# Untitled

December 8, 2023

[1]:
```
import numpy as np
from datascience import *

import matplotlib.pyplot as plt
plt.style.use("ggplot")
%matplotlib inline
```

[2]:
```
dataimport = Table.read_table("exams.csv")

data = dataimport.select('parental level of education', 'test preparation␣
 ↪course', 'math score', 'reading score', 'writing score' )
data
```

[2]:
```
parental level of education | test preparation course | math score | reading
score | writing score
high school                 | completed               | 67         | 67
| 63
some high school            | none                    | 40         | 59
| 55
some college                | none                    | 59         | 60
| 50
high school                 | none                    | 77         | 78
| 68
associate's degree          | completed               | 78         | 73
| 68
high school                 | none                    | 63         | 77
| 76
bachelor's degree           | none                    | 62         | 59
| 63
some college                | completed               | 93         | 88
| 84
high school                 | none                    | 63         | 56
| 65
some college                | none                    | 47         | 42
| 45
… (990 rows omitted)
```

Question: Does the education of a parrent have any correlation to their childs performance on

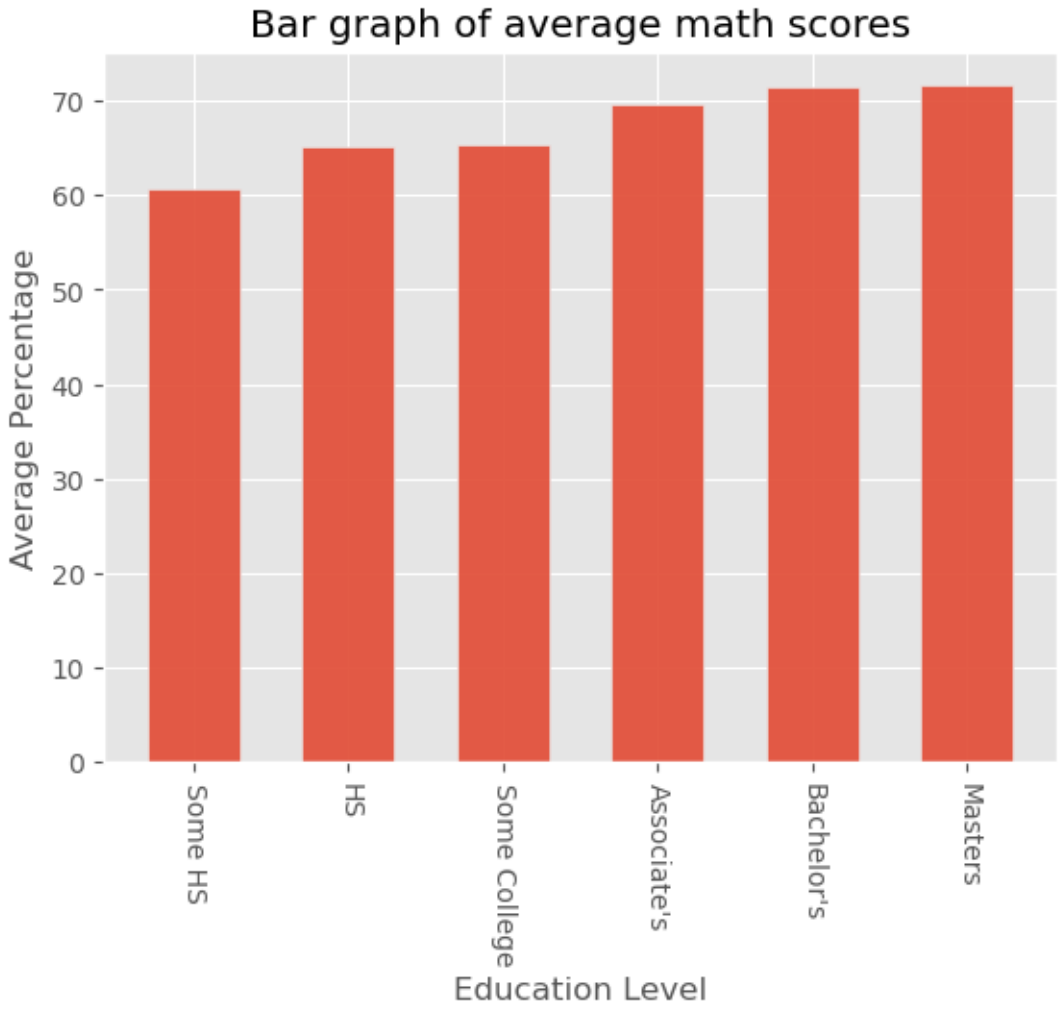
exams?

