data

(i.1)

In [3]:

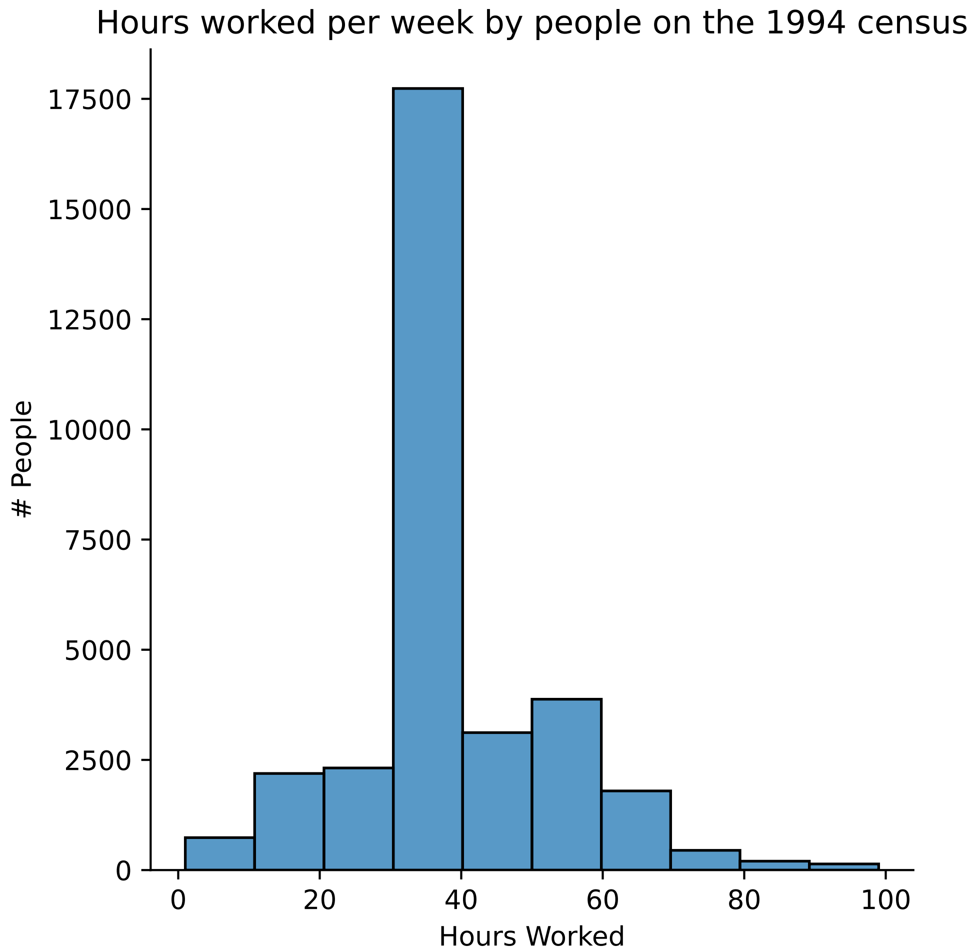
```
table = pd.read_csv("adult.data.txt", sep=", ", header=None, engine='python')
table.columns = ["age", "workclass", "fnlwgt", "education", "education-num", "ma
plot = sb.displot(data=table, x="age", bins=10)
plot.set(xlabel="Age", ylabel="# People")
plt.title("Age of people on 1994 census")
plt.show()
```



(i.2)

In [4]:

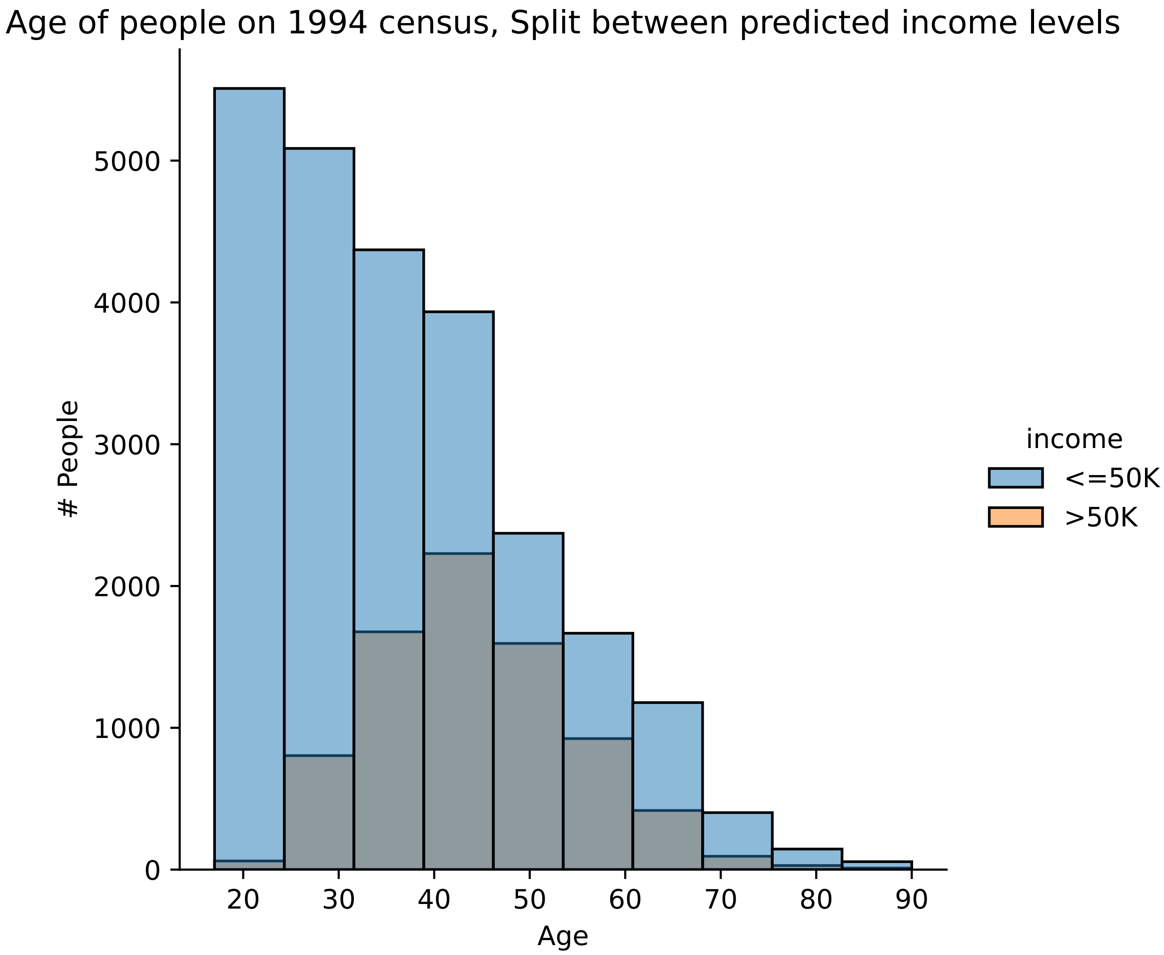
```
plot = sb.displot(data=table, x="hours-per-week", bins=10)
plot.set(xlabel="Hours Worked", ylabel="# People")
plt.title("Hours worked per week by people on the 1994 census")
plt.show()
```



