출석수업 과제물(평가결과물) 표지(온라인제출용)

**교과목명 : 파이썬과 R**

**학 번 : 202135-367895**

**성 명 : 김태정**

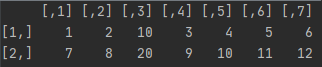
**강 의 실 : 지역대학 호**

**연 락 처 : 010-4172-4516**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**73페이지 4번 문제**

xm <- *matrix*(1:12, ncol = 6, byrow = T)  
*cbind*(xm[, 1:2], *c*(10, 20), xm[, 3:6])



cbind를 사용해서 행렬을 합치는데 가운데 끼워넣기 위해서 행렬을 1~2 슬라이싱 해서 앞에 두고 가운데 벡터를 넣고 마지막에 3~6을 슬라이싱해서 넣는다.

**73페이지 5번 문제**

import numpy as np  
  
mx = np.array(range(1, 13)).reshape(2, 6)  
print(np.insert(mx, 2, np.array([10, 20]), axis=1).reshape(2, 7))



np.array로 다차원 배열을 만들고 insert메소드를 사용해서10,20을 끼워넣는다.

**100페이지 7번 문제**

def mywage(time):  
 return time \* 10000 + ((time - 40) \* 5000 if time - 40 > 0 else 0)  
  
  
print(mywage(40))  
print(mywage(41))



시간당 만원이고 초과한 분은 1.5배인 15000원이 된다. 계산 법은 그냥 시간 별로 만원을 준다음에 삼항 연산자로 40이상에 5천원을 곱하면된다.

**100페이지 8번 문제**

mywage <- function(time) {  
 *list*(result = time \* 10000 + *ifelse*(time - 40 > 0, (time - 40) \* 5000, 0))  
}  
  
*sprintf*('%d', *mywage*(40)$result)  
*sprintf*('%d', *mywage*(41)$result)



40시간과 상관없이 일단 시간당 만원으로 계산하고 초과분은 ifelse함수를 사용해서 5000원을 추가해준다. 출력은 정수로 출력하기위해서 sprintf를 사용한다.

**100페이지 9번 문제**

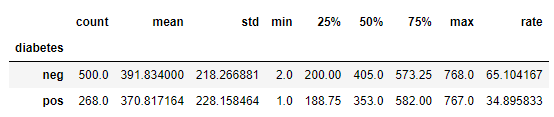
class example:  
 def \_\_init\_\_(self, name):  
 self.a = 'Hello ' + name + ' !'  
 self.b = 'Good-bye ' + name + ' !'  
  
  
name = 'David'  
aaa = example(name)  
print(aaa.a)  
print(aaa.b)



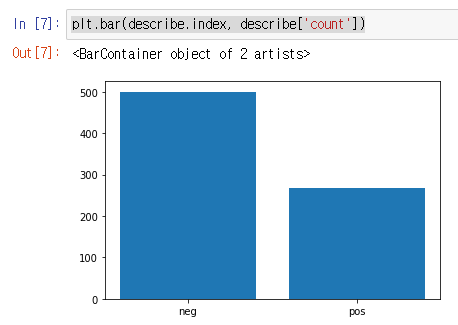
클래스 문법을 사용해서 값을 입력받았을 때 a와 b의 값을 설정하는 함수를 만들고 해당값의 a와 b를 출력한다.

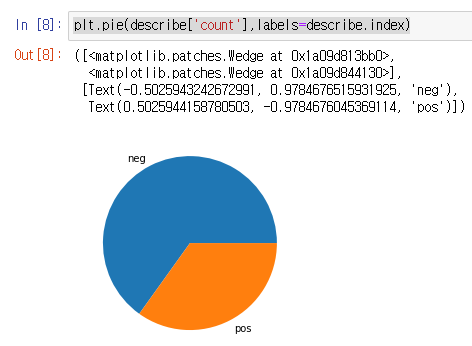
**269페이지 3.1번 파이썬**

import pandas as pd  
import matplotlib.pyplot as plt  
  
pima = pd.read\_csv('../data/pima2.csv')  
  
describe = pima.groupby('diabetes')['Unnamed: 0'].describe()  
describe['rate'] = pima.groupby('diabetes')['age'].describe()['count'] / sum(  
 pima.groupby('diabetes')['age'].describe()['count']) \* 100  
  
print(describe)  
  
plt.bar(describe.index, describe['count'])  
plt.show()  
plt.pie(describe['count'], labels=describe.index)  
plt.show()



빈도수는 count값을 알 수 있고 비율은 전체에 각 그룹의 개수로 나눠서 알 수 있다. 여기서는 rate라는 변수를 사용했다. 이 값을 이용해서 bar와 pie메소드를 사용해서 값을 출력할 수 있다.