[3]:
```
math = data.select('parental level of education', 'math score')
math
```

[3]:
```
parental level of education | math score
high school                 | 67
some high school            | 40
some college                | 59
high school                 | 77
associate's degree          | 78
high school                 | 63
bachelor's degree           | 62
some college                | 93
high school                 | 63
some college                | 47
… (990 rows omitted)
```

[4]:
```
# Filter the table based on parental level of education
SHS = math.where('parental level of education', 'some high school')
HS = math.where('parental level of education', 'high school')
SC = math.where('parental level of education', 'some college')
AS = math.where('parental level of education', "associate's degree")
BS = math.where('parental level of education', "bachelor's degree")
MS = math.where('parental level of education', "master's degree")

# Calculate the averages for each category
SHSAVG = SHS.column('math score').mean()
HSAVG = HS.column('math score').mean()
SCAVG = SC.column('math score').mean()
ASAVG = AS.column('math score').mean()
BSAVG = BS.column('math score').mean()
MSAVG = MS.column('math score').mean()

# Prepare the data for the bar chart
math_averages = [SHSAVG, HSAVG, SCAVG, ASAVG, BSAVG, MSAVG]
math_labels = ['Some HS', 'HS', 'Some College', "Associate's", "Bachelor's",␣
 ↪'Masters']

# Create an array for the position of each bar on the x-axis
x_pos = np.arange(len(math_labels))

# Width of the bars (default is 0.8)
width = .6

# Create bar chart with specified width
plt.bar(x_pos, math_averages, width=width, align='center', alpha=0.9)
```

```
# Replace the x ticks with the respective labels
plt.xticks(x_pos, math_labels)

# Adding title and labels
plt.xlabel('Education Level')
plt.ylabel('Average Percentage')
plt.title('Bar graph of average math scores')

plt.xticks(x_pos, math_labels, rotation=270)

# Display the plot
plt.show()
```

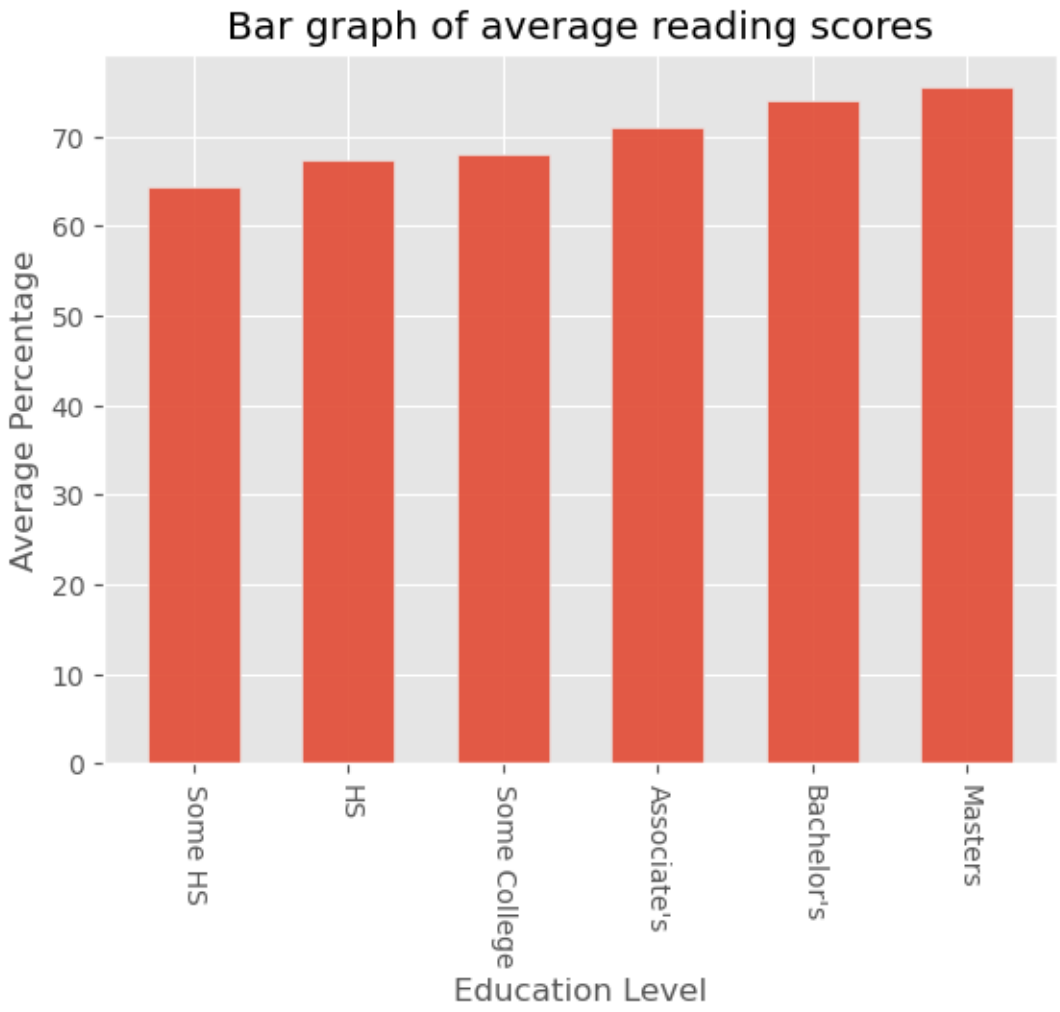



[5]:
```
reading = data.select('parental level of education', 'reading score')
reading
```

[5]:
```
parental level of education | reading score
high school                 | 67
some high school            | 59
some college                | 60
high school                 | 78
associate's degree          | 73
high school                 | 77
bachelor's degree           | 59
some college                | 88
high school                 | 56
some college                | 42
… (990 rows omitted)
```

[6]:
```
reading = data.select('parental level of education', 'reading score')

SHS = reading.where('parental level of education', 'some high school')
HS = reading.where('parental level of education', 'high school')
SC = reading.where('parental level of education', 'some college')
AS = reading.where('parental level of education', "associate's degree")
BS = reading.where('parental level of education', "bachelor's degree")
MS = reading.where('parental level of education', "master's degree")

SHSAVG = SHS.column('reading score').mean()
HSAVG = HS.column('reading score').mean()
SCAVG = SC.column('reading score').mean()
ASAVG = AS.column('reading score').mean()
BSAVG = BS.column('reading score').mean()
MSAVG = MS.column('reading score').mean()

reading_averages = [SHSAVG, HSAVG, SCAVG, ASAVG, BSAVG, MSAVG]
reading_labels = ['Some HS', 'HS', 'Some College', "Associate's", "Bachelor's",␣
 ↪'Masters']

x_pos = np.arange(len(reading_labels))

width = .6

plt.bar(x_pos, reading_averages, width=width, align='center', alpha=0.9)

plt.xticks(x_pos, reading_labels)

plt.xlabel('Education Level')
plt.ylabel('Average Percentage')
plt.title('Bar graph of average reading scores')

plt.xticks(x_pos, reading_labels, rotation=270)
```

```
plt.show()
```

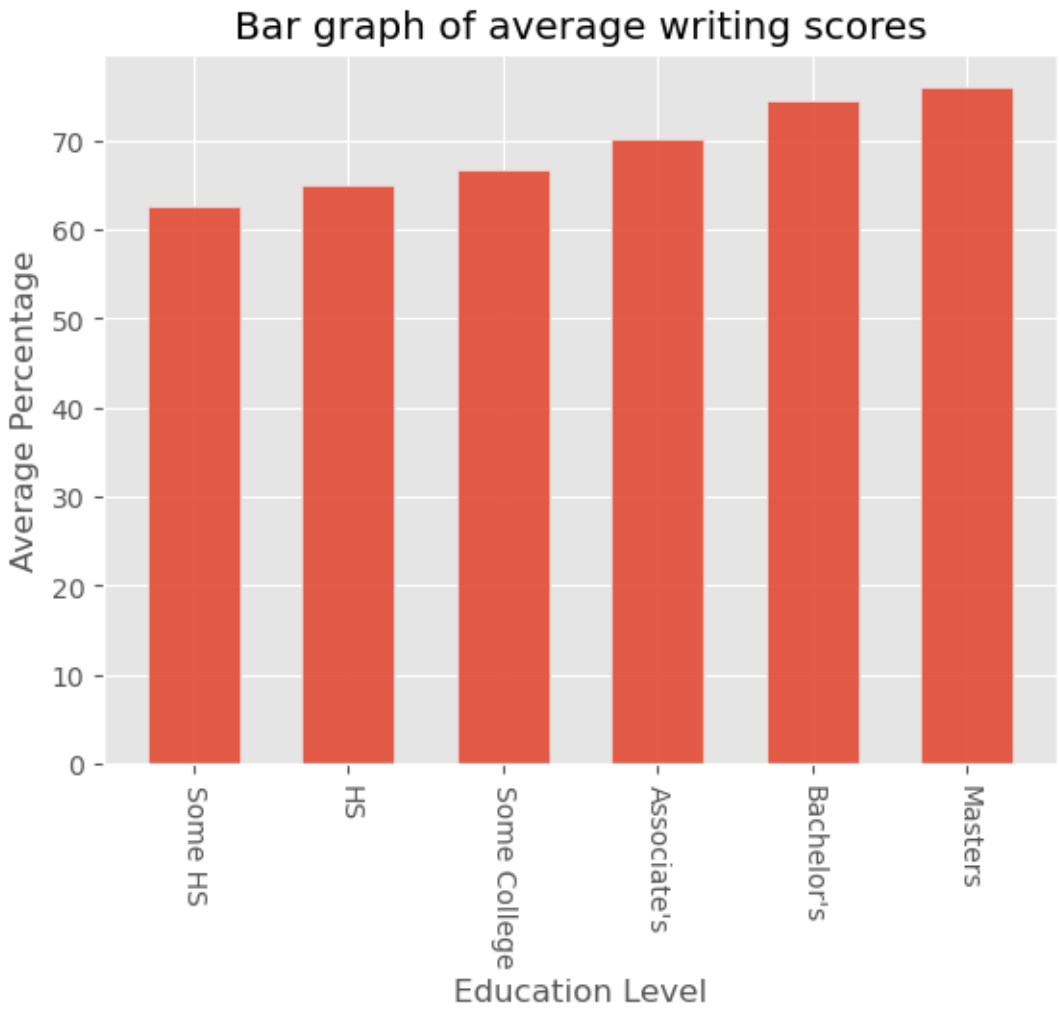



[7]:
```
writing = data.select('parental level of education', 'writing score')
writing
```

[7]:
```
parental level of education | writing score
high school                 | 63
some high school            | 55
some college                | 50
high school                 | 68
associate's degree          | 68
high school                 | 76
bachelor's degree           | 63
some college                | 84
high school                 | 65
```

```
some college                | 45
… (990 rows omitted)
```

[8]:
```
writing = data.select('parental level of education', 'writing score')

SHS = writing.where('parental level of education', 'some high school')
HS = writing.where('parental level of education', 'high school')
SC = writing.where('parental level of education', 'some college')
AS = writing.where('parental level of education', "associate's degree")
BS = writing.where('parental level of education', "bachelor's degree")
MS = writing.where('parental level of education', "master's degree")

SHSAVG = SHS.column('writing score').mean()
HSAVG = HS.column('writing score').mean()
SCAVG = SC.column('writing score').mean()
ASAVG = AS.column('writing score').mean()
BSAVG = BS.column('writing score').mean()
MSAVG = MS.column('writing score').mean()

writing_averages = [SHSAVG, HSAVG, SCAVG, ASAVG, BSAVG, MSAVG]
writing_labels = ['Some HS', 'HS', 'Some College', "Associate's", "Bachelor's",␣
 ↪'Masters']

x_pos = np.arange(len(writing_labels))

width = .6

plt.bar(x_pos, writing_averages, width=width, align='center', alpha=0.9)

plt.xticks(x_pos, writing_labels)

plt.xlabel('Education Level')
plt.ylabel('Average Percentage')
plt.title('Bar graph of average writing scores')

plt.xticks(x_pos, writing_labels, rotation=270)

plt.show()
```

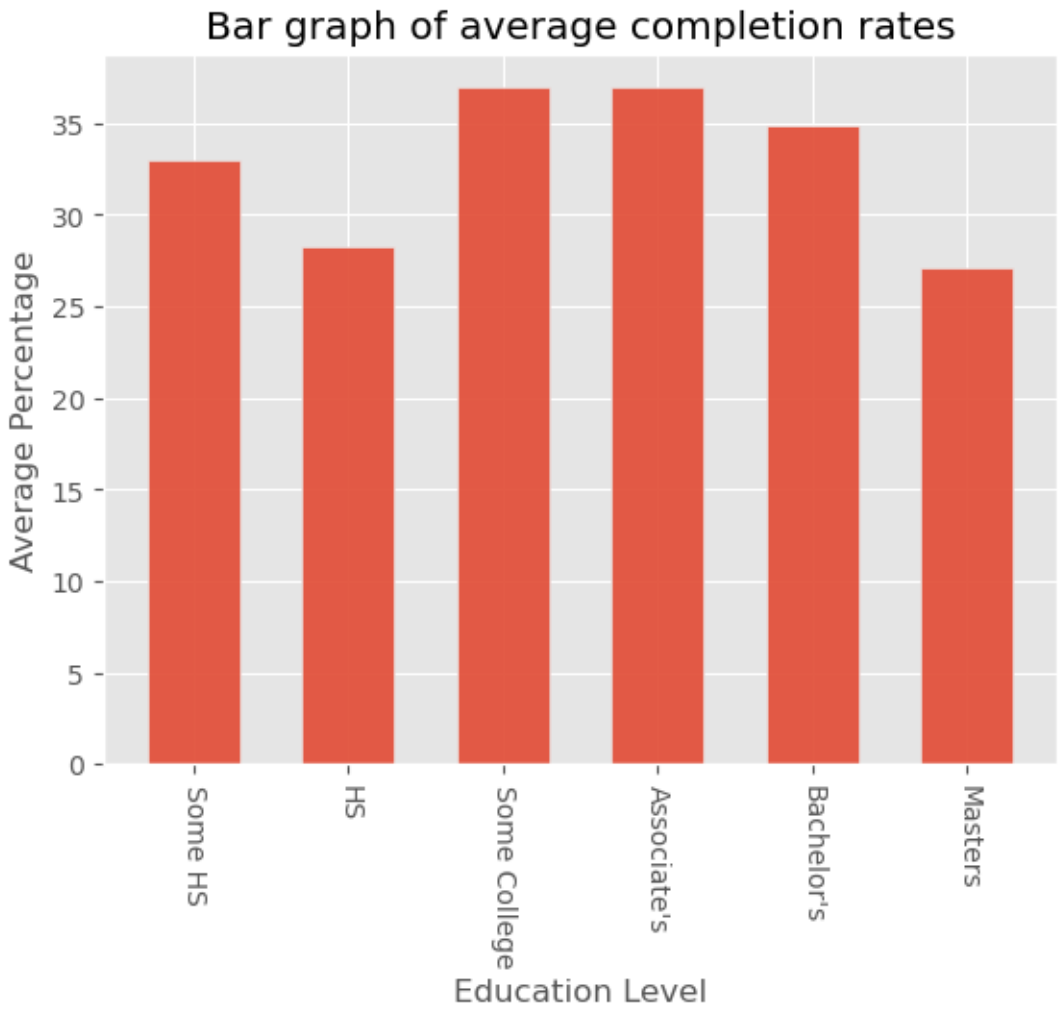



[9]:
```
prep_completed = data.where('test preparation course', 'completed')
percentage_completed = (prep_completed.num_rows / data.num_rows) * 100
percentage_completed
```

