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

---

73 페이지 4 번 문제

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

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7]
[1,]    1    2   10    3    4    5    6
[2,]    7    8   20    9   10   11   12
```

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

73 페이지 5 번 문제

```python
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))
```

```
[[ 1  2 10  3  4  5  6]
 [ 7  8 20  9 10 11 12]]
```

np.array 로 다차원 배열을 만들고 insert 메소드를 사용해서 10,20 을 끼워넣는다.

100 페이지 7 번 문제

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


print(mywage(40))
print(mywage(41))
```

```
400000
415000
```

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

100 페이지 8 번 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)
```

```
[1] "400000"
[1] "415000"
```

40 시간과 상관없이 일단 시간당 만원으로 계산하고 초과분은 ifelse 함수를 사용해서 5000 원을 추가해준다. 출력은 정수로 출력하기위해서 sprintf 를 사용한다.

100 페이지 9 번 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)
```

```
Hello David !
Good-bye David !
```

클래스 문법을 사용해서 값을 입력받았을 때 a 와 b 의 값을 설정하는 함수를 만들고 해당값의 a 와 b 를 출력한다.

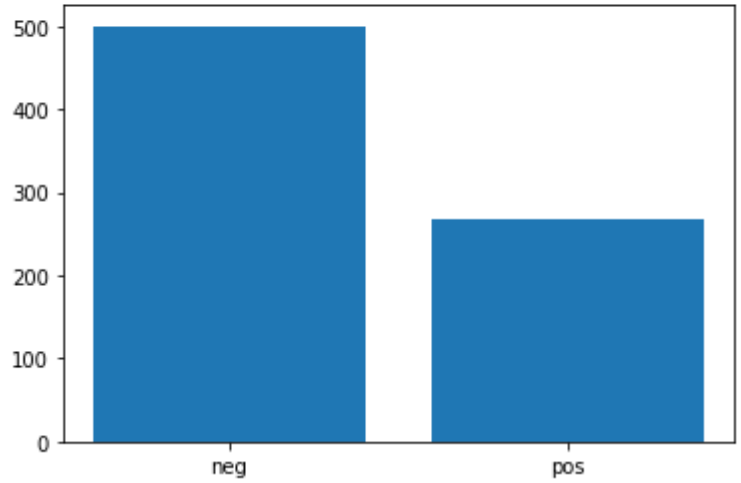
269 페이지 3.1 번 파이썬

```
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()
```

diabetes	count	mean	std	min	25%	50%	75%	max	rate
neg	500.0	391.834000	218.266881	2.0	200.00	405.0	573.25	768.0	65.104167
pos	268.0	370.817164	228.158464	1.0	188.75	353.0	582.00	767.0	34.895833

빈도수는 count 값을 알 수 있고 비율은 전체에 각 그룹의 개수로 나눠서 알 수 있다. 여기서는 rate 라는 변수를 사용했다. 이 값을 이용해서 bar 와 pie 메소드를 사용해서 값을 출력할 수 있다.

```
In [7]: plt.bar(describe.index, describe['count'])
Out[7]: <BarContainer object of 2 artists>
```




```
In [8]: plt.pie(describe['count'],labels=describe.index)
Out[8]: ([<matplotlib.patches.Wedge at 0x1a09d813bb0>,
  <matplotlib.patches.Wedge at 0x1a09d844130>],
 [Text(-0.5025943242672991, 0.9784676515931925, 'neg'),
  Text(0.5025944158780503, -0.9784676045369114, 'pos')])
```

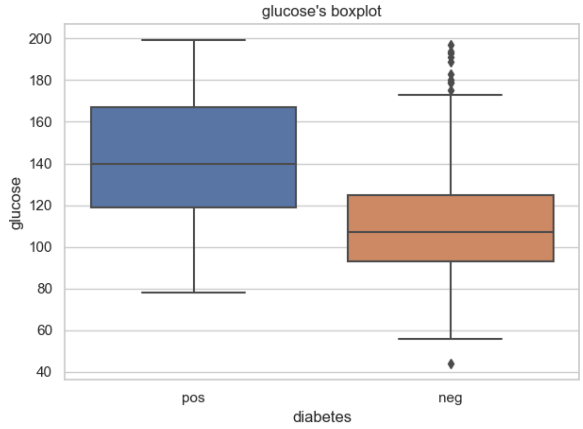
269 페이지 3.2 번 파이썬

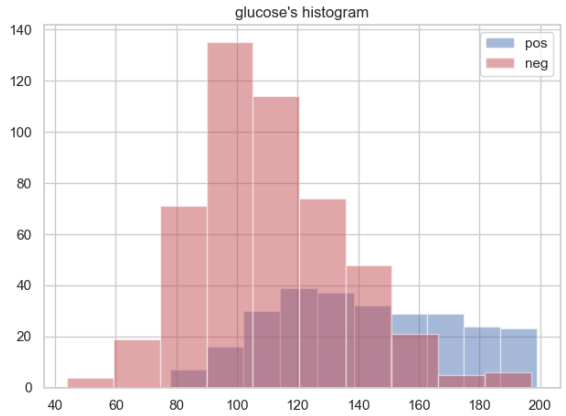
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

pima = pd.read_csv('../data/pima2.csv')

describe = pima.groupby('diabetes').describe()