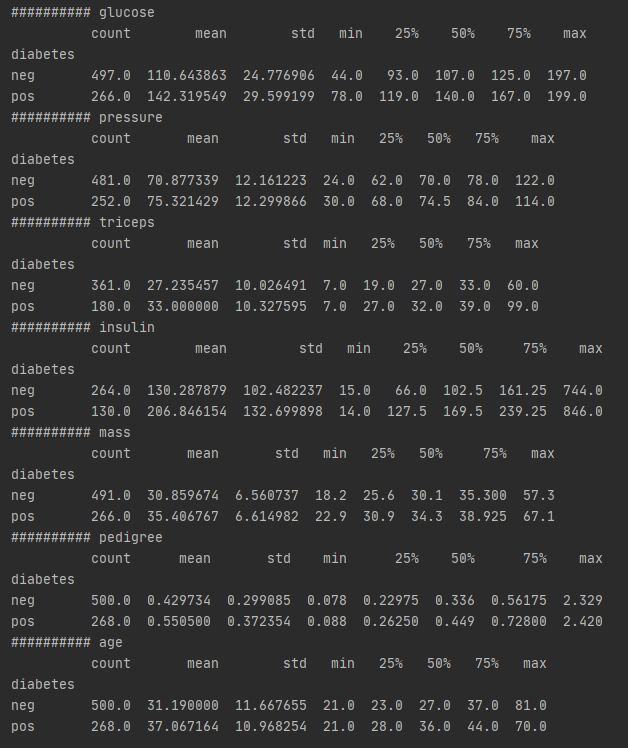


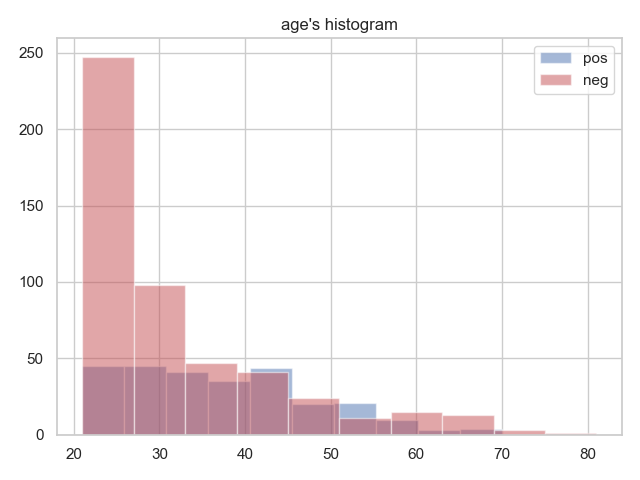
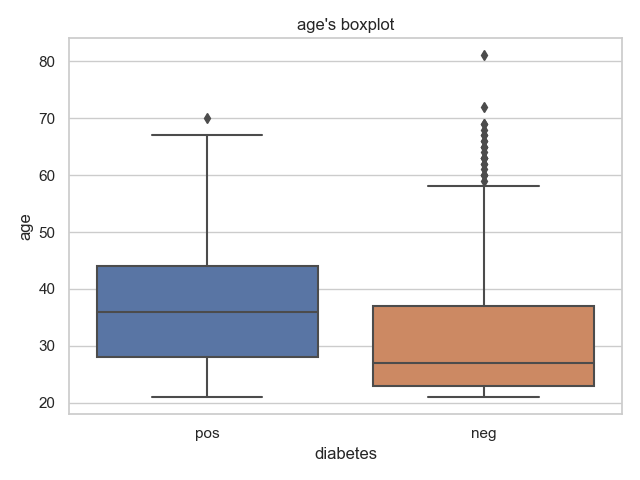
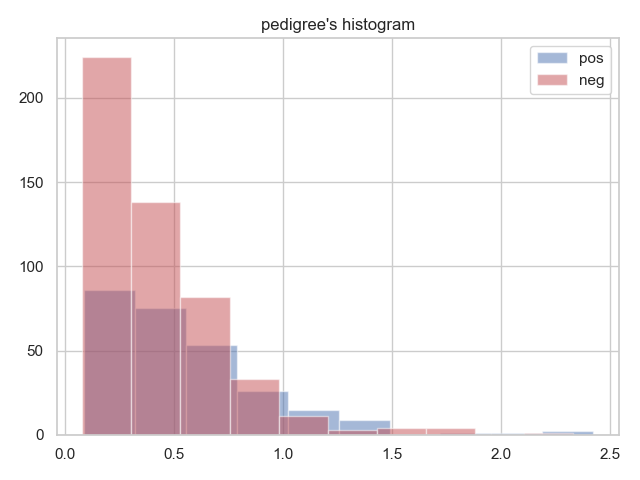
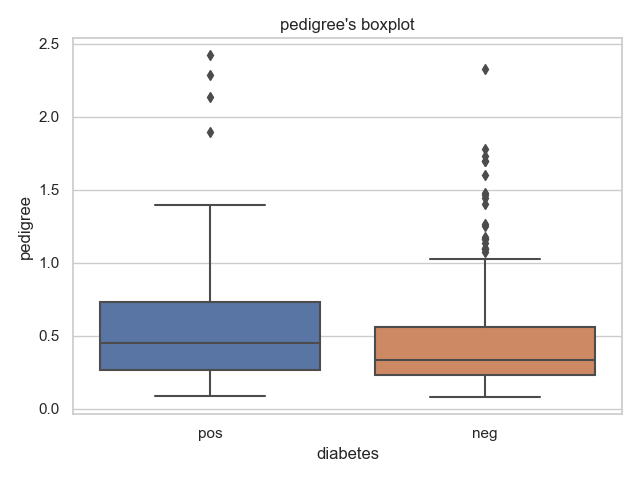
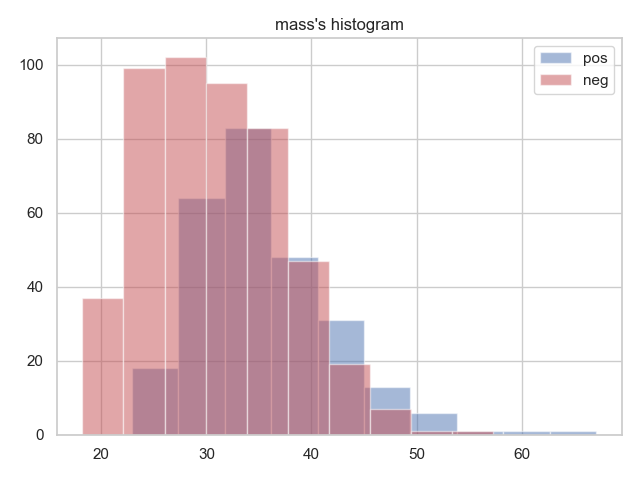
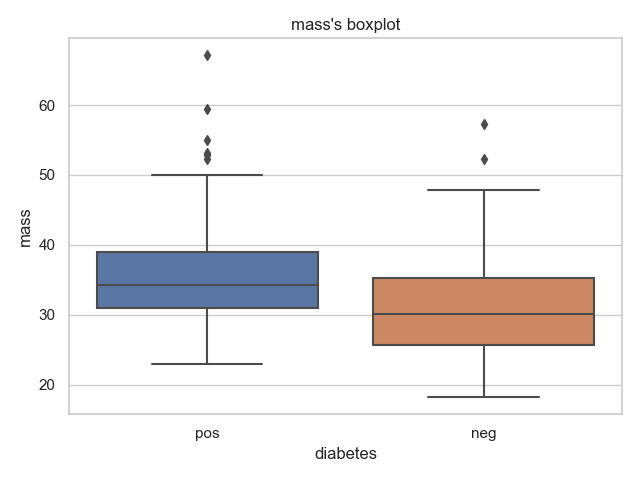