(ii.1)

In [5]:

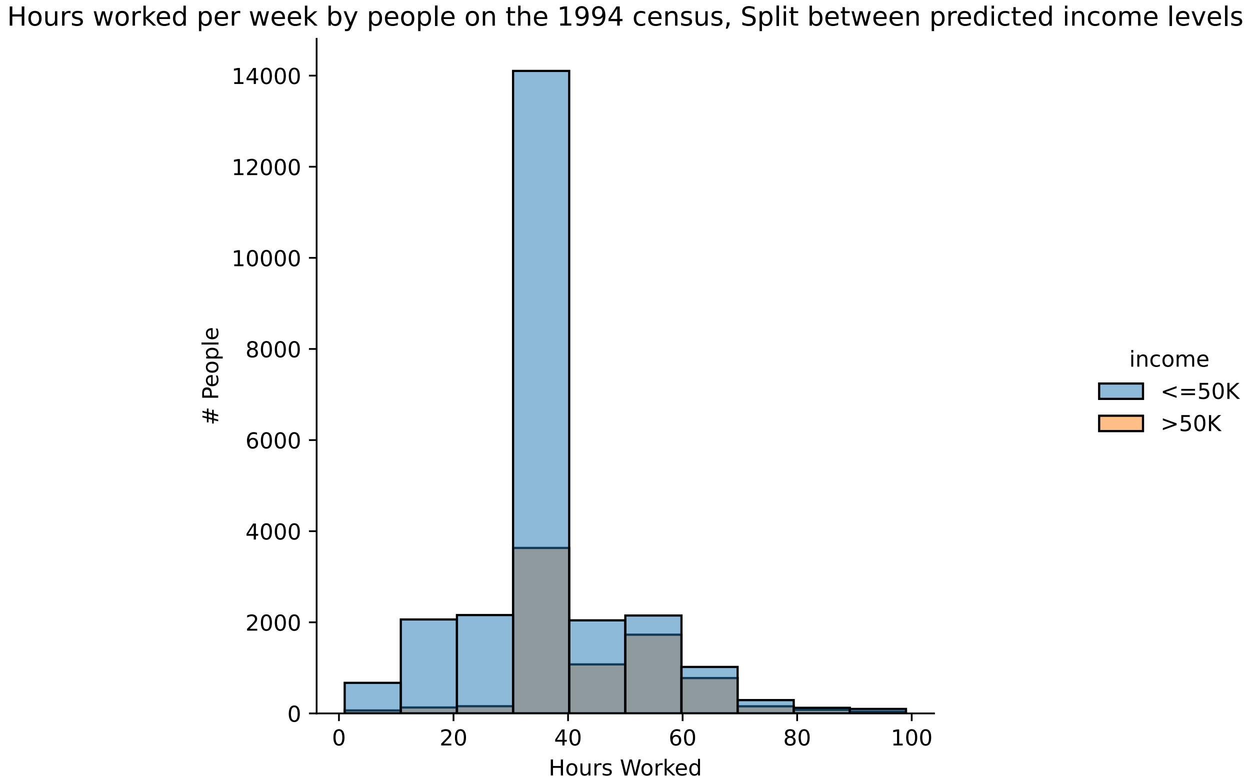
```
plot = sb.displot(data=table, x="age", hue="income", bins=10)
plot.set(xlabel="Age", ylabel="# People")
plt.title("Age of people on 1994 census, Split between predicted income levels")
plt.show()
```



(ii.2)

In [6]:

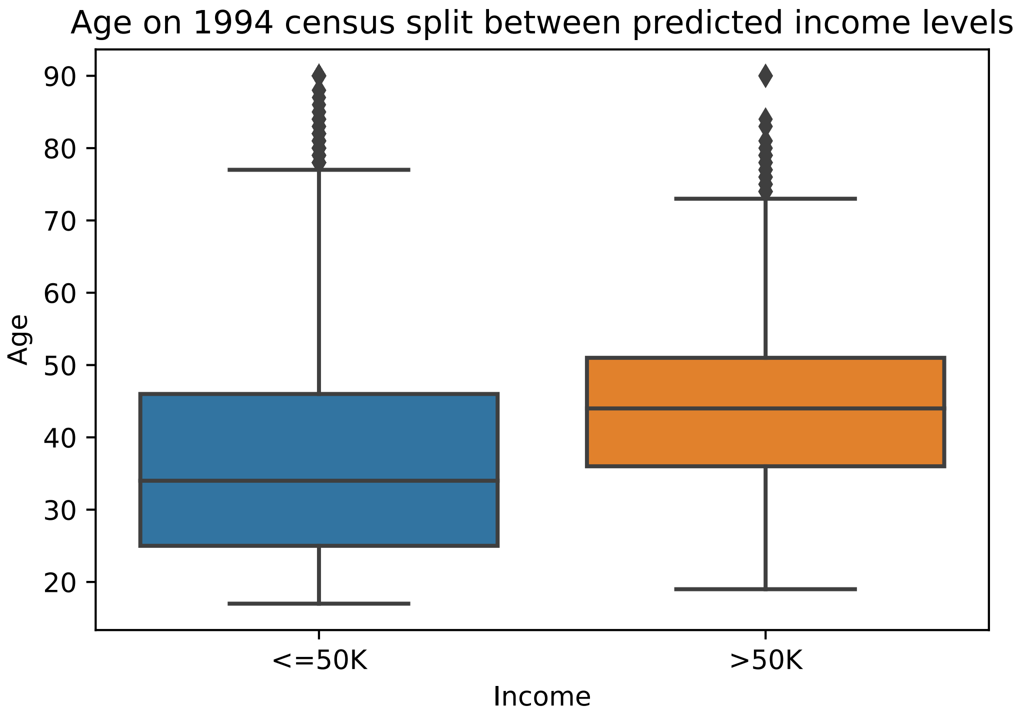
```
plot = sb.displot(data=table, x="hours-per-week", hue="income", bins=10)
plot.set(xlabel="Hours Worked", ylabel="# People")
plt.title("Hours worked per week by people on the 1994 census, Split between pre
plt.show()
```



(iii.1)

In [7]:

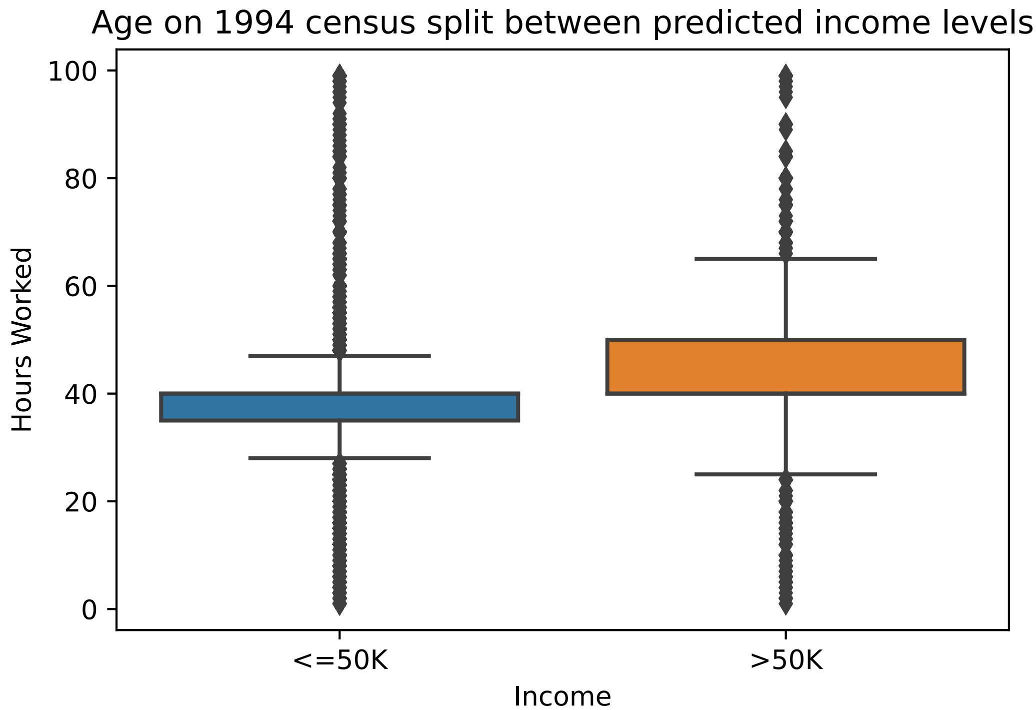
```
plot = sb.boxplot(data=table, y="age", x="income")
plot.set(xlabel="Income", ylabel="Age")
plt.title("Age on 1994 census split between predicted income levels")
plt.show()
```



(iii.2)

In [8]:

```
plot = sb.boxplot(data=table, y="hours-per-week", x="income")
plot.set(xlabel="Income", ylabel="Hours Worked")
plt.title("Age on 1994 census split between predicted income levels")
plt.show()
```



(iv)