for col in ['glucose', 'pressure', 'triceps', 'insulin', 'mass', 'pedigree',
```

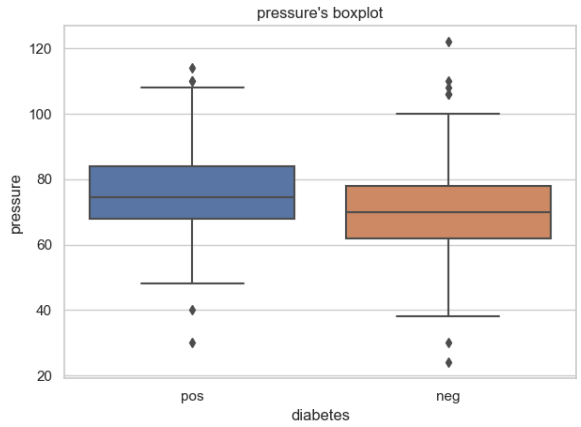
```
'age']:
    print('#' * 10 + ' ' + col)
    print(describe[col])
    sns.set(style='whitegrid')
    sns.boxplot(x='diabetes',y=col, data=pima)
    plt.gca().set(title='%s\'s boxplot'%(col))
    plt.show()
    pos_total = pima.loc[pima.diabetes=='pos'][col]
    neg_total = pima.loc[pima.diabetes=='neg'][col]
    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()
```

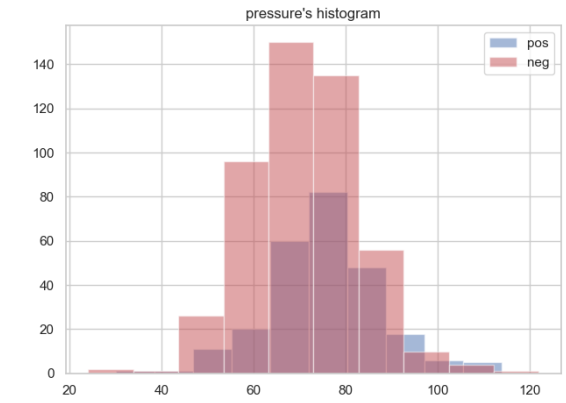
그룹화의 경우 groupby 를 사용해서 구해준다. 각 변수별로 출력을 해야하기 때문에 for 문을 돌려서 변수별로 실행한다. 세가지를 구해야하는데 기술통계, 히스토그램, 상자그림을 구해야한다. 기술통계는 describe 메소드를 사용하면 구할 수 있다. 히스토그램은 plt.hist 를 사용하면된다. 여기서 히스토그램을 따로 그릴 수도 있지만 같이 보기 위해서 여기서는 같이 그렸다.

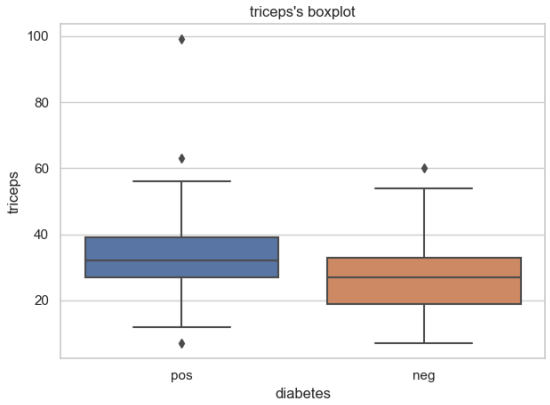
```
########## glucose
           count        mean        std   min    25%    50%    75%    max
diabetes
neg        497.0  110.643863  24.776906  44.0   93.0  107.0  125.0  197.0
pos        266.0  142.319549  29.599199  78.0  119.0  140.0  167.0  199.0
########## pressure
           count       mean        std   min   25%   50%   75%    max
diabetes
neg        481.0  70.877339  12.161223  24.0  62.0  70.0  78.0  122.0
pos        252.0  75.321429  12.299866  30.0  68.0  74.5  84.0  114.0
########## triceps
           count       mean        std  min   25%   50%   75%   max
diabetes
neg        361.0  27.235457  10.026491  7.0  19.0  27.0  33.0  60.0
pos        180.0  33.000000  10.327595  7.0  27.0  32.0  39.0  99.0
########## insulin
           count        mean         std   min    25%    50%     75%    max
diabetes
neg        264.0  130.287879  102.482237  15.0   66.0  102.5  161.25  744.0
pos        130.0  206.846154  132.699898  14.0  127.5  169.5  239.25  846.0
########## mass
           count       mean       std   min   25%   50%     75%   max
diabetes
neg        491.0  30.859674  6.560737  18.2  25.6  30.1  35.300  57.3
pos        266.0  35.406767  6.614982  22.9  30.9  34.3  38.925  67.1
########## pedigree
           count      mean       std    min      25%    50%      75%    max
diabetes
neg        500.0  0.429734  0.299085  0.078  0.22975  0.336  0.56175  2.329
pos        268.0  0.550500  0.372354  0.088  0.26250  0.449  0.72800  2.420
########## age
           count       mean        std   min   25%   50%   75%   max
diabetes
neg        500.0  31.190000  11.667655  21.0  23.0  27.0  37.0  81.0
pos        268.0  37.067164  10.968254  21.0  28.0  36.0  44.0  70.0
```

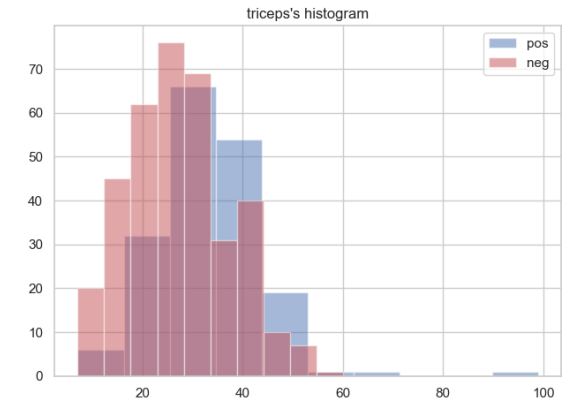

glucose's boxplot
glucose
diabetes
pos
neg

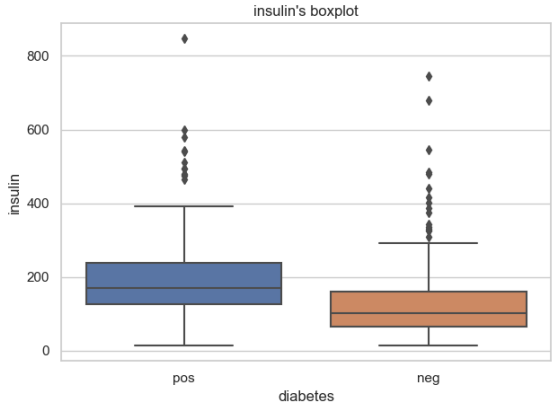

glucose's histogram
pos
neg

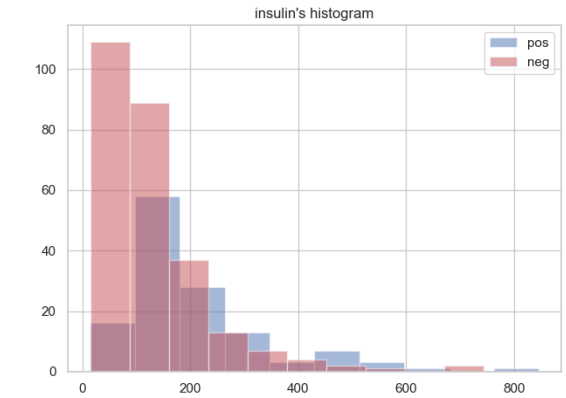

pressure's boxplot
pressure
diabetes
pos
neg

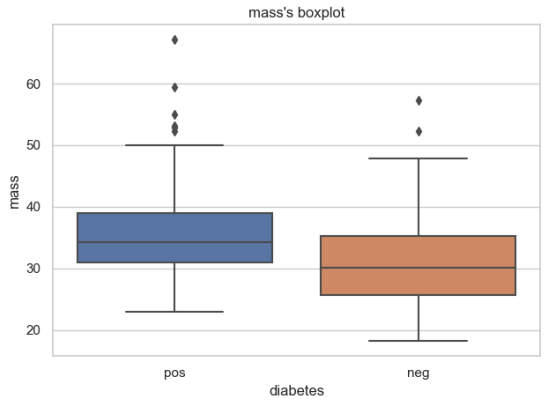

pressure's histogram
pos
neg

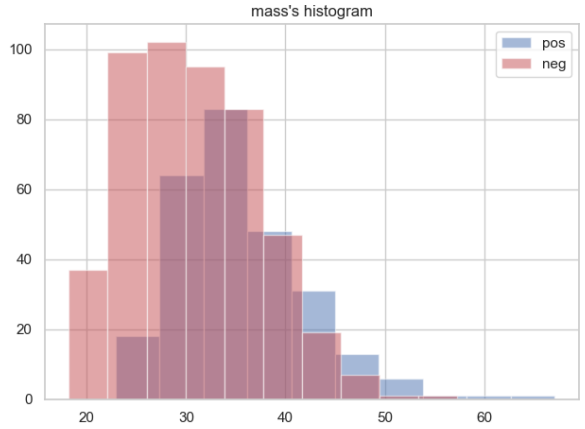

triceps's boxplot
triceps
diabetes
pos
neg

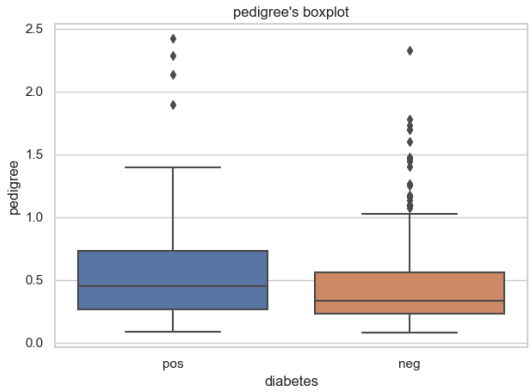

triceps's histogram
pos
neg

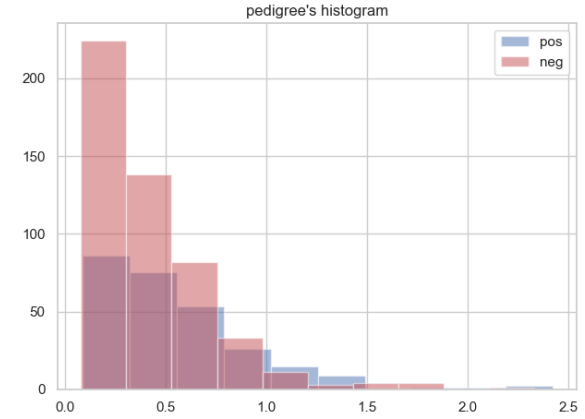

insulin's boxplot
insulin
diabetes
pos
neg

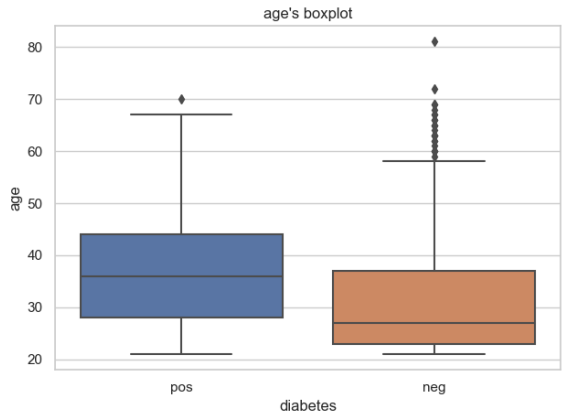

insulin's histogram
pos
neg

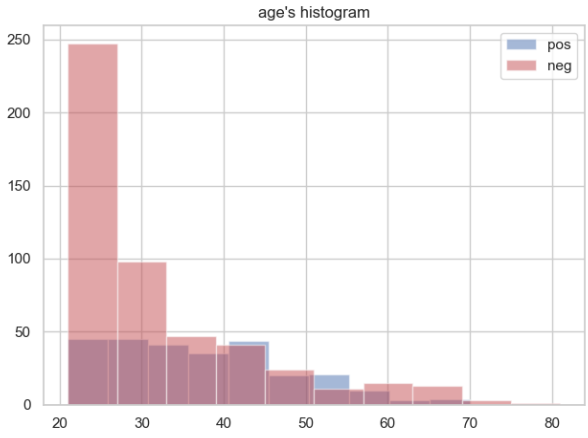












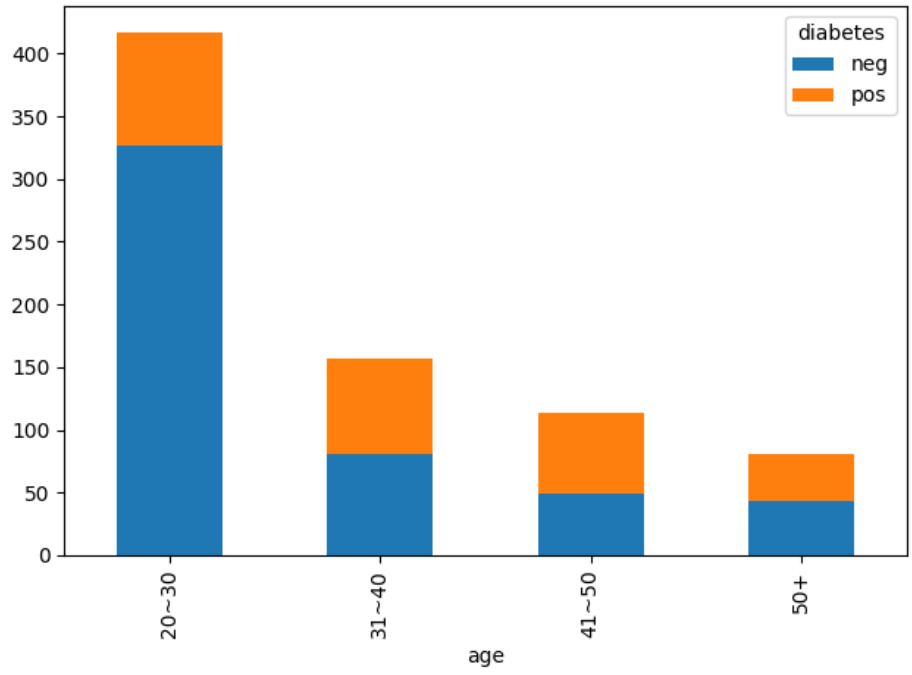
269 페이지 3.3 번 파이썬

```
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 classify 함수를 만들어준다. 그 다음 apply 메소드를 사용해서 각각 행의 나이를 확인해서 계급화 시켜주고 이를 이용하여 크로스탭을 만들어준다.

diabetes	neg	pos
age		
20~30	327	90
31~40	81	76
41~50	49	64
50+	43	38

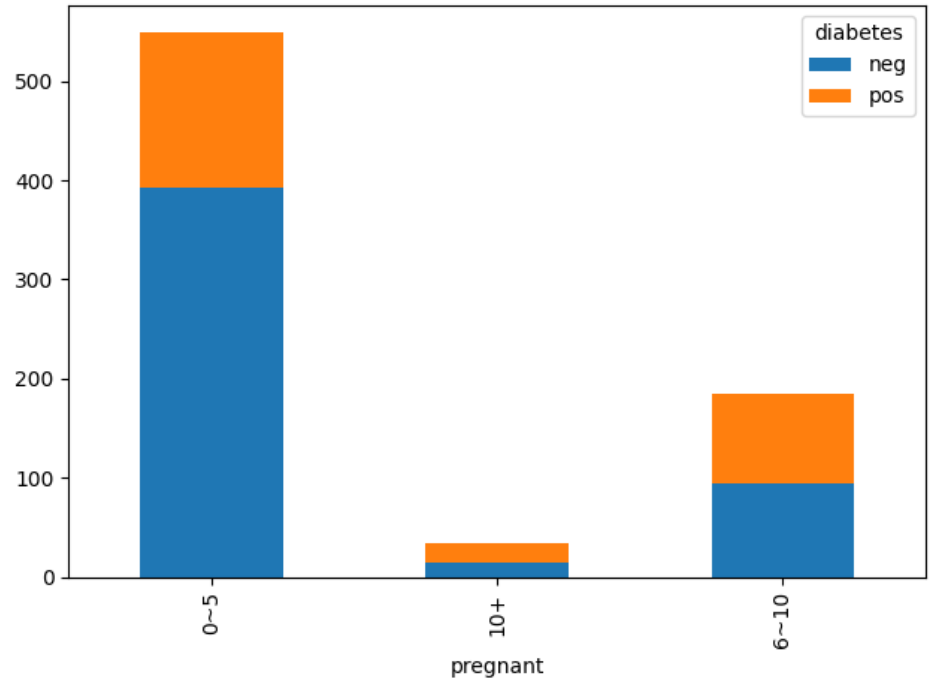


269 페이지 3.4 번 파이썬

```
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()
```

3.3 번과 동일한 방식으로 푼다. 등급화 시켜주는 함수만 바꿔준다.

diabetes	neg	pos
pregnant		
0~5	392	157
10+	14	20
6~10	94	91



269 페이지 3.5 번 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 describe 메소드를 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.

```
########## diabetes : glucose
          count        mean        std   min    25%    50%    75%    max
diabetes
neg       497.0  110.643863  24.776906  44.0   93.0  107.0  125.0  197.0
pos       266.0  142.319549  29.599199  78.0  119.0  140.0  167.0  199.0
########## diabetes : pressure
          count       mean        std   min   25%   50%   75%    max
diabetes
neg       481.0  70.877339  12.161223  24.0  62.0  70.0  78.0  122.0
pos       252.0  75.321429  12.299866  30.0  68.0  74.5  84.0  114.0
########## diabetes : triceps
          count       mean        std  min   25%   50%   75%   max
diabetes
neg       361.0  27.235457  10.026491  7.0  19.0  27.0  33.0  60.0
pos       180.0  33.000000  10.327595  7.0  27.0  32.0  39.0  99.0
########## diabetes : insulin
          count        mean         std   min    25%    50%     75%    max
diabetes
neg       264.0  130.287879  102.482237  15.0   66.0  102.5  161.25  744.0
pos       130.0  206.846154  132.699898  14.0  127.5  169.5  239.25  846.0
########## diabetes : mass
          count       mean       std   min   25%   50%     75%   max
diabetes
neg       491.0  30.859674  6.560737  18.2  25.6  30.1  35.300  57.3
pos       266.0  35.406767  6.614982  22.9  30.9  34.3  38.925  67.1
########## diabetes : pedigree
          count      mean       std    min      25%    50%      75%    max
diabetes
neg       500.0  0.429734  0.299085  0.078  0.22975  0.336  0.56175  2.329
pos       268.0  0.550500  0.372354  0.088  0.26250  0.449  0.72800  2.420
```

```
##########pregnant :glucose
          count        mean        std   min    25%    50%     75%    max
pregnant
0~5       545.0  118.968807  29.488173  44.0   97.0  114.0  137.00  199.0
10+        34.0  124.764706  25.601916  76.0  104.5  128.0  142.25  175.0
6~10      184.0  129.168478  33.157496  57.0  105.0  124.0  154.00  197.0
##########pregnant :pressure
          count       mean        std   min   25%   50%   75%    max
pregnant
0~5       525.0  70.897143  12.464641  24.0  62.0  70.0  78.0  122.0
10+        32.0  78.843750  11.676028  60.0  72.0  77.0  84.0  114.0
6~10      176.0  75.732955  11.242890  44.0  68.0  76.0  84.0  110.0
##########pregnant :triceps
          count       mean        std  min   25%   50%   75%   max
pregnant
0~5       408.0  28.365196  10.971326  8.0  20.0  28.0  36.0  99.0
10+        24.0  32.666667   9.608269  7.0  29.5  33.0  40.0  54.0
6~10      109.0  31.330275   8.104703  7.0  26.0  32.0  37.0  49.0
##########pregnant :insulin
          count        mean         std   min     25%    50%     75%    max
pregnant
0~5       308.0  148.623377  119.858634  14.0   73.75  115.0  182.00  846.0
10+        16.0  151.250000   77.024239  29.0  110.00  142.0  158.50  325.0
6~10       70.0  187.000000  117.992631  48.0  114.25  155.5  227.25  600.0
##########pregnant :mass
          count       mean       std   min     25%    50%     75%   max
pregnant
0~5       542.0  32.382103  7.199186  18.2  27.300  32.00  36.775  67.1
10+        34.0  35.691176  6.858146  22.2  30.375  36.35  40.425  52.3
6~10      181.0  32.075691  5.905794  19.6  27.600  32.40  35.500  50.0
##########pregnant :pedigree
          count      mean       std    min     25%    50%      75%    max
pregnant
0~5       549.0  0.472659  0.341328  0.078  0.2440  0.368  0.60500  2.420
10+        34.0  0.469824  0.308713  0.126  0.2465  0.395  0.58175  1.353
6~10      185.0  0.469930  0.305826  0.084  0.2380  0.382  0.67200  1.476
```