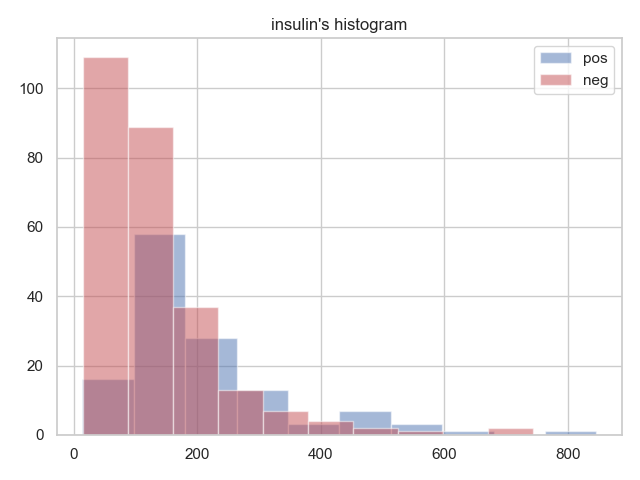
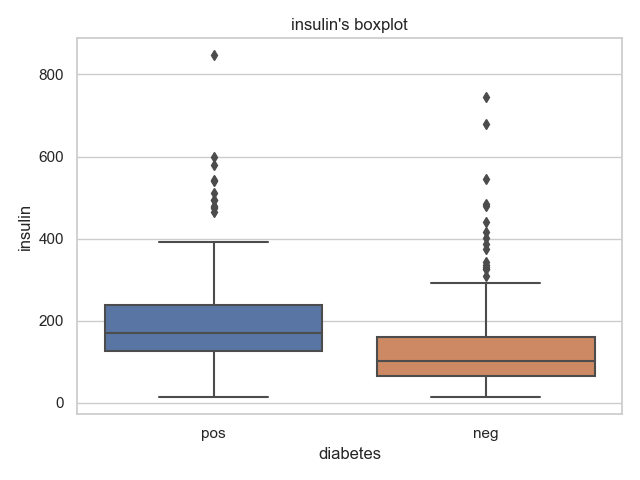
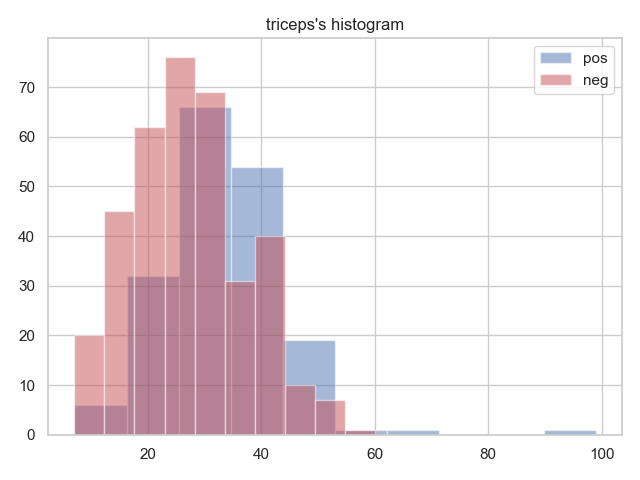
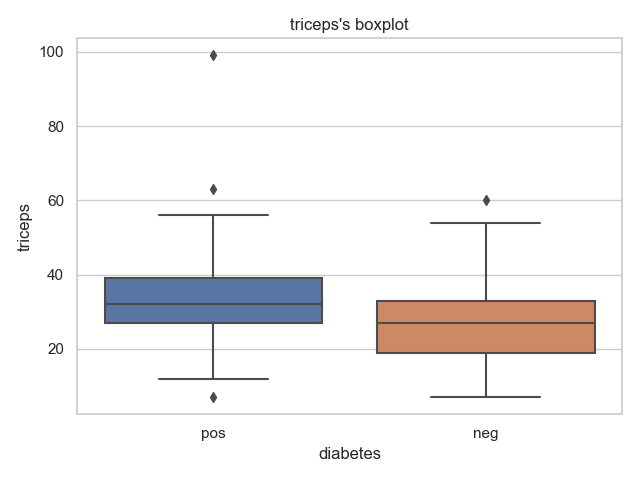
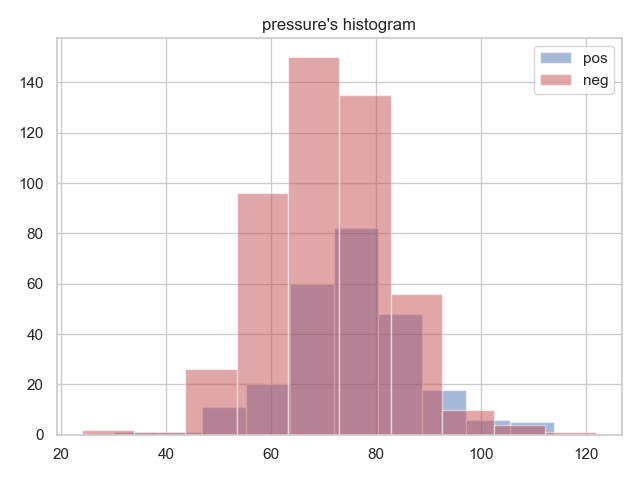
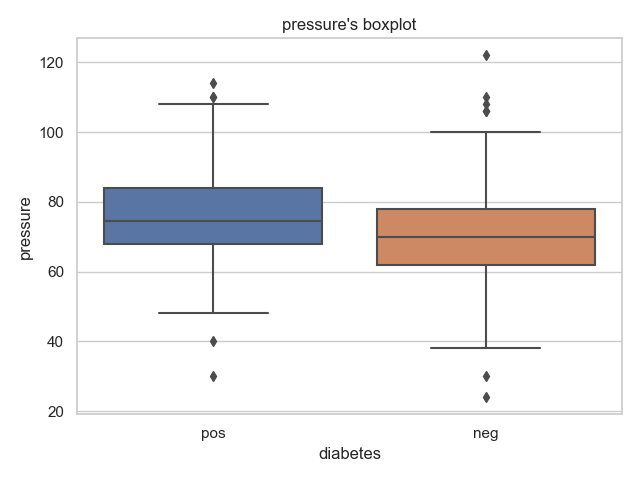
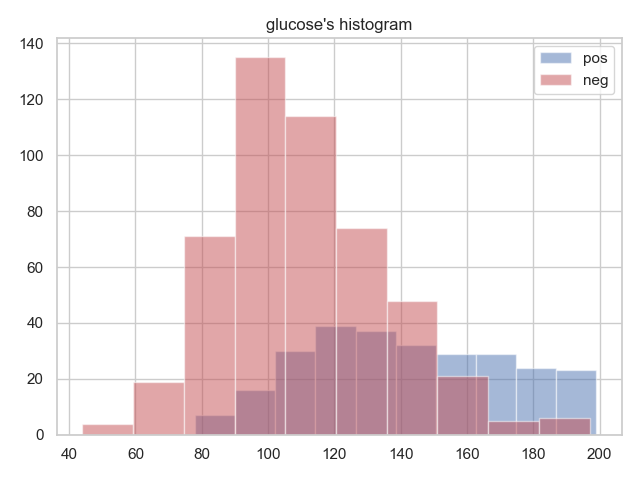
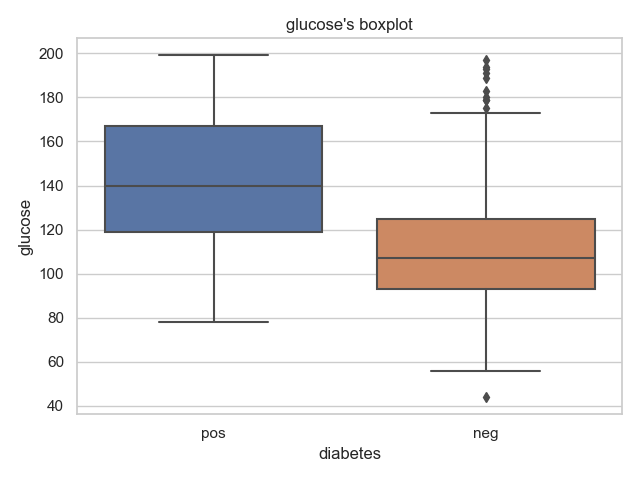