Summary of data

All data gathered from 1994 census

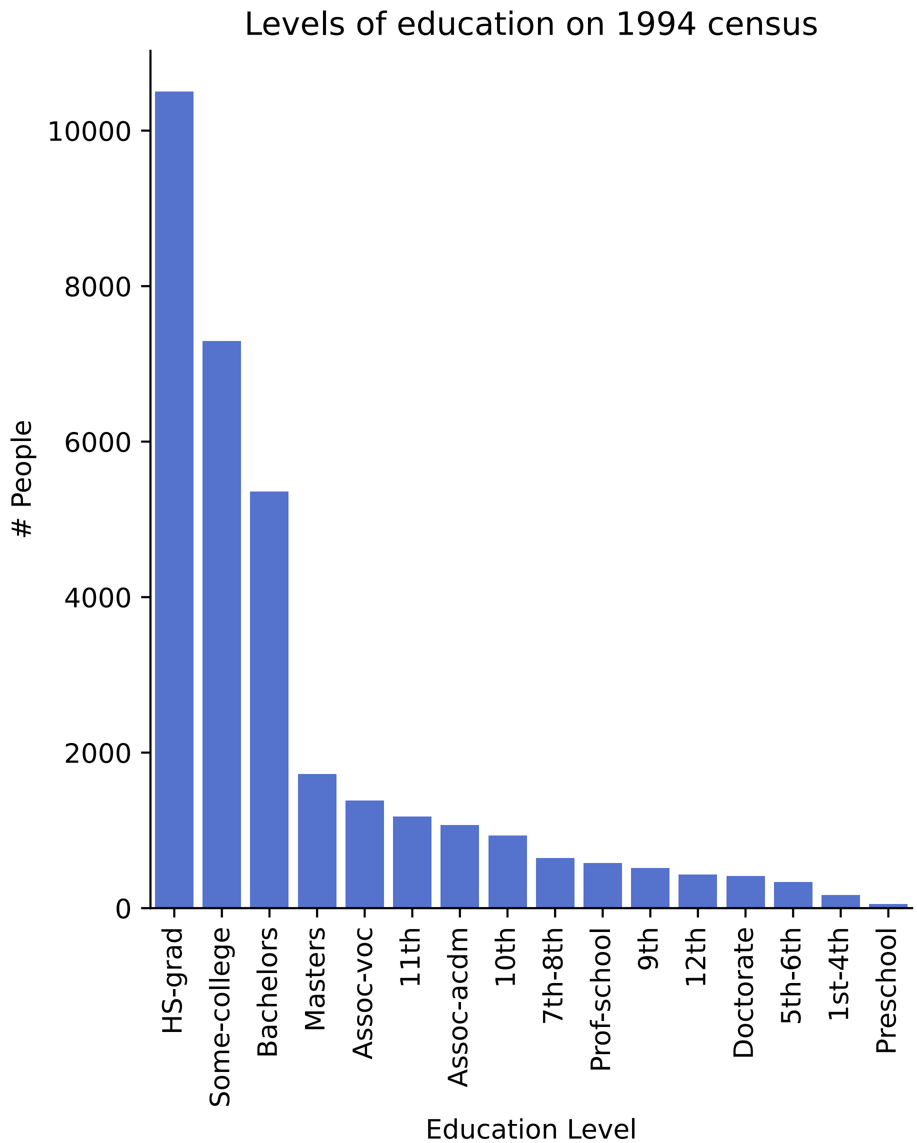
1. The majority of people were under 50
2. The majority of people worked 40 hours a week
3. People that made over 50k were likely to be older than those that made under 50k
4. People that made over 50k were more likely to work over 40 hours a week

(e) Categorical Data

(i.1)

In [9]:

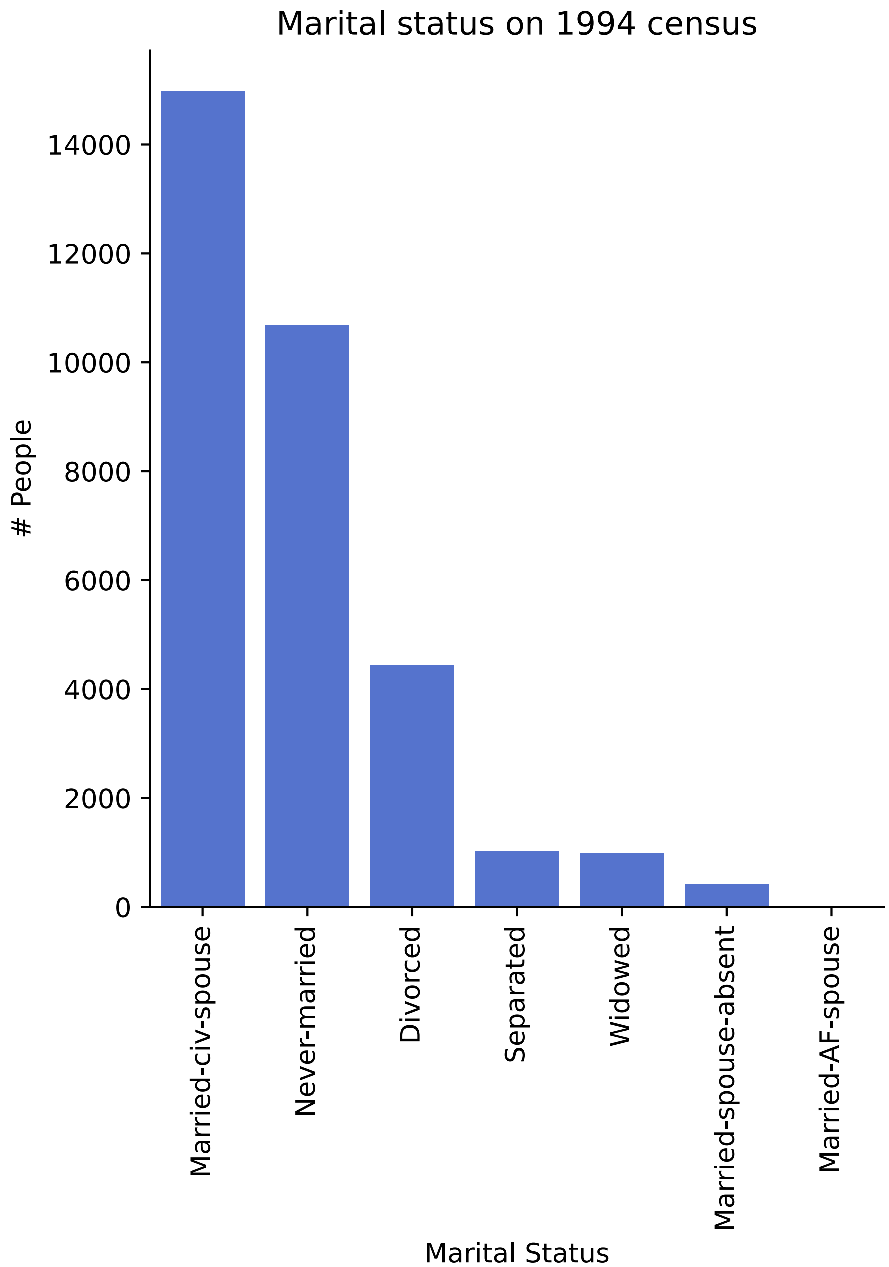
```
plot = sb.catplot(kind="count", data=table, x="education", order=table["educatio
plot.set_xticklabels(rotation=90)
plt.xlabel("Education Level")
plt.ylabel("# People")
plt.title("Levels of education on 1994 census")
plt.show()
```



