In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import re
import plotly.graph_objects as go
import seaborn as sns
import plotly.express as px
```

In [2]:
```
# Opens all the csv files for all the schools
```

In [3]:
```
ucsc_files = open('./184_project/UCSC files.csv')
df_ucsc = pd.read_csv(ucsc_files)

uci_files = open('./184_project/UCI/UCI files.csv')
df_uci = pd.read_csv(uci_files)

sdsu_files = open('./184_project/SDSU/SDSU files.csv')
df_sdsu = pd.read_csv(sdsu_files)

chico_files = open('./184_project/CHICO/CHICO files.csv')
df_chico = pd.read_csv(chico_files)

sfsu_files = open('./184_project/SF State/SF_State_Data.csv')
df_sfsu = pd.read_csv(sfsu_files)

harveymud_files = open('./184_project/Harvey_files.csv')
df_harvey = pd.read_csv(harveymud_files)

panoma_files = open('./184_project/Pomona_files.csv')
df_panoma = pd.read_csv(panoma_files)

rice_files = open('./184_project/RICE/RICE files.csv')
df_rice = pd.read_csv(rice_files)
```

In [4]:
```
#Goes through all the schools and cleans up the term to season + year
```

In [5]:
```
real_cols = ['num_people_enrolled', 'total_class_size', 'class_num', 'term',
'professor']
df_rice.columns = real_cols
df_rice
for row_num,i in enumerate(df_rice.iloc[:,3]):
    term_split = i.split('2')
    if term_split[0] == 'FALL':
        term_split[0] = 'Fall'
        term_split[1] = '2' + term_split[1]
    elif term_split[0] == 'SPRING':
        term_split[0] = 'Spring'
        term_split[1] = '2' + term_split[1]
    if len(term_split) == 3:
        term_split = [term_split[0], term_split[1] + '2']
    season_year = term_split[0] + ' ' + term_split[1]
    df_rice.at[row_num, 'term'] = season_year.upper()
```

In [6]:
```
for row_num, i in enumerate(df_harvey.iloc[:,3]):
    term_split = i.split(' ')
    if len(term_split[1]) < 4:
        year_last = term_split[1].split('0')
        term_split[1] = '200' + year_last[1]
    df_harvey.at[row_num, 'term'] = term_split[0] +' ' + term_split[1]
```

In [7]:
```
for row_num, i in enumerate(df_panoma.iloc[:,3]):
    term_split = i.split(' ')
    if len(term_split[1]) < 4:
        year_last = term_split[1].split('0')
        term_split[1] = '200' + year_last[1]
    df_panoma.at[row_num, 'term'] = term_split[0] + ' ' + term_split[1]
```

In [8]:
```
for row_num, i in enumerate(df_chico.iloc[:,3]):
    term_split = i.split('2')
    if term_split[0] == 'fa':
        term_split[0] = 'Fall'
        term_split[1] = '2' + term_split[1]
    elif term_split[0] == 'spr':
        term_split[0] = 'Spring'
        term_split[1] = '2' + term_split[1]
    if len(term_split) == 3:
        term_split = [term_split[0], term_split[1] + '2']
    season_year = term_split[0] + ' ' + term_split[1]
    df_chico.at[row_num, 'term'] = season_year.upper()
```

In [9]:
```
for row_num, i in enumerate(df_sfsu.iloc[:,1]):
    term_split = i.split(' ')
    season_year = term_split[0] + ' ' + term_split[1]
    df_sfsu.at[row_num, 'term'] = season_year.upper()

for row_num, class_num in enumerate(df_sfsu['class_number']):
    class_split = class_num.split(' ')
    df_sfsu.at[row_num, 'class_number'] = class_split[0] + class_split[1]
```

In [10]:
```
for row_num, i in enumerate(df_sdsu.iloc[:,3]):
    term_split = i.split(' ')
    if len(term_split[0]) == 1:
        term_split[0] = '200' + term_split[0]
    else:
        term_split[0] = '20' + term_split[0]
    df_sdsu.at[row_num, 'term'] = (term_split[1] + ' ' + term_split[0]).upper
()
```

In [11]:
```
for row_num, i in enumerate(df_uci.iloc[:,3]):
    term_split = i.split(' ')
    df_uci.at[row_num, 'term'] = (term_split[1] + ' ' + term_split[0]).upper()
```

In [12]:
```
for row_num, i in enumerate(df_ucsc.iloc[:, 3]):
    term_split = i.split(' ')
    if len(term_split[1]) < 4:
        year = term_split[1].split('0')
        new_year = year[0] + '00' + year[1]
        term_split[1] = new_year
    df_ucsc.at[row_num, 'term'] = (term_split[0] + ' ' + term_split[1]).upper
()
```

In [13]:
```
# Goes through all the data and adds the percentages of class to the end of th
e dataframe
```

In [14]:
```
def add_percentage(self, num_enroll, total_size):
    self['Percentage'] = 0.
    for row_num, i in enumerate(self.iloc[:, num_enroll]):
        try:
            float(self.iloc[row_num, total_size])
        except:
            self.iloc[row_num,total_size] = self.iloc[row_num, total_size].spl
it('W')[0]
        if float(self.iloc[row_num, total_size]) == 0:
            self.iloc[row_num, total_size] = self.iloc[row_num, num_enroll]
        if float(self.iloc[row_num, total_size]) == 0 and float(self.iloc[row_
num, num_enroll]) ==0:
            self.iloc[row_num, total_size ] = 1;
        percentage = float(self.iloc[row_num, num_enroll])/float(self.iloc[row
_num,total_size])
        if percentage > 1:
            percentage = 1
        if percentage <= 0:
            percentage = None
        self.at[row_num, 'Percentage'] = percentage
    return self
```

In [15]:
```
def clean_nan(self):
    self = self.replace(0.,np.NaN)
    self = self.dropna()
    return self
```

In [16]:
```
def clean_class_nums(self):
    for row_num,i in enumerate(self.iloc[:,2]):
        cs_classes = i.split(' ')
        self.iloc[row_num,2] = cs_classes[0].strip()
    return self
```

In [17]:
```
for num, i in enumerate(df_rice.iloc[:,0]):
    df_rice.iloc[num, 0] = re.sub('[^0-9]', '', i)
    if df_rice.iloc[num,0] == '500':
        df_rice.iloc[num,0] = 0
    df_rice.iloc[num, 1] = re.sub('[^0-9]', '', df_rice.iloc[num,1])
for num, i in enumerate(df_rice.iloc[:,2]):
    splitted = i.split()
    df_rice.iloc[num,2] = splitted[0] + splitted[1]
```

In [18]:
```
add_percentage(df_chico, 0, 1)
add_percentage(df_sfsu, 3, 4)
add_percentage(df_ucsc, 0, 1)
add_percentage(df_sdsu, 0, 1)
add_percentage(df_uci, 0, 1)
add_percentage(df_harvey, 0, 1)
add_percentage(df_panoma,0, 1)
add_percentage(df_rice, 0, 1)

