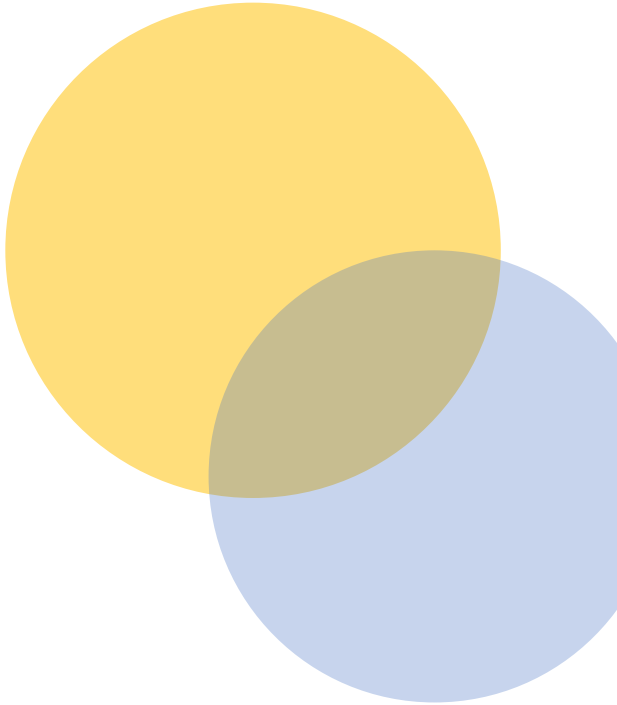
2020-2 컴퓨팅기초 기말 프로젝트

서울대학교 강의 데이터 분석 및 시각화

수업교시, 교과구분, 수업정원, 수강신청인원을 중심으로

2015-16294 김호연
2019-10056 김예진
2020-19088 이혜인
2019-18791 박성현
2019-13709 박나은

목차



1. 데이터 전처리

1. 분석 범위 내 데이터의 결측치 처리

```
print(raw_df.isna().sum())
```

```
==================================
컬럼별 결측치 개수 확인
==================================
교과구분                    0
개설대학                    2
개설학과                  497
이수과정                    0
학년                      199
교과목번호                   0
강좌번호                    0
교과목명                    0
부제명                  6807
학점                      0
강의                      0
실습                      0
수업교시                 2411
수업형태                 2411
강의실(동-호)(#연건, *평창)    2784
주담당교수                   0
정원                      0
수강신청인원                  0
비고                    4538
강의언어                    0
개설상태                    0
dtype: int64
```

```
# 수업 교시 결측치 행 제거
df = raw_df[raw_df['수업교시'].notnull()]
#개설대학 결측치->ROTC 수업 2개 제거
df[df['개설대학'].isnull()]
```

```
process_df.isna().sum()
```

```
교과구분        0
개설대학        0
이수과정        0
교과목번호       0
강좌번호        0
교과목명        0
학점          0
수업교시        0
정원          0
수강신청인원      0
dtype: int64
```

2. 분석 범위 외 특정 데이터 제외

```
#학부 강의 외 석사, 박사, 석박사통합 강의 제거
```

```
# 이수과정 목록 확인
df['이수과정'].unique()
```

```
array(['학사', '박사', '석사', '석박사통합'], dtype=object)
```

```
df = df[df['이수과정']=='학사']
```

```
#의과/약학/수의과/간호대학, 치의학대학원 전공 수업 제거
```

```
df[df['개설대학']=='의과대학']['교과구분'].unique()
```

```
array(['교양', '전선', '전필'], dtype=object)
```

```
medi_df = df[(df['개설대학']=='의과대학')&(df['교과구분']=='교양')]
```

```
df = df[(df['개설대학']!='의과대학')&(df['개설대학']!='약학대학')&(df['개설대학']
!='치의학대학원')&(df['개설대학']!='수의과대학')&(df['개설대학']!='간호대학')]
```

```
process_df = pd.concat([df,medi_df,phar_df,dent_df])
```

```
process_df
```

2939 rows × 21 columns

2.분석 및 시각화(1) 교과구분별

1. 교과구분 범주 데이터 확인

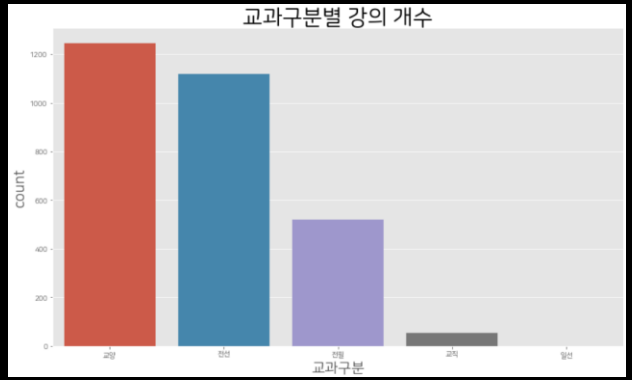
```
df['교과구분'].value_counts()
```

교양: 1246 /전선: 1119 / 전필:519
교직: 54 / 일선: 1

```
df['교과구분'].value_counts(normalize=True)
```

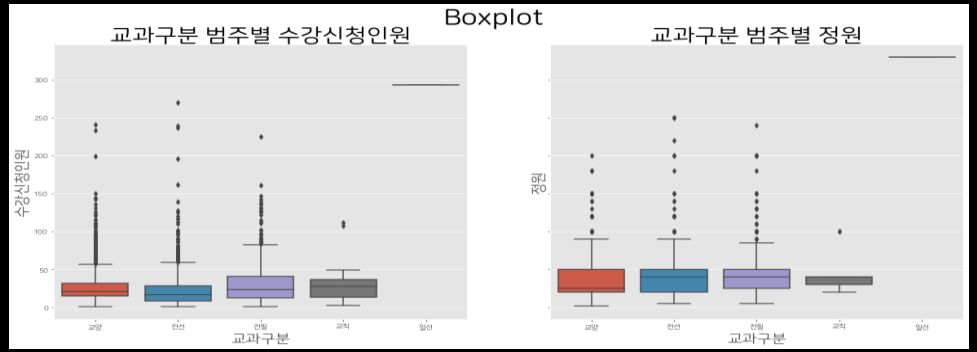
교양:0.423954/전선:0.380742/전필:0.176591/교직:0.018374/일선:0.000340

```
plt.figure()
sns.countplot(x='교과구분',data=df)
plt.title('교과구분별 강의 개수')
plt.show()
```



2.교과구분에 따른 수강신청인원 및 정원

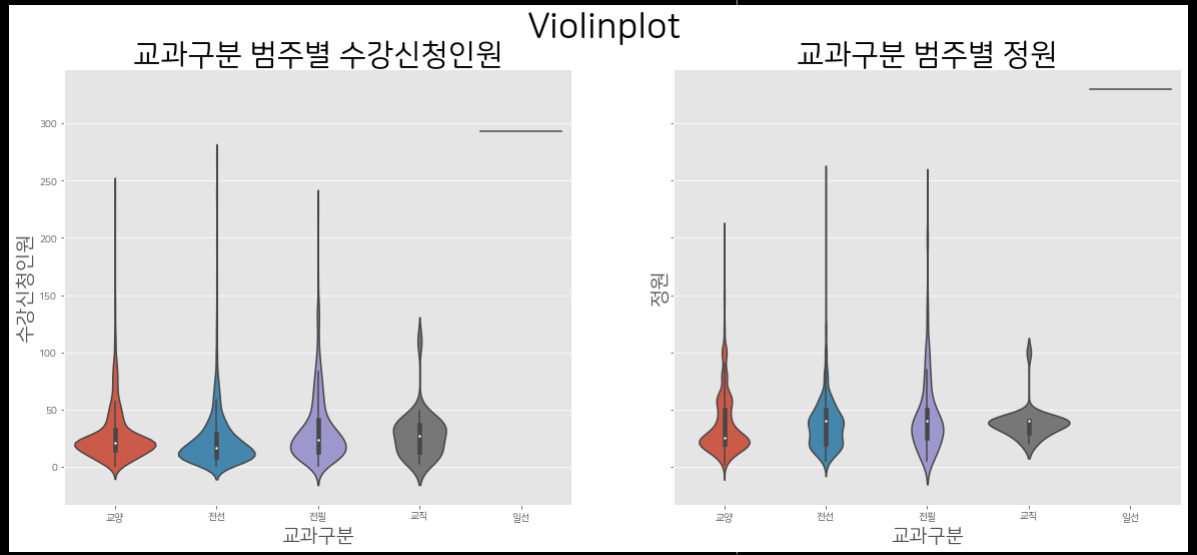
```
# 교과구분 범주별 수강신청인원과 정원 시각화(Boxplot)
fig, ax = plt.subplots(1,2,sharey=True,figsize=(20,8))
fig.suptitle('Boxplot')
sns.boxplot(ax=ax[0],x='교과구분',y='수강신청인원',data=df)
sns.boxplot(ax=ax[1],x='교과구분',y='정원',data=df)
ax[0].set_title('교과구분 범주별 수강신청인원')
ax[1].set_title('교과구분 범주별 정원')
plt.show()
```



2.분석 및 시각화(1) 교과구분별

2.교과구분에 따른 수강신청인원 및 정원

```
# 교과구분 범주별 수강신청인원과 정원 시각화(violinplot)
fig, ax = plt.subplots(1,2,sharey=True,figsize=(20,8))
fig.suptitle('Violinplot')
sns.violinplot(ax=ax[0],x='교과구분',y='수강신청인원',data=df)
sns.violinplot(ax=ax[1],x='교과구분',y='정원',data=df)
ax[0].set_title('교과구분 범주별 수강신청인원')
ax[1].set_title('교과구분 범주별 정원')
plt.show()
```



(일선 제외)

1) **수강신청인원의 평균에 큰 차이가 없었고** 분포 또한 비슷했음.

2) 전필 : 수강신청인원 **평균수**가 제일 큰 값

3) 전선 : 수강신청인원 **Max**가 제일 큰 값

4) 교과구분별 정원의 평균 : **전필이 가장 높은 값**, 그 외 범주에서는 비슷한 값을 보임. 전선에선 **정원의 Max**가 제일 큰 값을 보임.

5) 수강신청인원과 정원 그래프의 형태상의 차이는 정원이 다 채워지지 않았기 때문으로 추측됨.

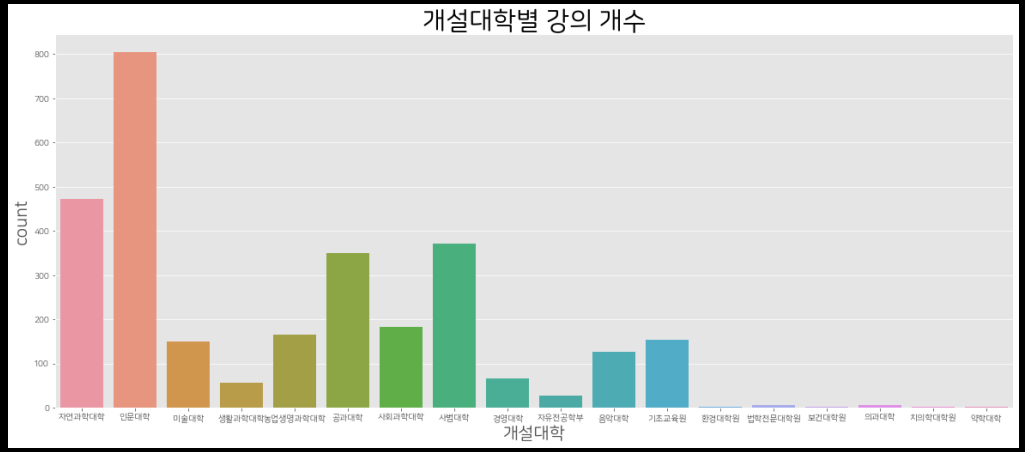
6) boxplot의 median이 수강신청인원의 경우 정원보다 전반적으로 내려가 있음.