(i.2)

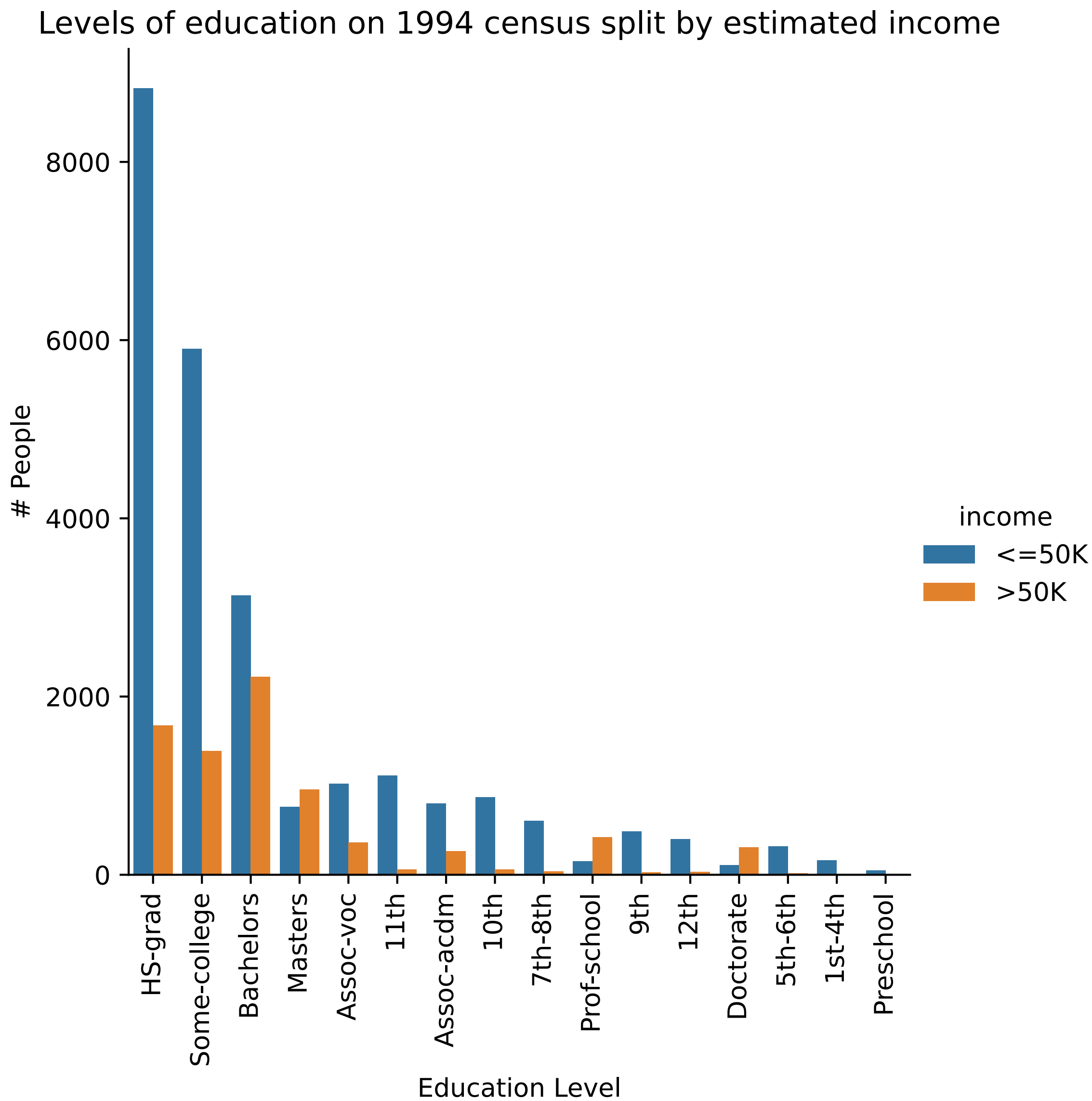
In [10]:

```
plot = sb.catplot(kind="count", data=table, x="marital-status", order=table["mar
plot.set_xticklabels(rotation=90)
plt.xlabel("Marital Status")
plt.ylabel("# People")
plt.title("Marital status on 1994 census")
plt.show()
```