col = ['num_people_enrolled', 'total_class_size', 'class_number', 'term', 'Pro
f Name', 'Percentage', 'Unnamed: 0']
df_sfsu = df_sfsu.loc[:,col]
df_sfsu.drop('Unnamed: 0', axis = 1, inplace = True)
df_sfsu.columns = ['num_people_enrolled', 'total_class_size', 'class_number',
'term' ,'professor', 'Percentage']

df_chico = clean_nan(df_chico)
df_sdsu= clean_nan(df_sdsu)
df_ucsc = clean_nan(df_ucsc)
df_uci = clean_nan(df_uci)
df_sfsu = clean_nan(df_sfsu)
df_rice = clean_nan(df_rice)
df_harvey = clean_nan(df_harvey)

df_panoma = clean_class_nums(df_panoma)
df_harvey = clean_class_nums(df_harvey)
```

In [19]:
```
# Splits the data into percentages for each term, so it makes it easier to plo
t
```

In [20]:
```
def getPercentages(self, term_location):
    unique_vals = []
    newData = []
    newTerm = []
    for i in self.iloc[:,term_location]:
        if i not in unique_vals:
            unique_vals.append(i)
    newTerm = list(unique_vals)
    for i in newTerm:
        newData.append(self.Percentage[self.term == i])
    return newData, newTerm
```

In [21]:
```
# Goes through each term and sorts the term with accordance to the data
```

In [22]:
```
def sort_term(term, data):
    to_sort = []
    for i in term:
        if 'FALL' in i:
            split = i.replace('FALL', '')
            split = split + '2'
        if 'SPRING' in i:
            split = i.replace('SPRING', '')
            split = split+'3'
        if 'WINTER' in i:
            split = i.replace('WINTER', '')
            split = split + '1'
        to_sort.append((int(int(split)%1000)))
    _, term, data = zip(*sorted(zip(to_sort, term, data)))
    return term, data
```

In [23]:
```
def term_to_num(term):
    quant = {'SPRING':.01, 'FALL':.02,  'WINTER':.03}
    df = pd.DataFrame()
    term_split = term.split(' ')
    return quant[term_split[0]] + int(term_split[1])
```

In [24]:
```
df_uci['term_num'] = df_uci['term'].apply(term_to_num)
df_ucsc['term_num'] = df_ucsc['term'].apply(term_to_num)
df_chico['term_num'] = df_chico['term'].apply(term_to_num)
df_sdsu['term_num'] = df_sdsu['term'].apply(term_to_num)
df_harvey['term_num'] = df_harvey['term'].apply(term_to_num)
df_panoma['term_num'] = df_panoma['term'].apply(term_to_num)
df_sfsu['term_num'] = df_sfsu['term'].apply(term_to_num)
df_rice['term_num'] = df_rice['term'].apply(term_to_num)

df_ucsc = df_ucsc.sort_values(by = ['term_num'])
df_uci = df_uci.sort_values(by = ['term_num'])
df_chico = df_chico.sort_values(by = ['term_num'])
df_sdsu = df_sdsu.sort_values(by = ['term_num'])
df_harvey = df_harvey.sort_values(by = ['term_num'])
df_sfsu = df_sfsu.sort_values(by=['term_num'])
df_rice = df_sfsu.sort_values(by=['term_num'])
```

In [25]:
```
## We took a look at the differernt types of computer science courses offered
 over time and how there enrollment has changed
## we focused on 6 subjects
## databases
## computer security
## AI
## Software Engineering
## Web Development
## Machine Learning
```

In [26]:
```
## index by year
def add_year(x):
    return int(x)
```

In [27]:
```
## add a column to the data frame to indcate the year
df_rice['year'] = df_rice['term_num'].apply(add_year)
df_sdsu['year'] = df_sdsu['term_num'].apply(add_year)
df_sfsu['year'] = df_sfsu['term_num'].apply(add_year)
df_ucsc['year'] = df_ucsc['term_num'].apply(add_year)
df_uci['year'] = df_uci['term_num'].apply(add_year)
df_chico['year'] = df_chico['term_num'].apply(add_year)
df_harvey['year'] = df_harvey['term_num'].apply(add_year)
```

In [28]:
```
## get class data from dataframes
```

In [29]:
```
## data_bases

ucsc_db= df_ucsc[df_ucsc['class_number'] == '180']

sdsu_db = df_sdsu[df_sdsu['class_number'] == 'CS-503']

uci_db = df_uci[df_uci['class_number'] == '']

sfsu_db = df_sfsu[df_sfsu['class_number'] == 'CSC675']

harvey_db = df_harvey[df_harvey['class_number'] == 'CSCI133']

chico_db = df_chico[df_chico['class_number'] == '370']

rice_db = df_rice[df_rice['class_number']=='CSC430']
```

```
C:\Users\terry\Anaconda3\lib\site-packages\pandas\core\ops.py:1649: FutureWar
ning:

elementwise comparison failed; returning scalar instead, but in the future wi
ll perform elementwise comparison
```

In [30]:
```
## computer securuty