**269페이지 3.2번 파이썬**

import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
pima = pd.read\_csv('../data/pima2.csv')  
  
describe = pima.groupby('diabetes').describe()  
  
for col in ['glucose', 'pressure', 'triceps', 'insulin', 'mass', 'pedigree', 'age']:  
 print('#' \* 10 + ' ' + col)  
 print(describe[col])  
 sns.set(style='whitegrid')  
 sns.boxplot(x='diabetes',y=col, data=pima)  
 plt.gca().set(title='%s\'s boxplot'%(col))  
 plt.show()  
 pos\_total = pima.loc[pima.diabetes=='pos'][col]  
 neg\_total = pima.loc[pima.diabetes=='neg'][col]  
 args = dict(alpha = 0.5, bins = 10)  
 plt.hist(pos\_total, \*\*args, color='b', label='pos')  
 plt.hist(neg\_total, \*\*args, color='r', label='neg')  
 plt.gca().set(title='%s\'s histogram'%(col))  
 plt.legend()  
 plt.show()

그룹화의 경우 groupby를 사용해서 구해준다. 각 변수별로 출력을 해야하기 때문에 for문을 돌려서 변수별로 실행한다. 세가지를 구해야하는데 기술통계, 히스토그램, 상자그림을 구해야한다. 기술통계는 describe메소드를 사용하면 구할 수 있다. 히스토그램은 plt.hist를 사용하면된다. 여기서 히스토그램을 따로 그릴 수도 있지만 같이 보기 위해서 여기서는 같이 그렸다.

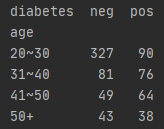


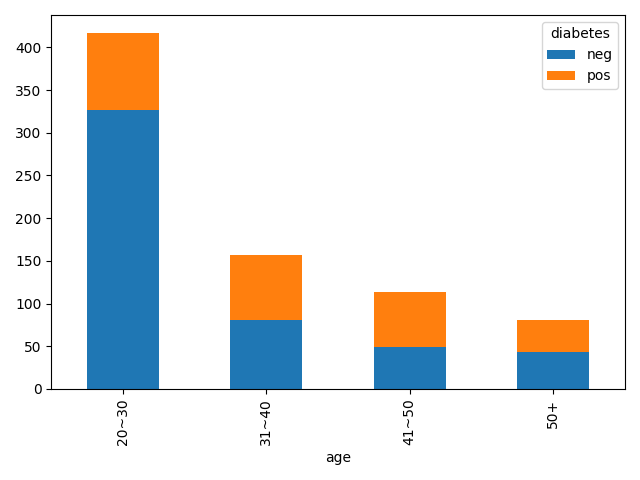


**269페이지 3.3번 파이썬**

import pandas as pd  
import matplotlib.pyplot as plt  
  
pima = pd.read\_csv('../data/pima2.csv')  
  
  
def classify(age):  
 if age <= 19:  
 return '0~19'  
 elif 20 <= age <= 30:  
 return '20~30'  
 elif 31 <= age <= 40:  
 return '31~40'  
 elif 41 <= age <= 50:  
 return '41~50'  
 else:  
 return '50+'  
  
  
pima['age'] = pima['age'].apply(classify)  
table = pd.crosstab(index=pima['age'], columns=pima['diabetes'])  
print(table)  
table.plot.bar(stacked=True)  
plt.show()

범위를 그룹화 하기 위해서 기준이 필요하기에 기준을 만들 classify함수를 만들어준다. 그 다음 apply메소드를 사용해서 각각 행의 나이를 확인해서 계급화 시켜주고 이를 이용하여 크로스탭을 만들어준다.

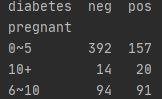


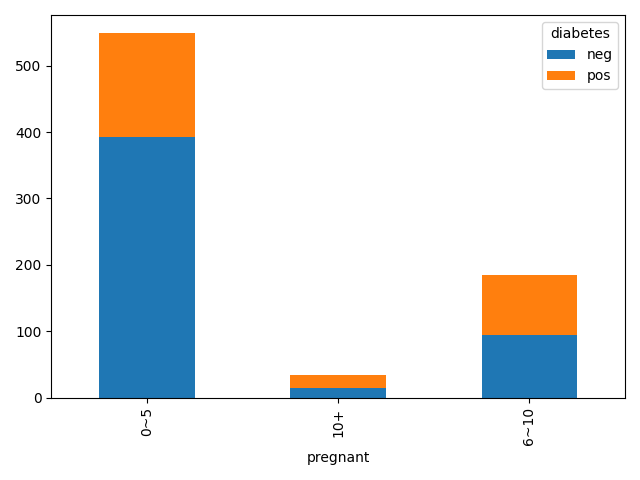


**269페이지 3.4번 파이썬**

import pandas as pd  
import matplotlib.pyplot as plt  
  
pima = pd.read\_csv('../data/pima2.csv')  
  
  
def classify(pregnant):  
 if pregnant <= 5:  
 return '0~5'  
 elif 6 <= pregnant <= 10:  
 return '6~10'  
 else:  
 return '10+'  
  
  
pima['pregnant'] = pima['pregnant'].apply(classify)  
table = pd.crosstab(index=pima['pregnant'], columns=pima['diabetes'])  
print(table)  
table.plot.bar(stacked=True)  
plt.show()

3.3번과 동일한 방식으로 푼다. 등급화 시켜주는 함수만 바꿔준다.





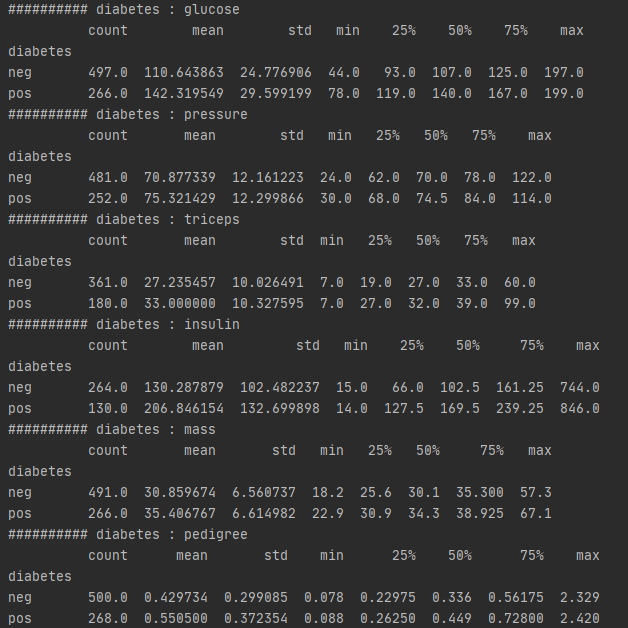
**269페이지 3.5번 파이썬**

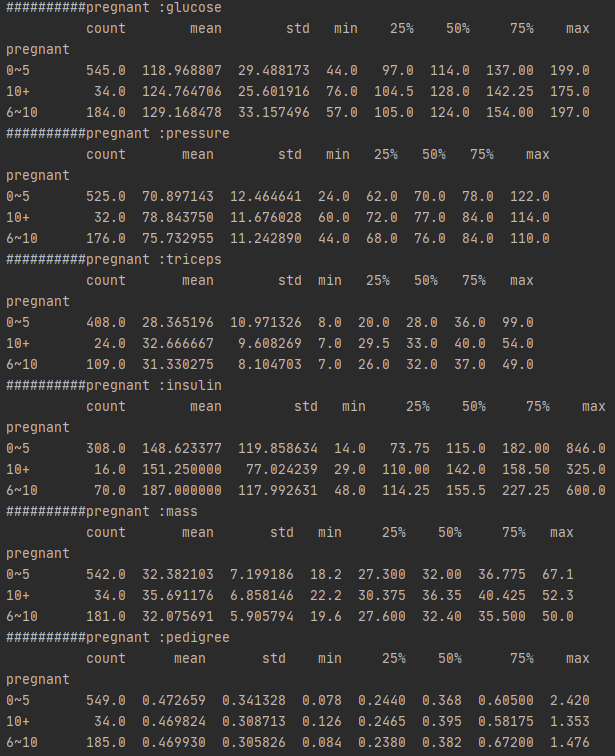
import pandas as pd  
  
pima = pd.read\_csv('../data/pima2.csv')  
for col in ['glucose', 'pressure', 'triceps', 'insulin', 'mass', 'pedigree']:  
 print('#' \* 10 + ' diabetes : ' + col)  
 print(pima.groupby('diabetes').describe()[col])  
  
  
def classify(pregnant):  
 if pregnant <= 5:  
 return '0~5'  
 elif 6 <= pregnant <= 10:  
 return '6~10'  
 else:  
 return '10+'  
  
  
pima['pregnant'] = pima['pregnant'].apply(classify)  
for col in ['glucose', 'pressure', 'triceps', 'insulin', 'mass', 'pedigree']:  
 print('#' \* 10 + 'pregnant :' + col)  
 print(pima.groupby('pregnant').describe()[col])

그룹에 대한 평균과 표준편차는 describe메소드를 통해서 추출해 낼 수 있다. 과제에서는 평균과 표준편차를 구하니 아래처럼 사용해서 추출할 수 있다.

for col in ['glucose', 'pressure', 'triceps', 'insulin', 'mass', 'pedigree']:  
 print('#' \* 10 + 'pregnant:' + col+':mean')  
 data = pima.groupby('pregnant').describe()[col]  
 print(data['mean'])  
 print('#' \* 10 + 'pregnant:' + col+':std')  
 print(data['std'])

코드에서 보면 그냥 나온 결과에 각각 지표를 사용하면 뽑아낼 수 있다.

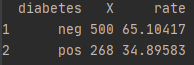




**269페이지 3.1번 R**

pima <- *read.csv*('./data/pima2.csv', header = T)  
result <- *aggregate*(pima['X'], *list*(diabetes = pima$diabetes), length)  
result['rate'] <- result['X'] / *sum*(result['X']) \* 100  
  
*print*(result)  
  
result\_table <- *table*(pima$diabetes)  
*par*(mfrow = *c*(1, 2))  
*barplot*(result\_table, col = 1:2)  
*pie*(result\_table, col = 1:2)

aggregate를 사용해서 diabetes를 기준으로 값을 그룹화한다. Length를 쓰는 이유는 개수만 새면 되기 때문이다. 그 다음 비율을 구해주기 위해서 전체의 값에서 각 그룹의 값으로 나눠준다. 그 후 barplot과 pie를 사용해서 차트를 출력한다.