7) **실제 수강신청인원은 정원보다 비슷비슷한 값**으로 수렴되는 경향성을 볼 수 있었고 이는 violinplot에서 보다 명확하게 파악 가능함.

2.분석 및 시각화(2) 개설대학별

1. 개설대학 별 강의 개수 시각화 (countplot)

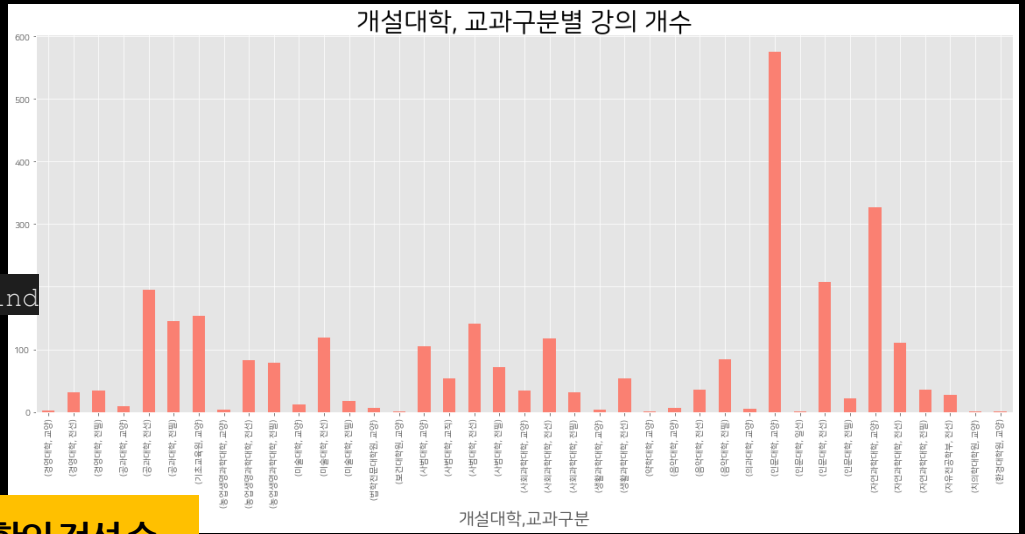
```
plt.figure(figsize=(20,8))
sns.countplot(x='개설대학',data=df)
plt.title('개설대학별 강의 개수')
plt.show()
```



인문대학> 자연과학대학> 사범대학 순

2. 개설대학 별 교과구분 별 강의 개수 시각화 (barplot)

```
plt.figure(figsize=(20,8))
df.groupby(['개설대학','교과구분']).count()['이수과정'].plot(kind
='bar',color='salmon')
plt.title('개설대학, 교과구분별 강의 개수')
plt.show()
```

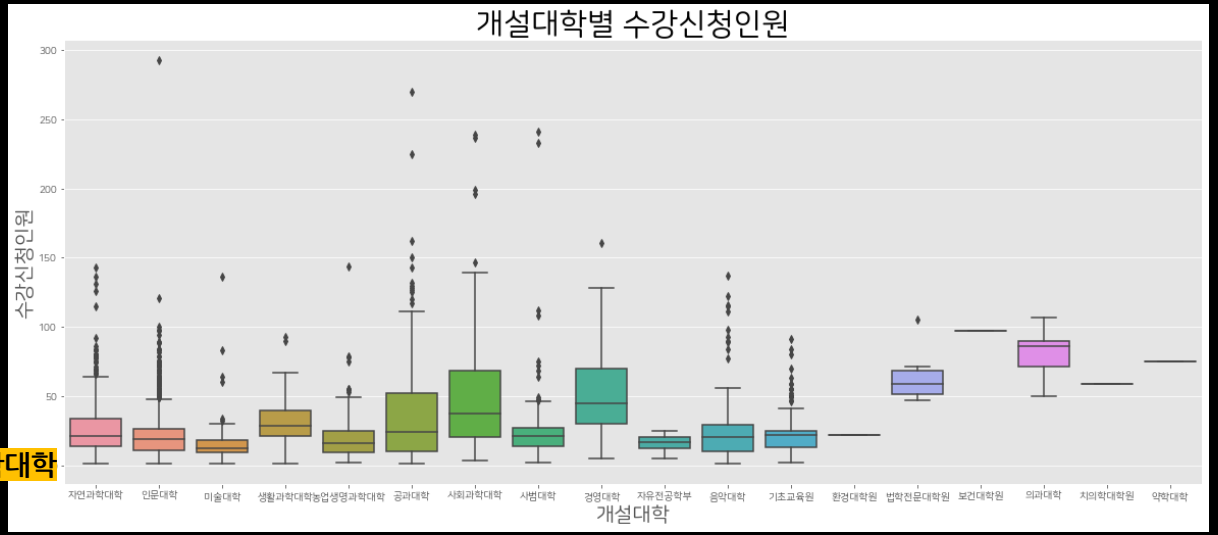


인문대학의 교양강의 > 자연과학대학의 교양> 인문대학의 전선> 공과대학의 전선 순

2.분석 및 시각화(2) 개설대학별

3.개설대학 별 수강신청인원 시각화 (boxplot)

```
plt.figure(figsize=(20,8))
sns.boxplot(x='개설대학',y='수강신청인원',data=df)
plt.title('개설대학별 수강신청인원')
plt.show()
```

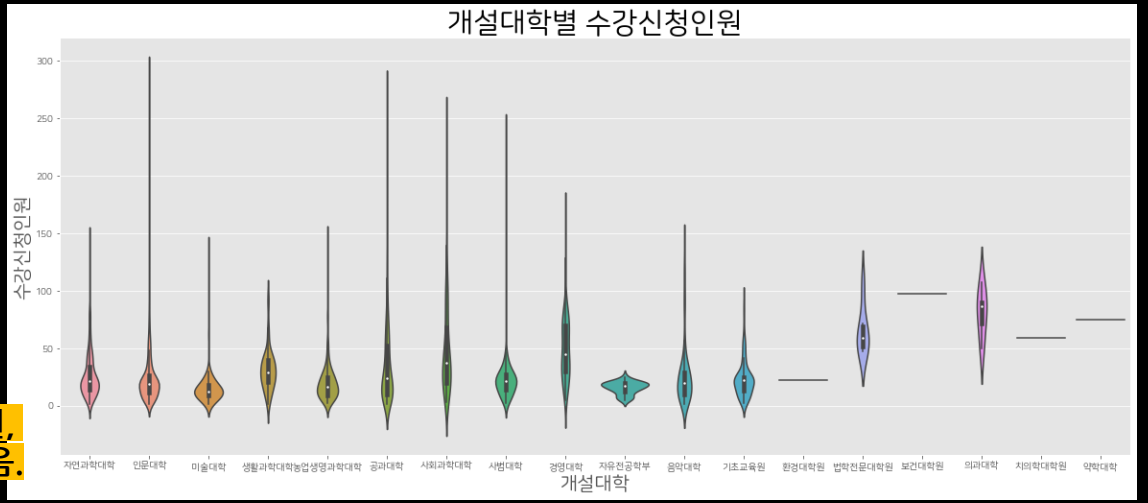


수강신청인원의 평균 : 경영대학 >사회과학대학>공과대학>생활과학대학

중앙값 : 경영대학 >사회과학대학 >생활과학대학 >공과대학

4. 개설대학 별 수강신청인원 시각화 (violinplot)

```
plt.figure(figsize=(20,8))
sns.violinplot(x='개설대학',y='수강신청인원',data=df)
plt.title('개설대학별 수강신청인원')
plt.show()
```

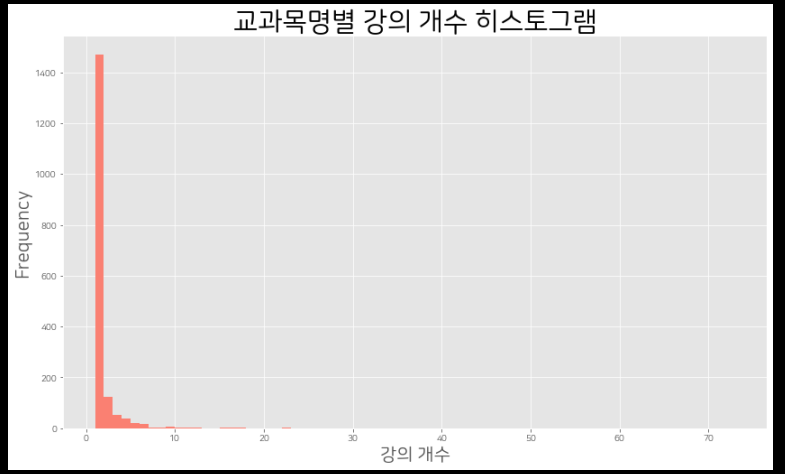


강의 개수가 1개로 확인된 환경대학원, 보건대학원, 치의학대학원, 약학대학을 제외하고 대부분 오른쪽 꼬리가 긴 분포를 보이고 있음.

2.분석 및 시각화(3) 교과목명별

1.교과목명별 강의개수 시각화 (histogram)

```
plt.figure()
df.groupby('교과목명').count().sort_values(by='교과구분',ascending=False)
                        ['교과구분'].plot(kind='hist',bins=72,color='salmon')
plt.title('교과목명별 강의 개수 히스토그램')
plt.xlabel('강의 개수')
plt.show()
```



1) 교과목명별로 개설된 강의는 대부분 1개였으며 Max값은 73개임.
2) 상위 1%에 해당하는 값은 약 14이며, 14개 이상 개설되는 강의는 19개 존재했음.

2.상위1%값이 14개임을 확인 -> 강의목록/개수 확인

```
 교과목명별 강의 개수의 상위 1% 값 확인
df.groupby('교과목명').count().sort_values(by='교과구분',ascending=False)['교과구분'].quantile(0.99)
```

```
14.3599999999999
```

```
# 강의 개수가 14개 이상인 강의 목록 및 강의 개수 확인
df.groupby('교과목명').count().sort_values(by='교과구분',ascending=False)[df.groupby('교과목명').count().sort_values
                                        (by='교과구분',ascending=False)['교과구분']>=14]
```

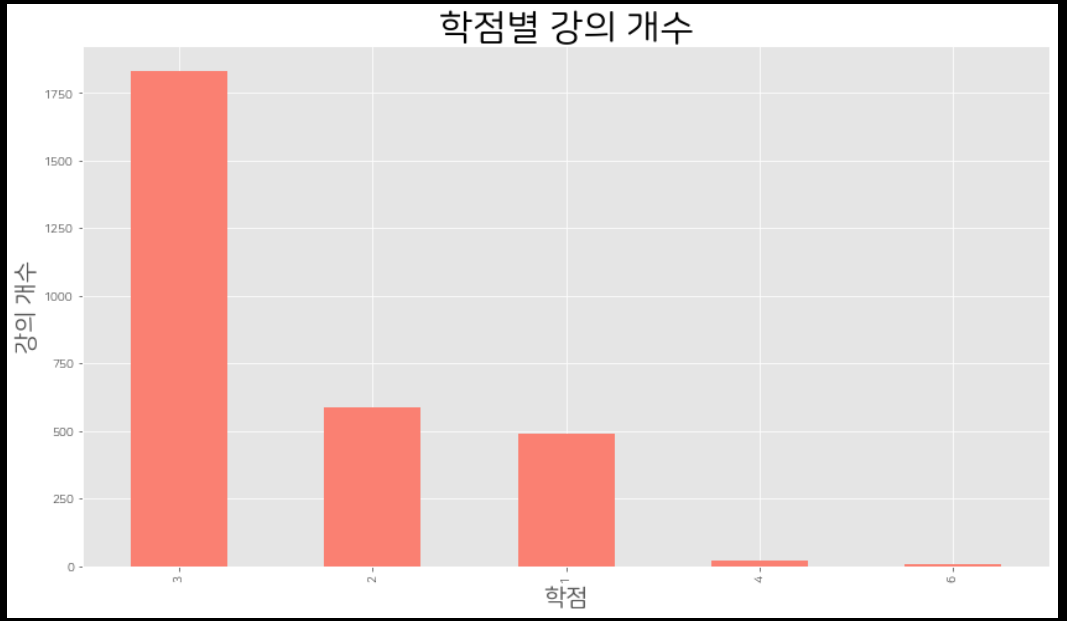
결과값 (표) 생략

2.분석 및 시각화(4) 학점별

1.학점 별 강의개수 시각화 (barplot)

```
plt.figure()
df['학점'].value_counts().plot(kind='bar',color='salmon')
plt.title('학점별 강의 개수')
plt.xlabel('학점')
plt.ylabel('강의 개수')
plt.show()
```

3학점 강의가 전체 강의의 절반이 넘는 62%를 차지함을 확인할 수 있음.



2.분석 및 시각화(5)수업교시별

1. 수업교시 칼럼을 수업요일, 수업횟수, 수업시간 컬럼으로 세분화

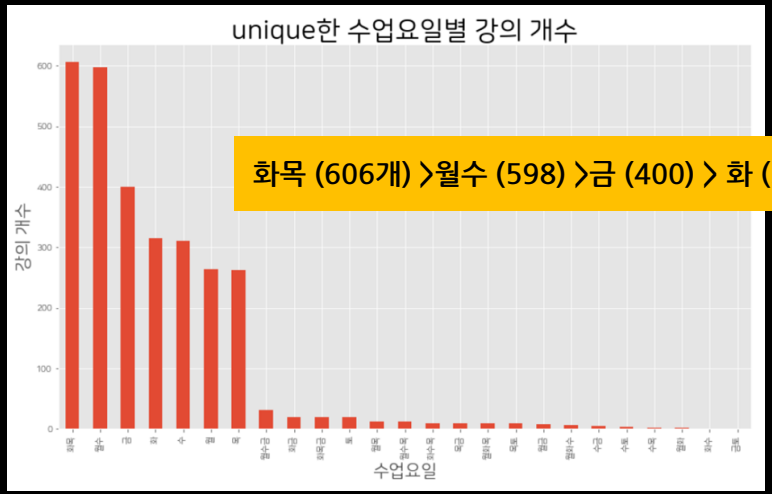
```
df['수업교시_list'] = df['수업교시'].str.split('/')
df['수업요일'] = np.nan
df['수업횟수'] = np.nan
df['수업시간'] = np.nan


for i in range(len(df)):
    df['수업횟수'][i] = len(df['수업교시_list'][i])
    for j in range(len(df['수업교시_list'][i])):
        if j == 0:
            df['수업요일'][i] = df['수업교시_list'][i][j][0]
            df['수업시간'][i] = [df['수업교시_list'][i][j][2:-1]]
        else:
            df['수업요일'][i] += df['수업교시_list'][i][j][0]
            df['수업시간'][i] += [df['수업교시_list'][i][j][2:-1]]
```

2.분석 및 시각화(5)수업교시별

2.수업요일별 강의 개수(barplot)

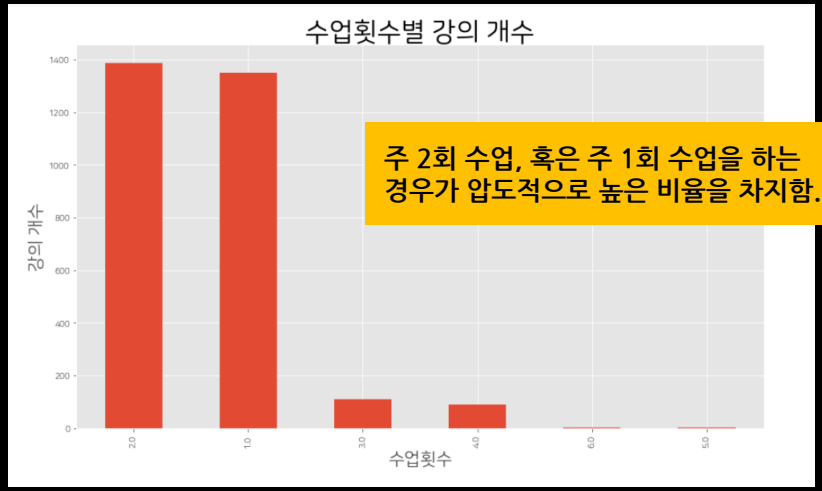
```
#중복된 수업요일을 합쳐 수업요일_unique 컬럼 생성
plt.figure()
df['수업요일_unique'].value_counts().plot(kind='bar')
plt.title('unique한 수업요일별 강의 개수')
plt.xlabel('수업요일')
plt.ylabel('강의 개수')
plt.show()
```



화목 (606개) >월수 (598) >금 (400) > 화 (315) >수 (311) 순으로 많았음.

3. 수업횟수별 강의 개수(barplot)

```
plt.figure()
df['수업횟수'].value_counts().plot(kind='bar')
plt.title('수업횟수별 강의 개수')
plt.xlabel('수업횟수')
plt.ylabel('강의 개수')
plt.show()
```



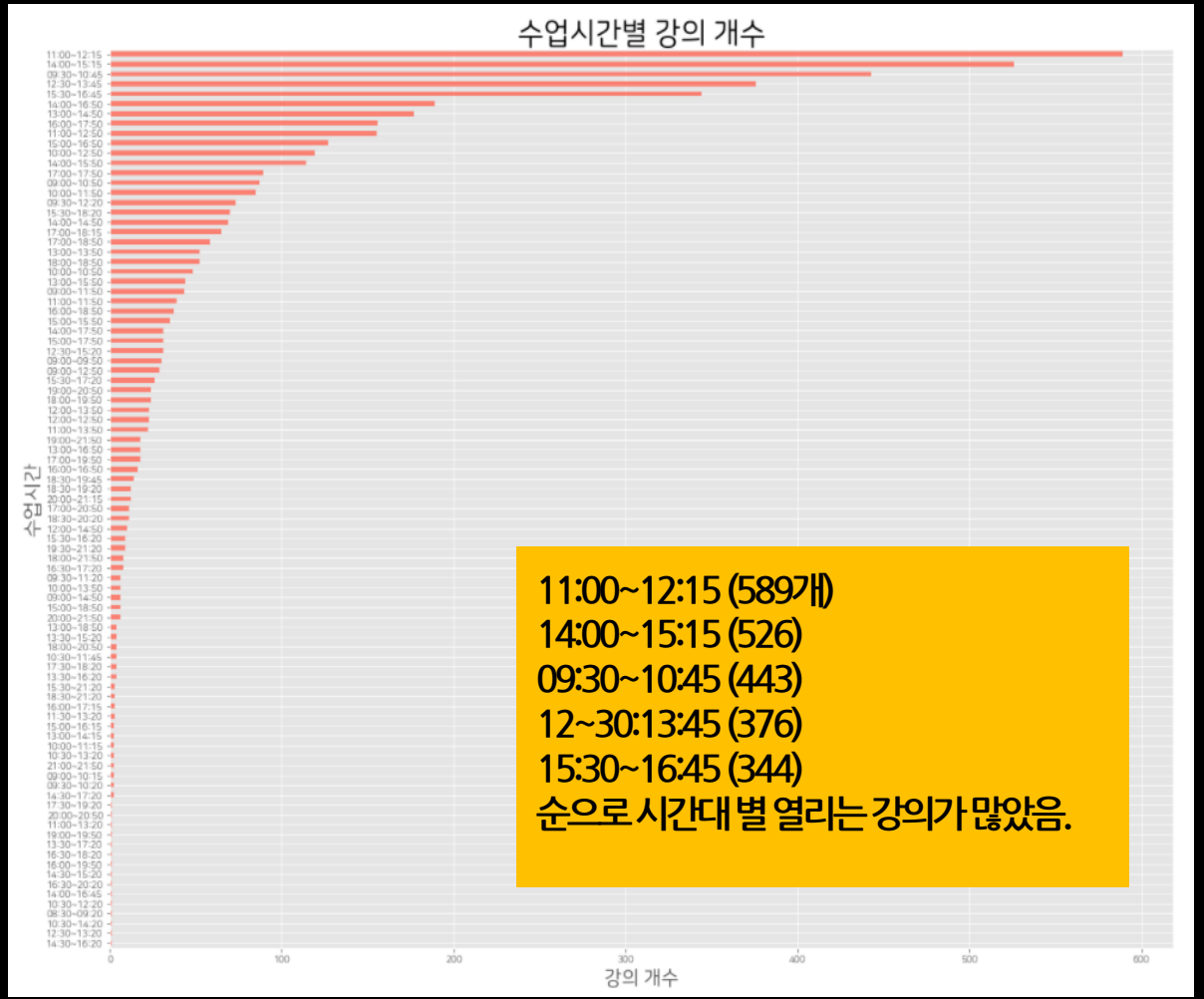
주 2회 수업, 혹은 주 1회 수업을 하는 경우가 압도적으로 높은 비율을 차지함.

2.분석 및 시각화(5)수업교시별

4. 수업시간별 강의 개수(barplot)

```
#강의시간 범주별 개수 count
time_count={}
for x in lecture_time:
        try: time_count[x]+=1
        except: time_count[x]=1
time_count

#수업시간별 강의 개수 시각화
pit.figure(figsize=(20,16))
lecture_num_df['강의 개수'].plot(kind='
barh',color='salmon')
plt.title('수업시간별 강의 개수')
plt.xlabel('강의 개수')
plt.ylabel('수업시간')
plt.show()
```



11:00~12:15 (589개)
14:00~15:15 (526)
09:30~10:45 (443)
12~30:13:45 (376)
15:30~16:45 (344)
순으로 시간대 별 열리는 강의가 많았음.

2.분석 및 시각화 (6) 정원별

1.정원 별 강의 개수 시각화 (barplot)

```
plt.figure()
df['정원'].value_counts().plot(kind='bar',color='salmon')
plt.title('정원별 강의 개수 시각화')
plt.xlabel('정원')
plt.ylabel('강의 개수')
plt.show()
```

2. 정원 별 수업교시 강의 개수

```
df.groupby(['정원','수업교시']).count().sort_values(by='교과구분',ascending=False)
```

```
#결과 표는 생략
```

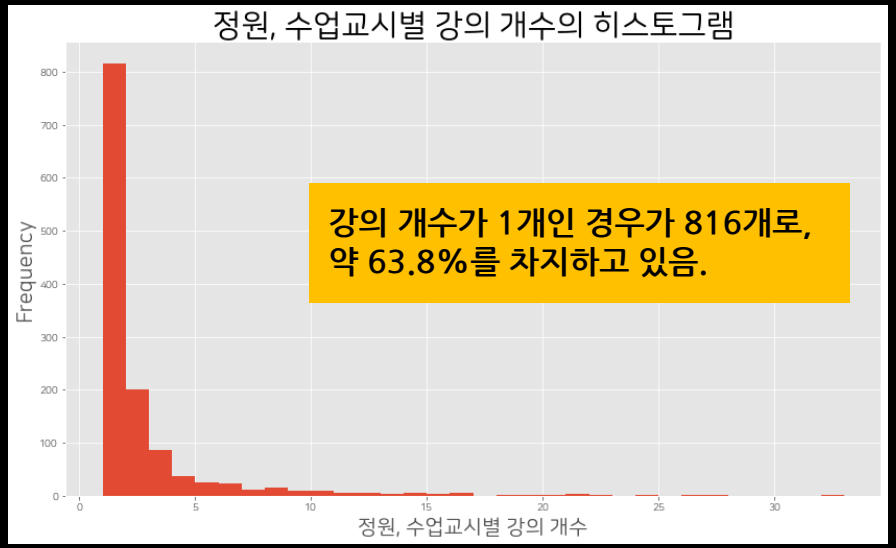
1)정원별 수업교시 강의 개수를 확인해본 결과,
정원별 수업교시별 범주의 수가 1278개임을 확인할 수 있었음.

2)정원, 수업교시별로 groupby해보니 정원이 40명이고 화/목 오전 11시에 시작하는 강의의 개수가 가장 많음을 확인할 수 있었음.

2.분석 및 시각화 (6) 정원별

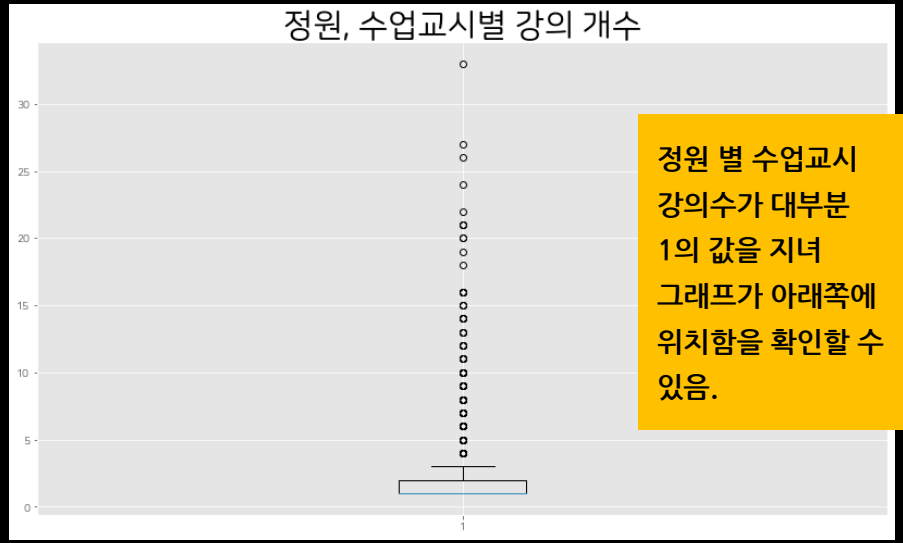
3.정원별 수업교시 강의 개수 시각화 (histogram)

```
plt.figure()
df.groupby(['정원','수업교시']).count()['이수과정']
.plot(kind='hist',bins=32)
plt.title('정원, 수업교시별 강의 개수의 히스토그램')
plt.xlabel('정원, 수업교시별 강의 개수')
plt.show()
```



4. 정원별 수업교시 강의 개수 시각화 (boxplot)

```
plt.figure()
plt.boxplot(df.groupby(['정원','수업교시']).coun
t()['이수과정'])
plt.title('정원, 수업교시별 강의 개수')
plt.show()
```



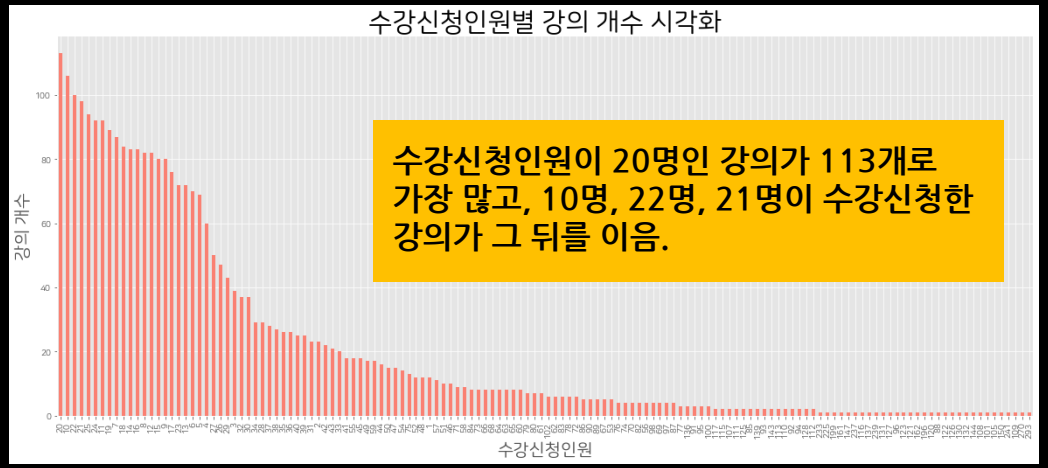
2.분석 및 시각화(7)수강신청인원별

1.수강신청인원별 강의 개수 시각화 (barplot)

```
plt.figure(figsize=(20,8))
df['수강신청인원'].value_counts().plot(kind='bar',color=
                'salmon')
plt.title('수강신청인원별 강의 개수 시각화')
plt.xlabel('수강신청인원')
plt.ylabel('강의 개수')
plt.show()
```

```
# 수강신청인원과 정원의 상관계수
df['수강신청인원'].corr(df['정원'])
```

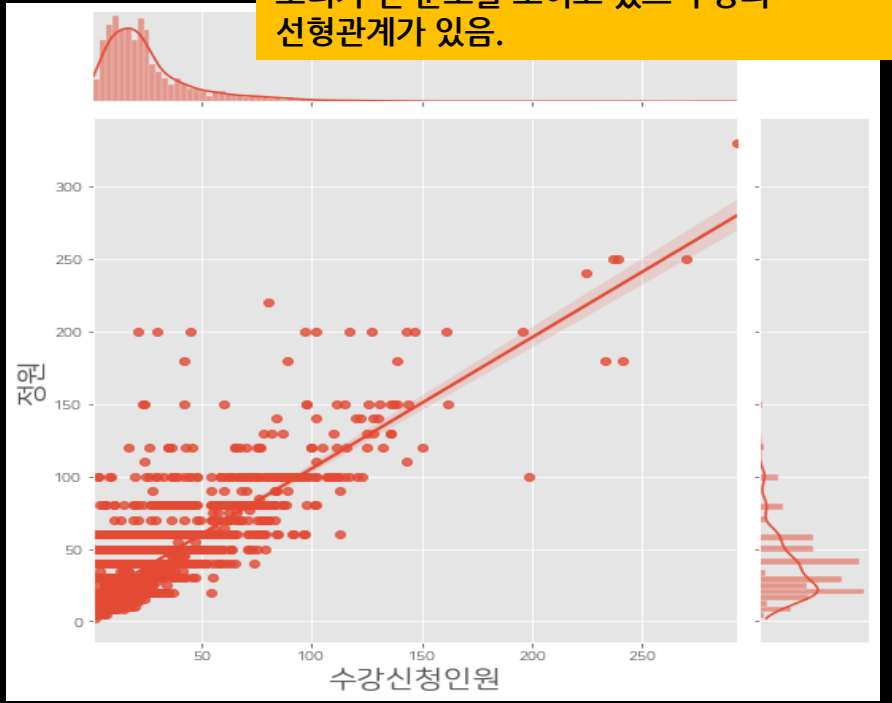
```
0.7984847199922184
```



2. 수강신청인원과 정원 시각화 (jointplot)

```
plt.figure()
g = sns.jointplot (x='수강신청인원', y='정원', kind='reg', data=df)
g.fig.set_figwidth(8)
g.fig.set_figheight(8)
plt.show()
```

1) 수강신청인원과 정원이 약 0.80의 높은 양의 상관관계를 보임
2) 수강신청인원, 정원 두 값모두 오른쪽으로 꼬리가 긴 분포를 보이고 있으며 양의 선형관계가 있음.



print (‘감사합니다!’)

〈역할분담〉

2015-16294 김호연 : 데이터 전처리, 전체 코드 수합 및 정리, 보고서 담당
2019-10056 김예진 : 정원별 데이터 처리 및 시각화 담당, PPT 제작
2020-19088 이혜인 : 수업교시별 데이터 처리 및 시각화 담당, PPT 제작
2019-18791 박성현 : 교과구분별 데이터 처리 및 시각화 담당, 영상 발표자
2019-13709 박나은 : 수강신청 인원 별 데이터 처리 및 시각화 담당, PPT 제작

<부록> 프로젝트를 마치며

호연 : 매 학기마다 수강신청하며 봐온 실제 강의 데이터를 직접 추출하여 분석한다는 것이 실생활과 밀접한 관련이 있어 프로젝트에 더 열심히 참여할 수 있었다. 좋은 주제를 제안해주신 팀원분들께 감사하고, 컴퓨팅 기초 수업에서 배운 내용을 토대로 분석 및 시각화를 진행함으로써 강의의 분포를 알 수 있어서 흥미로웠다. 또한 코드 수합 및 정리, 부가적인 컬럼별 분석을 진행하며 수업 내용을 응용해 볼 수 있어 좋았다.

예진 : 교과구분, 정원, 수강신청인원, 수업교시에 따라 강의를 직접 분류하고 시각화함으로써 강의들이 많이 개설되는 특정 요일/시간대가 있는지, 각 칼럼 간의 상관관계는 어떠한지 등에 대한 인사이트를 얻을 수 있었다. 정제되지 않은 데이터로부터 원하는 데이터를 추출하고 Dataframe을 만들고자 코드를 작성하는 과정에서, 여러번 실패를 경험하며 구글링을 하거나 튜터님으로부터 도움을 얻는 과정에서 코딩 실력이 크게 성장한 것 같아 만족스럽다. 특히 다른 팀원분들이 작성한 코드와 나의 코드를 비교해보며 사고를 확장할 수 있었다. 이번 기말프로젝트는 그동안 컴퓨팅 기초 수업에서 배운 내용을 종합적으로 적용하고 나의 실력을 점검할 수 있는 좋은 기회였기에 의미가 큰 것 같다. 한 학기 동안 많은 가르침 주신 교수님, 튜터님들, 그리고 우리 팀원들 사랑합니다!!

성현 : 처음에는 막막해 보이기만 했던 프로젝트였지만, 팀원들과 소통하며 주제를 정하고, 데이터 분석 방식에 대해 토의하며 코딩에 대한 나의 시각을 넓힐 수 있었다. 파이썬을 이용하여 데이터를 편리하게 분석하는 것도 흥미로웠고, 코드의 결과값을 바탕으로 의미를 도출해내는 작업도 즐거웠다. 팀프로젝트가 순조롭게 마무리될 수 있도록 각자의 역할을 충실히 수행하신 모든 팀원 분들께 감사의 말씀을 전하고 싶다.

혜인 : 기말 프로젝트를 통해 평소에 궁금했던 수강편람 강의의 수업시간과 수업요일에 대한 정보를 분석하고 시각화 해볼 수 있어서 좋았다. 맨 땅에서 시작했다면 몇년이 걸릴 것 같은 방대한 데이터를 파이썬을 이용해 빠르게 분석하고 결과를 도출할 수 있다는 점을 배웠다. 무엇보다 팀원 분들 덕분에 기말 프로젝트를 진행하고 완성할 수 있었다는 점에 감사한다.

나은 : 기말 프로젝트를 통해 그동안 수업에서 배웠던 이론을 관심있는 주제에 적용하는 연습을 해볼 수 있었다. 주제 설정, 데이터 수집, 처리, 시각화의 과정을 거치며 직접 설정한 목표에 맞게 실행해나가는 과정은 사실 생각보다 많이 막연하고 어려웠다. 그러나 팀원들과 함께 고민을 공유하고, 피드백을 해나가며 어느 정도 의미있는 결과를 도출해냈을 때의 뿌듯함도 매우 컸다. 컴퓨팅 기초 수업은 이번 기말 프로젝트를 끝으로 마무리되겠지만, 앞으로 내가 하는 공부와 관심있는 분야에도 코딩을 활용하며 의미있는 데이터를 만들어나가고 싶다. 기말 프로젝트를 성공적으로 마무리하기까지 가르쳐주신 교수님과 튜터님들, 그리고 팀원들에게 정말 감사하다.