## ucsc == 122
ucsc_sec = df_ucsc[df_ucsc['class_number'] == '122']
## sdsu == 574
sdsu_sec = df_sdsu[df_sdsu['class_number'] == 'CS-574']
## chico == 546
chico_sec = df_chico[df_chico['class_number'] == '546']
## harvey == 125
harvey_sec = df_harvey[df_harvey['class_number'] == 'CSCI125']
# sf state = 650
sfsu_sec = df_sfsu[df_sfsu['class_number'] == 'CSC650']
## rice == 527
rice_sec = df_rice[df_rice['class_number']== 'CSC527']
## uci
uci_sec = df_uci[df_uci['class_number'] == '']
```

In [31]:
```
#34340 UCI AI
#140 UCSC AI
ucsc_ai = df_ucsc[df_ucsc['class_number'] == '140']
#8719 CSC 665 [01] SFSU AI
sfsu_ai = df_sfsu[df_sfsu['class_number'] == 'CSC665']
#CS 151. Artificial Intelligence Harvey Mudd
harvey_ai = df_harvey[df_harvey['class_number'] == 'CSCI151']
#CS-550 SDSU
sdsu_ai = df_sdsu[df_sdsu['class_number'] == 'CS-550']
## chico Ai == 580
chico_ai = df_chico[df_chico['class_number'] == '580']
## rice == 440
rice_ai = df_rice[df_rice['class_number']=='CSC440']
##uci
uci_ai = df_uci[df_uci['class_number'] == '']
```

In [32]:
```
#software engineering

#ucsc==115 and 115A
uscs_se_classes = ['115', '115A']
ucsc_se = df_ucsc.loc[df_ucsc['class_number'].isin(uscs_se_classes)]
#sf state == 413
sfsu_se = df_sfsu[df_sfsu['class_number'] == 'CSC413']
#chico==430
chico_se = df_chico[df_chico['class_number'] == '430']
#sdsu == 532
sdsu_se = df_sdsu[df_sdsu['class_number'] == 'CS-532']
#harvey == 121
harvey_se = df_harvey[df_harvey['class_number'] == 'CSCI121']
## rice ==410
rice_se = df_rice[df_rice['class_number']=='CSC410']
## uci
uci_se = df_uci[df_uci['class_number'] == '']
```

In [33]:
```
#web_dev

## harvey == 121
harvey_web = df_harvey[df_harvey['class_number'] == 'CSCI121']
## rice == 431
rice_web = df_rice[df_rice['class_number'] == 'CSC431']
## sf_state == 317
sfsu_web = df_sfsu[df_sfsu['class_number'] == 'CSC317']
##ucsc == 183
ucsc_web = df_ucsc[df_ucsc['class_number'] == '183']
##sdsu == none
sdsu_web = df_sdsu[df_sdsu['class_number'] == '']
## chico == 465
chico_web = df_chico[df_chico['class_number'] == '465']
##uci
uci_web = df_uci[df_uci['class_number'] == '']
```

In [34]:
```
## machine learning

## rice ==  502, 540, 542
rice_ml_classes = ['CSC502', 'CSC540', 'CSC540']
rice_ml = df_rice[df_rice['class_number'].isin(rice_ml_classes)]
## ucsc == 142 and 143
ucsc_ml_classes = ['142', '143']
ucsc_ml = df_ucsc[df_ucsc['class_number'].isin(ucsc_ml_classes)]
##harvey == 158
harvey_ml = df_harvey[df_harvey['class_number'] == 'CSCI121']
##sfsu == none
sfsu_ml = df_harvey[df_harvey['class_number'] == '']
## chico == 585
chico_ml = df_chico[df_chico['class_number'] == 'CSCI121']
## sdsu == none
sdsu_ml = df_harvey[df_harvey['class_number'] == '']
## uci
uci_ml = df_uci[df_uci['class_number'] == '']
```

In [35]:
```
### Total Enrollment ###
```

In [36]:
```
def merge_data_totals(sfsu, sdsu, chico, ucsc, harvey, rice, uci):
    df = pd.DataFrame({"sfsu": sfsu['num_people_enrolled'] })
    df2 = pd.DataFrame({"sdsu": sdsu['num_people_enrolled'] })
    df3 = pd.DataFrame({"chico": chico['num_people_enrolled'] })
    df4 = pd.DataFrame({"ucsc": ucsc['num_people_enrolled'] })
    df5 = pd.DataFrame({"harvey": harvey['num_people_enrolled'] })
    df6 = pd.DataFrame({"rice": rice['num_people_enrolled'] })
    df7 = pd.DataFrame({"uci": uci['num_people_enrolled'] })
    df_8 = df.merge(df2, right_index = True, left_index = True, how="outer")
    df_9 =df_8.merge(df3, right_index = True, left_index = True, how="outer")
    df_10 =df_9.merge(df4, right_index = True, left_index = True, how="outer")
    df_11 =df_10.merge(df5, right_index = True, left_index = True, how="outer"
)
    df_12 =df_11.merge(df6, right_index = True, left_index = True, how="outer"
)
    data =df_12.merge(df7, right_index = True, left_index = True, how="outer")
    data =data.transpose()
    return data
```

In [37]:
```
## data bases total sums

## aggregate by years
sfsu_db_sum = sfsu_db.groupby(['year']).sum()
sdsu_db_sum = sdsu_db.groupby(['year']).sum()
chico_db_sum = chico_db.groupby(['year']).sum()
harvey_db_sum = harvey_db.groupby(['year']).sum()
ucsc_db_sum = ucsc_db.groupby(['year']).sum()
rice_db_sum = rice_db.groupby(['year']).sum()
uci_db_sum = uci_db.groupby(['year']).sum()

## merge data to one frame
data_bases = merge_data_totals(sfsu_db_sum, sdsu_db_sum, chico_db_sum, ucsc_db
_sum, harvey_db_sum, rice_db_sum, uci_db_sum)
```

In [38]:
```
## AI

## aggregate by years total sums
sfsu_ai_sum= sfsu_ai.groupby(['year']).sum()
sdsu_ai_sum = sdsu_ai.groupby(['year']).sum()
chico_ai_sum = chico_ai.groupby(['year']).sum()
harvey_ai_sum = harvey_ai.groupby(['year']).sum()
ucsc_ai_sum = ucsc_ai.groupby(['year']).sum()
rice_ai_sum = rice_ai.groupby(['year']).sum()
uci_ai_sum = uci_ai.groupby(['year']).sum()

## merge data to one frame
AI = merge_data_totals(sfsu_ai_sum, sdsu_ai_sum, chico_ai_sum, ucsc_ai_sum, ha
rvey_ai_sum, rice_ai_sum, uci_ai_sum)
```

In [39]:
```
## Software Engineering

## aggregate by years total sum
sfsu_se_sum = sfsu_se.groupby(['year']).sum()
sdsu_se_sum = sdsu_se.groupby(['year']).sum()
chico_se_sum = ucsc_se.groupby(['year']).sum()
harvey_se_sum = harvey_se.groupby(['year']).sum()
ucsc_se_sum = ucsc_se.groupby(['year']).sum()
rice_se_sum = rice_se.groupby(['year']).sum()
uci_se_sum = uci_se.groupby(['year']).sum()

## merge data to one frame
s_eng = merge_data_totals(sfsu_se_sum, sdsu_se_sum, chico_se_sum, ucsc_se_sum,
harvey_se_sum, rice_se_sum, uci_se_sum)
```

In [40]:
```
## Machine learning

## aggregate by years total sum
chico_ml_sum = chico_ml.groupby(['year']).sum()
harvey_ml_sum = harvey_ml.groupby(['year']).sum()
ucsc_ml_sum = ucsc_ml.groupby(['year']).sum()
rice_ml_sum = rice_ml.groupby(['year']).sum()
sfsu_ml_sum = sfsu_ml.groupby(['year']).sum()
sdsu_ml_sum = sdsu_ml.groupby(['year']).sum()
uci_ml_sum = uci_ml.groupby(['year']).sum()
## merge data to one frame
ML = merge_data_totals(sfsu_ml_sum, sdsu_ml_sum, chico_ml_sum, ucsc_ml_sum, ha
rvey_ml_sum, rice_ml_sum, uci_ml_sum)
```

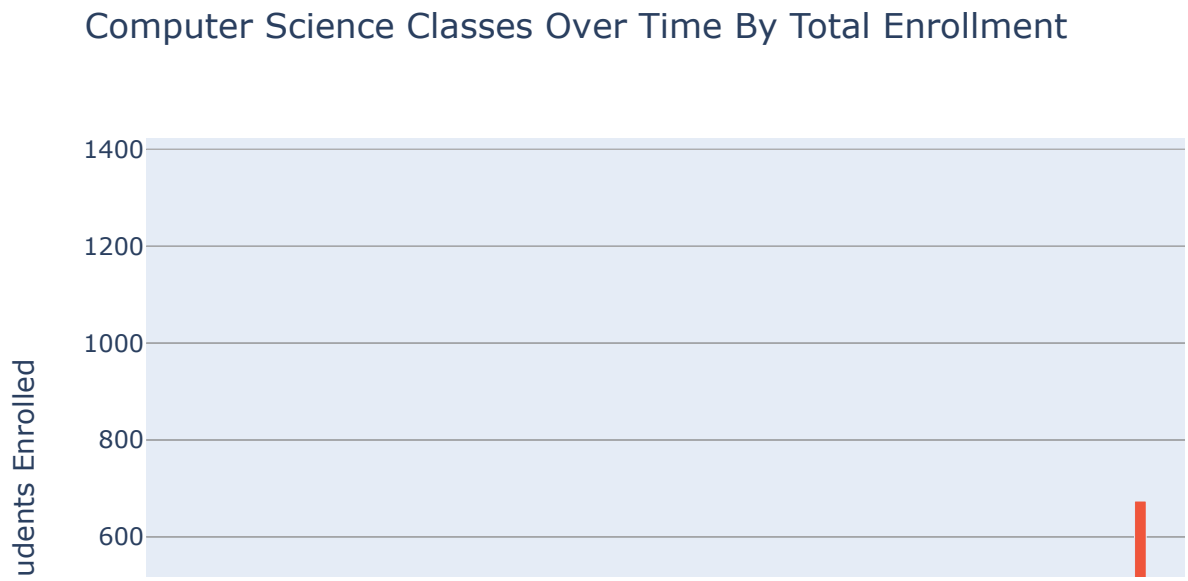
In [41]:
```
## Computer Securuty

## aggregate by years total sum
sfsu_sec_sum = sfsu_sec.groupby(['year']).sum()
sdsu_sec_sum = sdsu_sec.groupby(['year']).sum()
chico_sec_sum = chico_sec.groupby(['year']).sum()
harvey_sec_sum = harvey_sec.groupby(['year']).sum()
ucsc_sec_sum = ucsc_sec.groupby(['year']).sum()
rice_sec_sum = rice_sec.groupby(['year']).sum()
uci_sec_sum = uci_sec.groupby(['year']).sum()

## merge data to one frame
SEC = merge_data_totals(sfsu_sec_sum, sdsu_sec_sum, chico_sec_sum, ucsc_sec_su
m, harvey_sec_sum, rice_sec_sum, uci_sec_sum)
```

In [42]:
```
## Web Development

## aggregate by years total sum
sfsu_web_sum = sfsu_web.groupby(['year']).sum()
sdsu_web_sum = sdsu_web.groupby(['year']).sum()
chico_web_sum = chico_web.groupby(['year']).sum()
harvey_web_sum = harvey_web.groupby(['year']).sum()
ucsc_web_sum = ucsc_web.groupby(['year']).sum()
rice_web_sum = rice_web.groupby(['year']).sum()
uci_web_sum = uci_web.groupby(['year']).sum()

## merge data to one frame
web = merge_data_totals(sfsu_sec_sum, sdsu_sec_sum, chico_sec_sum, ucsc_sec_su
m, harvey_sec_sum, rice_sec_sum, uci_sec_sum)
```

In [43]:
```
## merge totals
db_sum =  pd.DataFrame({"data base":data_bases.sum()})
sec_sum = pd.DataFrame({"computer Security": SEC.sum()})
AI_sum = pd.DataFrame({"AI":AI.sum()})
s_eng_sum = pd.DataFrame({"Software Engineering":s_eng.sum()})
mL_sum = pd.DataFrame({"Machine Learning":ML.sum()})
web_sum = pd.DataFrame({"Web Developement":web.sum()})

df = db_sum.merge(sec_sum, right_index = True, left_index = True, how="outer")
df1 = df.merge(s_eng_sum, right_index = True, left_index = True, how="outer")
df2 = df1.merge(web_sum, right_index = True, left_index = True, how="outer")
df3 = df2.merge(mL_sum, right_index = True, left_index = True, how="outer")
out = df3.merge(AI_sum, right_index = True, left_index = True, how="outer")
out = out.fillna(0)
```

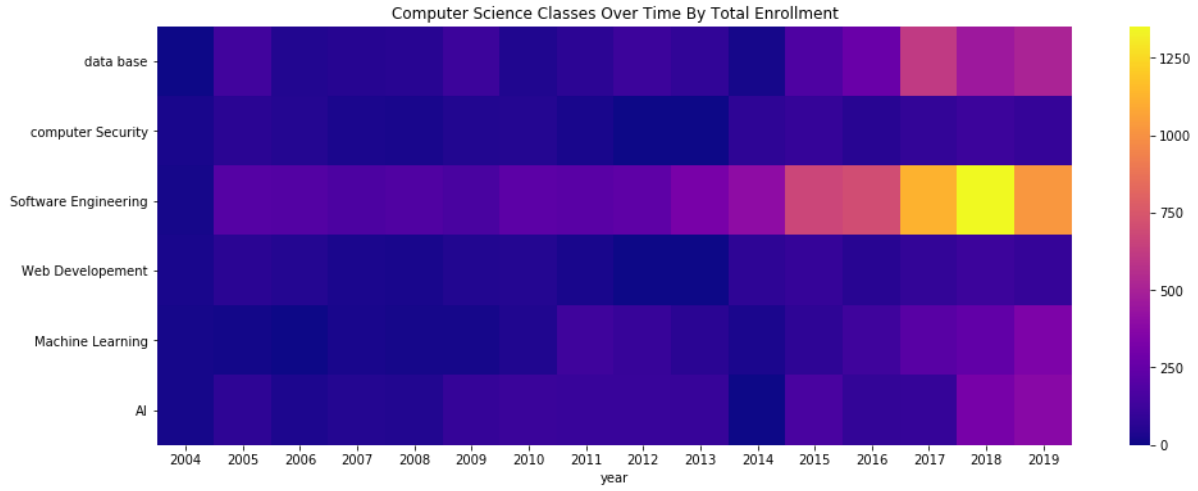
In [44]:
```
## bar graph and heat map of the total class enrolments from 2005 to 2019
## this shows how the popularity of computer science clases have changed over
 time
## from these visualizatons we conclude the subjests that have grown the most
 in popularity are
## Software Engineering
## Machine learning
## Databases
## one thing we found surprising is that artificial intellicence showed very l
ittle growth
```

In [45]:
```
## bar grpah of total class enrolment
fig = go.Figure(data=[
    go.Bar(name='AI', x= out.index, y=out['AI']),
    go.Bar(name='Software Engineering',x= out.index, y=out['Software Engineeri
ng']),
    go.Bar(name='Data Bases', x= out.index,y=out['data base']),
    go.Bar(name='computer Security',x= out.index, y=out['computer Security']),
    go.Bar(name='Web Development ', x= out.index,y=out['Web Developement']),
    go.Bar(name='Machine Learning', x= out.index, y=out['Machine Learning'])
],)
fig.update_layout(barmode='group',  title="Computer Science Classes Over Time
 By Total Enrollment",
    xaxis_title="Year",
    yaxis_title="Students Enrolled")
fig.show()
```



In [46]:
```
## heat map of toral class enrolement
import seaborn as sns
out = out.transpose()
plt.figure(figsize=(16, 6))
sns.heatmap(out, cmap = 'plasma').set_title('Computer Science Classes Over Tim
e By Total Enrollment')
```

Out[46]:
```
Text(0.5, 1, 'Computer Science Classes Over Time By Total Enrollment')
```



In [47]:
```
### percentages ###
```

In [48]:
```
def merge_data_perc(sfsu, sdsu, chico, ucsc, harvey, rice, uci):
    df = pd.DataFrame({"sfsu": sfsu['Percentage'] })
    df2 = pd.DataFrame({"sdsu": sdsu['Percentage'] })
    df3 = pd.DataFrame({"chico": chico['Percentage'] })
    df4 = pd.DataFrame({"ucsc": ucsc['Percentage'] })
    df5 = pd.DataFrame({"harvey": harvey['Percentage'] })
    df6 = pd.DataFrame({"rice": rice['Percentage'] })
    df7 = pd.DataFrame({"uci": uci['Percentage'] })
    df_8 = df.merge(df2, right_index = True, left_index = True, how="outer")
    df_9 =df_8.merge(df3, right_index = True, left_index = True, how="outer")
    df_10 =df_9.merge(df4, right_index = True, left_index = True, how="outer")
    df_11 =df_10.merge(df5, right_index = True, left_index = True, how="outer"
)
    df_12 =df_11.merge(df6, right_index = True, left_index = True, how="outer"
)
    data =df_12.merge(df7, right_index = True, left_index = True, how="outer")
    data =data.transpose()
    return data
```

In [49]:
```
## Software Developemnt

## aggregate by years means
sfsu_se_mean= sfsu_se.groupby(['year']).mean()
sdsu_se_mean = sdsu_se.groupby(['year']).mean()
chico_se_mean = chico_se.groupby(['year']).mean()
harvey_se_mean = harvey_se.groupby(['year']).mean()
ucsc_se_mean = ucsc_se.groupby(['year']).mean()
rice_se_mean = rice_se.groupby(['year']).mean()
uci_se_mean = uci_se.groupby(['year']).mean()
## merge data to one frame
s_eng_per = merge_data_perc(sfsu_se_mean, sdsu_se_mean, chico_se_mean, ucsc_se
_mean, harvey_se_mean, rice_se_mean, uci_se_mean)
```

In [50]:
```
## AI

## aggregate by years means
sfsu_ai_mean= sfsu_ai.groupby(['year']).mean()
sdsu_ai_mean = sdsu_ai.groupby(['year']).mean()
chico_ai_mean = chico_ai.groupby(['year']).mean()
harvey_ai_mean = harvey_ai.groupby(['year']).mean()
ucsc_ai_mean = ucsc_ai.groupby(['year']).mean()
rice_ai_mean = rice_ai.groupby(['year']).mean()
uci_ai_mean = uci_ai.groupby(['year']).mean()

## merge data to one frame
AI_per = merge_data_perc(sfsu_ai_mean, sdsu_ai_mean, chico_ai_mean, ucsc_ai_me
an, harvey_ai_mean, rice_ai_mean, uci_ai_mean)
```

In [51]:
```
## Computer Secuirty

## aggregate by years means
sfsu_sec_mean= sfsu_sec.groupby(['year']).mean()
sdsu_sec_mean = sdsu_sec.groupby(['year']).mean()
chico_sec_mean = chico_sec.groupby(['year']).mean()
harvey_sec_mean = harvey_sec.groupby(['year']).mean()
ucsc_sec_mean = ucsc_sec.groupby(['year']).mean()
rice_sec_mean = rice_sec.groupby(['year']).mean()
uci_sec_mean = uci_sec.groupby(['year']).mean()

## merge data to one frame
SEC_per = merge_data_perc(sfsu_sec_mean, sdsu_sec_mean, chico_sec_mean, ucsc_s
ec_mean, harvey_sec_mean, rice_sec_mean, uci_sec_mean)
```

In [52]:
```
## Data Bases

## aggregate by years means
sfsu_db_mean= sfsu_db.groupby(['year']).mean()
sdsu_db_mean = sdsu_db.groupby(['year']).mean()
chico_db_mean = chico_db.groupby(['year']).mean()
harvey_db_mean = harvey_db.groupby(['year']).mean()
ucsc_db_mean = ucsc_db.groupby(['year']).mean()
rice_db_mean = rice_db.groupby(['year']).mean()
uci_db_mean = uci_db.groupby(['year']).mean()

## merge data to one frame
data_bases_per = merge_data_perc(sfsu_db_mean, sdsu_db_mean, chico_db_mean, uc
sc_db_mean, harvey_db_mean, rice_db_mean, uci_db_mean)
```

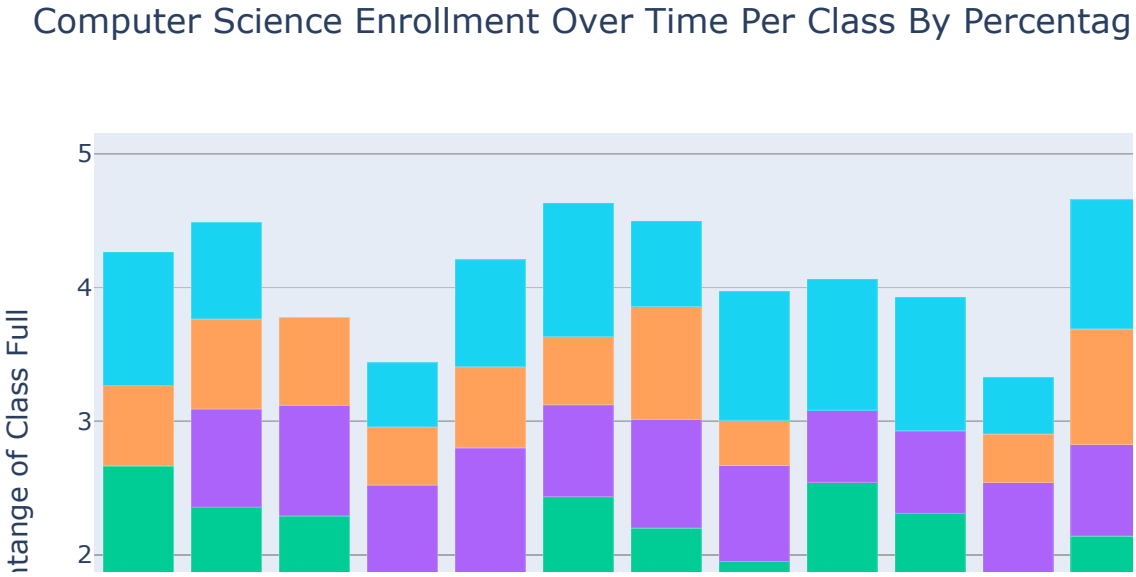
In [53]:
```
## Machine Learning

## aggregate by years means
sfsu_ml_mean= sfsu_ml.groupby(['year']).mean()
sdsu_ml_mean = sdsu_ml.groupby(['year']).mean()
chico_ml_mean = chico_ml.groupby(['year']).mean()
harvey_ml_mean = harvey_ml.groupby(['year']).mean()
ucsc_ml_mean = ucsc_ml.groupby(['year']).mean()
rice_ml_mean = rice_ml.groupby(['year']).mean()
uci_ml_mean = uci_ml.groupby(['year']).mean()

## merge data to one frame
ML_per = merge_data_perc(sfsu_ml_mean, sdsu_ml_mean, chico_ml_mean, ucsc_ml_me
an, harvey_ml_mean, rice_ml_mean, uci_ml_mean)
```

In [54]:
```
## Web Development

## aggregate by years means
sfsu_web_mean= sfsu_web.groupby(['year']).mean()
sdsu_web_mean = sdsu_web.groupby(['year']).mean()
chico_web_mean = chico_web.groupby(['year']).mean()
harvey_web_mean = harvey_web.groupby(['year']).mean()
ucsc_web_mean = ucsc_web.groupby(['year']).mean()
rice_web_mean = rice_web.groupby(['year']).mean()
uci_web_mean = uci_web.groupby(['year']).mean()

## merge data to one frame
web_per = merge_data_perc(sfsu_web_mean, sdsu_web_mean, chico_web_mean, ucsc_w
eb_mean, harvey_web_mean, rice_web_mean, uci_web_mean)
```

In [55]:
```
## aggregate percentages

db_mean =  pd.DataFrame({"data base":data_bases_per.mean()})
sec_mean = pd.DataFrame({"computer Security": SEC_per.mean()})
AI_mean = pd.DataFrame({"AI":AI_per.mean()})
s_eng_mean = pd.DataFrame({"Software Engineering":s_eng_per.mean()})
ML_mean = pd.DataFrame({"Machine Leanring":ML_per.mean()})
web_mean = pd.DataFrame({"Web Development":web_per.mean()})

df = db_mean.merge(sec_mean, right_index = True, left_index = True, how="oute
r")
df1 = df.merge(s_eng_mean, right_index = True, left_index = True, how="outer")
df2 = df1.merge(ML_mean, right_index = True, left_index = True, how="outer")
df3 = df2.merge(web_mean, right_index = True, left_index = True, how="outer")
out = df3.merge(AI_mean, right_index = True, left_index = True, how="outer")
out_per= out.fillna(0)
```

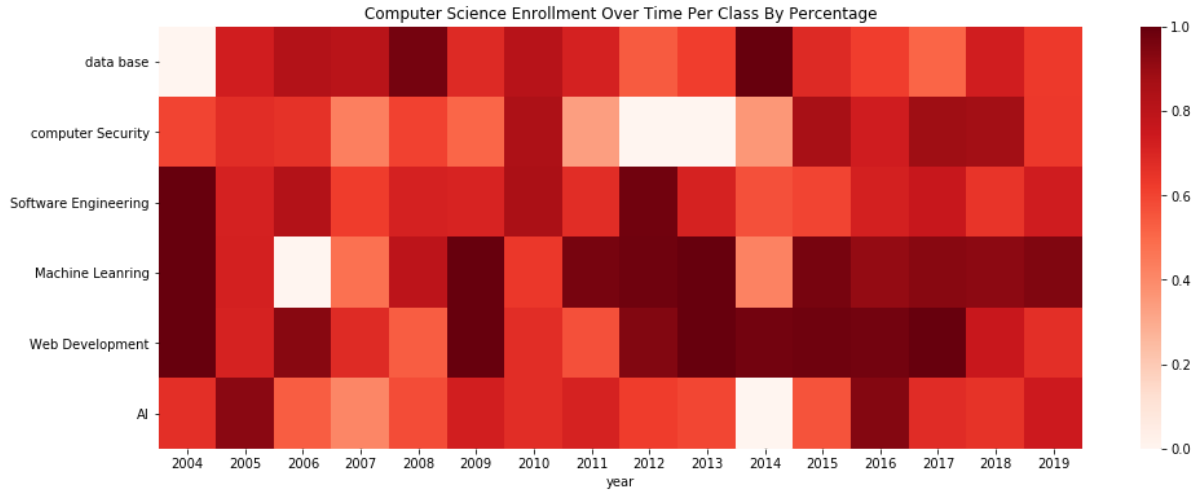
In [56]:
```
## this bar chart and heat map show the average fullness of a class per a give
n year
## from this we can see this the distribution of how full classes are over tim
e
## the most highly enrolled classea are generally not the most filled classes
    ## this implys that colleges are not always supplying the right number of
 seats in the right courses
## One thing that stands out about this chart is that web development classe a
re almost always full
    ## this subject might be more popular if there were more seats in the clas
ss
```

In [57]:
```
## bar chart for percentage

fig = go.Figure(data=[
    go.Bar(name='AI', x= out.index, y=out['AI']),
    go.Bar(name='Software Engineering',x= out_per.index, y=out_per['Software E
ngineering']),
    go.Bar(name='Web Development', x= out_per.index,y=out_per['Web Developmen
t']),
    go.Bar(name='Data Bases', x= out_per.index,y=out_per['data base']),
    go.Bar(name='computer Security',x= out_per.index, y=out_per['computer Secu
rity']),
    go.Bar(name='Machine Learing',x= out_per.index, y=out_per['Machine Leanrin
g']),
],)
fig.update_layout(barmode='stack', title="Computer Science Enrollment Over Tim
e Per Class By Percentage",
    xaxis_title="Year",
    yaxis_title="Pecentange of Class Full")
fig.show()
```



In [58]:
```
## heat map for percentage
out_per = out_per.transpose()
plt.figure(figsize=(16, 6))
sns.heatmap(out_per, cmap = 'Reds').set_title('Computer Science Enrollment Ove
r Time Per Class By Percentage')
```

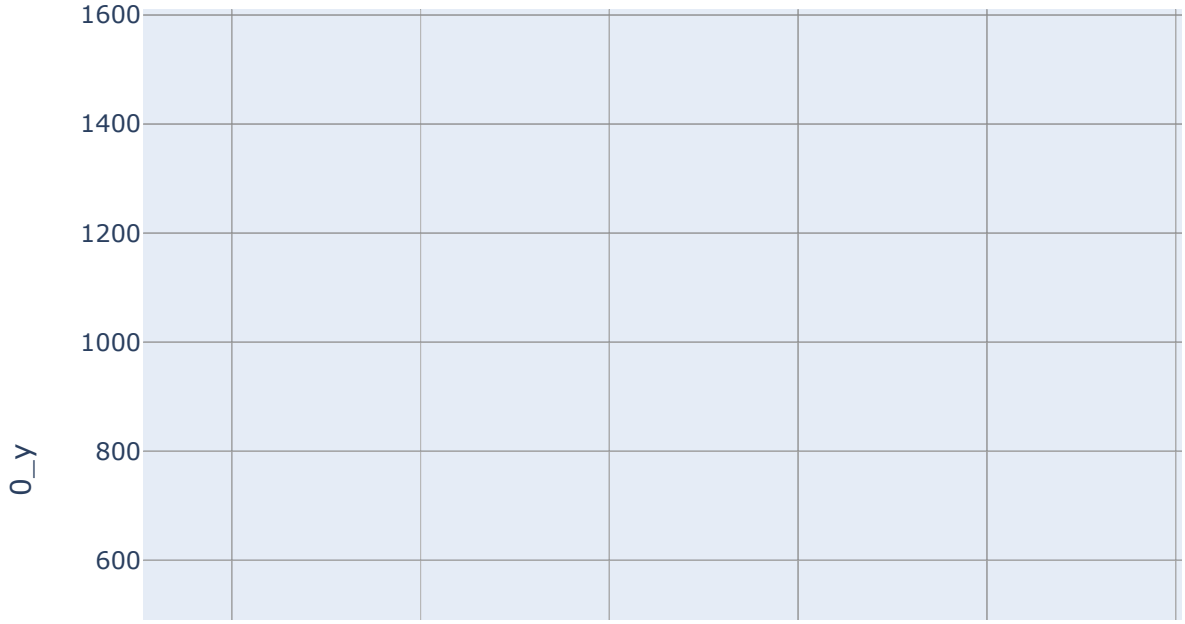
Out[58]: Text(0.5, 1, 'Computer Science Enrollment Over Time Per Class By Percentage')



In [59]:
```
### standard deviation
    ## measure the standard devation of class enrolment between colleges with
 total enrollment
```

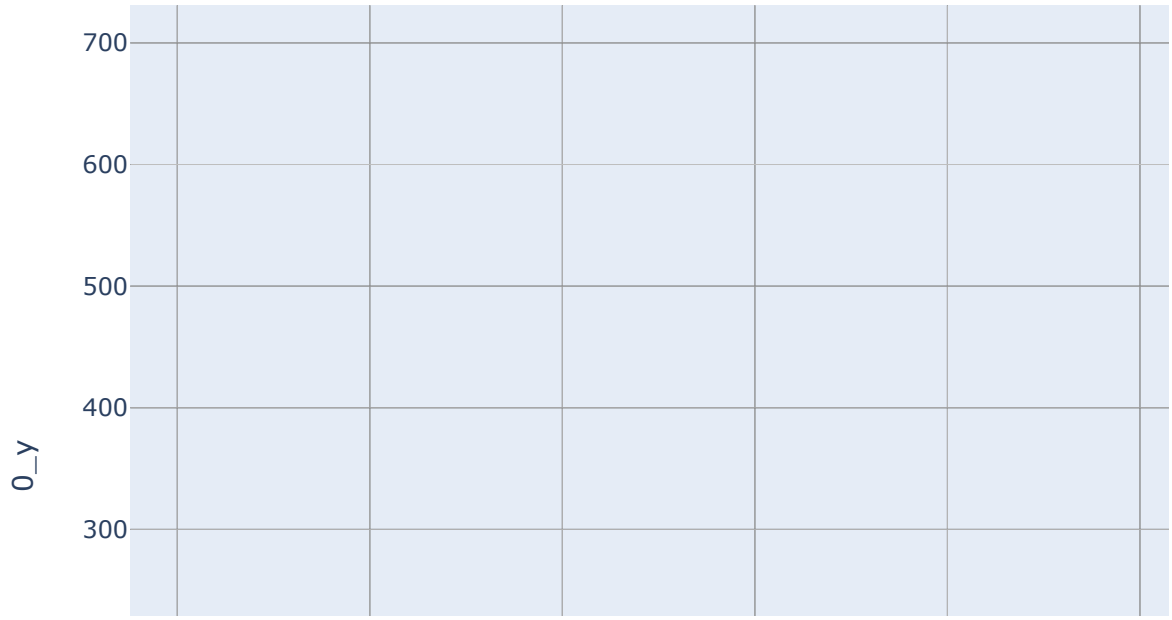
In [60]:
```
## software engineering
s_eng_var = pd.DataFrame(s_eng.var())
s_eng_totals = pd.DataFrame(s_eng.sum())
s_eng_var = s_eng_var.merge(s_eng_totals, left_index =True, right_index= True)
s_eng_var = s_eng_var.fillna(0)
s_eng_var = s_eng_var.reset_index()
```

In [61]:
```
fig = px.scatter(s_eng_var,x='year', y= '0_y', size='0_x', size_max=60)
fig.show()
```



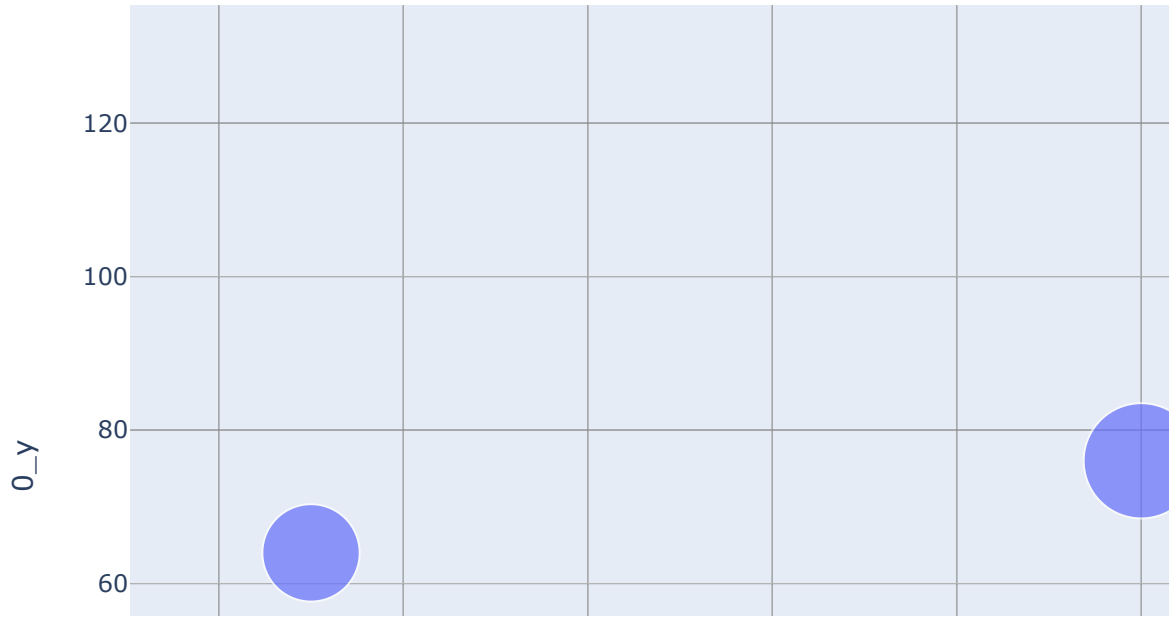
In [62]:
```
## database
db_sd = pd.DataFrame(data_bases.std())
db_totals = pd.DataFrame(data_bases.sum())
db_sd = db_sd.merge(db_totals, left_index =True, right_index= True)
db_sd = db_sd.fillna(0)
db_sd = db_sd.reset_index()
```

In [63]:
```
fig = px.scatter(db_sd,x='year', y= '0_y', size='0_x', size_max=60)
fig.show()
```



In [64]:
```
## Computer Security
sec_sd = pd.DataFrame(SEC.std())
sec_totals = pd.DataFrame(SEC.sum())
sec_sd = sec_sd.merge(sec_totals, left_index =True, right_index= True)
sec_sd = sec_sd.fillna(0)
sec_sd = sec_sd.reset_index()
```

In [65]:
```
fig = px.scatter(sec_sd,x='year', y= '0_y', size='0_x', size_max=60)
fig.show()
```



In [ ]:

In [ ]: