UCSD Students: Time and Course Satisfaction

Permissions

Place an `X` in the appropriate bracket below to specify if you would like your group's project to be made available to the public. (Note that student names will be included (but PIDs will be scraped from any groups who include their PIDs).

- [x] YES - make available
- [ ] NO - keep private

Names

- Henry Lam
- Alex Truong
- Dianne Natanauan
- Darryl Remulla
- Sam Hormozian

Abstract

We did research on the relationship between the amount of hours spent per week in courses at UCSD and course satisfaction because we wanted to know if spending more or less time would change if a student's perception of a course was positive or negative. We used a public dataset called CAPEs (short for Course and Professor Evaluations), which shows the recorded responses to a survey that UCSD used to release every quarter in order for students to evaluate different parts of the courses they were enrolled in for that quarter. We calculated the correlation between hours spent per week and course satisfaction of all reported courses taken at UCSD while taking into account department and upper division standing. Through this research, we found that there is enough information to conclude that the amount of hours spent per week in courses at UCSD has a statistically significant correlation with course satisfaction, but some of the data may be biased due to some courses and departments having way less representation in the dataset.

To further explore this bias and its effects on our analysis, we also reviewed which courses had the strongest correlations within the upper and lower division departments. We found that most of the top 3 courses with the strongest correlations within upper and lower

division had a very small pool of classes to represent them and considered how this skewed their correlation score. Overall, our major conclusion was that despite there being a few department courses that might cause a bit of bias in our data, there were still enough valid data points to determine a statistically significant correlation between hours spent per week and course satisfaction, supporting our hypothesis that more time spent leads to less course satisfaction.

Research Question

What is the relationship between time spent in courses and UCSD undergraduates' satisfaction with courses over the time period of Summer Session 2007 to Spring 2023? Does this relationship differ by departments and are there confounding variables?

Background and Prior Work

Our study involves the relationship between students' time allocation in courses with their overall satisfaction with both the course content as well as the professors. Investigating the factors that influence students' decisions on how they distribute their time is important for understanding academic engagement.

Relevant research performed by Brian K. Coffey and his team, professors at Kansas State University in the Agricultural Economics department, in which they observed students' time allocation and the significance of GPA compared to other values in their article: "How Do Students Allocate Their Time? An Application of Prospect Theory to Trade-offs between Time Spent to Improve GPA Versus Time Spent on Other Activities"[1]. The study focused on students enrolled in required intermediate microeconomic theory courses for agricultural economics and agribusiness majors and minors. Through surveys, Coffey and his team identified the priority order of university goals, with graduating being the highest, followed by networking, academic achievement, income, and social experiences. Choice experiment analysis and prospect theory were utilized to understand decision-making processes involving risk and uncertainty. The study revealed that student GPA significantly and positively influenced the probability of choosing a particular alternative, emphasizing the importance students place on GPA in their time allocation decisions. Beyond academic pursuits, Coffey observed various alternative activities, providing insights into students' weekly time commitments and their correlation with GPA. These findings contribute to understanding the complex interplay between time management, academic priorities, and overall student satisfaction.[1]

Further studying student satisfaction in higher learning, in "The Adult Student and Course Satisfaction: What Matters Most?" Howell and Buck investigate the factors that affect satisfaction for adult students in a business degree program. They gathered survey data

from 1,725 students and 214 instructors from five different institutions of higher education, and performed various multiple regression analyses to determine which factors have significant relationships with course satisfaction. Of the 11 factors they explored, including class size, location, subject, and more, they found that there are four main factors that have a significant effect on course satisfaction: subject matter relevancy, faculty competency, classroom management, and student workload. They go on to explain that a higher course load can be perceived as too demanding and therefore have a negative impact on satisfaction. This guides our research to further explore the relationship between student workload and course satisfaction.[4]

Furthermore, in another study by Janet M. Ferguson and Amy E. DeFelice, titled "Length of Online Course and Student Satisfaction, Perceived Learning, and Academic Performance,"[2] the authors investigated the impact of the course format (five-week intensive online course vs. full-semester online course) on student satisfaction, perceived learning, and academic performance. For who participated in the study, Ferguson and DeFelice observed graduate students enrolled in online courses for pre-service teachers. The study, consisting of a course evaluation survey and an analysis of final grades, revealed that students in the intensive five-week course exhibited higher satisfaction with student-student communication but lower satisfaction with communication with the instructor compared to full-semester students.Students in the shorter format demonstrated significantly stronger academic performance than their counterparts in the full-semester course, which may suggest a correlation between the time allocated to a course and students' satisfaction and performance. This highlights the importance of considering course format and duration when evaluating student satisfaction and academic outcomes in online education as a condensed format may positively influence academic performance based on the length of the online course.

The paper "Students' Satisfaction in Higher Education Literature Review" by M Salinda Weerasinghe, R. Lalitha and, S. Fernando, explores the multidimensional nature of students' satisfaction in higher education, observing influential factors such as the quality of lecturers, availability of resources, and the effectiveness of technology. Various satisfaction models and frameworks, including SERVQUAL (standing for Service and Quality--designed to measure the perceived quality of services provided by an organization from the customer's perspective) and investment theory, are reviewed, providing a theoretical background for understanding satisfaction in higher education. Empirical findings from diverse studies emphasize the significance of teaching quality, curriculum, and infrastructure in shaping students' satisfaction levels, which may also be relevant in aiding us in answering our question.[3]

To observe UCSD undergraduates' satisfaction with professors and courses, UCSD promoted a system known as CAPES, which stands for Course and Professor Evaluations, where students would provide their opinions and feedback on the course and professor near the final weeks of taking part in the course. Similarly to Coffey's observations, CAPEs surveyed

students on how much time they put into the course, whether or not they recommend the course and professor, what their expected grade was, and afterwards when grades were finalized, included the grade received. Although this does not observe every time allocation a particular student decides to spend their time on, it provides us a stepping stone into answering our research question regarding student satisfaction being correlated to their time spent in the respective course with the respective professor.

1. ^ https://ageconsearch.umn.edu/record/303903
2. ^https://www.researchgate.net/publication/44099000_Length_of_Online_Course_and_Stude
3. ^https://deliverypdf.ssrn.com/delivery.php?ID=782003117027015015122124091020125089052016006059021006029098015127094079
4. ^https://link.springer.com/article/10.1007/s10755-011-9201-0

Hypothesis

We believe as the amount of time spent by UCSD undergraduate students increases, students will be less satisfied by their course. This outcome is supported by the idea that students do not prefer courses that assign time-consuming work or that time-consuming work would make a class be perceived as tougher.

Data

Data overview

- Dataset #1
  - Dataset Name: UCSD CAPEs Data
  - Link to the dataset: https://www.kaggle.com/datasets/sanbornpnguyen/ucsdcapes?utm_medium=social&utm_campaign=kaggle-dataset-share
  - Number of observations: 63363
  - Number of variables: 11

We are using a dataset of UCSD CAPEs Data from Summer Session 2007 to Spring 2023. The data includes the ratings of 63,363 courses and the respective professors teaching those courses during the duration of that time. The dataset measures total number of students enrolled in the course, the total number of CAPEs given for the course, the percentage of students that would recommend the class, the percentage of students that recommended the professor, the average study hours per week reported by students, average grade expected by students, average grade received by students, and the evaluation URL for that course/professor.

Per our hypothesis, we wanted to focus on the average study hours per week reported by the students and the percentage of students that would recommend the class/professor. Since we assume that there may be a bias in how the course is perceived due to its department or if it is lower or upper division, we want to wrangle our dataset to include two new variable columns: the department a course is in and whether that course is upper or lower division. We also want to remove the evaluation URL column, the letter grade represented in "average grade expected" and "average grade received" by students, and percentage symbol in the related columns since it did not seem valuable to our analysis.

To create our two new columns, we can split each string in the "Course" column by word, leaving us with the department code in the first split word and the course number in the second split word. Each department code can be put into a separate column for each course, and each course number can be compared to the number 100 so that the "Upper Division" column would show False for course numbers less than 100 and True for course numbers greater than or equal to it.

Dataset: UCSD Capes Data from Summer Session 2007 to Spring 2023

Setup:

In [2]:
```
# Imports
%matplotlib inline

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

import seaborn as sns
sns.set()
sns.set_context('talk')

import warnings
warnings.filterwarnings('ignore')

import patsy
import statsmodels.api as sm
import scipy.stats as stats
from scipy.stats import ttest_ind, chisquare, normaltest
# Note: the statsmodels import may print out a 'FutureWarning'. Thats fine.

df = pd.read_csv("capes_data.csv")
```

Data Cleaning

- Our data will become clean in the sense that there will be no nan values.

- To get the data into usable format, made a new column that specified the respective department of each course, made a new column for boolean values that determined Upper_Division, removed excess symbols as well as converted to floats in regards to the data in Recommend_Class and Recommend_Professor and Avg_Grade_Received and Avg_Grade_Expected, and removed nan values.
  - We determined these steps were necessary to get the data into usable format because comparing relationships will be done between quantitative variables and separating and comparing distributions separated by departments and upper division standing will be easier with these changes.
- Pre-processing steps: we did check data distributions, but no transformations were required since the outliers were dense and might be explained later in the EDA section where we take into account departments and upperdivision standing for courses.

In [3]:
```
def standardize_Upper_Division(st):
    st = st.lower()
    st = st.strip()
    while (st.isnumeric()==False):
        st = st[: len(st)-1]
    st = int(st)
    if (st>99):
        return True
    else:
        return False
def standardize_Percentages(st):
    st = st.lower()
    st = st.strip()
    st = st[: len(st)-1]
    st = float(st)
    return st
def standardize_Grades(st):
    st = str(st)
    if(st == str(df['Average Grade Received'][2])):
        return np.nan
    else:
        return st[len(st)-5:len(st)-1]
```

In [4]:
```
#time for some data wrangling: add a Department column and Upper_Division column
df = df.assign(Department = df['Course'].str.split(pat = ' ').str[0])
df = df.assign(Upper_Division = df['Course'].str.split(pat = ' ').str[1])
df['Upper_Division'] = df['Upper_Division'].apply(standardize_Upper_Division)

#remove percentage signs
df['Percentage Recommended Class'] = df['Percentage Recommended Class'].apply(stand
df['Percentage Recommended Professor'] = df['Percentage Recommended Professor'].app

#remove letter grade, leaving only floating points and nan values
df['Average Grade Received'] = df['Average Grade Received'].apply(standardize_Grade
df['Average Grade Expected'] = df['Average Grade Expected'].apply(standardize_Grade

#remove 'Evaluation URL' because useless column
df = df[['Instructor', 'Course', 'Quarter', 'Total Enrolled in Course',
       'Total CAPEs Given', 'Percentage Recommended Class',
```

```
       'Percentage Recommended Professor', 'Study Hours per Week',
       'Average Grade Expected', 'Average Grade Received',
       'Department', 'Upper_Division']]

#rename some columns to make them easier to work with
df.columns = ['Instructor', 'Course', 'Quarter', 'Total_Enrolled',
       'Total_CAPEs_Given', 'Recommend_Class',
       'Recommended_Professor', 'Weekly_Hours',
       'Avg_Grade_Expected', 'Avg_Grade_Received',
       'Department', 'Upper_Division']

#remove all rows with nan values
df = df.dropna(axis = 'rows')

#change data types for grades expected and received
df = df.astype({'Avg_Grade_Expected': 'float'})
df = df.astype({'Avg_Grade_Received': 'float'})
df
```

Out[4]:

	Instructor	Course	Quarter	Total_Enrolled	Total_CAPEs_Given
0	Butler Elizabeth Annette	AAS 10 - Intro/African-American Studies (A)	SP23	66	48
1	Butler Elizabeth Annette	AAS 170 - Legacies of Research (A)	SP23	20	7
3	Shtienberg Gilad	ANAR 115 - Coastal Geomorphology/Environ (A)	SP23	26	6
4	Braswell Geoffrey E.	ANAR 155 - Stdy Abrd: Ancient Mesoamerica (A)	SP23	22	9
5	Hrvoj Mihic Branka	ANBI 111 - Human Evolution (A)	SP23	22	4
...	...	...	...	...	...
63005	Newsome Elizabeth Ann	VIS 21A - Int/ArtAmericas/Africa/Oceania (A)	FA07	230	160
63006	Steinbach Haim	VIS 3 - Intr/Art-Making:3-D Practices (A)	FA07	300	220
63007	Wallen Ruth	VIS 60 - Introduction to Photography (A)	FA07	66	44
63008	Trigilio Michael	VIS 70N - Introduction to Media (A)	FA07	226	77
63009	Mangolte Babette	VIS 84 - History Of Film (A)	FA07	224	153