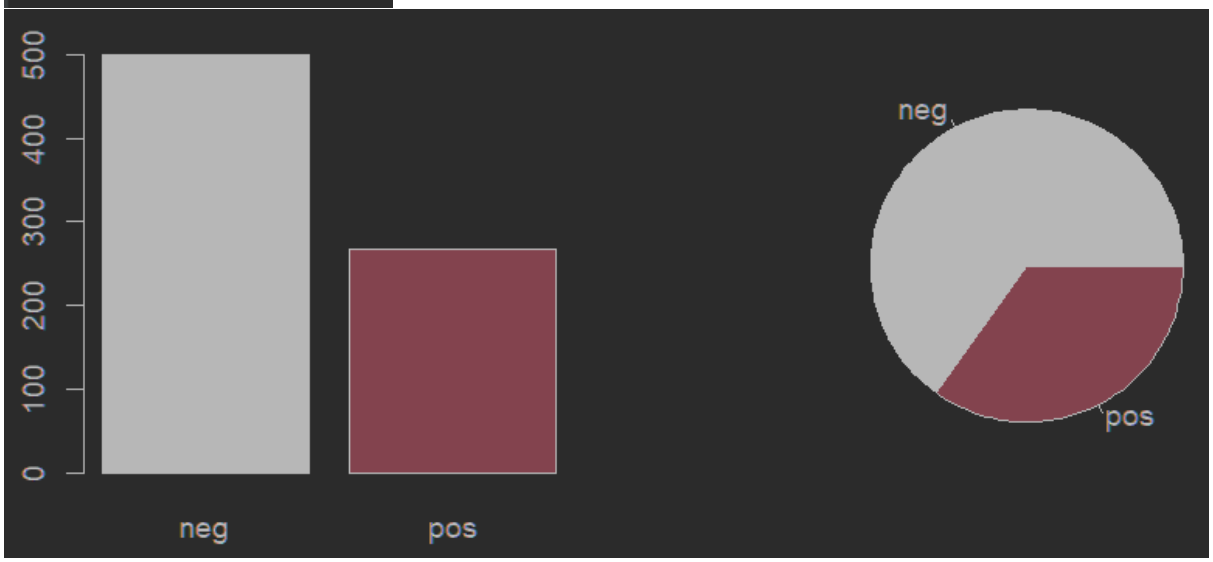
269 페이지 3.1 번 R

```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)
```

```r
result_table <- table(pima$diabetes)
par(mfrow = c(1, 2))
barplot(result_table, col = 1:2)
pie(result_table, col = 1:2)
```

aggregate 를 사용해서 diabetes 를 기준으로 값을 그룹화한다. Length 를 쓰는 이유는 개수만 새면 되기 때문이다. 그 다음 비율을 구해주기 위해서 전체의 값에서 각 그룹의 값으로 나눠준다. 그 후 barplot 과 pie 를 사용해서 차트를 출력한다.

```
  diabetes   X     rate
1      neg 500 65.10417
2      pos 268 34.89583
```



269 페이지 3.2 번 R

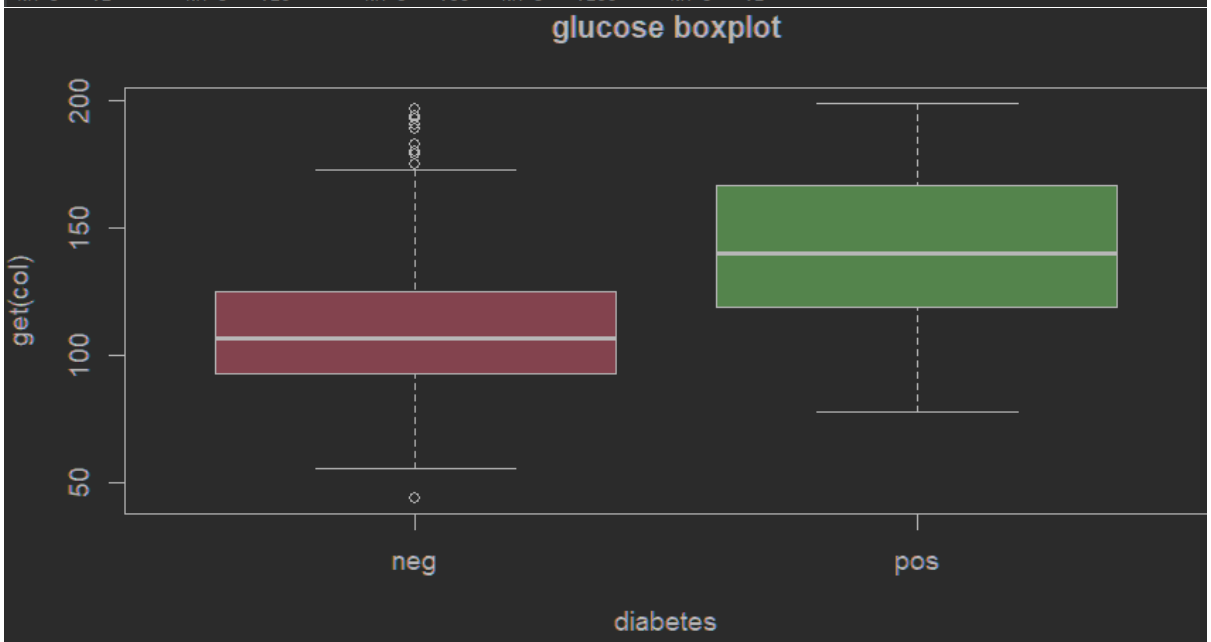
```r
pima <- read.csv('./data/pima2.csv', header = T)
par(mfrow = c(2, 1))
cols <- c('glucose', 'pressure', 'triceps', 'insulin', 'mass', 'pedigree',
'age')
by(pima[cols],pima$diabetes,summary)
for (col in cols) {
    boxplot(get(col) ~ diabetes, data = pima, main = paste0(col, '
boxplot'), col = 1:2)
    hist_pos <- hist(pima[[col]][pima$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'))
}
```

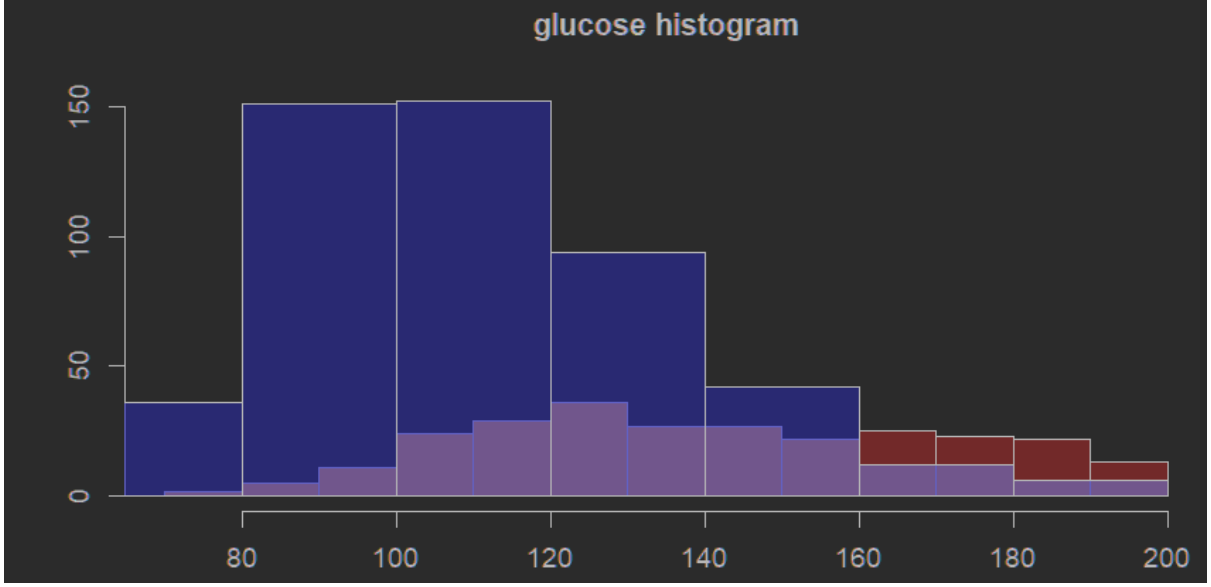
by 를 사용해서 기술통계를 구한다. 그 이후 각 열을 순회하면서 히스토그램과 파이그래프에서 값에 쓸 값을 구한다.

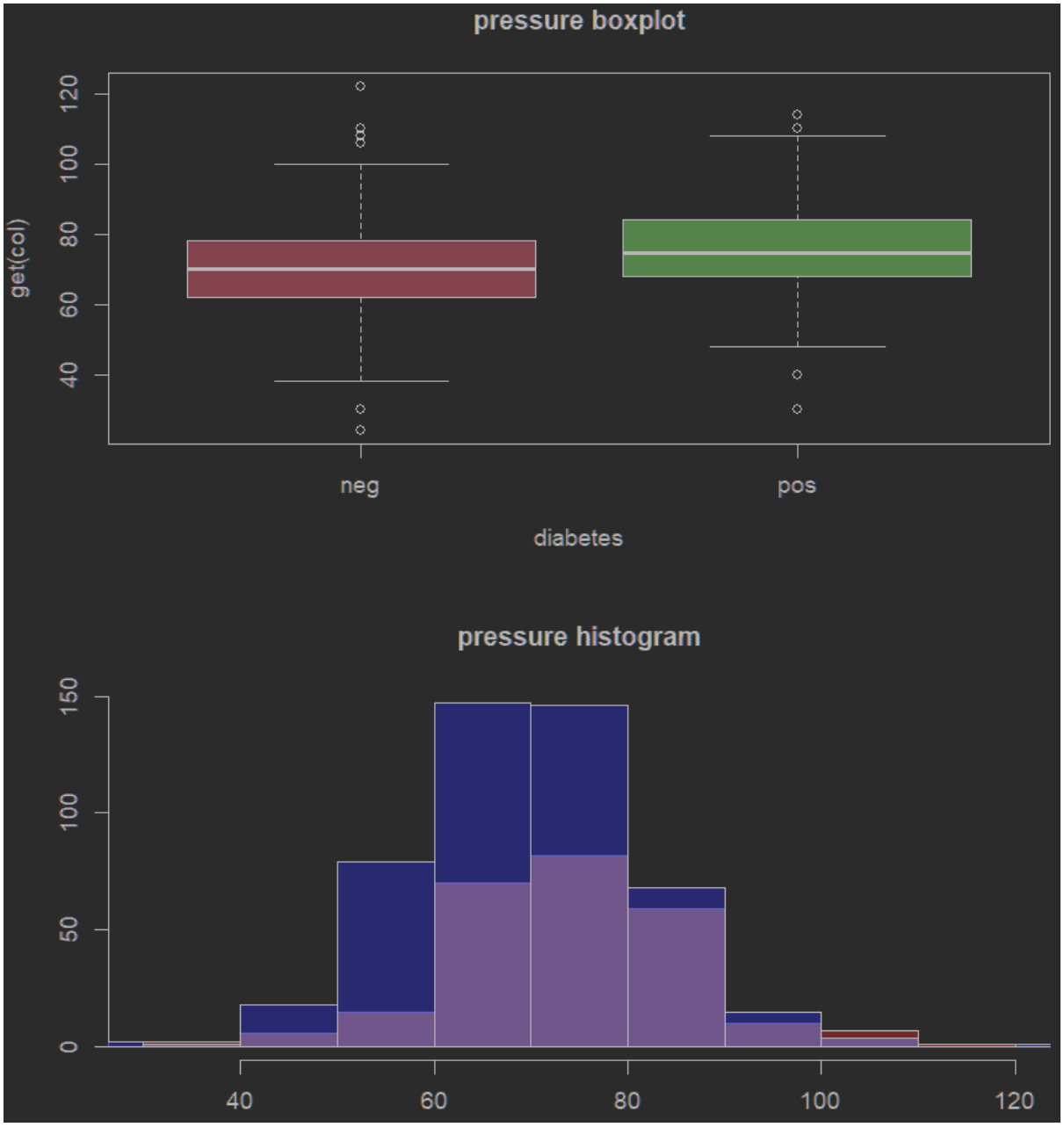
```
pima[[col]][pima$diabetes == 'pos']
```

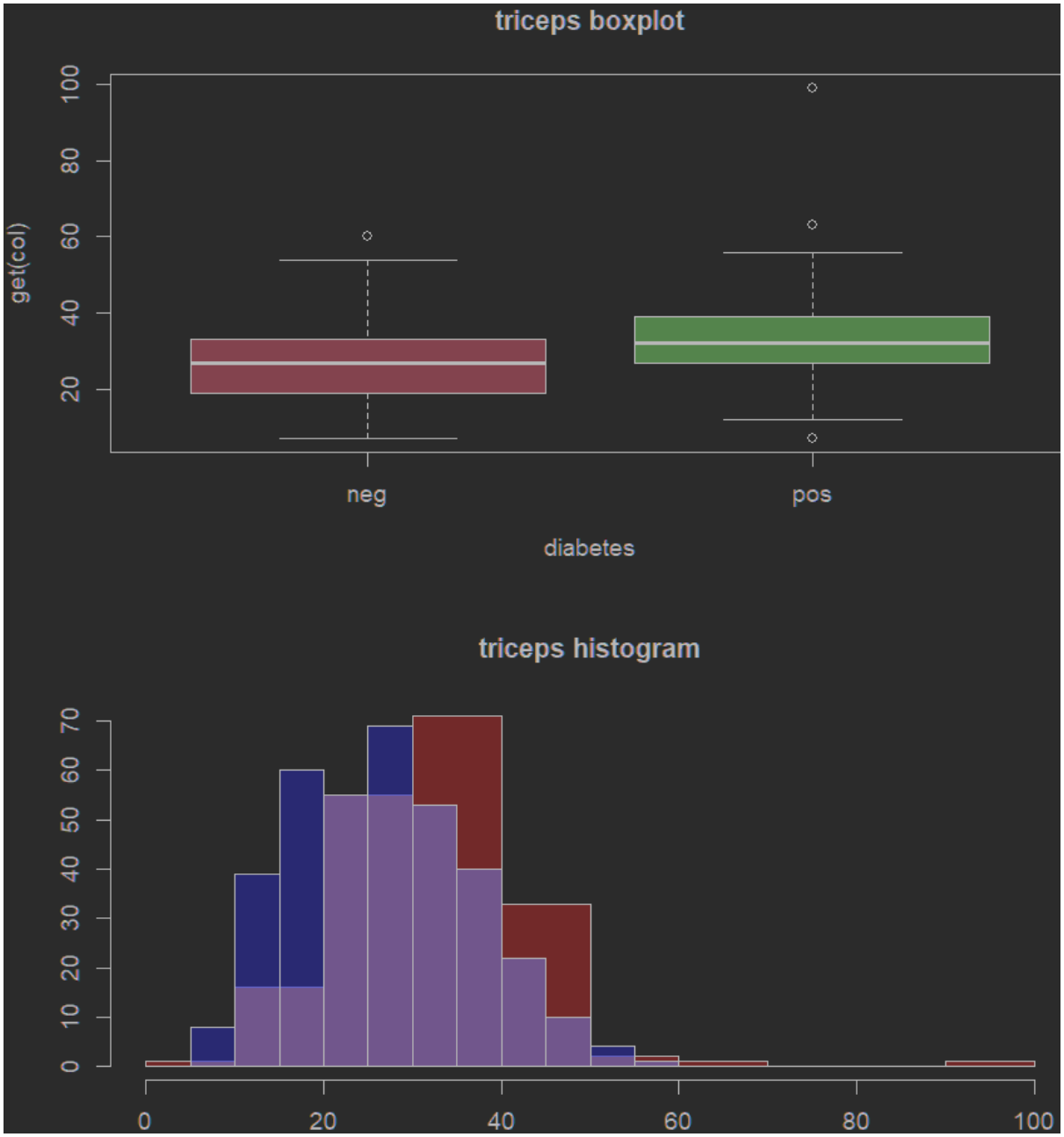
그 이후 히스토그램과 그래프를 출력한다.

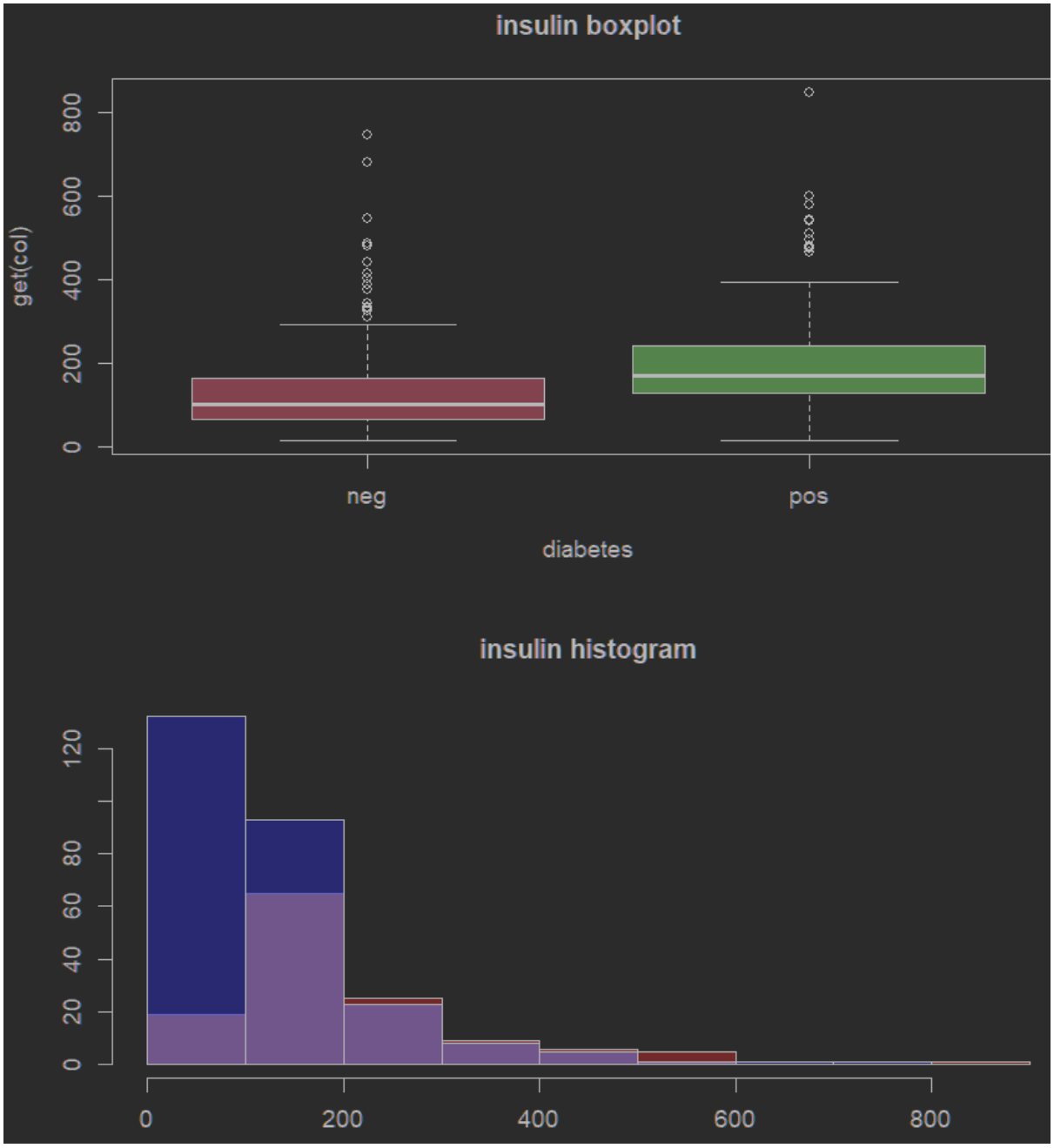
```
pima$diabetes: neg
    glucose         pressure         triceps         insulin           mass          pedigree           age
 Min.   : 44.0   Min.   : 24.00   Min.   : 7.00   Min.   : 15.0   Min.   :18.20   Min.   :0.0780   Min.   :21.00
 1st Qu.: 93.0   1st Qu.: 62.00   1st Qu.:19.00   1st Qu.: 66.0   1st Qu.:25.60   1st Qu.:0.2298   1st Qu.:23.00
 Median :107.0   Median : 70.00   Median :27.00   Median :102.5   Median :30.10   Median :0.3360   Median :27.00
 Mean   :110.6   Mean   : 70.88   Mean   :27.24   Mean   :130.3   Mean   :30.86   Mean   :0.4297   Mean   :31.19
 3rd Qu.:125.0   3rd Qu.: 78.00   3rd Qu.:33.00   3rd Qu.:161.2   3rd Qu.:35.30   3rd Qu.:0.5617   3rd Qu.:37.00
 Max.   :197.0   Max.   :122.00   Max.   :60.00   Max.   :744.0   Max.   :57.30   Max.   :2.3290   Max.   :81.00
 NA's   :3       NA's   :19       NA's   :139     NA's   :236     NA's   :9
------------------------------------------------------------------------------------------------------------------------
pima$diabetes: pos
    glucose         pressure         triceps       insulin           mass          pedigree           age
 Min.   : 78.0   Min.   : 30.00   Min.   : 7   Min.   : 14.0   Min.   :22.90   Min.   :0.0880   Min.   :21.00
 1st Qu.:119.0   1st Qu.: 68.00   1st Qu.:27   1st Qu.:127.5   1st Qu.:30.90   1st Qu.:0.2625   1st Qu.:28.00
 Median :140.0   Median : 74.50   Median :32   Median :169.5   Median :34.30   Median :0.4490   Median :36.00
 Mean   :142.3   Mean   : 75.32   Mean   :33   Mean   :206.8   Mean   :35.41   Mean   :0.5505   Mean   :37.07
 3rd Qu.:167.0   3rd Qu.: 84.00   3rd Qu.:39   3rd Qu.:239.2   3rd Qu.:38.92   3rd Qu.:0.7280   3rd Qu.:44.00
 Max.   :199.0   Max.   :114.00   Max.   :99   Max.   :846.0   Max.   :67.10   Max.   :2.4200   Max.   :70.00
 NA's   :2       NA's   :16       NA's   :88   NA's   :138     NA's   :2
```

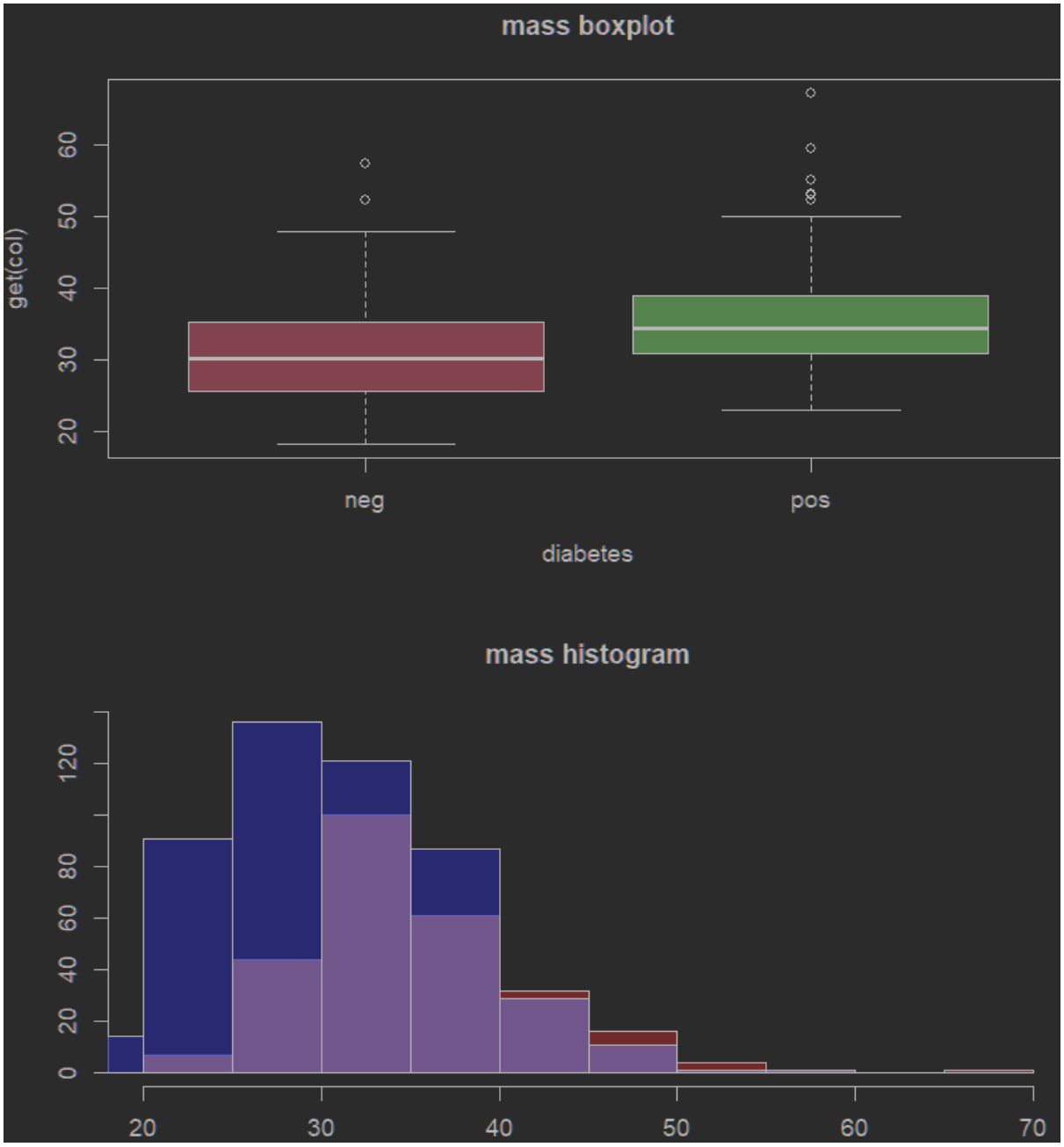


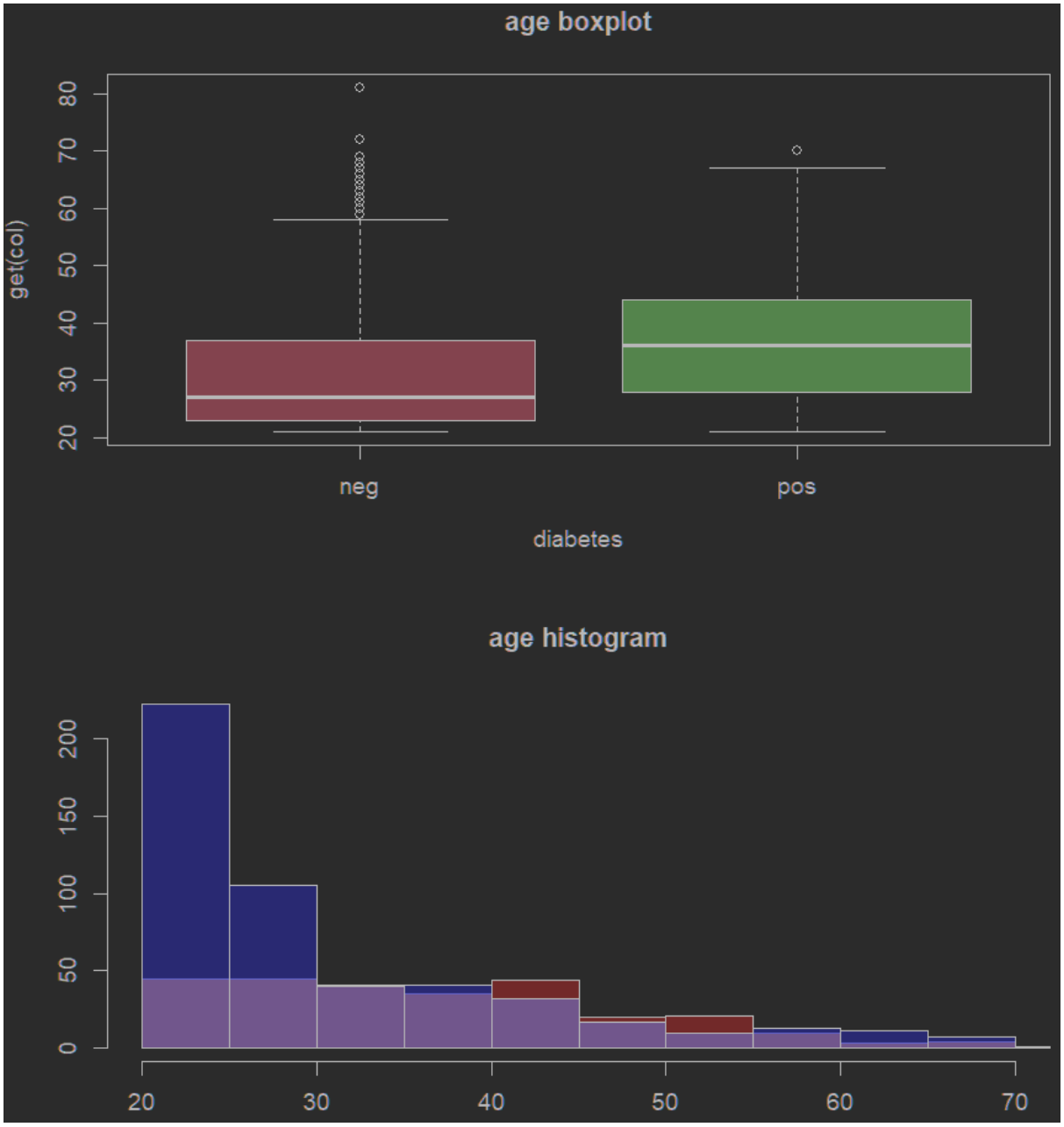


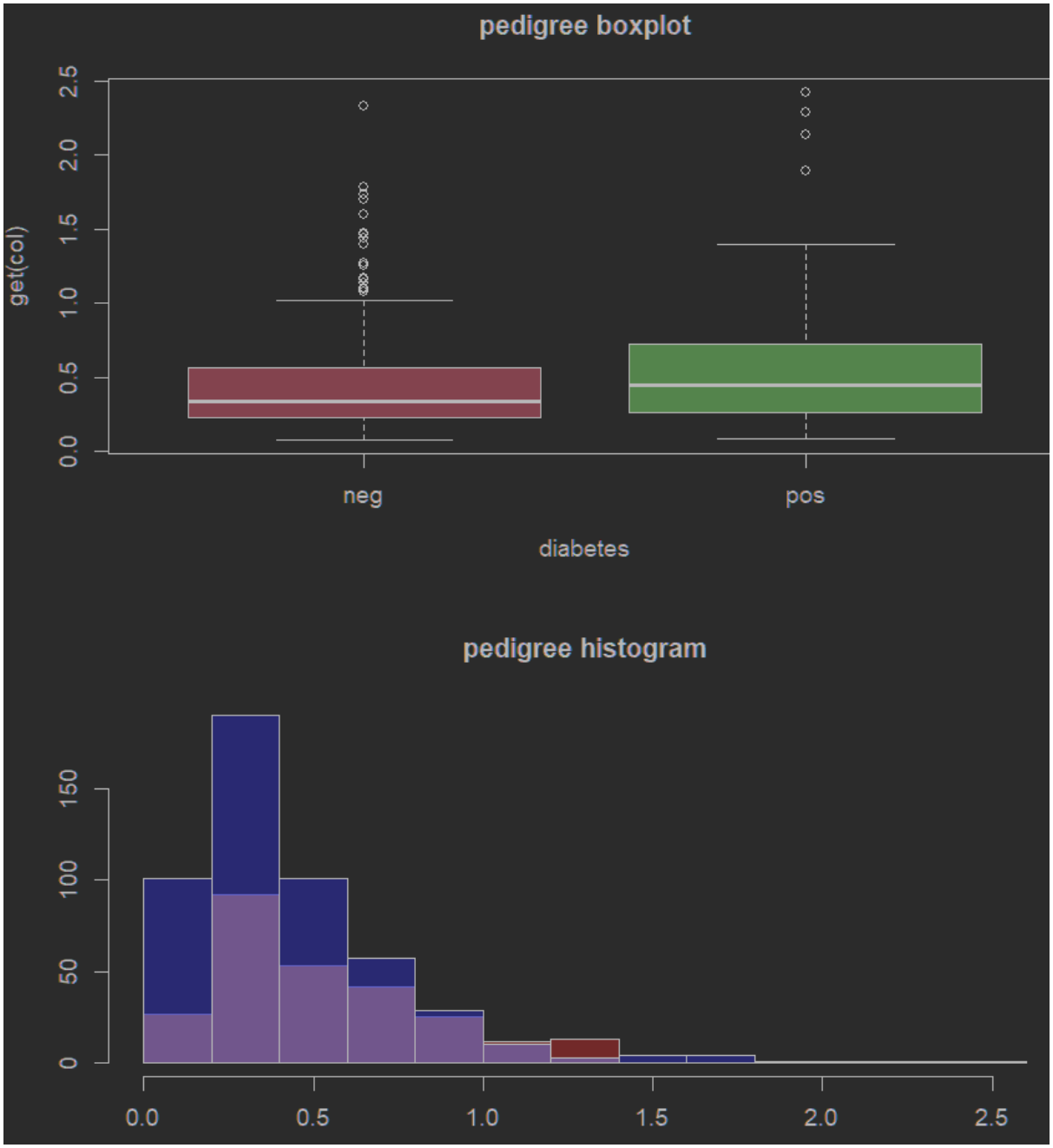

pressure boxplot
get(col)
40
60
80
100
120
neg
pos
diabetes
pressure histogram
0
50
100
150
40
60
80
100
120


triceps boxplot
get(col)
20
40
60
80
100
neg
pos
diabetes
triceps histogram
0
10
20
30
40
50
60
70
0
20
40
60
80
100


insulin boxplot
get(col)
0
200
400
600
800
neg
pos
diabetes
insulin histogram
0
20
40
60
80
120
0
200
400
600
800


mass boxplot
get(col)
20
30
40
50
60
neg
pos
diabetes
mass histogram
0
20
40
60
80
120
20
30
40
50
60
70


age boxplot
get(col)
20
30
40
50
60
70
80
neg
pos
diabetes
age histogram
0
50
100
150
200
20
30
40
50
60
70



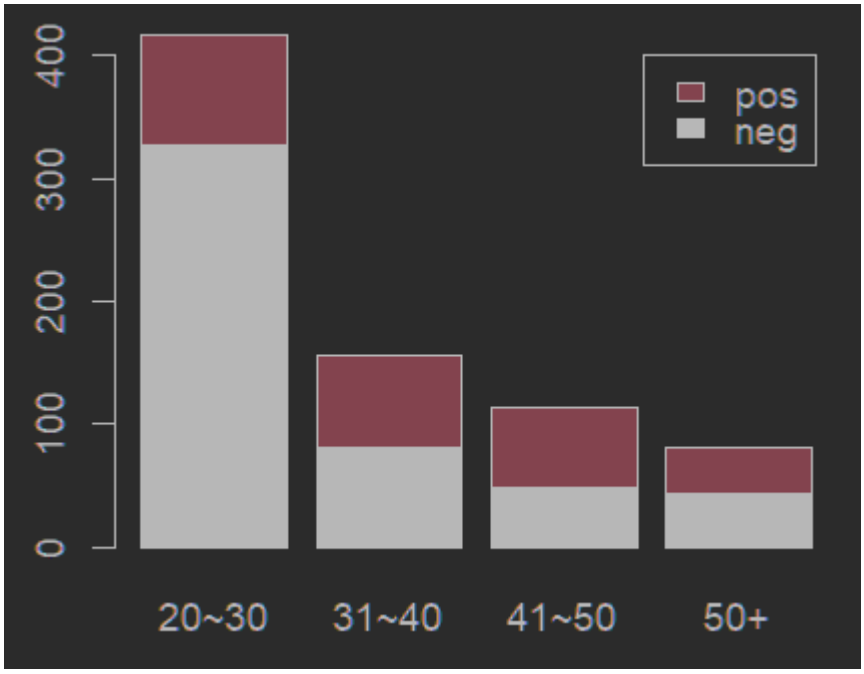
269 페이지 3.3 번 R

```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))
```

lapply 로 각열을 순회하면서 그룹화에 기준이 될변수를 정해서 age 에 덮어쓴다. 그 이후 table 을 제작하고 출력한 후 그래프를 출력한다.

```
      neg pos
20~30 327  90
31~40  81  76
41~50  49  64
50+    43  38
```



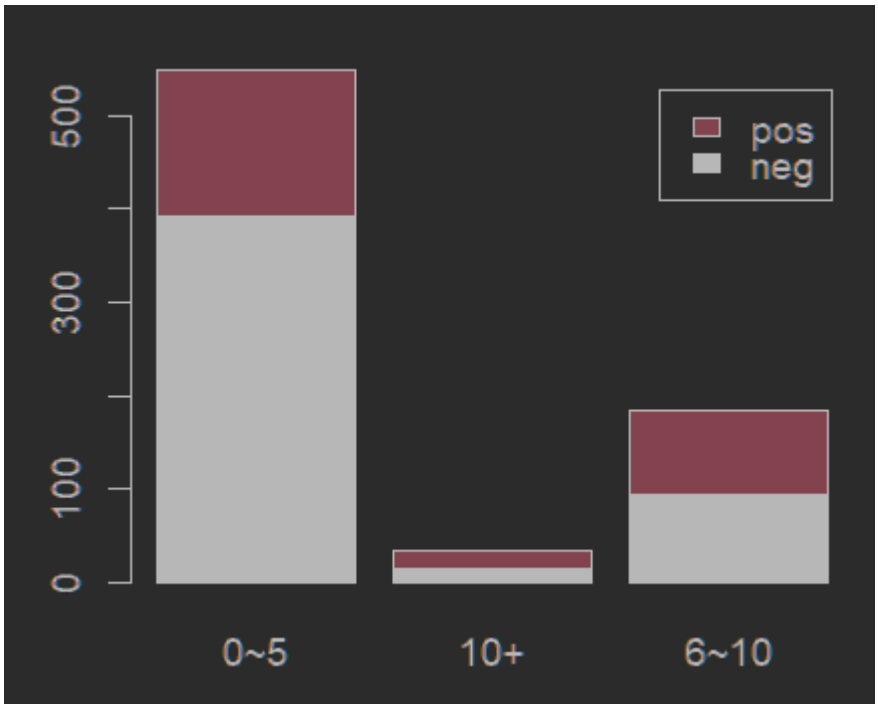
269 페이지 3.4 번 R

```
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+')
    }
}
```

```r
pima$pregnant <- unlist(lapply(pima$pregnant, classify))
tbl <- table(pima$pregnant, pima$diabetes)
tbl
barplot(t(tbl), col = 1:2, legend = colnames(tbl))
```

lapply 로 각열을 순회하면서 그룹화에 기준이 될변수를 정해서 pregnant 에 덮어쓴다. 그 이후 table 을 제작하고 출력한 후 그래프를 출력한다.

```
     neg pos
0~5  392 157
10+   14  20
6~10  94  91
```



269 페이지 3.5 번 R

```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))
}
```

aggregate 를 이용해서 집계함수를 출력한다. 이때는 mean 과 sd 를 따로넣어서 출력해주는데 전체적으로 구하고 싶다면 그냥 간편하게 summary 를 넣어주면된다. Pregnant 는 등급이 필요하기에 등급을 구해준후에 반복문을 돌려서 출력한다.

```
[1] "**********:diabetes:glucose"
[1] "##########:MEAN"
  diabetes  glucose
1      neg 110.6439
2      pos 142.3195
[1] "##########:SD"
  diabetes  glucose
1      neg 24.77691
2      pos 29.59920
[1] "**********:diabetes:pressure"
[1] "##########:MEAN"
  diabetes pressure
1      neg 70.87734
2      pos 75.32143
[1] "##########:SD"
  diabetes pressure
1      neg 12.16122
2      pos 12.29987
[1] "**********:diabetes:triceps"
[1] "##########:MEAN"
  diabetes  triceps
1      neg 27.23546
2      pos 33.00000
[1] "##########:SD"
  diabetes  triceps
1      neg 10.02649
2      pos 10.32759
[1] "**********:diabetes:insulin"
[1] "##########:MEAN"
  diabetes  insulin
1      neg 130.2879
2      pos 206.8462
[1] "##########:SD"
  diabetes  insulin
1      neg 102.4822
2      pos 132.6999
[1] "**********:diabetes:mass"
[1] "##########:MEAN"
  diabetes     mass
1      neg 30.85967
2      pos 35.40677
[1] "##########:SD"
  diabetes     mass
1      neg 6.560737
2      pos 6.614982
[1] "**********:diabetes:pedigree"
[1] "##########:MEAN"
  diabetes pedigree
1      neg 0.429734
2      pos 0.550500
[1] "##########:SD"
  diabetes  pedigree
1      neg 0.2990853
2      pos 0.3723545
```

```
[1] "**********:pregnant:glucose"
[1] "##########:MEAN"
  diabetes  glucose
1      0~5 118.9688
2      10+ 124.7647
3     6~10 129.1685
[1] "##########:SD"
  diabetes  glucose
1      0~5 29.48817
2      10+ 25.60192
3     6~10 33.15750
[1] "**********:pregnant:pressure"
[1] "##########:MEAN"
  diabetes pressure
1      0~5 70.89714
2      10+ 78.84375
3     6~10 75.73295
[1] "##########:SD"
  diabetes pressure
1      0~5 12.46464
2      10+ 11.67603
3     6~10 11.24289
[1] "**********:pregnant:triceps"
[1] "##########:MEAN"
  diabetes  triceps
1      0~5 28.36520
2      10+ 32.66667
3     6~10 31.33028
[1] "##########:SD"
  diabetes   triceps
1      0~5 10.971326
2      10+  9.608269
3     6~10  8.104703
[1] "**********:pregnant:insulin"
[1] "##########:MEAN"
  diabetes  insulin
1      0~5 148.6234
2      10+ 151.2500
3     6~10 187.0000
[1] "##########:SD"
  diabetes   insulin
1      0~5 119.85863
2      10+  77.02424
3     6~10 117.99263
[1] "**********:pregnant:mass"
[1] "##########:MEAN"
  diabetes     mass
1      0~5 32.38210
2      10+ 35.69118
3     6~10 32.07569
[1] "##########:SD"
  diabetes     mass
1      0~5 7.199186
2      10+ 6.858146
3     6~10 5.905794
```

```
[1] "**********:pregnant:pedigree"
[1] "##########:MEAN"
  diabetes  pedigree
1      0~5 0.4726594
2      10+ 0.4698235
3     6~10 0.4699297
[1] "##########:SD"
  diabetes  pedigree
1      0~5 0.3413279
2      10+ 0.3087132
3     6~10 0.3058258
```