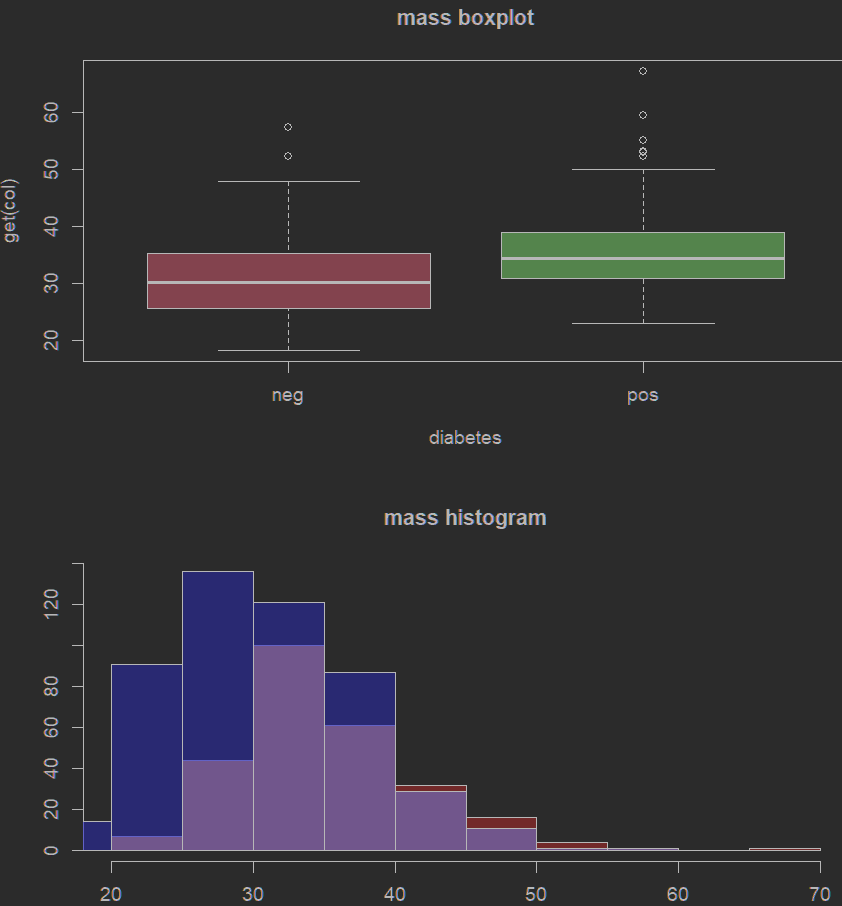
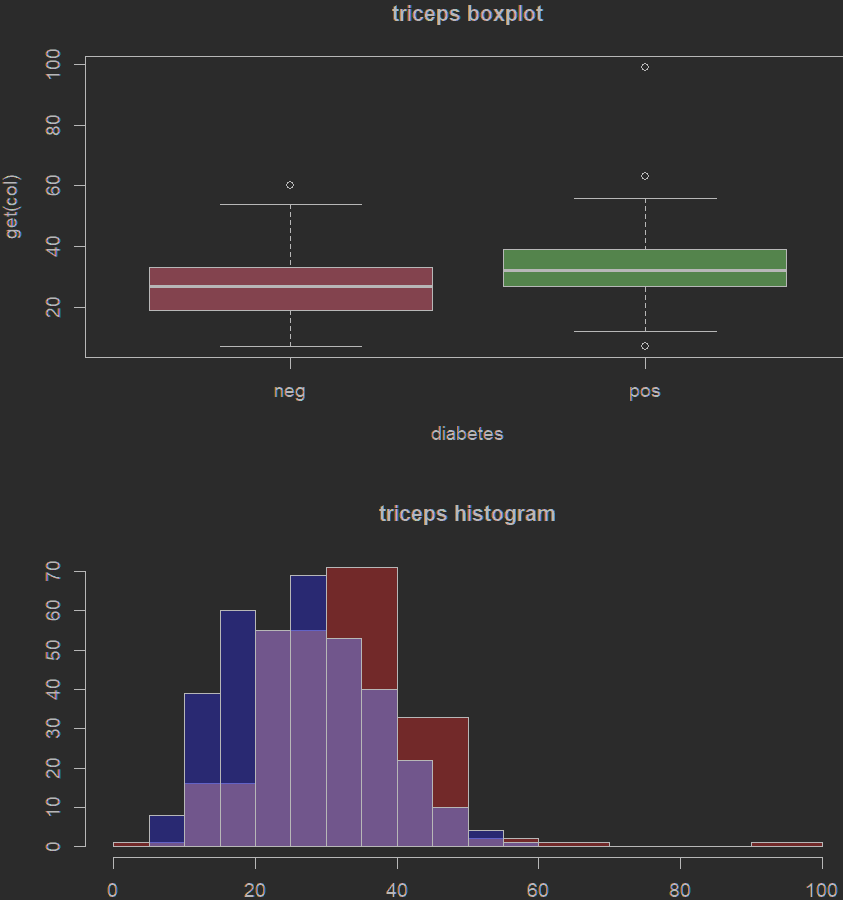
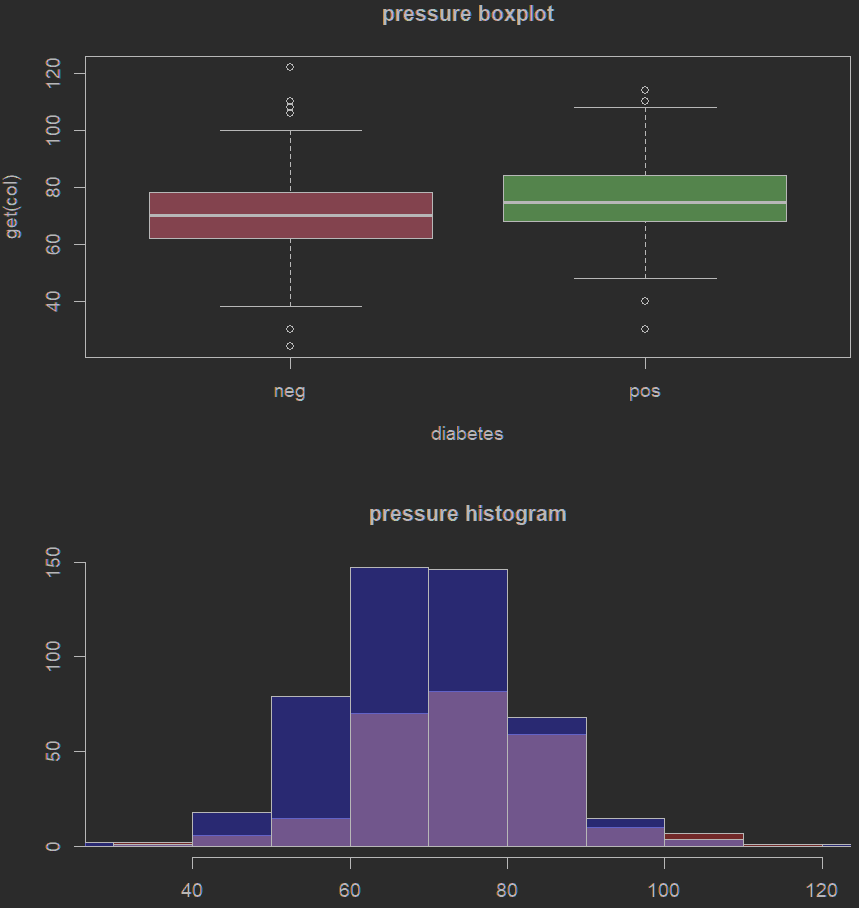
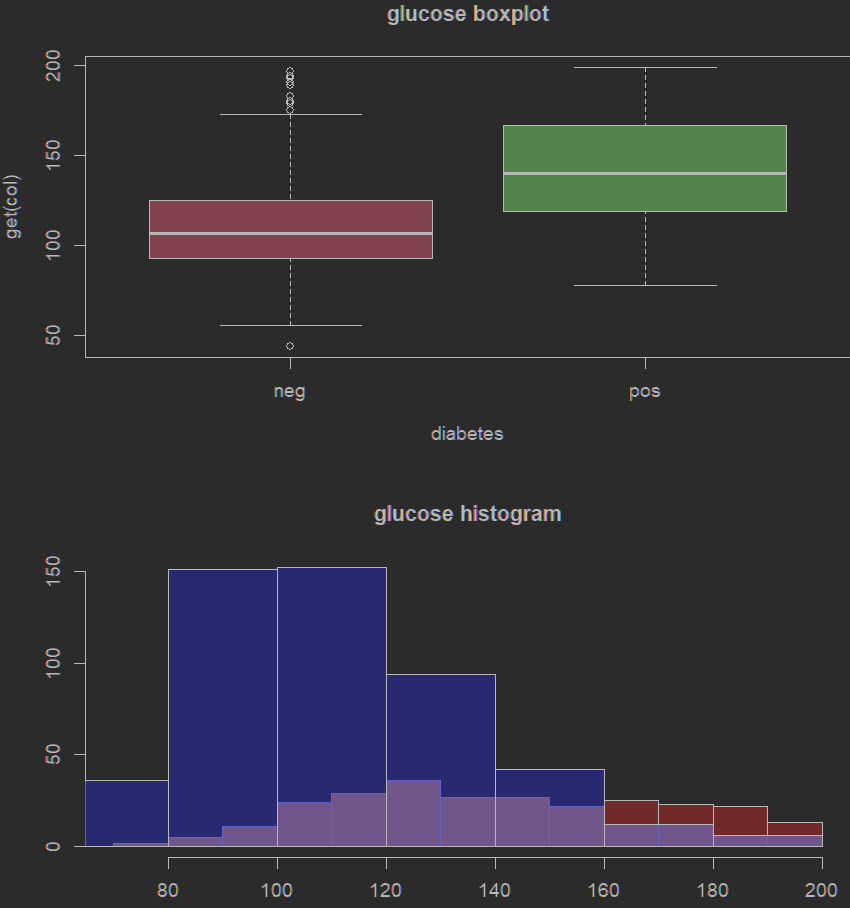
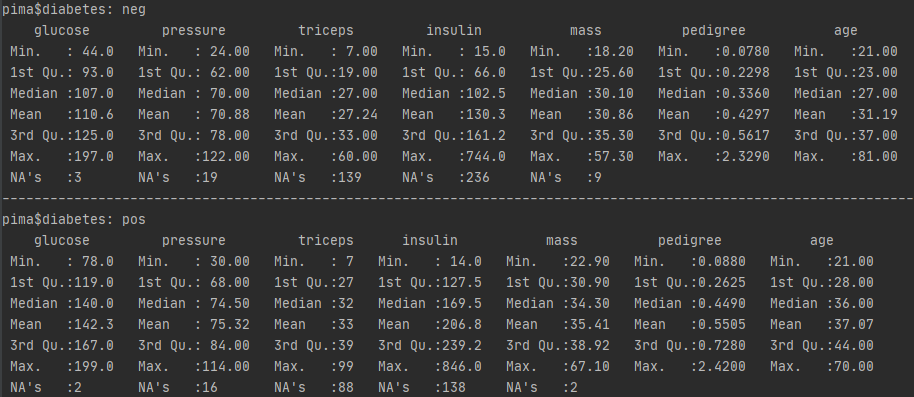
**269페이지 3.2번 R**

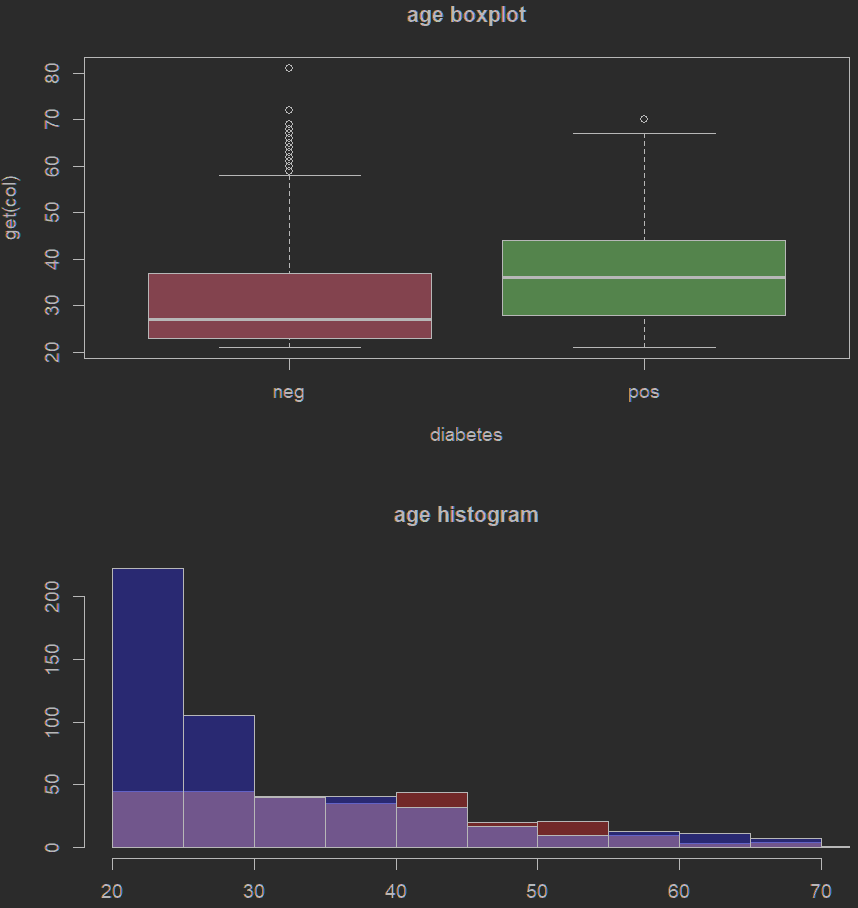
pima <- *read.csv*('./data/pima2.csv', header = T)  
*par*(mfrow = *c*(2, 1))  
cols <- *c*('glucose', 'pressure', 'triceps', 'insulin', 'mass', 'pedigree', 'age')  
*by*(pima[cols],pima$diabetes,summary)  
for (col in cols) {  
 *boxplot*(*get*(col) ~ diabetes, data = pima, main = *paste0*(col, ' boxplot'), col = 1:2)  
 hist\_pos <- *hist*(pima[[col]][pima$diabetes == 'pos'], plot = F)  
 hist\_neg <- *hist*(pima[[col]][pima$diabetes == 'neg'], plot = F)  
 *plot*(hist\_pos,  
 col = *adjustcolor*("red", alpha = 0.5),  
 ann = FALSE,  
 ylim = *c*(0, *max*(hist\_pos$counts, hist\_neg$counts)))  
 *plot*(hist\_neg,  
 col = *adjustcolor*("blue", alpha = 0.5),  
 add = TRUE,  
 ylim = *c*(0, *max*(hist\_pos$counts, hist\_neg$counts)))  
 *title*(main = *paste0*(col, ' histogram'))  
}

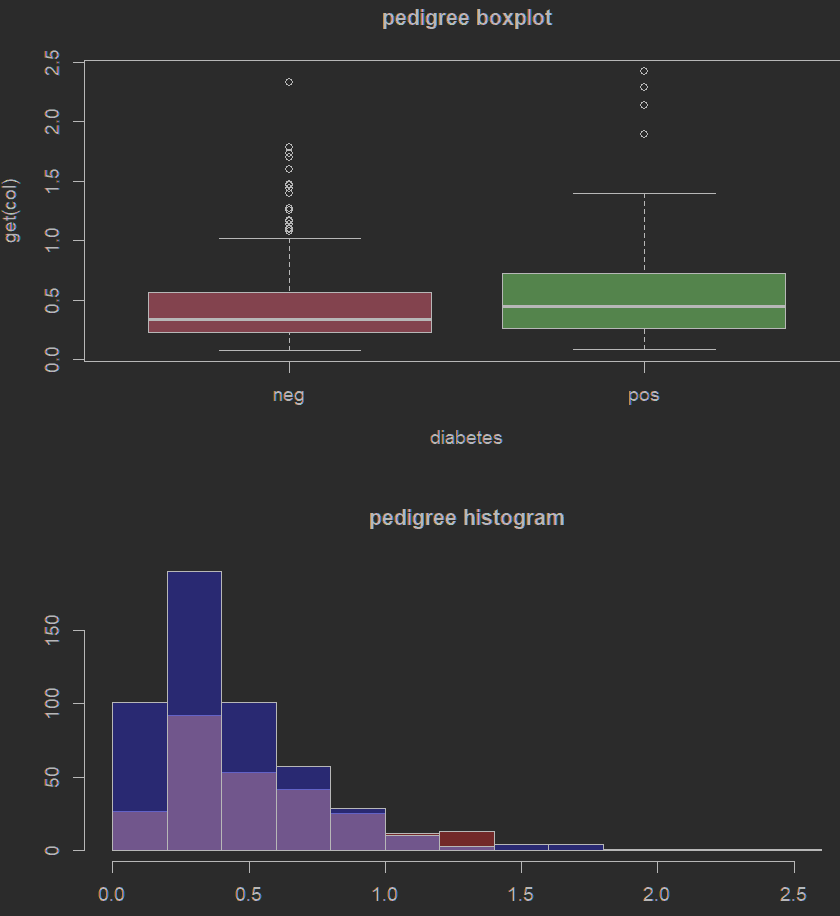
by를 사용해서 기술통계를 구한다. 그 이후 각 열을 순회하면서 히스토그램과 파이그래프에서 값에 쓸 값을 구한다.

pima[[col]][pima$diabetes == 'pos']

그 이후 히스토그램과 그래프를 출력한다.



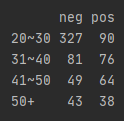


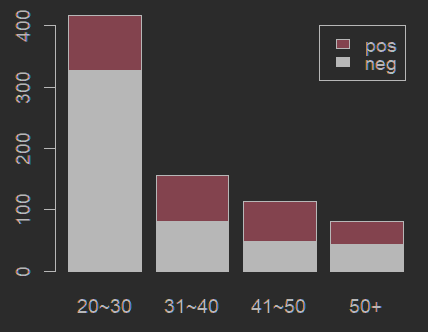


**269페이지 3.3번 R**

pima <- *read.csv*('./data/pima2.csv', header = T)  
  
classify <- function(age) {  
 if (age <= 19) {  
 *return*('0~19')  
 }else if (20 <= age && age <= 30) {  
 *return*('20~30')  
 }else if (31 <= age && age <= 40) {  
 *return*('31~40')  
 }else if (41 <= age && age <= 50) {  
 *return*('41~50')  
 }else {  
 *return*('50+')  
 }  
}  
  
pima$age <- *unlist*(*lapply*(pima$age, classify))  
tbl <- *table*(pima$age, pima$diabetes)  
tbl  
*barplot*(*t*(tbl), col = 1:2, legend = *colnames*(tbl))

lapply로 각열을 순회하면서 그룹화에 기준이 될변수를 정해서 age에 덮어쓴다. 그 이후 table을 제작하고 출력한 후 그래프를 출력한다.

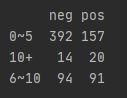


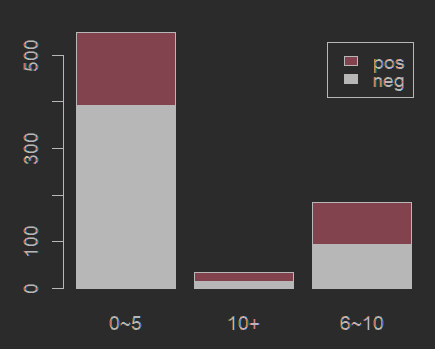


**269페이지 3.4번 R**

pima <- *read.csv*('./data/pima2.csv', header = T)  
  
classify <- function(pregnant) {  
 if (pregnant <= 5) {  
 *return*('0~5')  
 }else if (6 <= pregnant && pregnant <= 10) {  
 *return*('6~10')  
 }else {  
 *return*('10+')  
 }  
}  
  
pima$pregnant <- *unlist*(*lapply*(pima$pregnant, classify))  
tbl <- *table*(pima$pregnant, pima$diabetes)  
tbl  
*barplot*(*t*(tbl), col = 1:2, legend = *colnames*(tbl))

lapply로 각열을 순회하면서 그룹화에 기준이 될변수를 정해서 pregnant에 덮어쓴다. 그 이후 table을 제작하고 출력한 후 그래프를 출력한다.





**269페이지 3.5번 R**

pima <- *read.csv*('./data/pima2.csv', header = T)  
  
classify <- function(pregnant) {  
 if (pregnant <= 5) {  
 *return*('0~5')  
 }else if (6 <= pregnant && pregnant <= 10) {  
 *return*('6~10')  
 }else {  
 *return*('10+')  
 }  
}  
  
  
for (col in *c*('glucose', 'pressure', 'triceps', 'insulin', 'mass', 'pedigree')) {  
 *print*(*paste0*('\*\*\*\*\*\*\*\*\*\*:', col))  
 *print*('##########:MEAN')  
 *print*(*aggregate*(pima[col], *list*(diabetes = pima$diabetes), mean, na.rm = T))  
 *print*('##########:SD')  
 *print*(*aggregate*(pima[col], *list*(diabetes = pima$diabetes), sd, na.rm = T))  
}  
pima$pregnant <- *unlist*(*lapply*(pima$pregnant, classify))  
  
for (col in *c*('glucose', 'pressure', 'triceps', 'insulin', 'mass', 'pedigree')) {  
 *print*(*paste0*('\*\*\*\*\*\*\*\*\*\*:', col))  
 *print*('##########:MEAN')  
 *print*(*aggregate*(pima[col], *list*(diabetes = pima$pregnant), mean, na.rm = T))  
 *print*('##########:SD')  
 *print*(*aggregate*(pima[col], *list*(diabetes = pima$pregnant), sd, na.rm = T))  
}

aggregate를 이용해서 집계함수를 출력한다. 이때는 mean과 sd를 따로넣어서 출력해주는데 전체적으로 구하고 싶다면 그냥 간편하게 summary를 넣어주면된다. Pregnant는 등급이 필요하기에 등급을 구해준후에 반복문을 돌려서 출력한다.