45393 rows × 12 columns

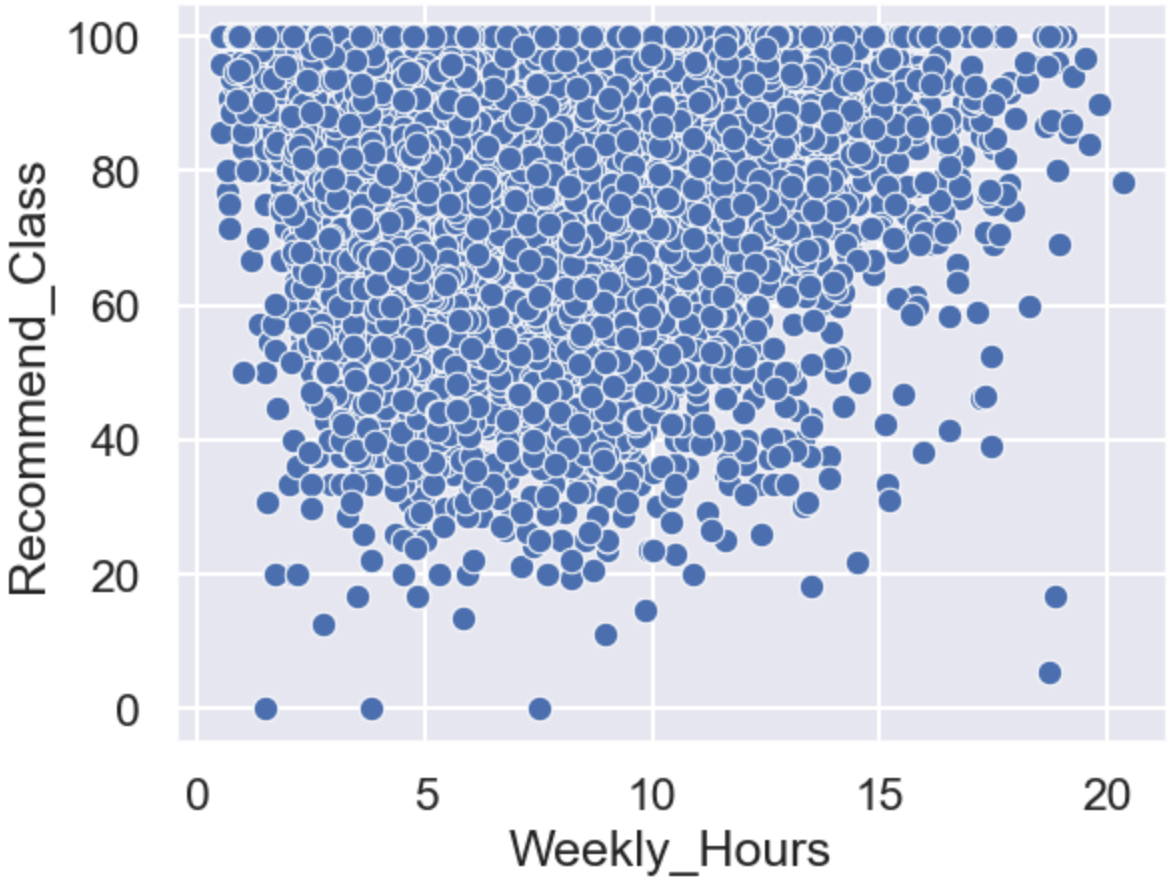
Data Analysis

Checking the Relationship between Weekly_Hours and Recommend_Class as well as potential confounding variable Avg_Grade_Expected

- We will use scatterplots to get an idea of how the data looks and use linear_regression to show the relationship between Weekly_Hours and Recommend_Class.

In [5]:
```
sns.scatterplot(data = df, x = 'Weekly_Hours', y = 'Recommend_Class')
```

```
plt.show()
```

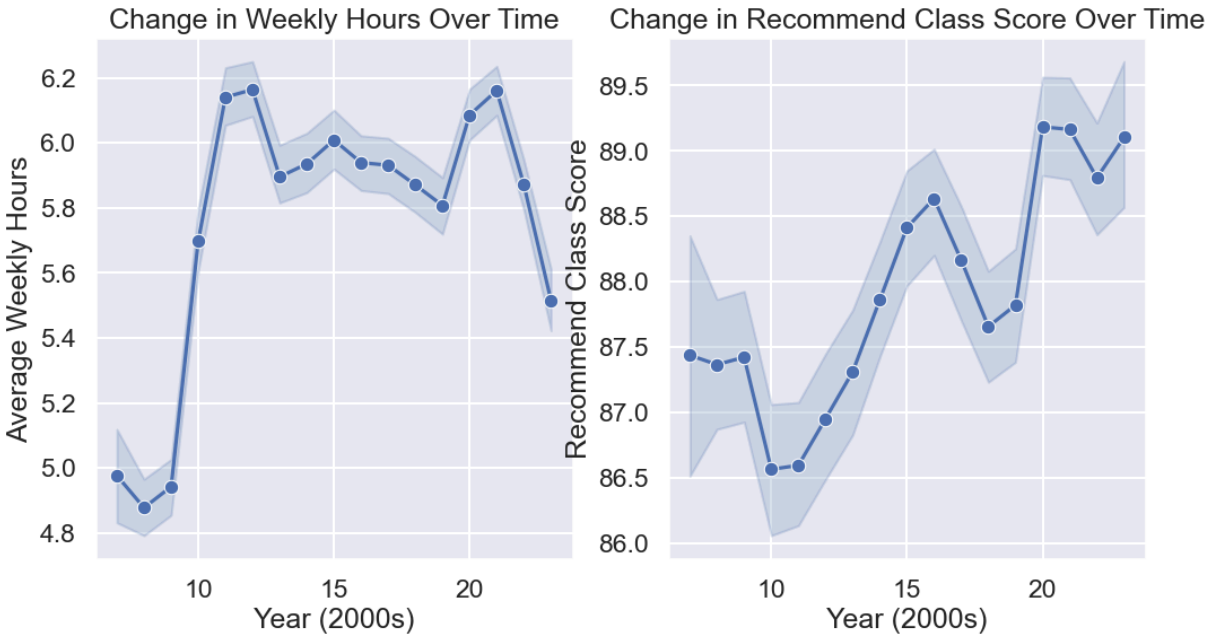


We see that the data has high variability, but that in general as weekly hours spent on a class increases, the recommendation score decreases. To know with more certainty we can conduct a linear regression between the two variable to assess their relationship.

We can also take a look at changes in weekly hours reported and recommended class score over the years within our dataset to see there could be any significant changes as time has passed that would affect our analysis.

In [20]:
```
#create new dataframe where years are extracted to show changes over time
df_year = df
df_year['Year'] = df['Quarter'].str[2:].astype(int)

#show two lineplots to compare weekly hours and course satisfaction
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 6))

#graph to view changes in weekly hours over time
plt.figure(figsize=(8, 6))
sns.lineplot(data=df_year, x='Year', y='Weekly_Hours', marker='o', ax=axes[0])
axes[0].set_title('Change in Weekly Hours Over Time')
axes[0].set_xlabel('Year (2000s)')
axes[0].set_ylabel('Average Weekly Hours')
axes[0].grid(True)

#graph to view changes in course satisfaction over time
plt.figure(figsize=(8, 6))
```

```
sns.lineplot(data=df_year, x='Year', y='Recommend_Class', marker='o', ax=axes[1])
axes[1].set_title('Change in Recommend Class Score Over Time')
axes[1].set_xlabel('Year (2000s)')
axes[1].set_ylabel('Recommend Class Score')
axes[1].grid(True)
```




```
<Figure size 800x600 with 0 Axes>
<Figure size 800x600 with 0 Axes>
```

Taking a look at the lineplots above, there does not seem to be too much of a significant change in weekly hours over time. There is a small spike in reported weekly hours on average in 2010, but the increase is only around 1.2 hours. Taking a look at the recommended class score over the years, it is unclear if there is any relationship between this data and average weekly hours. The recommend class score has increased over time, but the line plot also shows that there is a great degree in variability of these scores. Since it isn't exactly clear what relationship these two variables have to each other is by looking at their changes over time, we will first look at a linear regression overall of the data and observe other factors that may affect their relationship.

Checking assumptions about linear regression:

- Linear relationship?
  - Recommend_Class and Weekly_Hours does not seem to have a relationship that is too strongly linear.
- Multicollinearity:
  - The independent variables are not too highly correlated with each other and we will explain the proposed confounding variable (Avg_Grade_Expected) further down in the document.
- Autocorrelation:

    - Observations are independent of one another as each course's Weekly_Hours and Recommend_Class do not affect each other courses' Weekly_Hours and Recommend_Class.
- Not homoscedastic:
    - We have points spread throughout both ends of the graph according to our scatterplot shown above.

In [5]:
```
outcome_1, predictors_1 = patsy.dmatrices('Recommend_Class ~ Weekly_Hours', df)
mod_1 = sm.OLS(outcome_1, predictors_1)
res_1 = mod_1.fit()
print(res_1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Recommend_Class   R-squared:                       0.030
Model:                            OLS   Adj. R-squared:                  0.030
Method:                 Least Squares   F-statistic:                     1421.
Date:                Wed, 20 Mar 2024   Prob (F-statistic):          3.10e-306
Time:                        16:00:45   Log-Likelihood:            -1.7748e+05
No. Observations:               45393   AIC:                         3.550e+05
Df Residuals:                   45391   BIC:                         3.550e+05
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      93.3155      0.152    615.136      0.000      93.018      93.613
Weekly_Hours   -0.9102      0.024    -37.697      0.000      -0.958      -0.863
==============================================================================
Omnibus:                    12728.979   Durbin-Watson:                   1.543
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            34831.089
Skew:                          -1.502   Prob(JB):                         0.00
Kurtosis:                       6.065   Cond. No.                         17.2
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
```

After performing a linear regression, our results show us that our regression has a y-intercept of 93.3155 and weekly hours has an effect size of -0.9102. This tells us that:

1. If weekly hours spent in a class was 0 hours, the recommended class score would be roughly 93.3155.
2. For every 1 hour increase in weekly hours, we expect to see a -0.9102, or a 0.9102 decrease in recommended class score.

Essentially, the linear regression results suggest that if we saw that a class had a weekly hour average of 0, we could expect a relatively high recommended class score of roughly 93%. As the amount of average weekly hours for a class increases by one hour, we can expect the recommended class score to decrease by almost 1 point.

Consider Avg_Grade_Expected

- With consideration for Avg_Grade_Expected, is Weekly_Hours still statistically significant? Let's perform an OLS regression test to check.

In [6]:
```
outcome_2, predictors_2 = patsy.dmatrices('Recommend_Class ~ Weekly_Hours + Avg_Gra
mod_2 = sm.OLS(outcome_2, predictors_2)
res_2 = mod_2.fit()
print(res_2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Recommend_Class   R-squared:                       0.225
Model:                            OLS   Adj. R-squared:                  0.225
Method:                 Least Squares   F-statistic:                     6592.
Date:                Wed, 20 Mar 2024   Prob (F-statistic):               0.00
Time:                        16:00:46   Log-Likelihood:            -1.7239e+05
No. Observations:               45393   AIC:                         3.448e+05
Df Residuals:                   45390   BIC:                         3.448e+05
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
==
                         coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
--
Intercept             18.8936      0.710     26.614      0.000      17.502      20.2
85
Weekly_Hours           0.0689      0.023      2.936      0.003       0.023       0.1
15
Avg_Grade_Expected    19.7481      0.185    106.799      0.000      19.386      20.1
11
==============================================================================
Omnibus:                    13116.010   Durbin-Watson:                   1.616
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            42472.806
Skew:                          -1.472   Prob(JB):                         0.00
Kurtosis:                       6.713   Cond. No.                         103.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
```

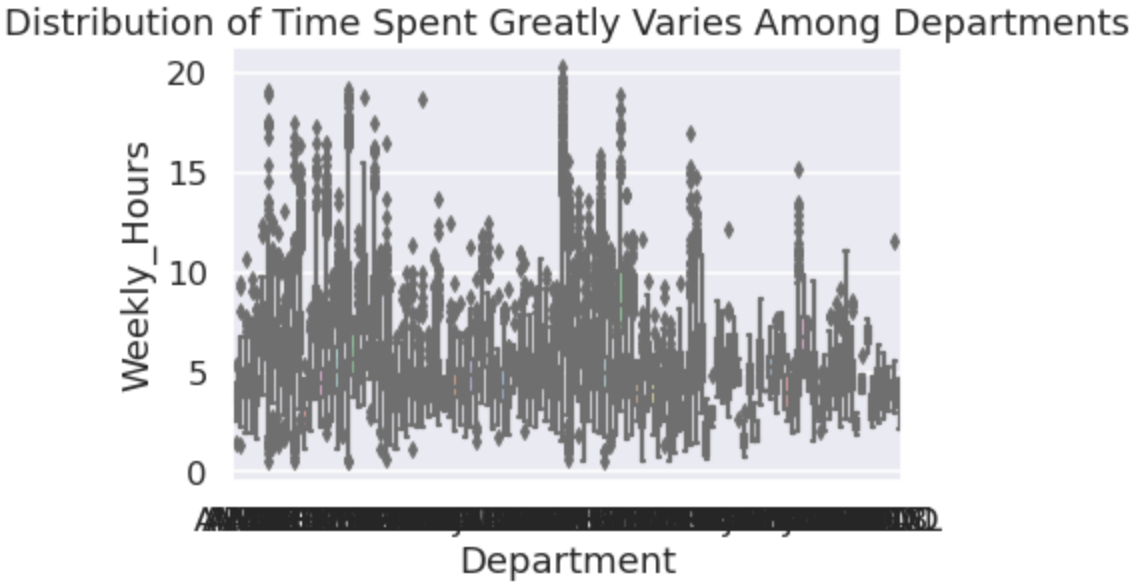
With an alpha value of 0.01, the Weekly_Hours's p-value of 0.003 passes the statistical test for significance. This means that the observed results would occur at a 0.3% chance even with Avg_Grade_Expected taken into account.

Addressing the Concern for the Argument that Different Departments Require Different Time Commmitments for Student Learning:

- Let's see how the distributions of time spent(weekly_hours) as well as course satisfaction(recommend_class) differ when departments are taken into account.

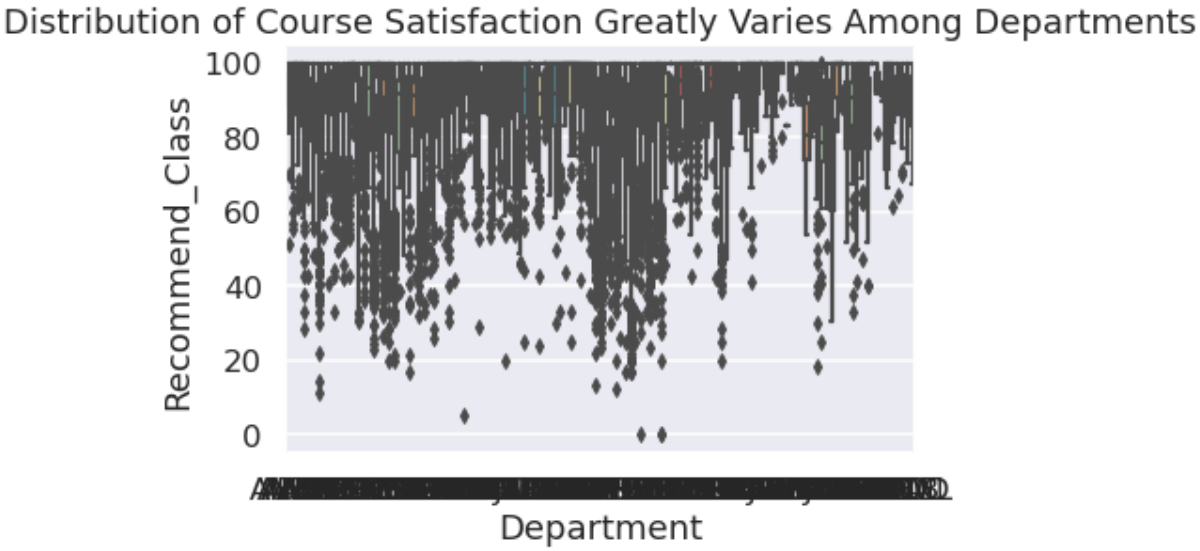
In [7]:
```
boxplot_weekly_hours = sns.boxplot(x = 'Department', y = 'Weekly_Hours', data = df,
boxplot_weekly_hours.set_title('Distribution of Time Spent Greatly Varies Among Dep
```

Out[7]:
```
Text(0.5, 1.0, 'Distribution of Time Spent Greatly Varies Among Departments')
```



In [8]:
```
boxplot_recommend_class = sns.boxplot(x = 'Department', y = 'Recommend_Class', data
boxplot_recommend_class.set_title('Distribution of Course Satisfaction Greatly Vari
```

Out[8]:
```
Text(0.5, 1.0, 'Distribution of Course Satisfaction Greatly Varies Among Departmen
ts')
```

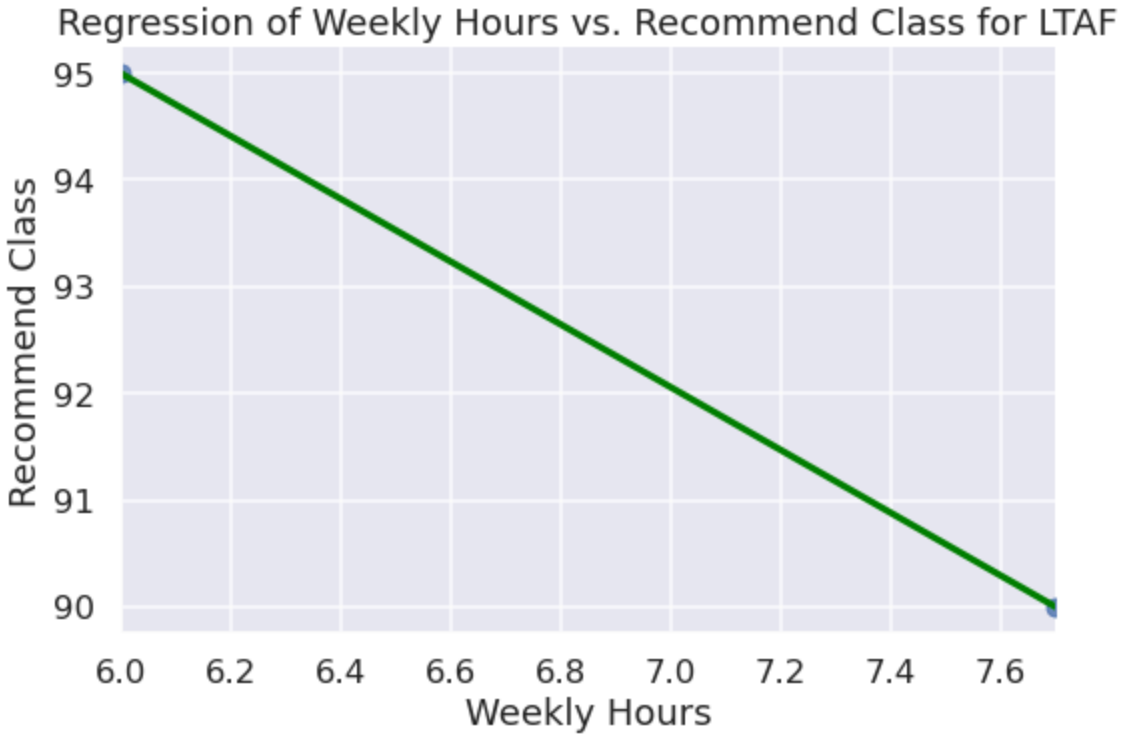


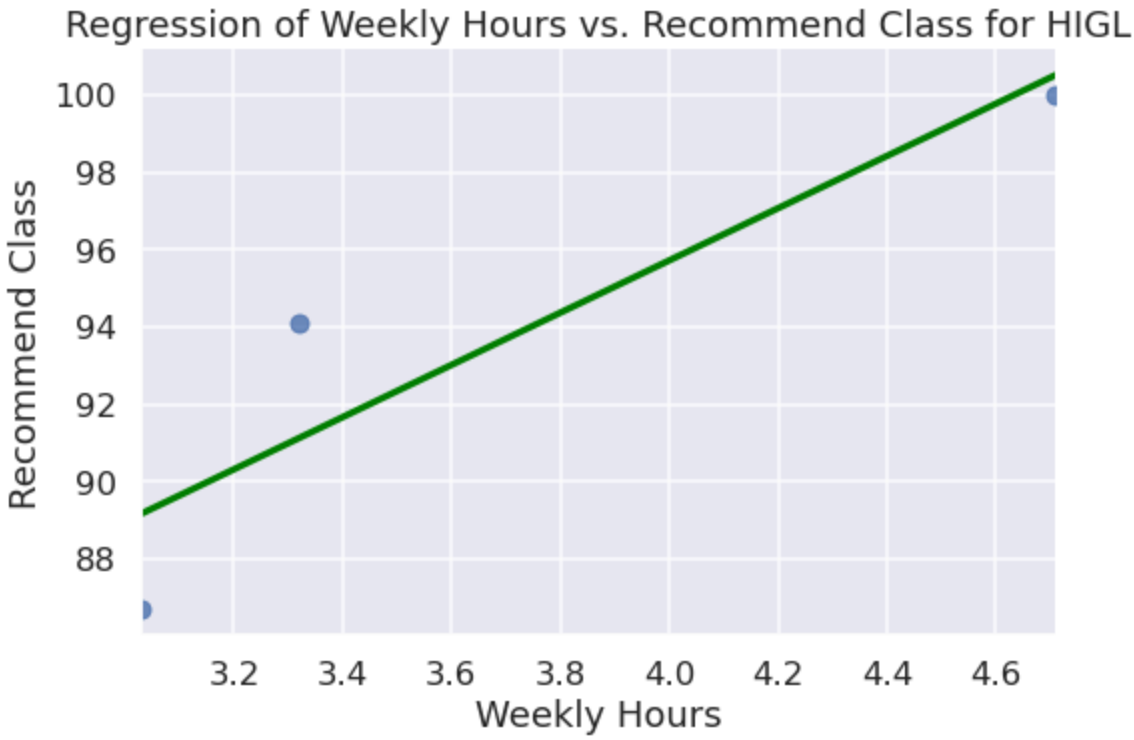
As we can tell from how the distributions of both weekly_hours and recommend_class greatly varies when departments are taken into account, we should separate by departments to closer analyze the relationship between weekly_hours and recommend_class.

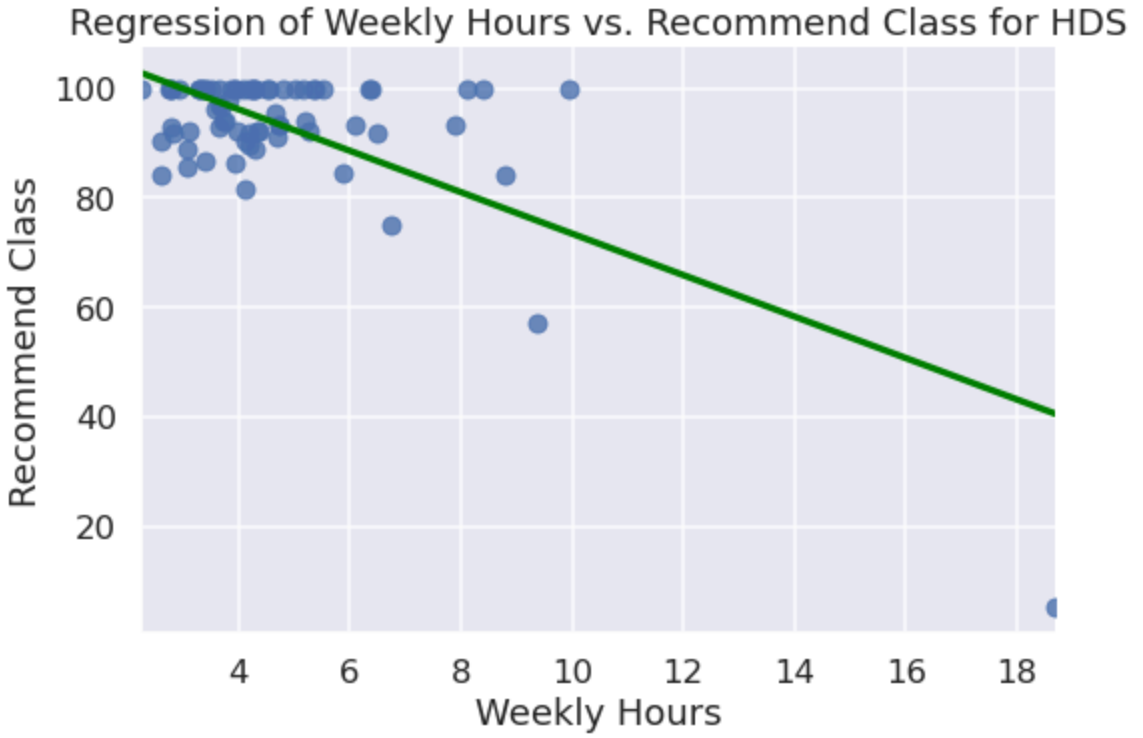
Analysis on Correlation of Weekly_Hours vs Recommend_Class Separated by Departments

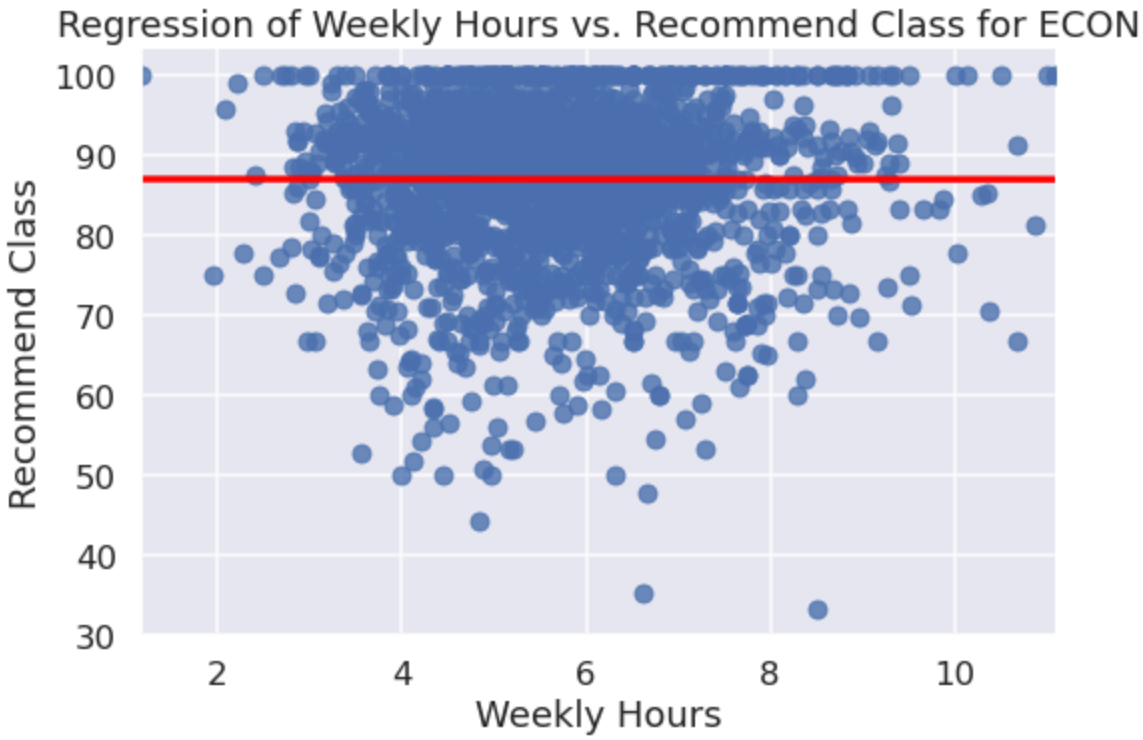
- We will use scatterplots to get an idea of how the data looks and use linear_regression to show the relationship between Weekly_Hours and Recommend_Class with courses separated into upper division or not upper division courses.
- To better visualize the data, we'll color the line of regression for the top 3 correlations as green and the lowest 3 correlations as red.

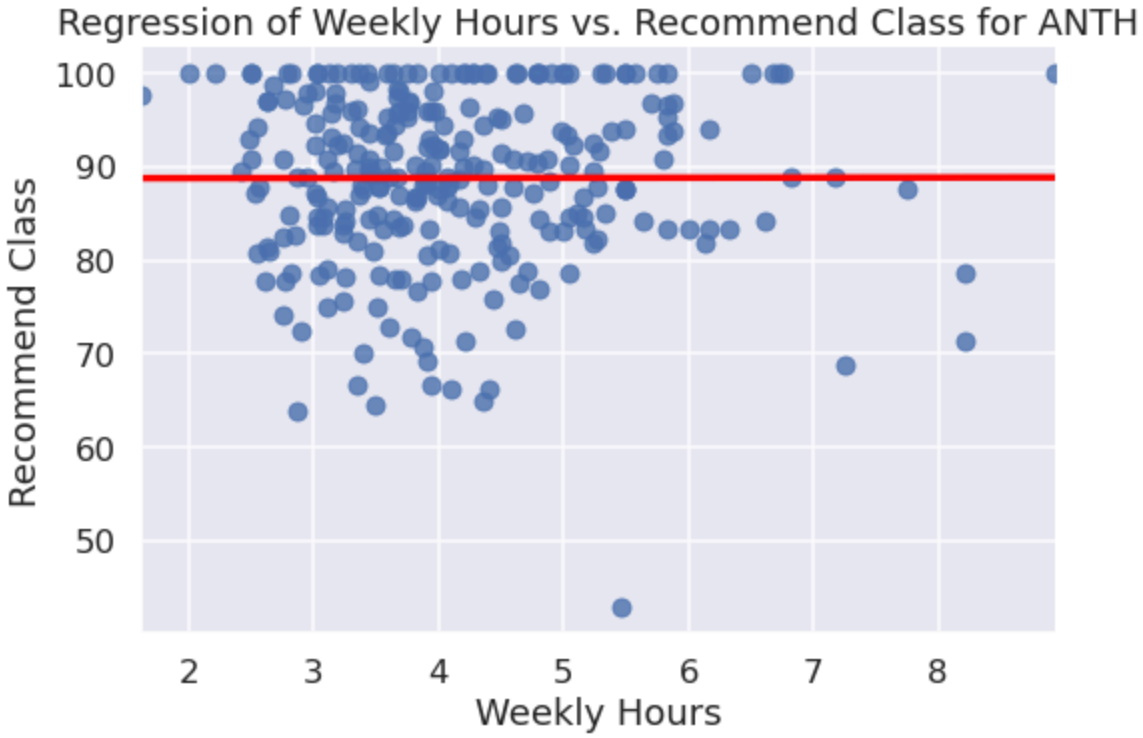
In [9]:
```
# Calculate correlation coefficients for each department
correlations = df.groupby('Department').apply(lambda x: x['Weekly_Hours'].corr(x['R

#apply fix to correlations to properly measure strongest (closest to -1 or +1) and
correlations = np.absolute(correlations)

# Sort to find top 3 strongest and lowest 3 correlations
top_3 = correlations.sort_values(ascending=False).head(3)
lowest_3 = correlations.sort_values().head(3)

# Function to perform linear regression and plot
def plot_regression(data, title, color):
    sns.lmplot(x='Weekly_Hours', y='Recommend_Class', data=data, aspect=1.5, ci=Non
    plt.title(f'Regression of Weekly Hours vs. Recommend Class for {title}')
    plt.xlabel('Weekly Hours')
    plt.ylabel('Recommend Class')

# Plotting top 3 strongest correlations
for dept in top_3.index:
    plot_regression(df[df['Department'] == dept], dept, 'green')

# Plotting lowest 3 correlations
for dept in lowest_3.index:
    plot_regression(df[df['Department'] == dept], dept, 'red')

plt.show()
```

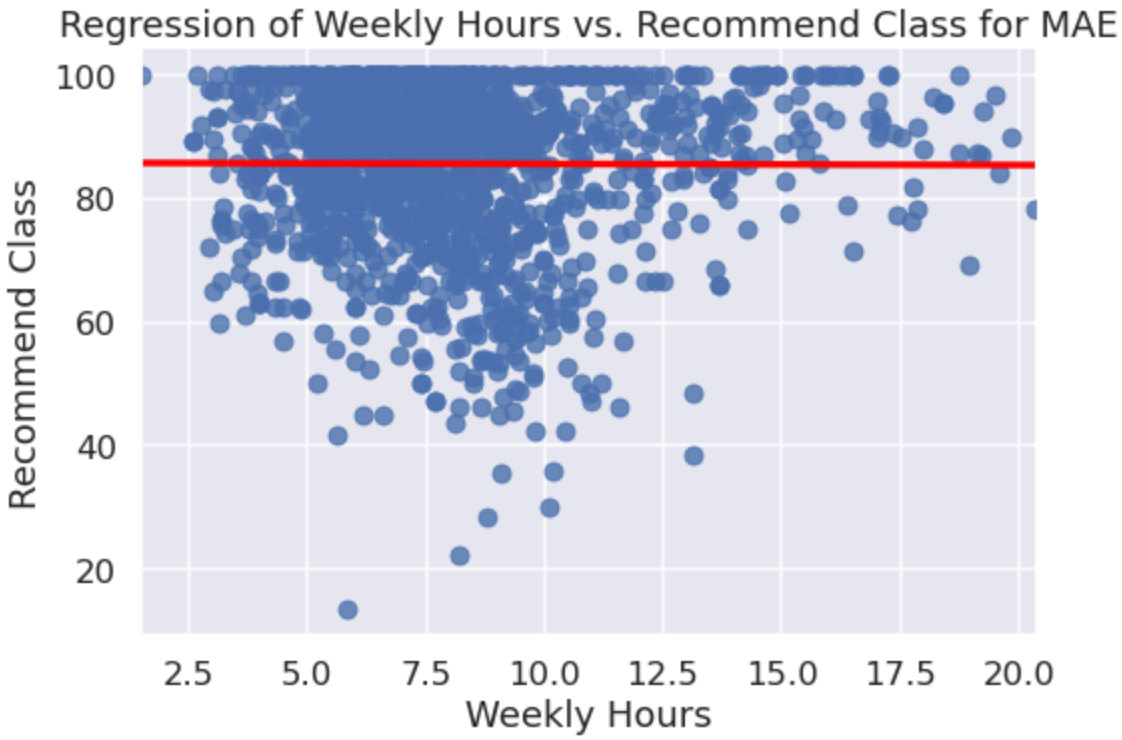

Regression of Weekly Hours vs. Recommend Class for LTAF
Recommend Class
95
94
93
92
91
90
6.0
6.2
6.4
6.6
6.8
7.0
7.2
7.4
7.6
Weekly Hours


Regression of Weekly Hours vs. Recommend Class for HIGL
Recommend Class
100
98
96
94
92
90
88
3.2
3.4
3.6
3.8
4.0
4.2
4.4
4.6
Weekly Hours


Regression of Weekly Hours vs. Recommend Class for HDS
Recommend Class
100
80
60
40
20
4
6
8
10
12
14
16
18
Weekly Hours


Regression of Weekly Hours vs. Recommend Class for ECON
Recommend Class
100
90
80
70
60
50
40
30
2
4
6
8
10
Weekly Hours





From our plots above, we can see that the top 3 strongest correlations we found were within the African Literature (LTAF) department, the Global History (HIGL) department, and the Human Developmental Sciences (HDS) department. Their linear regression lines show a significantly stronger relationship than the lowest 3 correlations within our dataset and look much cleaner in terms of plotting, so we can further investigate these specific departments to see why that might be the case.

In [10]:
```
#filter dataframe by LTAF department
df_LTAF = df[df['Department'] == 'LTAF']
df_LTAF
```

Out[10]:

	Instructor	Course	Quarter	Total_Enrolled	Total_CAPEs_Given	Recommend_Class
49663	Cancel Robert	LTAF 120 - Lit & Film of Modern Africa (A)	SP11	21	10	90.0
60579	Cancel Robert	LTAF 120 - Lit & Film of Modern Africa (A)	SP08	21	20	95.0

Here we can see that there are only two LTAF courses within our dataframe. So when we plot their data, a very strong correlation between weekly hours and recommend class score follows since there are only two datapoints to compare. In our data visualization above, it looks like there is a significant correlation between weekly hours and recommend class since there is a very obvious negative slope to the line. But when we examine the data this way, it turns out that the recommend class scores for the two LTAF courses in our dataset are not actually that far apart - 90% and 95%!

In [11]:
```
#filter dataframe by HIGL department
df_HIGL = df[df['Department'] == 'HIGL']
df_HIGL
```

Out[11]:

	Instructor	Course	Quarter	Total_Enrolled	Total_CAPEs_Given	Recommend_Class	F
522	Ivey James Alexander	HIGL 127 - Sport in the Modern World (A)	SP23	56	17	94.1	
3218	Edington Claire Ellen	HIGL 114 - History of Modern Vietnam (A)	FA22	58	19	100.0	
6556	Edelman Robert S.	HIGL 127 - Sport in the Modern World (A)	WI22	50	15	86.7	

Similar to our table for the LTAF courses, there are only three HIGL courses for us to analyze, leading us to compute a very strong correlation between weekly hours and recommended class score for this department. We can also see that between these three courses, the overall recommend class score is relatively high with 86.7% being the lowest. We can also roughly see a pattern where the higher scores also have a greater average weekly hours, and with only three points representing the HIGL department in the model, would contribute to what seems to be a positive correlation between average weekly hours and recommend class score.

Something else to note from this filtered dataframe is that HIGL 114, a course offerred in Fall Quarter 2022, has an outstanding recommend class score of 100%. It seems quite unrealistic for any course to reach 100% recommend class, especially if 19/58 students filled out the evaluation survey for it. The likelihood of all 19 of these students recommending this class wholeheartedly is low but could be attributed to the fact that some courses and professors at UCSD offer extra credit if a target number of students fill out the CAPES survey. This might be the case for this course and if it is true, would be skewing their actual opinion of the class to be higher than it theoretically would be if the survey responses were given without incentive.

In [12]:
```
#filter dataframe by HDS department
df_HDS = df[df['Department'] == 'HDS']
df_HDS
```

Out[12]:

	Instructor	Course	Quarter	Total_Enrolled	Total_CAPEs_Giv
503	Ackerman Farrell	HDS 1 - Intro/HumanDevelopmentScience (A)	SP23	191	
504	Kamowski-Shakibai Margare	HDS 120 - Language Development (A)	SP23	73	
505	Rodriguez Victoria Calip	HDS 122 - Development/Social Cognition (A)	SP23	94	
506	Ishizuka Katie E	HDS 173 - Race Media and Identity (A)	SP23	44	
507	Bjorklund Peter	HDS 181 - Exper Proj/Human Dev. Resear (A)	SP23	24	
...	...	...	...	...	
17295	Wilder Linnea Lorene	HDS 111 - Evolution in Human Development (A)	FA19	63	
17296	Deak Gedeon O.	HDS 121 - The Developing Mind (A)	FA19	29	
17297	Blaise Jean G	HDS 171 - Diversity in Human Development (A)	FA19	37	
17298	Blaise Jean G	HDS 175 - PowerWealth&Inequality/HumDev (A)	FA19	39	
17299	Ark Wendy S	HDS 181 - Exper Proj/Human Dev. Resear (A)	FA19	36	

71 rows × 12 columns

Compared to the other two departments, the HDS department actually has a larger sample of 71 different representative courses in the dataset, so it more accurately represents a correlation between the average weekly hours a student reported for the courses in this department and their overall recommended score for this class.

Further analysis of the HDS department's original linear correlation model shows that there is an outlier within the HDS department that is skewing our analysis of its data.

In [13]:
```
#find out what the outlier is within the HDS department
df_HDS[df_HDS['Weekly_Hours'] > 17]
```

Out[13]:

	Instructor	Course	Quarter	Total_Enrolled	Total_CAPEs_Given	Recommend_Cla
17299	Ark Wendy S	HDS 181 - Exper Proj/Human Dev. Resear (A)	FA19	36	20	5

In [14]:
```
#summary overview of HDS weekly hours
df_HDS['Weekly_Hours'].describe()
```

Out[14]:
```
count    71.000000
mean      4.725493
std       2.350058
min       2.250000
25%       3.455000
50%       4.170000
75%       5.190000
max      18.700000
Name: Weekly_Hours, dtype: float64
```

We can see that HDS 181 had a reported average weekly hours of 18.7 hours! 20/36 students filled out the survey for this course, so over half the class reported a huge amount of hours needed and also reported a very low average recommend class score of 5.3%. If we take a look at the descriptive statistics of the HDS department, the mean of the average weekly hours reported in CAPEs was around 4.7 hours with a standard deviation of 2.35, making this a huge outlier in the HDS data. We can see that in this course, with a significantly average amount of weekly hours reported, the recommend class score was also remarkably low. Though this datapoint is an outlier and one of out of many, it does give us an interesting insight into whether or not our hypothesis is supported by the data.

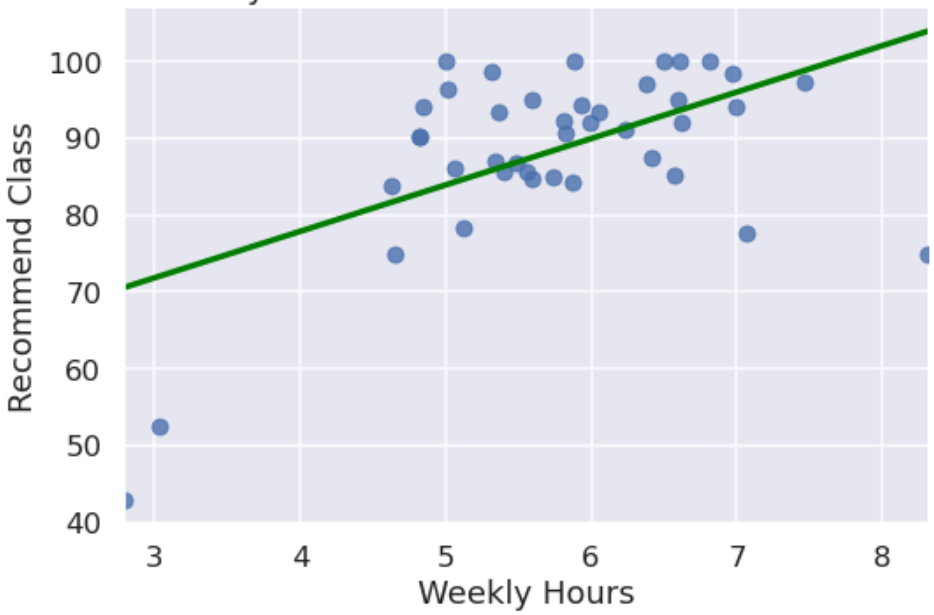
Now we take into account upper division standing:

- The reason we need to account for upper division standing is that upper division courses are generally seen as more rigorous and time consuming than lower division courses. Upper division courses tend to be more difficult than lower division courses as well.
- To visualize the correlation of weekly hours to course satisfaction(measured through Recommend_Class), we will graph the top 3 strongest correlations and lowest 3 correlations for both the non-upper division courses and the upper division courses.
- In order to better understand and compare the graphs, we'll color the line of regression for the top 3 correlations as green and the lowest 3 correlations as red for each respective section(non-upper division as well as upper division)

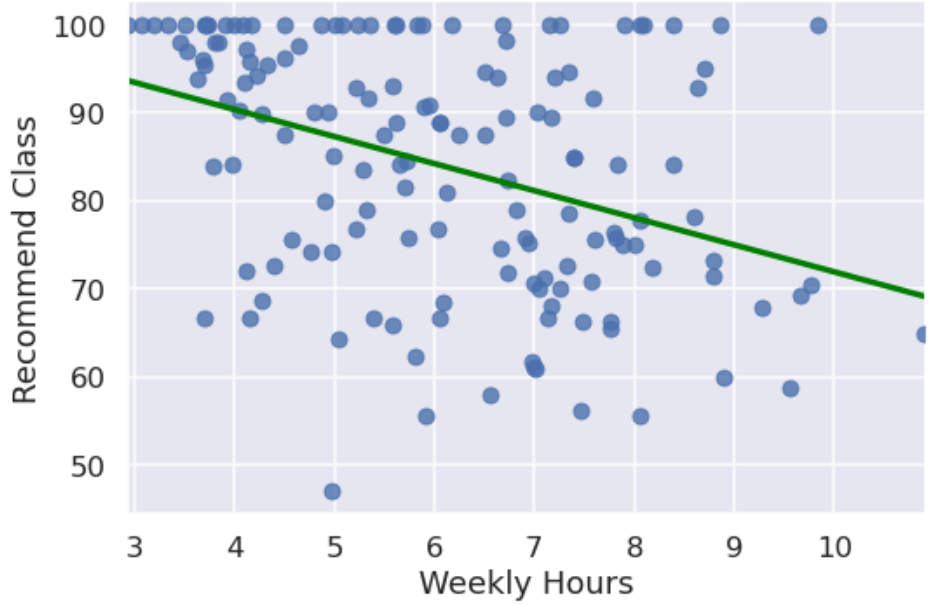
In [15]:
```
#sort into upper and lower division to see if departments are consistent
correlationsLower = df[df['Upper_Division'] == False]
```

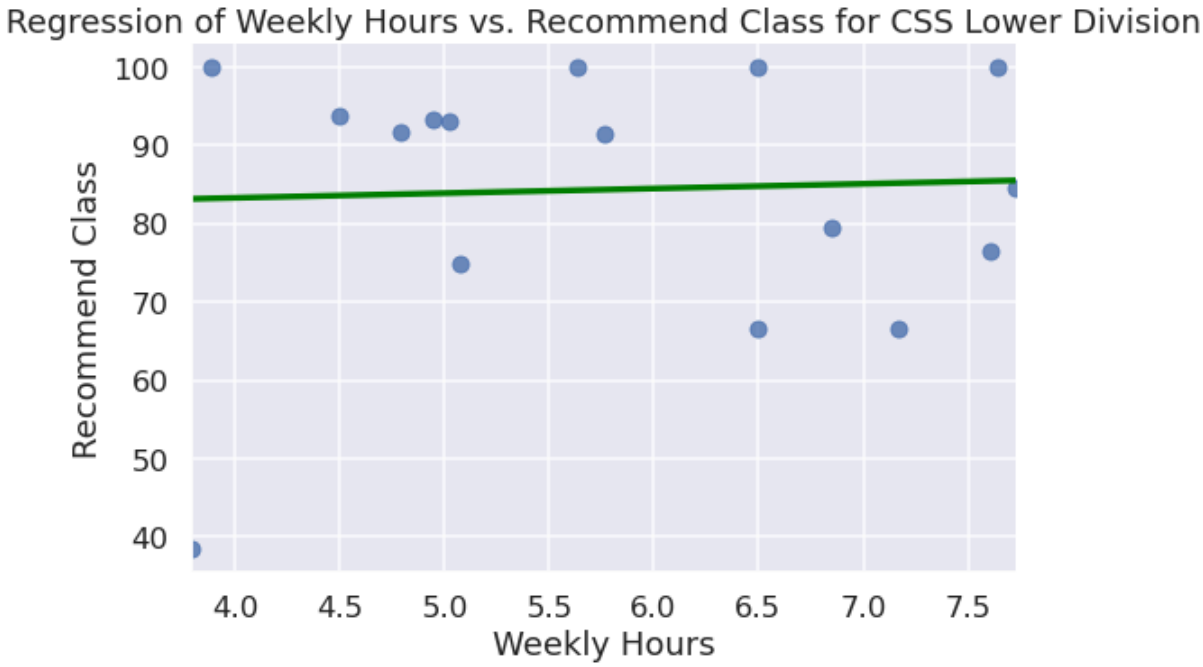
```
correlationsUpper = df[df['Upper_Division'] == True]

#calculate the correlation for upper and lower separately
correlationsLower = correlationsLower.groupby('Department').apply(lambda x: x['Week
correlationsUpper = correlationsUpper.groupby('Department').apply(lambda x: x['Week

correlationsLower = np.absolute(correlationsLower)
correlationsUpper = np.absolute(correlationsUpper)

# Sort to find top 3 strongest and lowest 3 correlations
top_3 = correlationsLower.sort_values(ascending=False).head(3)
lowest_3 = correlationsLower.sort_values().head(3)

# Function to perform linear regression and plot
def plot_regression_lowdiv(data, title, color):
    sns.lmplot(x='Weekly_Hours', y='Recommend_Class', data=data, aspect=1.5, ci=Non
    plt.title(f'Regression of Weekly Hours vs. Recommend Class for {title} Lower Di
    plt.xlabel('Weekly Hours')
    plt.ylabel('Recommend Class')

# Plotting top 3 strongest correlations
for dept in top_3.index:
    plot_regression_lowdiv(df[df['Department'] == dept], dept, 'green')

# Plotting lowest 3 correlations
for dept in lowest_3.index:
    plot_regression_lowdiv(df[df['Department'] == dept], dept, 'red')

plt.show()



# Sort to find top 3 strongest and lowest 3 correlations
top_3 = correlationsUpper.sort_values(ascending=False).head(3)
lowest_3 = correlationsUpper.sort_values().head(3)

# Function to perform linear regression and plot
def plot_regression_updiv(data, title, color):
    sns.lmplot(x='Weekly_Hours', y='Recommend_Class', data=data, aspect=1.5, ci=Non
    plt.title(f'Regression of Weekly Hours vs. Recommend Class for {title} Upper Di
    plt.xlabel('Weekly Hours')
    plt.ylabel('Recommend Class')

# Plotting top 3 strongest correlations
for dept in top_3.index:
    plot_regression_updiv(df[df['Department'] == dept], dept, 'green')

# Plotting lowest 3 correlations
for dept in lowest_3.index:
    plot_regression_updiv(df[df['Department'] == dept], dept, 'red')

plt.show()
```

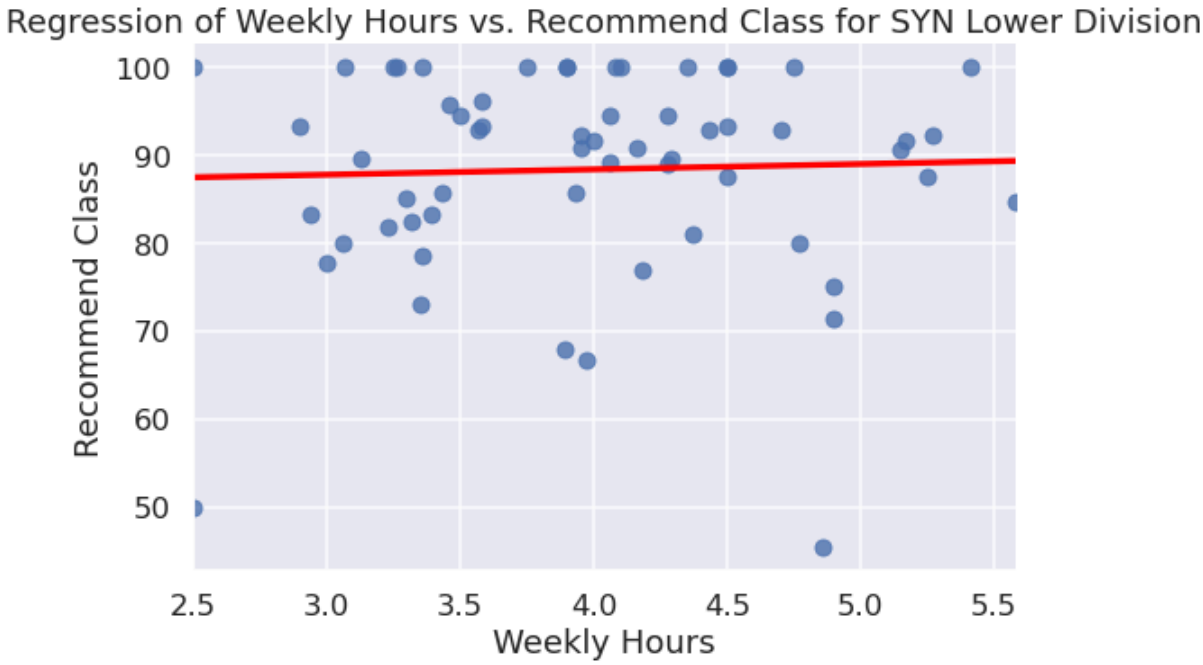
Regression of Weekly Hours vs. Recommend Class for DSGN Lower Division

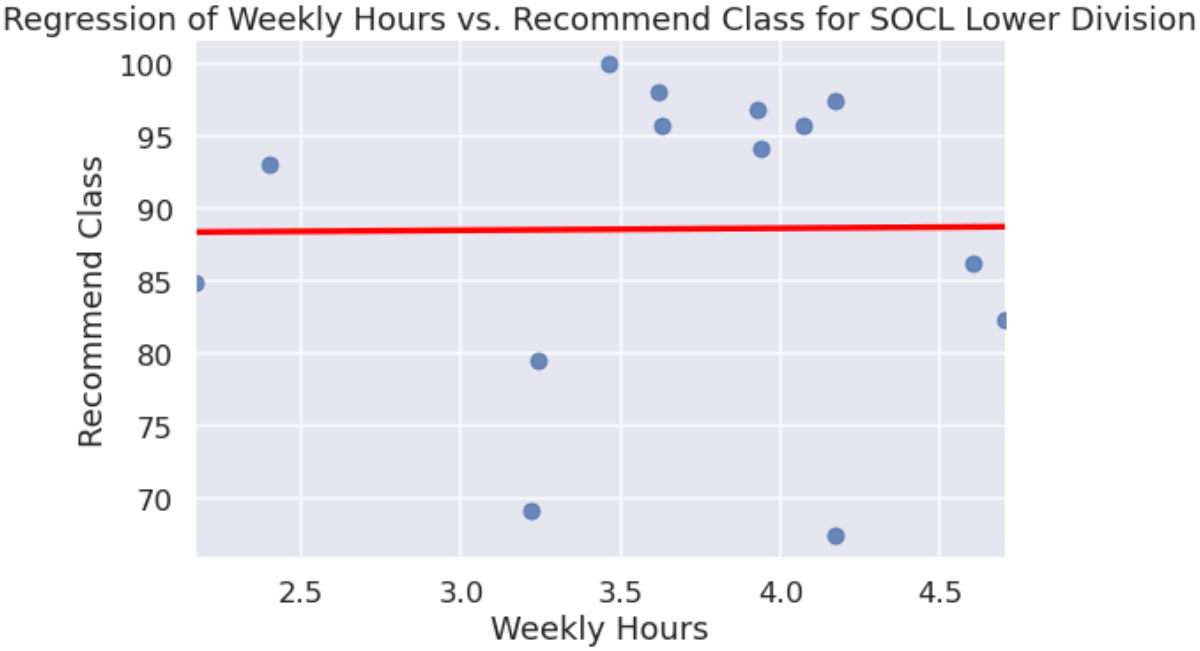


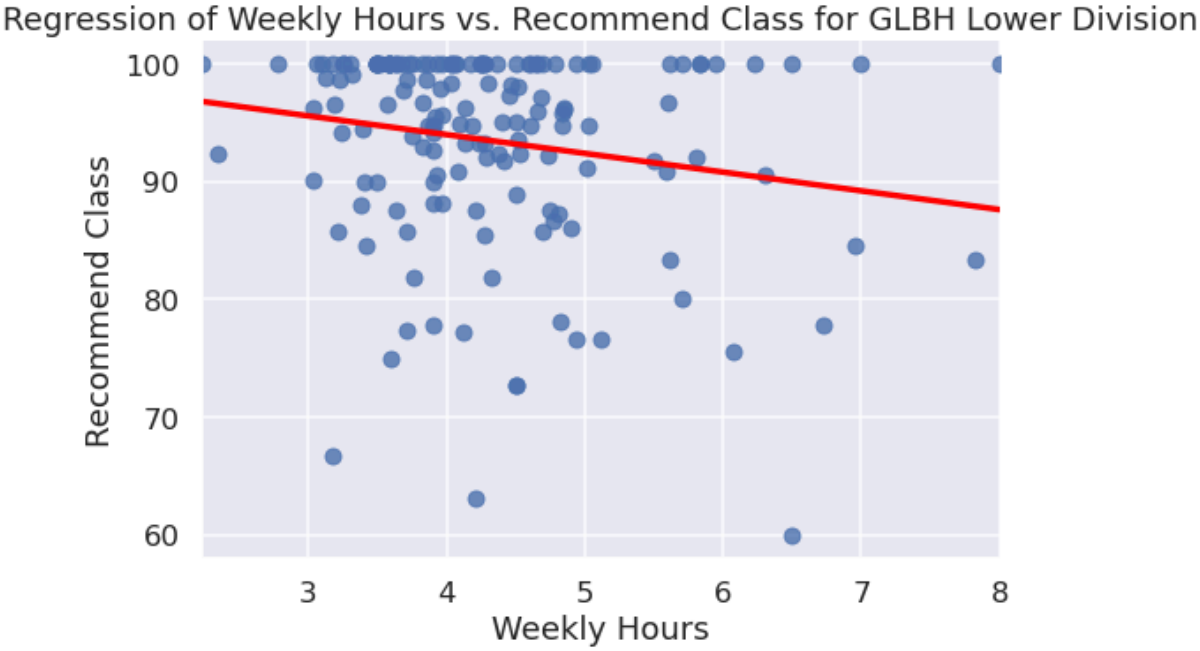
Regression of Weekly Hours vs. Recommend Class for WCWP Lower Division

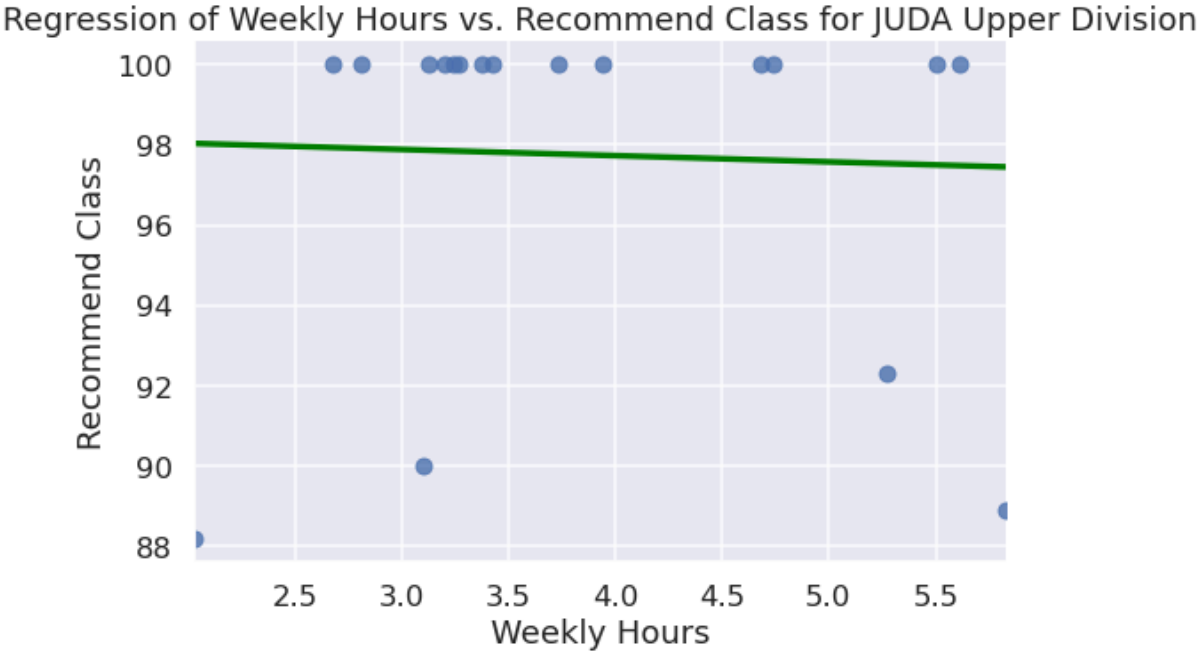


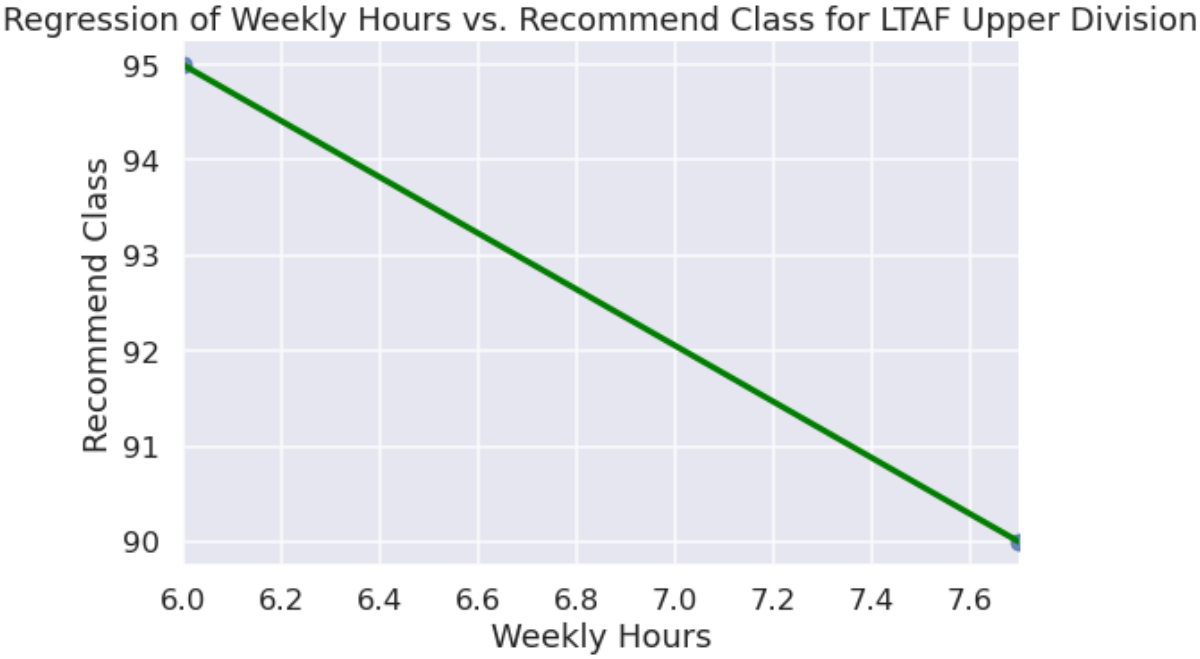

Regression of Weekly Hours vs. Recommend Class for CSS Lower Division
Recommend Class
100
90
80
70
60
50
40
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
Weekly Hours


Regression of Weekly Hours vs. Recommend Class for SYN Lower Division
Recommend Class
100
90
80
70
60
50
2.5
3.0
3.5
4.0
4.5
5.0
5.5
Weekly Hours

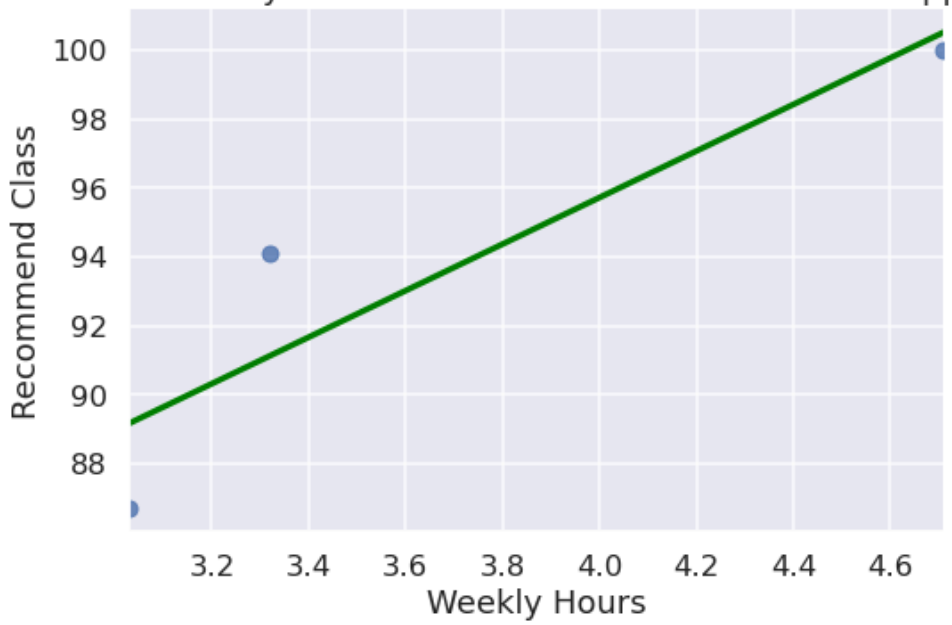

Regression of Weekly Hours vs. Recommend Class for SOCL Lower Division
Recommend Class
100
95
90
85
80
75
70
2.5
3.0
3.5
4.0
4.5
Weekly Hours


Regression of Weekly Hours vs. Recommend Class for GLBH Lower Division
Recommend Class
100
90
80
70
60
3
4
5
6
7
8
Weekly Hours