(ii.1)

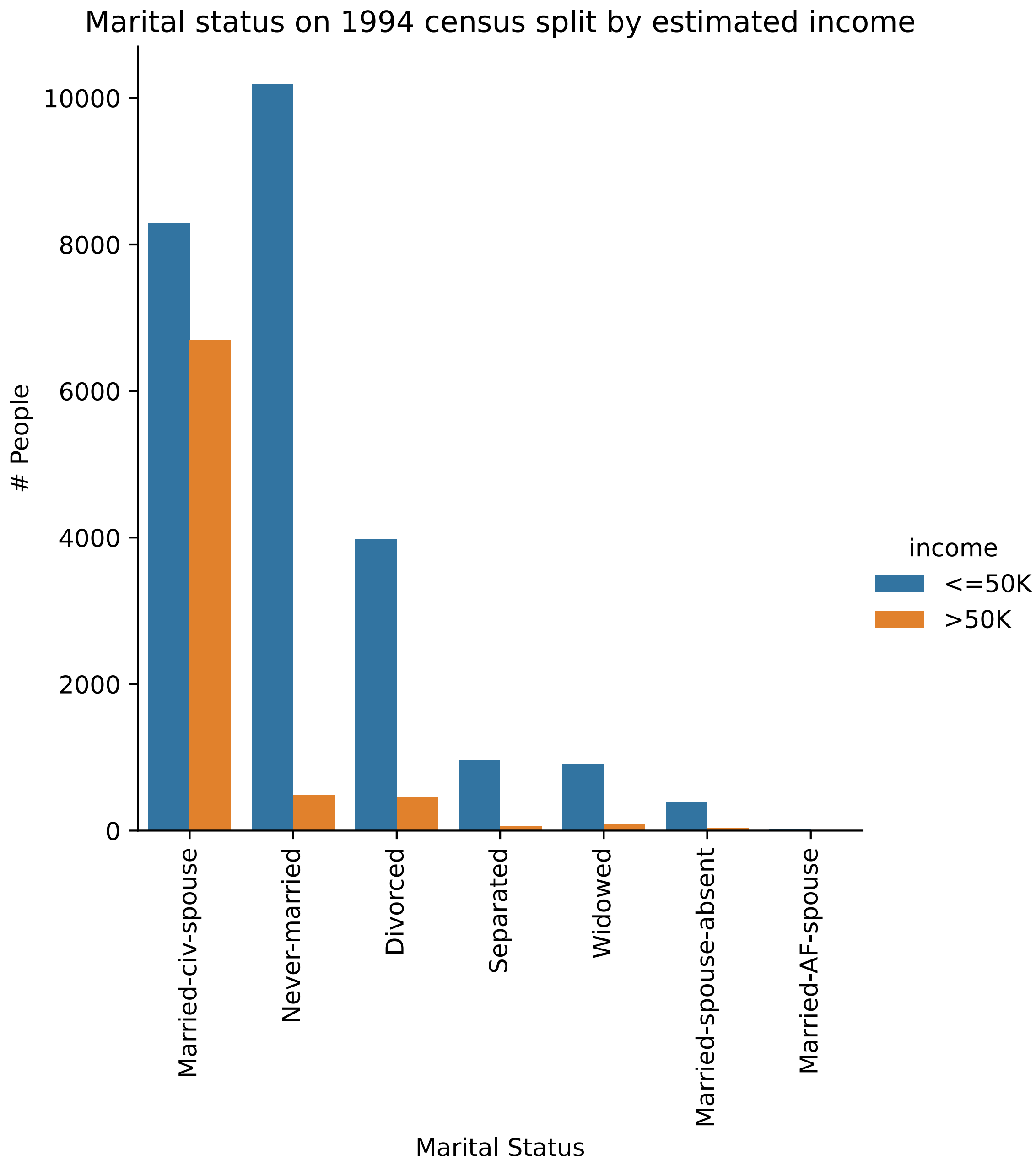
In [11]:
```
plot = sb.catplot(kind="count", data=table, hue="income", x="education", order=t
plot.set_xticklabels(rotation=90)
plt.xlabel("Education Level")
plt.ylabel("# People")
plt.title("Levels of education on 1994 census split by estimated income")
plt.show()
```



(ii.2)

In [12]:

```
plot = sb.catplot(kind="count", data=table, hue="income", x="marital-status", or
plot.set_xticklabels(rotation=90)
plt.xlabel("Marital Status")
plt.ylabel("# People")
plt.title("Marital status on 1994 census split by estimated income")
plt.show()
```



(iii)

Summary of data

All data gathered from 1994 census

1. The majority of people stopped with a high school education
2. The majority of people were marriedto civil spouses
3. People with a masters degree or better were most likely to make over 50k
4. Married people were significantly more likely to make over 50k

Sports Data

(a)

In [13]:

```
table2 = pd.read_csv("nfl-20-running-stats.csv", sep=",", engine='python').query
print("Total players considered: {0}".format(len(table2.index)))
```

```
Total players considered: 80
```

(b)

In [14]:

```
print("TD")
print("Mean:\t{0}".format(table2["TD"].mean()))
print("Median:\t{0}".format(table2["TD"].median()))
print("Mode:\t{0}".format(table2["TD"].mode()))
print()
print("FMB")
print("Mean:\t{0}".format(table2["Fmb"].mean()))
print("Median:\t{0}".format(table2["Fmb"].median()))
print("Mode:\t{0}".format(table2["Fmb"].mode()))
```

```
TD
Mean:   4.1125
Median: 3.0
Mode:   0    0
dtype: int64

FMB
Mean:   1.1875
Median: 1.0
Mode:   0    0
dtype: int64
```

(c)

In [15]:

```
print("YDS")
print("Q1:\t{0}".format(np.percentile(table2["Yds"], 25)))
print("Q3:\t{0}".format(np.percentile(table2["Yds"], 75)))
print("37th:\t{0}".format(np.percentile(table2["Yds"], 37)))
print()
print("1D")
print("Q1:\t{0}".format(np.percentile(table2["1D"], 25)))
print("Q3:\t{0}".format(np.percentile(table2["1D"], 75)))
print("37th:\t{0}".format(np.percentile(table2["1D"], 37)))
```

```
YDS
Q1:     221.5
Q3:     698.5
37th:   367.23

1D
Q1:     11.75
Q3:     40.0
37th:   19.23
```

(d)

In [16]:

```
print(table2["Y/G"].describe())
print()
print(table2["Lng"].describe())
```

```
count     80.000000
mean      38.456250
std       27.684405
min       -0.100000
25%       18.300000
50%       38.950000
75%       57.000000
max      126.700000
Name: Y/G, dtype: float64

count    80.000000
mean     36.525000
std      23.449366
min      -1.000000
25%      16.500000
50%      34.000000
75%      51.750000
max      98.000000
Name: Lng, dtype: float64
```

In [17]:

```
print("Five-Number Summary")
fiveSum = pd.DataFrame({'Column' : ["Y/G", "Lng"], 'Min' : [-0.1, -1.0], 'Q1' :
fiveSum
```

```
Five-Number Summary
```

Out[17]:

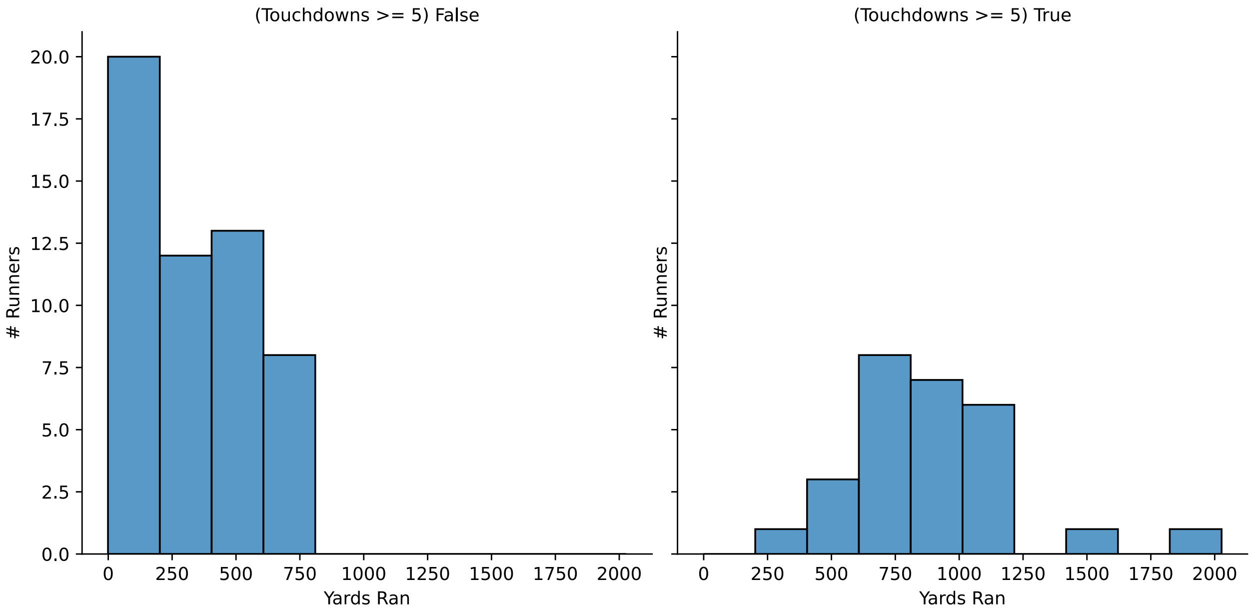
	Column	Min	Q1	Median	Q3	Max
0	Y/G	-0.1	18.3	38.95	57.00	126.7
1	Lng	-1.0	16.5	34.00	51.75	98.0

(e)

Multiple visuals given for same data

Visual (i)

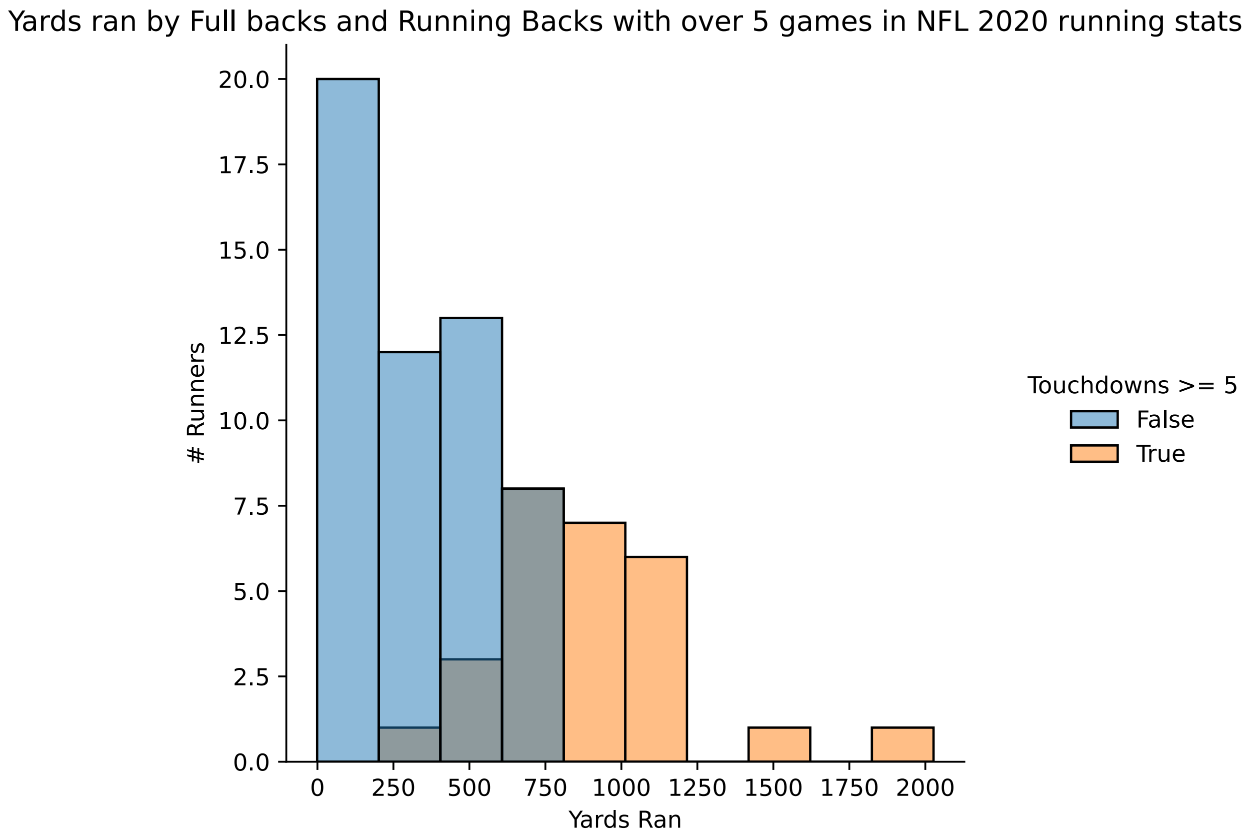
In [18]:
```
plot = sb.displot(data=table2, x="Yds", col=(table2["TD"] > 5) )
plot.set(xlabel="Yards Ran", ylabel="# Runners")
plot.set_titles("(Touchdowns >= 5) {col_name}")
plt.show()
```



Visual (ii)

In [19]:

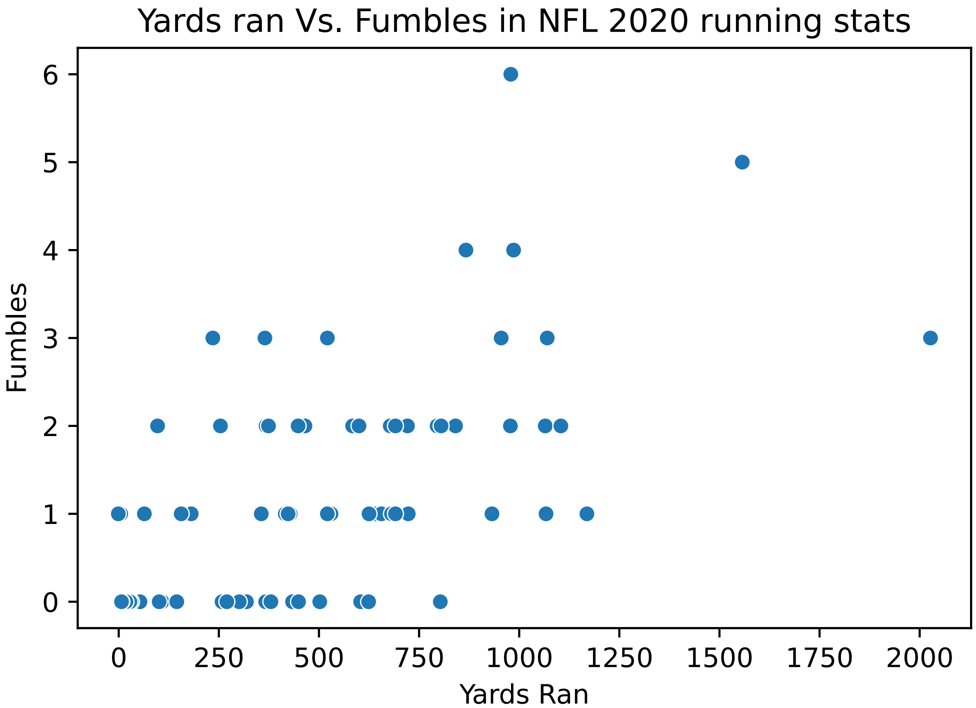
```
plot = sb.displot(data=table2, x="Yds", hue=(table2["TD"] > 5) )
plot.set(xlabel="Yards Ran", ylabel="# Runners")
plt.title("Yards ran by Full backs and Running Backs with over 5 games in NFL 20
plot._legend.set_title("Touchdowns >= 5")
plt.show()
```



(f)

In [20]:

```
sb.scatterplot(data=table2, x="Yds", y="Fmb")
plt.title("Yards ran Vs. Fumbles in NFL 2020 running stats")
plt.xlabel("Yards Ran")
plt.ylabel("Fumbles")
plt.show()
```



(g)

In [21]:

```
sb.scatterplot(data=table2, x="1D", y="Y/A")
plot.set(xlabel="First Downs", ylabel="Rushing Yards Per Attempt")
plt.title("First Downs Vs. Rushing Yards Per Attempt in NFL 2020 running stats")
plt.xlabel("First Downs")
plt.ylabel("Rushing Yards Per Attempt")
plt.show()
```