[9]: 33.5

[10]:
```
prep_completed = data.where('test preparation course', 'completed')

percentage_completed = (prep_completed.num_rows / data.num_rows) * 100

SHS_completed = prep_completed.where('parental level of education', 'some high␣
 ↪school')
HS_completed = prep_completed.where('parental level of education', 'high␣
 ↪school')
```

```
SC_completed = prep_completed.where('parental level of education', 'some␣
 ↪college')
AS_completed = prep_completed.where('parental level of education', "associate's␣
 ↪degree")
BS_completed = prep_completed.where('parental level of education', "bachelor's␣
 ↪degree")
MS_completed = prep_completed.where('parental level of education', "master's␣
 ↪degree")

SHS_percentage = (SHS_completed.num_rows / SHS.num_rows) * 100
HS_percentage = (HS_completed.num_rows / HS.num_rows) * 100
SC_percentage = (SC_completed.num_rows / SC.num_rows) * 100
AS_percentage = (AS_completed.num_rows / AS.num_rows) * 100
BS_percentage = (BS_completed.num_rows / BS.num_rows) * 100
MS_percentage = (MS_completed.num_rows / MS.num_rows) * 100

prep_averages = [SHS_percentage, HS_percentage, SC_percentage, AS_percentage,␣
 ↪BS_percentage, MS_percentage]
prep_labels = ['Some HS', 'HS', 'Some College', "Associate's", "Bachelor's",␣
 ↪'Masters']

x_pos = np.arange(len(prep_labels))

width = .6

plt.bar(x_pos, prep_averages, width=width, align='center', alpha=0.9)

plt.xticks(x_pos, prep_labels)

plt.xlabel('Education Level')
plt.ylabel('Average Percentage')
plt.title('Bar graph of average completion rates')

plt.xticks(x_pos, prep_labels, rotation=270)

plt.show()

print(f"The overall percentage of people who completed the 'test preparation␣
 ↪course' is {percentage_completed}%")
```




The overall percentage of people who completed the 'test preparation course' is 33.5%

[ ]:

[ ]:
```
prep_completed = data.where('test preparation course', 'completed')

SHS_completed = prep_completed.where('parental level of education', 'some high␣
 ↪school')
HS_completed = prep_completed.where('parental level of education', 'high␣
 ↪school')
SC_completed = prep_completed.where('parental level of education', 'some␣
 ↪college')
AS_completed = prep_completed.where('parental level of education', "associate's␣
 ↪degree")
```

```
BS_completed = prep_completed.where('parental level of education', "bachelor's␣
 ↪degree")
MS_completed = prep_completed.where('parental level of education', "master's␣
 ↪degree")

SHS_AVG = SHS_completed.column('math score').mean()
HS_AVG = HS_completed.column('math score').mean()
SC_AVG = SC_completed.column('math score').mean()
AS_AVG = AS_completed.column('math score').mean()
BS_AVG = BS_completed.column('math score').mean()
MS_AVG = MS_completed.column('math score').mean()

completed_averages = [SHS_AVG, HS_AVG, SC_AVG, AS_AVG, BS_AVG, MS_AVG]
labels = ['Some HS', 'HS', 'Some College', "Associate's", "Bachelor's",␣
 ↪'Masters']

x_pos = np.arange(len(labels))

width = .6

plt.figure(figsize=(10, 6))
plt.bar(x_pos, completed_averages, width=width, align='center', alpha=0.9)
plt.xticks(x_pos, labels, rotation=270)
plt.xlabel('Education Level')
plt.ylabel('Average Math Score')
plt.title('Bar graph of average math scores for those who completed the␣
 ↪preparation course')
plt.show()
```

[ ]:
```
prep_not_completed = data.where('test preparation course', 'none')

SHS_not_completed = prep_not_completed.where('parental level of education',␣
 ↪'some high school')
HS_not_completed = prep_not_completed.where('parental level of education',␣
 ↪'high school')
SC_not_completed = prep_not_completed.where('parental level of education',␣
 ↪'some college')
AS_not_completed = prep_not_completed.where('parental level of education',␣
 ↪"associate's degree")
BS_not_completed = prep_not_completed.where('parental level of education',␣
 ↪"bachelor's degree")
MS_not_completed = prep_not_completed.where('parental level of education',␣
 ↪"master's degree")

SHS_AVG = SHS_not_completed.column('math score').mean()
HS_AVG = HS_not_completed.column('math score').mean()
```

```
SC_AVG = SC_not_completed.column('math score').mean()
AS_AVG = AS_not_completed.column('math score').mean()
BS_AVG = BS_not_completed.column('math score').mean()
MS_AVG = MS_not_completed.column('math score').mean()

not_completed_averages = [SHS_AVG, HS_AVG, SC_AVG, AS_AVG, BS_AVG, MS_AVG]
labels = ['Some HS', 'HS', 'Some College', "Associate's", "Bachelor's",␣
 ↪'Masters']

x_pos = np.arange(len(labels))

width = .6

plt.figure(figsize=(10, 6))
plt.bar(x_pos, not_completed_averages, width=width, align='center', alpha=0.9)
plt.xticks(x_pos, labels, rotation=270)
plt.xlabel('Education Level')
plt.ylabel('Average Math Score')
plt.title('Bar graph of average math scores for those who did not complete the␣
 ↪preparation course')
plt.show()
```

[ ]:
```
prep_completed = data.where('test preparation course', 'completed')

SHS_completed = prep_completed.where('parental level of education', 'some high␣
 ↪school')
HS_completed = prep_completed.where('parental level of education', 'high␣
 ↪school')
SC_completed = prep_completed.where('parental level of education', 'some␣
 ↪college')
AS_completed = prep_completed.where('parental level of education', "associate's␣
 ↪degree")
BS_completed = prep_completed.where('parental level of education', "bachelor's␣
 ↪degree")
MS_completed = prep_completed.where('parental level of education', "master's␣
 ↪degree")

SHS_AVG = SHS_completed.column('writing score').mean()
HS_AVG = HS_completed.column('writing score').mean()
SC_AVG = SC_completed.column('writing score').mean()
AS_AVG = AS_completed.column('writing score').mean()
BS_AVG = BS_completed.column('writing score').mean()
MS_AVG = MS_completed.column('writing score').mean()

completed_averages = [SHS_AVG, HS_AVG, SC_AVG, AS_AVG, BS_AVG, MS_AVG]
```

```
labels = ['Some HS', 'HS', 'Some College', "Associate's", "Bachelor's",␣
 ↪'Masters']

x_pos = np.arange(len(labels))

width = .6

plt.figure(figsize=(10, 6))
plt.bar(x_pos, completed_averages, width=width, align='center', alpha=0.9)
plt.xticks(x_pos, labels, rotation=270)
plt.xlabel('Education Level')
plt.ylabel('Average Writing Score')
plt.title('Bar graph of average writing scores for those who completed the␣
 ↪preparation course')
plt.show()
```

[ ]:
```
prep_not_completed = data.where('test preparation course', 'none')

SHS_not_completed = prep_not_completed.where('parental level of education',␣
 ↪'some high school')
HS_not_completed = prep_not_completed.where('parental level of education',␣
 ↪'high school')
SC_not_completed = prep_not_completed.where('parental level of education',␣
 ↪'some college')
AS_not_completed = prep_not_completed.where('parental level of education',␣
 ↪"associate's degree")
BS_not_completed = prep_not_completed.where('parental level of education',␣
 ↪"bachelor's degree")
MS_not_completed = prep_not_completed.where('parental level of education',␣
 ↪"master's degree")

SHS_AVG = SHS_not_completed.column('writing score').mean()
HS_AVG = HS_not_completed.column('writing score').mean()
SC_AVG = SC_not_completed.column('writing score').mean()
AS_AVG = AS_not_completed.column('writing score').mean()
BS_AVG = BS_not_completed.column('writing score').mean()
MS_AVG = MS_not_completed.column('writing score').mean()

not_completed_averages = [SHS_AVG, HS_AVG, SC_AVG, AS_AVG, BS_AVG, MS_AVG]
labels = ['Some HS', 'HS', 'Some College', "Associate's", "Bachelor's",␣
 ↪'Masters']

x_pos = np.arange(len(labels))

width = .6
```

```
plt.figure(figsize=(10, 6))
plt.bar(x_pos, not_completed_averages, width=width, align='center', alpha=0.9)
plt.xticks(x_pos, labels, rotation=270)
plt.xlabel('Education Level')
plt.ylabel('Average Writing Score')
plt.title('Bar graph of average writing scores for those who did not complete␣
 ↪the preparation course')
plt.show()
```

[ ]:
```
prep_completed = data.where('test preparation course', 'completed')

SHS_completed = prep_completed.where('parental level of education', 'some high␣
 ↪school')
HS_completed = prep_completed.where('parental level of education', 'high␣
 ↪school')
SC_completed = prep_completed.where('parental level of education', 'some␣
 ↪college')
AS_completed = prep_completed.where('parental level of education', "associate's␣
 ↪degree")
BS_completed = prep_completed.where('parental level of education', "bachelor's␣
 ↪degree")
MS_completed = prep_completed.where('parental level of education', "master's␣
 ↪degree")

SHS_AVG = SHS_completed.column('reading score').mean()
HS_AVG = HS_completed.column('reading score').mean()
SC_AVG = SC_completed.column('reading score').mean()
AS_AVG = AS_completed.column('reading score').mean()
BS_AVG = BS_completed.column('reading score').mean()
MS_AVG = MS_completed.column('reading score').mean()

completed_averages = [SHS_AVG, HS_AVG, SC_AVG, AS_AVG, BS_AVG, MS_AVG]
labels = ['Some HS', 'HS', 'Some College', "Associate's", "Bachelor's",␣
 ↪'Masters']

x_pos = np.arange(len(labels))

width = .6

plt.figure(figsize=(10, 6))
plt.bar(x_pos, completed_averages, width=width, align='center', alpha=0.9)
plt.xticks(x_pos, labels, rotation=270)
plt.xlabel('Education Level')
plt.ylabel('Average Reading Score')
plt.title('Bar graph of average reading scores for those who completed the␣
 ↪preparation course')
```

```
plt.show()
```

[ ]:
```
prep_not_completed = data.where('test preparation course', 'none')

SHS_not_completed = prep_not_completed.where('parental level of education',␣
 ↪'some high school')
HS_not_completed = prep_not_completed.where('parental level of education',␣
 ↪'high school')
SC_not_completed = prep_not_completed.where('parental level of education',␣
 ↪'some college')
AS_not_completed = prep_not_completed.where('parental level of education',␣
 ↪"associate's degree")
BS_not_completed = prep_not_completed.where('parental level of education',␣
 ↪"bachelor's degree")
MS_not_completed = prep_not_completed.where('parental level of education',␣
 ↪"master's degree")

SHS_AVG = SHS_not_completed.column('reading score').mean()
HS_AVG = HS_not_completed.column('reading score').mean()
SC_AVG = SC_not_completed.column('reading score').mean()
AS_AVG = AS_not_completed.column('reading score').mean()
BS_AVG = BS_not_completed.column('reading score').mean()
MS_AVG = MS_not_completed.column('reading score').mean()

not_completed_averages = [SHS_AVG, HS_AVG, SC_AVG, AS_AVG, BS_AVG, MS_AVG]
labels = ['Some HS', 'HS', 'Some College', "Associate's", "Bachelor's",␣
 ↪'Masters']

x_pos = np.arange(len(labels))

width = .6

plt.figure(figsize=(10, 6))
plt.bar(x_pos, not_completed_averages, width=width, align='center', alpha=0.9)
plt.xticks(x_pos, labels, rotation=270)
plt.xlabel('Education Level')
plt.ylabel('Average Reading Score')
plt.title('Bar graph of average reading scores for those who did not complete␣
 ↪the preparation course')
plt.show()
```

[ ]:
```

```

[ ]:
```

```