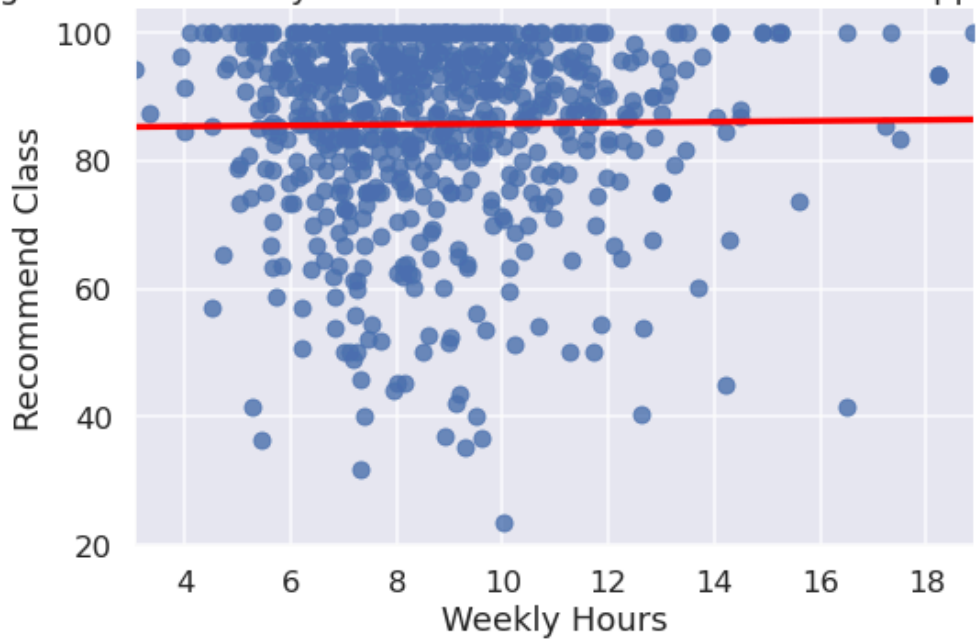

Regression of Weekly Hours vs. Recommend Class for JUDA Upper Division
Recommend Class
100
98
96
94
92
90
88
2.5
3.0
3.5
4.0
4.5
5.0
5.5
Weekly Hours

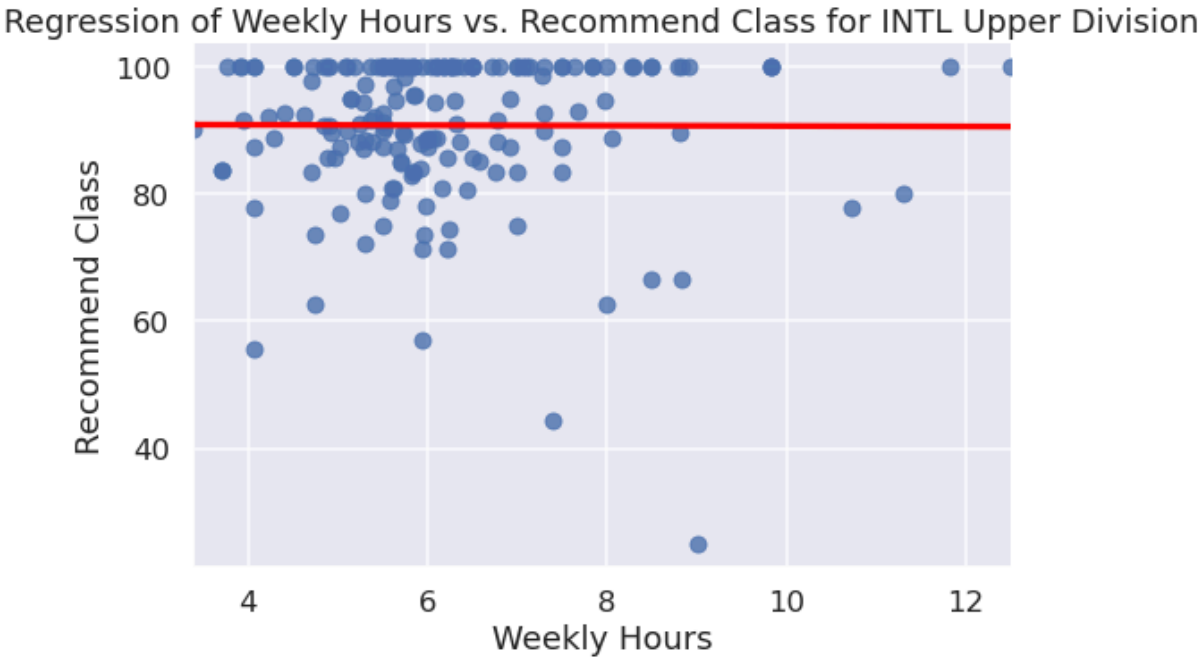

Regression of Weekly Hours vs. Recommend Class for LTAF Upper Division
Recommend Class
95
94
93
92
91
90
6.0
6.2
6.4
6.6
6.8
7.0
7.2
7.4
7.6
Weekly Hours

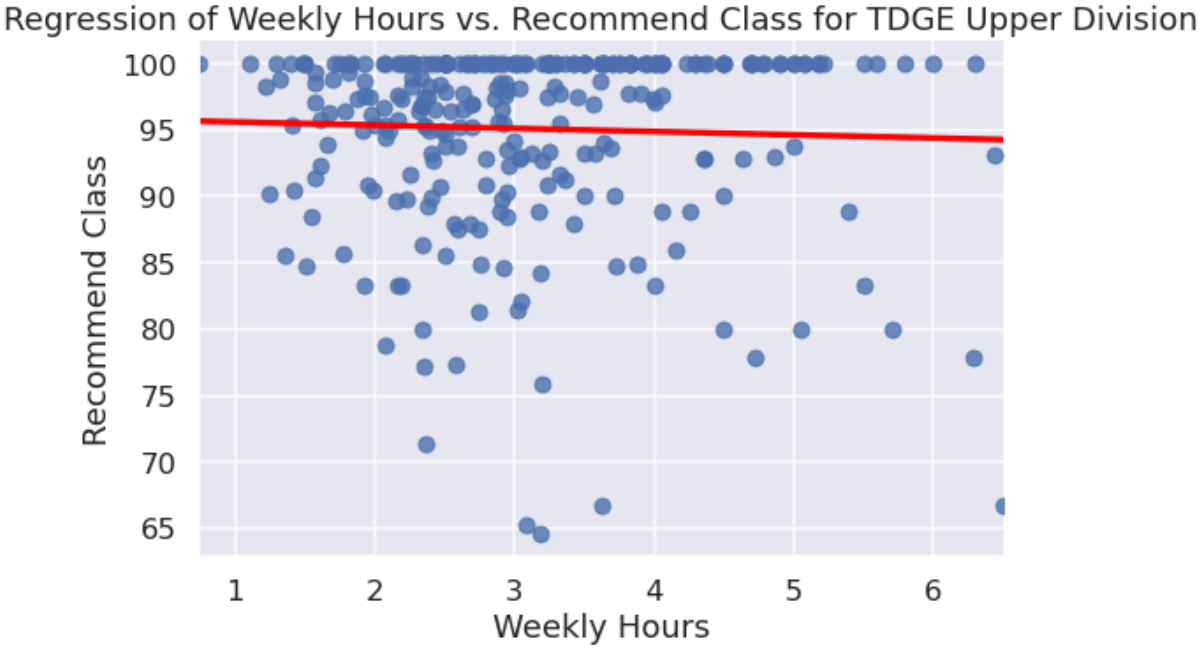
Regression of Weekly Hours vs. Recommend Class for HIGL Upper Division



Regression of Weekly Hours vs. Recommend Class for SE Upper Division







Findings For Correlation with Consideration for Upper Division Standing

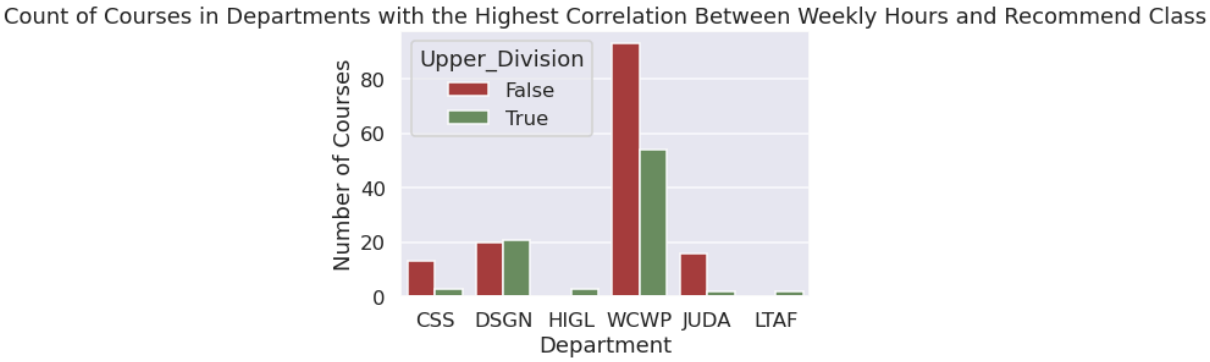
After sorting the data into upper and lower division, we found that the departments with the strongest correlation scores in lower division courses regards to Weekly_Hours and Recommend_Class were DSGN, WCWP, and CSS while the departments with the strongest correlation scores in upper divison courses were JUDA, LTAF, and HIGL. In terms of top correlation scores, none of the departments remained consistent when separated into upper and lower division.

Inconsistencies with Correlation Graphs of Courses by Department and Upper-Division Standing

- We could note that some of the top correlation scores were from departments that had only a handful of courses, let's explore how this could affect our correlation scores.

In [16]:
```
#subset of dataframe showing the 3 departments with the  highest correlations in up
highest_lower_upper = ['DSGN','WCWP','CSS','JUDA','LTAF','HIGL']
HLowUpper_filtered = df[df['Department'].isin(highest_lower_upper)]

#create countplot of filtered dataframe
sns.countplot(x='Department', data = HLowUpper_filtered, hue='Upper_Division', pale
plt.title('Count of Courses in Departments with the Highest Correlation Between Wee
plt.ylabel('Number of Courses')
```

Out[16]:
```
Text(0, 0.5, 'Number of Courses')
```



Similar to what we investigated in section 3 (weekly hours vs recommend class, filtered by department), if we take a look at the course count for top 3 highest correlations in the lower division departments (DSGN, WCWP, CSS), we see that there are around 20 or less DSGN and CSS courses. This lower class count may contribute to why their correlation scores were so high. If we take a look at the course count for the top 3 highest correlations in the upper division departments (JUDA, LTAF, HIGL), we see that there are less than 10 courses for each of those departments. Taking this into account, we can see that for the upper division classes in our analysis, such a low proportion of representation in our dataset may be skewing our results and suggesting a higher correlation between weekly hours and recommended class than there should be when filtering by upper division departments.

Consider departments separated by upper division status that might have NaN correlation scores

In [17]:
```
# Check for NaN correlation scores
nan_lower = correlationsLower[correlationsLower.isna()]
print("Departments with NaN correlation scores in lower division:", nan_lower)
nan_upper = correlationsUpper[correlationsUpper.isna()]
print("Departments with NaN correlation scores in upper division:", nan_upper)
```

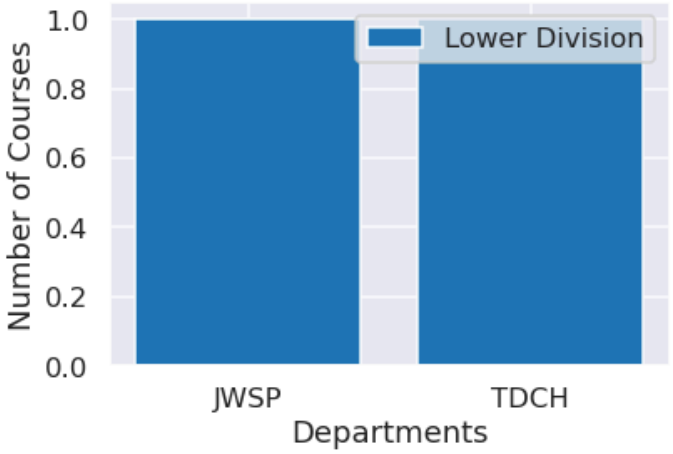
```
Departments with NaN correlation scores in lower division: Department
JWSP   NaN
TDCH   NaN
dtype: float64
Departments with NaN correlation scores in upper division: Department
LTRU   NaN
STPA   NaN
TDCH   NaN
TDDE   NaN
TDDM   NaN
TDTR   NaN
dtype: float64
```

We can note that there are several departments with lower division courses that have NaN correlation scores. This also applies to the departments with upper division courses. We will explore and analyze these NaN correlation scores below:

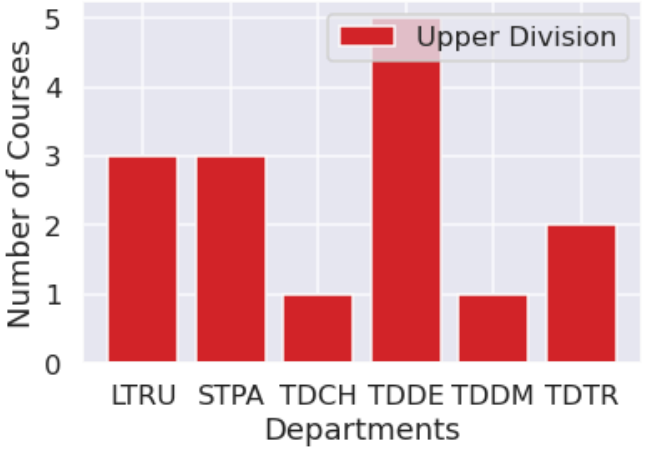
In [18]:
```
departments_with_nan_lower = nan_lower.index.tolist()
departments_with_nan_upper = nan_upper.index.tolist()

num_courses_lower_nan = df[(df['Upper_Division'] == False) & (df['Department'].isin
num_courses_upper_nan = df[(df['Upper_Division'] == True) & (df['Department'].isin(

# Plot number of courses in departments with NaN correlation scores
plt.bar(num_courses_lower_nan.index, num_courses_lower_nan, color='tab:blue', label
plt.title('Number of Courses in Lower Division Departments with NaN Correlation Sco
plt.xlabel('Departments')
plt.ylabel('Number of Courses')
plt.legend()
plt.show()

plt.bar(num_courses_upper_nan.index, num_courses_upper_nan, color='tab:red', label=
plt.title('Number of Courses in Upper Division Departments with NaN Correlation Sco
plt.xlabel('Departments')
plt.ylabel('Number of Courses')
plt.legend()
plt.show()


# Function to plot histograms to see variability in Recommended Class Scores
def plotUniqueScoresCountLower(depts):
    unique_scores_counts_lower = []

    for dept in depts:
        lower_data = df[(df['Department'] == dept) & (df['Upper_Division'] == False

        # Count unique recommended class scores in lower division
        unique_scores_count_lower = lower_data['Recommend_Class'].nunique()
        unique_scores_counts_lower.append(unique_scores_count_lower)

    # Plot bar plot for lower division
    plt.bar(depts, unique_scores_counts_lower, color='blue', label='Lower Division'
    plt.title('Count of Unique Recommended Class Scores for Departments with NaN Co
    plt.xlabel('Departments')
    plt.ylabel('Count of Unique Scores')
```

```
    plt.legend()
    plt.show()

def plotUniqueScoresCountUpper(depts):
    unique_scores_counts_upper = []

    for dept in depts:
        upper_data = df[(df['Department'] == dept) & (df['Upper_Division'] == True)

        # Count unique recommended class scores in upper division
        unique_scores_count_upper = upper_data['Recommend_Class'].nunique()
        unique_scores_counts_upper.append(unique_scores_count_upper)

    # Plot bar plot for upper division
    plt.bar(depts, unique_scores_counts_upper, color='orange', label='Upper Divisio
    plt.title('Variation in Recommended Class Scores for Departments with NaN Corre
    plt.xlabel('Departments')
    plt.ylabel('Count of Unique Scores')
    plt.legend()
    plt.show()

# Plot the bar plots for count of unique recommended class scores in lower and uppe
plotUniqueScoresCountLower(departments_with_nan_lower)
plotUniqueScoresCountUpper(departments_with_nan_upper)

# Remove departments with only 1 course in their respective division
df = df[~((df['Upper_Division'] == False) & (df['Department'].isin(['JWSP', 'TDCH']
df = df[~((df['Upper_Division'] == True) & (df['Department'].isin(['TDDM', 'TDCH']))

# Set Correlation scores for departments with no variability to be 0
departments_with_one_recommend_score_upper = df[df['Upper_Division'] == True].group
departments_with_one_recommend_score_upper = departments_with_one_recommend_score_u
correlationsUpper.loc[departments_with_one_recommend_score_upper] = 0

# Plot lower division histogram
plt.hist(correlationsLower, bins=20, color='blue')
plt.title('Histogram of Correlation Coefficients (Lower Division)')
plt.xlabel('Correlation Coefficient')
plt.ylabel('Number of Departments')
plt.show()

# Plot upper division histogram
plt.hist(correlationsUpper, bins=20, color='orange')
plt.title('Histogram of Correlation Coefficients (Upper Division)')
plt.xlabel('Correlation Coefficient')
plt.ylabel('Number of Departments')
plt.show()
```

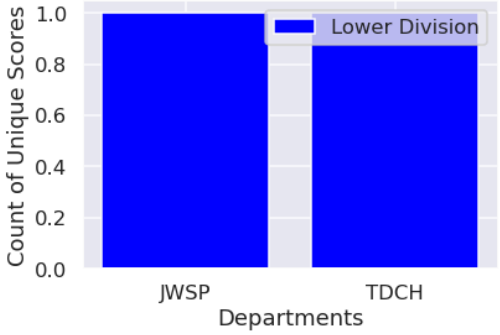
Number of Courses in Lower Division Departments with NaN Correlation Scores



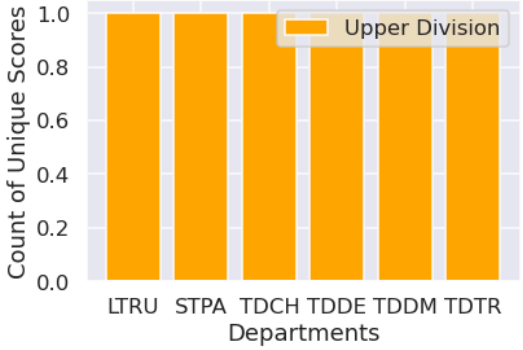
Number of Courses in Upper Division Departments with NaN Correlation Scores

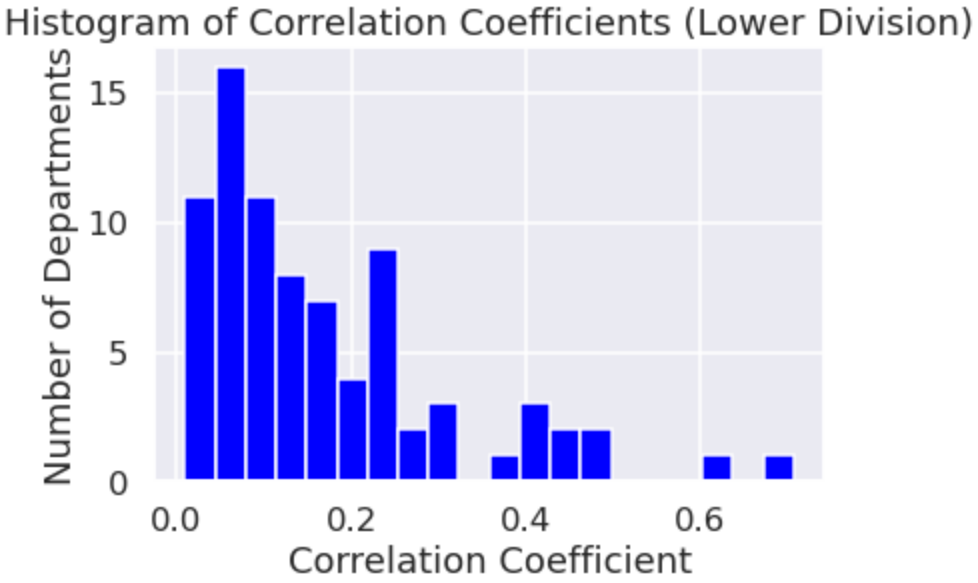


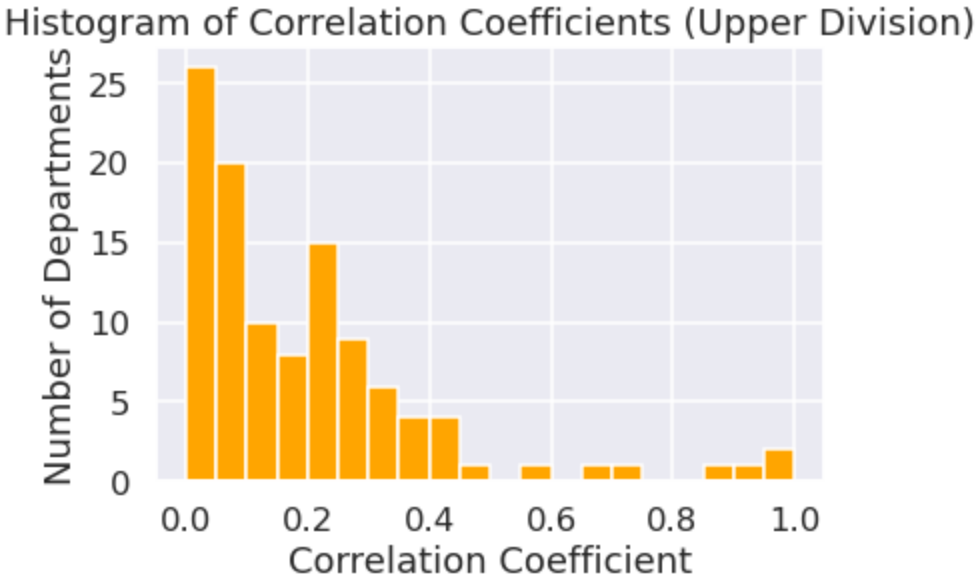
Count of Unique Recommended Class Scores for Departments with NaN Correlation Scores (Lower Division)



Variation in Recommended Class Scores for Departments with NaN Correlation Scores (Upper Division)







After observing the departments that turned out to have NaN correlation coefficients, we can see that for lower division departments, JWSP had NaN correlation coefficients. For upper division departments, LTRU, STPA, TDCH, TDDE, TDDM, and TDTR had NaN correlation coefficients. For lower division JWSP and TDCH and upper division TDCH and TDDM, this can be explained since they only had 1 course in their respective departments, which can be observed in the plot titled "Number of Courses in Each Department with NaN Correlation Scores", making it unable to form a correlation. Since it is unable to form a correlation for these departments, we have decided to remove them from our observation.

For upper division departments LTRU, STPA, TDDE, and TDTR, there was no variability amongst the 'Recommended Class' column, which can be observed in the graphs labeled "Variation in Recommended Class Scores for Departments with NaN Correlation Scores". In this case, we replaced the correlation coefficient for the following departments to be 0. This can be noticed in the histograms for the recommended classes for each department as there is only 1 unique value per department. Then we plotted histograms of the number of

departments that had which correlation coefficient for both lower division and upper division and it can be noted that both generally have a right skewesd normal distribution centered around 0.

Ethics & Privacy

Potential Biases:

- Students might perceive different departments as more difficult or the departments have specific courses that are overall more difficult that might skew the correlation statistics (confounding bias that may result from different student experiences and familarity with specific topics).
- Additionally, students may over-estimate or under-estimate how many hours they needed per week (response bias).
- Upper-division courses are generally accepted as more difficult than lower division courses, so the difficulty of the course would lower the satisfaction of the course (confounding bias, expectation bias).
- Some courses might inherently require more time for students to learn the material but might not necessarily be more difficult or affect overall satisfaction with the course, which would not be properly represented in our data (measurement bias, since we have no way of measuring if such courses would require more time with the survey data that we have).
- We don't believe that there would be issues from our topic area, data, and/or analyses that would be potentially problematic in terms of data privacy and equitable impact.

Handling biases:

- We could analyze the data separated by department and further separate the data by lower and upper division courses to address the issue of certain departments/upper division classes being perceived as more difficult (matching variables that are more similar to isolate possible confounding variables).
- One way we could account for students potentially over or underestimating their study hours is to observe how their hours impacts their Average Grade Expected as well, so we have another variable to compare their overall satisfaction to.
    - Additionally, taking a look at the Average Grade Expected will help us account for if a course may require more learning time by design but is not considered too difficult, then a student is likely to expect a higher grade in the course.

Discusison and Conclusion

Earlier, we discussed research and studies completed by other researchers like Brian K. Coffey, Howell and Buck, Janet M. Ferguson, Amy E. DeFelice, M Salinda Weerasinghe, R.

Lalitha and, S. Fernando. Their respective findings include the relationships of time management with student satisfaction, course load with satisfaction, course format and duration with satisfaction, and quality of teaching with satisfaction. Our findings extend the previous research and studies by discussing the relationship between time spent, which is similar to course load, and satisfaction with courses. Instead of discussing satisfaction of students in general, our work focuses on a more specific area of satisfaction, which is how satisfied are students with their courses. Previous findings on the link between quality of teaching, course format, and satisfaction led us to closely examine and account for other factors that may skew our findings, such as the difficulty of the courses, which are gauged through data showing student grade received and upper division standing of the courses.

We used CAPEs data, which is closely described in the background section, and we related time spent and course satisfaction with weekly_hours and recommend_class respectively, so we will use these terms interchangeably. According to OLS regression done with weekly_hours and recommend_class, with an alpha value of 0.01, weekly_hours is statistically significant in determining recommend_class. Even with Avg_Grade_Expected, which could be related to course difficulty, taken into account as a potential confounding variable, weekly_hours remained statistically significant. After separating the data into departments and upper division/lower division courses, we found that the correlation of time spent and course satisfaction differed among departments as well as their respective upper and lower division courses. Upon closer inspection, some departments had very strong correlation scores or even nan correlation scores due to the lack of data or variability among data.

We can conclude that although time spent has a statistically significant relationship with course satisfaction, and this holds true when taking into account certain potentially confounding variables, some departments have a stronger correlation of time spent and course satisfaction while a majority of the departments seem to have a weak correlation. Overall, our findings could be improved with the addition of more data to increase the varaibility of course data from existing departments, and this necessity is shown in the nan correlation scores in certain departments. More data would also lead the overall distribution of correlation scores among the departments to be less right skewed; this is because some departments have strong correlation scores due to having only a few datapoints. This work may impact society by leading UCSD students to spend less time on courses from certain departments. Ethically speaking, UCSD students should not proceed in this manner because factors about the students in each of the courses may vary and cannot be accounted for. This research only serves as a possible explanation for decreasing or increasing course satisfaction as a result of time spent.

Team Contributions

- **Henry Lam**: Abstract, data setup and cleaning, data analysis, scatterplot, boxplots, linear regression assumptions, OLS regression, regression graphs with upper division standing split, conclusion, ethics section

- **Alex Truong**: Background, histograms of correlations and course distributions
- **Diannie Natanauan**: Abstract, data description, signficiant amount of data analysis, lineplot over time, countplot, ethics section
- **Darryl Remulla**: Some data analysis, background
- **Sam Hormozian**: Regression graphs

In [ ]: