

3장 기술통계(Descriptive Statistics)

U S I N G



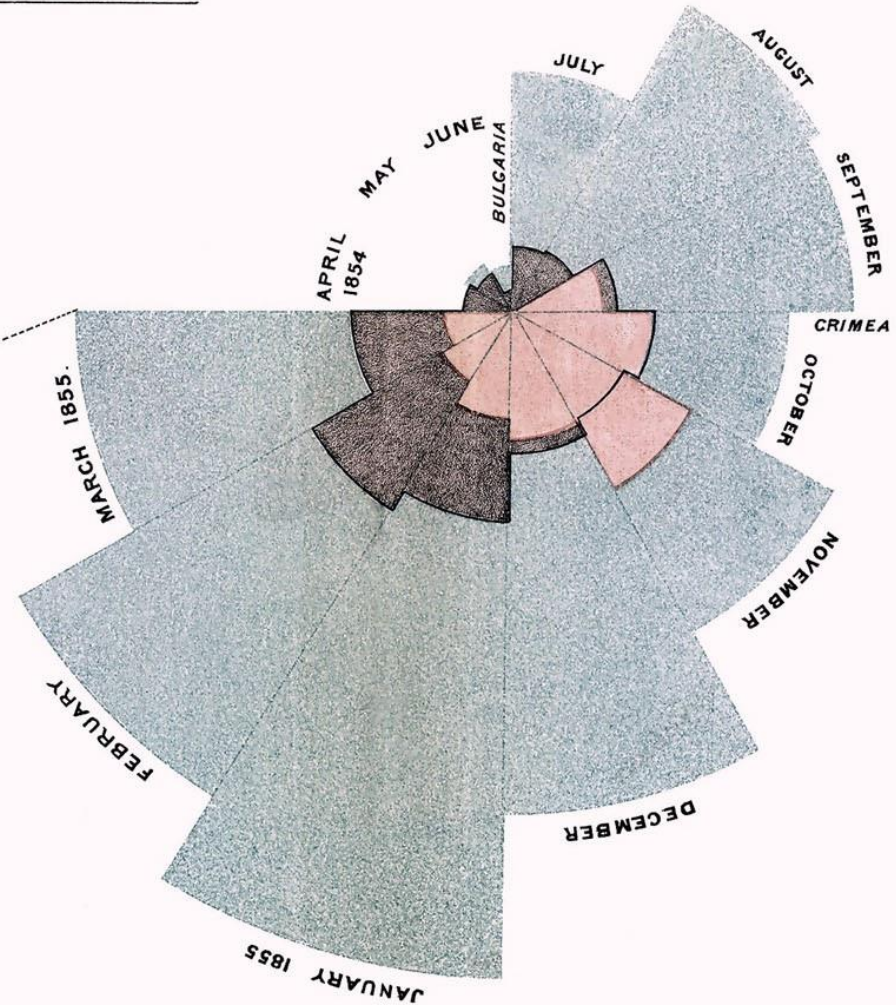
STATISTICAL ANALYSIS FOR
SOCIAL SCIENCE USING R

1. 빈도표 출력
2. 기술통계량 출력
3. 여러 형태의 도표 그리기

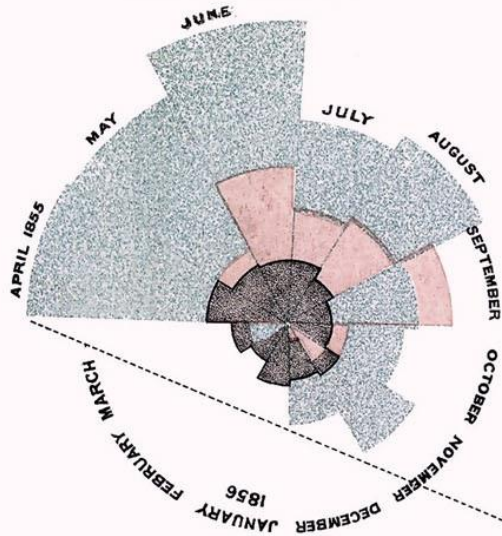
통계로 세상을 구한 나이팅게일

DIAGRAM OF THE CAUSES OF MORTALITY IN THE ARMY IN THE EAST.

1.
APRIL 1854 TO MARCH 1855.



2.
APRIL 1855 TO MARCH 1856.



The Areas of the blue, red, & black wedges are each measured from the centre as the common vertex.

The blue wedges measured from the centre of the circle represent area for area the deaths from Preventable or Mitigable Zymotic diseases; the red wedges measured from the centre the deaths from wounds; & the black wedges measured from the centre the deaths from all other causes.

The black line across the red triangle in Nov. 1854 marks the boundary of the deaths from all other causes during the month.

In October 1854, & April 1855, the black area coincides with the red; in January & February 1856, the blue coincides with the black.

The entire areas may be compared by following the blue, the red & the black lines enclosing them.

- 데이터 정리
- 데이터 요약
- 통계량 계산
- 시각화



- 측정된 표본 데이터를 분석하여 모집단의 특성을 추측
- 모수추정
- 가설검정
- 분산분석
- 회귀분석





1. 빈도표 출력

1. table 함수를 이용한 빈도표 출력

<분석 순서>

- 1) R에서 기본적으로 제공하는 table 함수를 이용하여 빈도표를 출력한다.
- 2) 청소년들의 삶의 만족도(q50w1)에 대한 빈도분포를 구해본다.

```
# table 함수를 이용한 방법  
table(spssdata$q50w1)
```

- ✓ table 함수는 기본적인 **빈도표**나 교차표를 출력할 수 있는 함수
- ✓ table 함수의 괄호 안에 빈도표를 출력하고자 하는 변수명을 입력

```
> table(spssdata$q50w1)  
 1    2    3    4    5  
5   70 165 298  57
```

- ✓ table 함수를 이용한 결과에는 **변수값과 변수값에 따른 빈도가** 출력
 - 빈도분포를 청소년들이 자신들의 삶에 대해서 만족한다는 응답이 더 많은 것을 알 수 있음



1. 빈도표 출력

2. 'plyr'패키지를 이용한 빈도표 출력

<분석 순서>

- 1) 'plyr' 패키지를 설치하고, 불러온다.
- 2) count 함수를 이용하여 빈도표를 출력한다.

```
# 'plyr' 패키지를 이용한 방법  
library(plyr)  
count(spssdata, 'q50w1')
```

- ✓ 'plyr' 패키지를 설치하고 불러옴
- ✓ count 함수에는 빈도표를 출력하고자 하는 변수가 있는 데이터명을 먼저 입력하고, 다음에 해당 변수를 입력

```
> count(spssdata, 'q50w1')  
  q50w1 freq  
1      1    5  
2      2   70  
3      3  165  
4      4  298  
5      5   57
```



1. 빈도표 출력

3. cbind를 이용한 빈도표 출력

<분석 순서>

- 1) cbind 함수를 이용하여 빈도, 누적 빈도, 비율, 누적 비율을 출력한다.
- 2) round 함수를 이용하여 백분율과 소수점을 변경한다.

① cbind를 이용한 방법

```
cbind(Freq=table(spssdata$q50w1),           # 빈도  
      Cumul=cumsum(table(spssdata$q50w1)),  # 누적 빈도  
      relative=prop.table(table(spssdata$q50w1)), # 빈도에 따른 비율  
      Cum.prop=cumsum(prop.table(table(spssdata$q50w1))))# 누적 빈도
```

- ✓ cbind 함수는 개별적으로 빈도, 누적 빈도, 빈도에 따른 비율, 그리고 누적 비율을 각각 계산해서 횡으로 연결해서 출력
- ✓ 입력된 인자의 순서대로 결과를 출력해주며, 각 인자는 "로 구분



1. 빈도표 출력

✓ 빈도 구하기

- cbind 함수에서 첫 번째 인자는 빈도를 구함
- table 함수를 이용해서 삶의 만족도(q50w1) 변수의 빈도분포를 구한 후에 '=' 기호를 사용해서 출력될 결과의 이름('Freq')을 지정
- table 인자에는 빈도를 출력할 대상 변수명(spssdata\$q50w1)을 입력

✓ 누적 빈도 구하기

- 누적 빈도를 출력하기 위해 누적 빈도를 구해주는 **cumsum** 함수를 사용
- 누적빈도 분석결과를 출력할 이름을 'Cumul'라고 지정하는데, 이를 위해 '=' 기호를 사용
- cumsum 함수를 출력하기 위해서 괄호 안에 table 함수를 사용하며, 이 때 table 함수의 내용은 빈도를 구할 때와 동일함
- table 함수를 통해 변수의 빈도분포를 우선 구하고, cumsum 함수로 table 함수로 계산한 각 변수값의 빈도를 누적 빈도로 계산하여 출력하게 하는 것임



1. 빈도표 출력

- ✓ 빈도에 따른 비율 구하기
 - 원리는 누적빈도를 구한 방법과 동일
 - table 함수를 사용해서 구한 빈도분포를 **prop.table** 함수를 이용하여 비율로 바꾼 후에 결과를 출력할 이름을 'relative'라고 지정
 - prop.table 함수는 table 함수로 계산한 변수의 빈도분포를 비율로 전환시켜줌
- ✓ 누적 비율 구하기
 - 명령어가 복잡한 것처럼 보이지만 동일한 원리가 적용됨
 - 누적 비율을 출력하기 위해서는 비율을 출력했던 결과를 **cumsum** 함수를 사용해서 변환시키고, 그 결과를 출력하기 위해 Cum.prop이라는 이름을 붙임

1. 빈도표 출력

```
> cbind(Freq=table(spssdata$q50w1),
+ Cumul=cumsum(table(spssdata$q50w1)),
+ relative=prop.table(table(spssdata$q50w1))),
+ Cum.prop=cumsum(prop.table(table(spssdata$q50w1))))
```

	Freq	Cumul	relative	Cum.prop
1	5	5	0.008403361	0.008403361
2	70	75	0.117647059	0.126050420
3	165	240	0.277310924	0.403361345
4	298	538	0.500840336	0.904201681
5	57	595	0.095798319	1.000000000

- ✓ 삶의 만족도(q50w1) 변수에 대해서 cbind에서 이름을 붙인 대로 Freq(빈도), Cumul(누적 빈도), relative(비율), 그리고 Cum.prop(누적 비율)이 출력
- ✓ 이름은 원하는 대로 지정해서 출력할 수 있으며, 한글도 사용할 수 있음

1. 빈도표 출력

② 백분율 구하기와 round 함수를 이용한 소수점 변경

```
cbind(Freq=table(spssdata$q50w1),
      percentage=100*prop.table(table(spssdata$q50w1)),
      relative=round(100*prop.table(table(spssdata$q50w1)), 3))
```

- ✓ cbind 함수를 이용하여 빈도분포와 백분율을 구하고, 소수점을 변경
 - cbind 함수의 첫 번째 인자로 table 함수를 사용해서 빈도를 구함
 - 비율 대신 백분율로 결과를 출력하기 위해서 앞에서 비율을 구하기 위해 사용했던 prop.table 인자에 100을 곱하고, 그 결과를 출력하기 위해서 percentage라는 이름을 붙임
 - **round** 함수를 이용하여 소수점 자리수를 변경

```
> cbind(Freq=table(spssdata$q50w1),
+       percentage=100*prop.table(table(spssdata$q50w1))),
+       relative=round(100*prop.table(table(spssdata$q50w1)), 3))
```

	Freq	percentage	relative
1	5	0.8403361	0.840
2	70	11.7647059	11.765
3	165	27.7310924	27.731
4	298	50.0840336	50.084
5	57	9.5798319	9.580



1. 빈도표 출력

4. 'Hmisc'패키지 describe 함수를 이용한 빈도표

```
# 'Hmisc' 패키지를 이용한 방법  
library(Hmisc)  
describe(spssdata$q50w1)
```

- ✓ 'Hmisc' 패키지의 describe 함수를 이용하여 기술통계량을 출력
- ✓ 'Hmisc' 패키지를 불러온다.
- ✓ describe 함수에 기술통계량을 출력할 대상 변수명을 입력하여 결과를 출력할 수 있음

```
> library(Hmisc)  
> describe(spssdata$q50w1)  
spssdata$q50w1 : 50. 학생은 학생의 삶에 전반적으로 얼마나 만족하고 있습니까?  


|  | n   | missing | distinct | Info  | Mean  | Gmd    |
|--|-----|---------|----------|-------|-------|--------|
|  | 595 | 0       | 5        | 0.851 | 3.558 | 0.8931 |


| value      | 1     | 2     | 3     | 4     | 5     |
|------------|-------|-------|-------|-------|-------|
| Frequency  | 5     | 70    | 165   | 298   | 57    |
| Proportion | 0.008 | 0.118 | 0.277 | 0.501 | 0.096 |

  
> detach("package:Hmisc") #sjPlot uses describe() from Psych  
package, not from Hmisc package!
```



1. 빈도표 출력

- ✓ 'Hmisc' 패키지의 describe 함수에서는 사례수, 결측값 수, 속성 수 (distinct), Info (10이상이면 연속변수), 평균, GMD(Gini's mean difference), 그리고 빈도분포가 출력
- ✓ 빈도분포를 위주로 보여주며, 이에 필요한 기본적인 기술통계량 정보를 출력
- ✓ 변수 설명까지 출력되어 좀 더 쉽게 분석결과를 이해할 수 있음

1. 빈도표 출력

5. 'sjmisc' 패키지를 이용한 빈도표 출력

<분석 순서>

- 1) 'sjmisc' 패키지를 설치하고 불러온다.
- 2) 'sjmisc' 패키지의 `frq` 함수를 이용하여 빈도표를 출력한다.

```
> library(sjmisc)
> frq(spssdata$q50w1)
```

```
# x <integer>
```

```
# total N=595   valid N=595   mean=3.56   sd=0.85
```

val	frq	raw.prc	valid.prc	cum.prc
1	5	0.84	0.84	0.84
2	70	11.76	11.76	12.61
3	165	27.73	27.73	40.34
4	298	50.08	50.08	90.42
5	57	9.58	9.58	100.00
<NA>	0	0.00	NA	NA

```
> frq(spssdata$q50w1, out = "v", title = "Satisfaction level")
> frq(spssdata$q50w1, out = "v", title = "만족도 수준")
> frq(spssdata$q50w1, out = "b", title = "Satisfaction level")
> frq(spssdata$q50w1, out = "b", title = "만족도 수준")
```



2. 기술통계량 출력

1. summary 함수를 이용한 기술통계량 출력

```
# summary 함수를 이용한 방법  
summary(spssdata$q50w1)
```

- ✓ summary 함수의 괄호 안에 해당 변수명을 입력하면 그 변수의 가장 기본적인 기술통계량이 출력됨
- ✓ 괄호 안에 데이터프레임명을 입력하면 데이터프레임에 들어 있는 모든 변수의 기술통계량을 출력

```
> summary(spssdata$q50w1)  
  Min. 1st Qu. Median  Mean 3rd Qu.  Max.   
1.000   3.000   4.000 3.558   4.000  5.000
```



2. 기술통계량 출력

2. 'psych' 패키지를 이용한 기술통계량 출력

<분석 순서>

- 1) 'psych' 패키지를 설치하고 불러온다.
- 2) 'psych' 패키지의 `describe` 함수를 이용하여 기술통계량을 출력한다.

```
# 'psych' 패키지를 이용한 방법  
library(psych)  
describe(spssdata$q50w1)
```

- ✓ 'psych' 패키지의 `describe` 함수를 이용하여 기술통계량을 출력
- ✓ 'psych' 패키지를 설치하고 불러옴
- ✓ `describe` 함수의 괄호 안에 기술통계량을 출력할 대상 변수명을 입력

2. 기술통계량 출력

```
> library(psych)
> describe(spssdata$q50w1)
vars    n mean    sd median trimmed mad min max range  skew
1  1  595 3.56 0.85      4    3.59   0  1  5      4 -0.49
kurtosis  se
-0.12 0.03
```

- ✓ describe 함수를 이용하여 출력되는 기술통계량은 변수의 수(vars), 유효 사례수(n), 평균(mean), 표준편차(sd), 중위수(median), 절삭평균(trimmed mean, 디폴트는 0.1), 중위수 절대 편차(mad: median absolute deviation, 각 사례값이 중위수로부터 떨어진 거리인 절대편차의 평균), 최소값(min), 최대값(max), 범위(range), 왜도(skew), 첨도(kurtosis), 표준오차(se: standard error)가 제시됨

2. 기술통계량 출력

3. 'sjmisc' 패키지를 이용한 기술통계량 출력

<분석 순서>

- 1) 기술통계량을 출력하기 위한 변수를 선택하고, 해당 변수만으로 구성된 데이터프레임을 만든다.
- 2) 'sjmisc' 패키지의 `descr` 함수를 이용하여 기술통계량을 출력한다.

```
# ① 기술통계량을 출력할 변수를 선택하고, 해당 변수만으로 구성된 데이터 만들기
var1 <- c("q33a01w1", "q50w1")
# tab1: 'q33a01w1'와 'q50w1' 변수만 있는 데이터프레임
tab1 <- spssdata[var1]
# ② 기술통계량 출력
# viewer에 직접 출력하는 방법
library(sjmisc)
descr(tab1)
descr(tab1, out = "v") #viewer로 출력
descr(tab1, out = "b") #browser로 출력
```

- ✓ `descr` 함수는 데이터프레임 단위로 기술통계량을 출력
 - `spssdata$q50w1`와 같이 기술통계량을 출력할 대상 변수를 지정할 수 없고, 오직 데이터프레임만 지정할 수 있음

3. 여러 형태의 도표 그리기

1. 바 도표(Bar chart)

1) barplot 함수를 이용한 바 도표 출력

<분석 순서>

- 1) barplot 함수를 이용하여 속성별 빈도에 따른 바 도표를 출력한다.
- 2) barplot 함수를 이용하여 속성별 비율에 따른 바 도표를 출력한다.
- 3) 필요한 경우 'RColorBrewer' 패키지를 이용하여 바 도표에 색상을 지정한다.

```
# ① 속성별 빈도에 따른 바 도표(빈도)
barplot(table(spssdata$q50w1),
        names.arg=c("전혀 만족하지 못한다", "만족하지 못하는 편이다",
                    "보통이다", "만족하는 편이다", "매우 만족한다"),
        space=1.5, border=NA, cex.names=0.5, ylim=c(0, 350))

# ② 속성별 비율에 따른 바 도표(비율)
barplot(prop.table(table(spssdata$q50w1)),
        names.arg=c("전혀 만족하지 못한다", "만족하지 못하는 편이다",
                    "보통이다", "만족하는 편이다", "매우 만족한다"),
        space=1.5, border=NA, cex.names=0.5, ylim=c(0, 1.0))
```

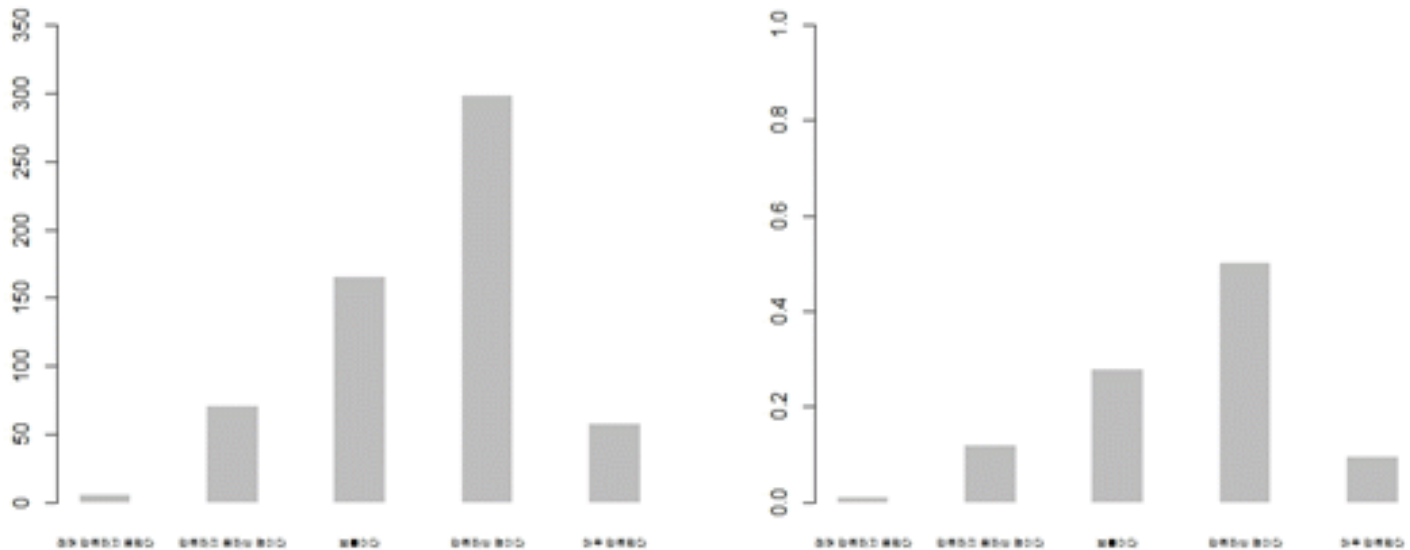
#Error in plot.new() : figure margins too large

```
par("mar")
```

```
par(mar=c(1,1,1,1)) #혹은 plot 창 확대
```

3. 여러 형태의 도표 그리기

- ✓ barplot 함수는 바 도표로 출력하고자 하는 대상 변수에 대해 **table** 함수를 사용하여 계산된 빈도를 사용
 - barplot 함수가 주어진 데이터를 직접 계산하여 빈도나 비율을 계산해 주지 않기 때문
- ✓ prop.table 함수를 사용해 바 도표로 출력하고자 하는 대상 변수에 대해 계산된 비율을 barplot 함수에 사용

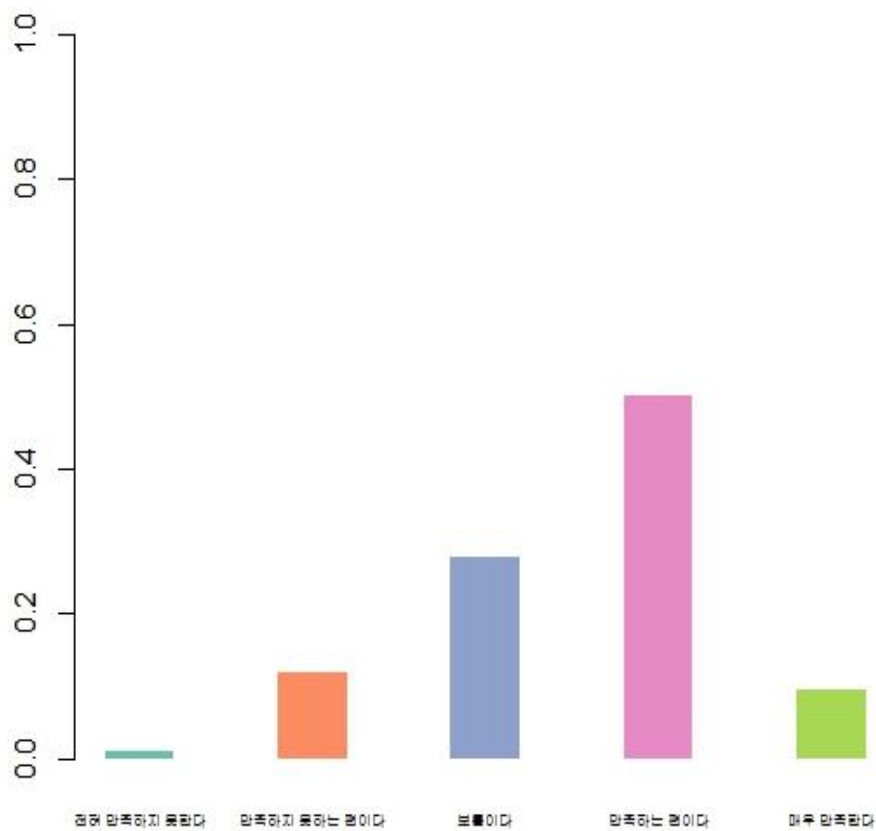


3. 여러 형태의 도표 그리기

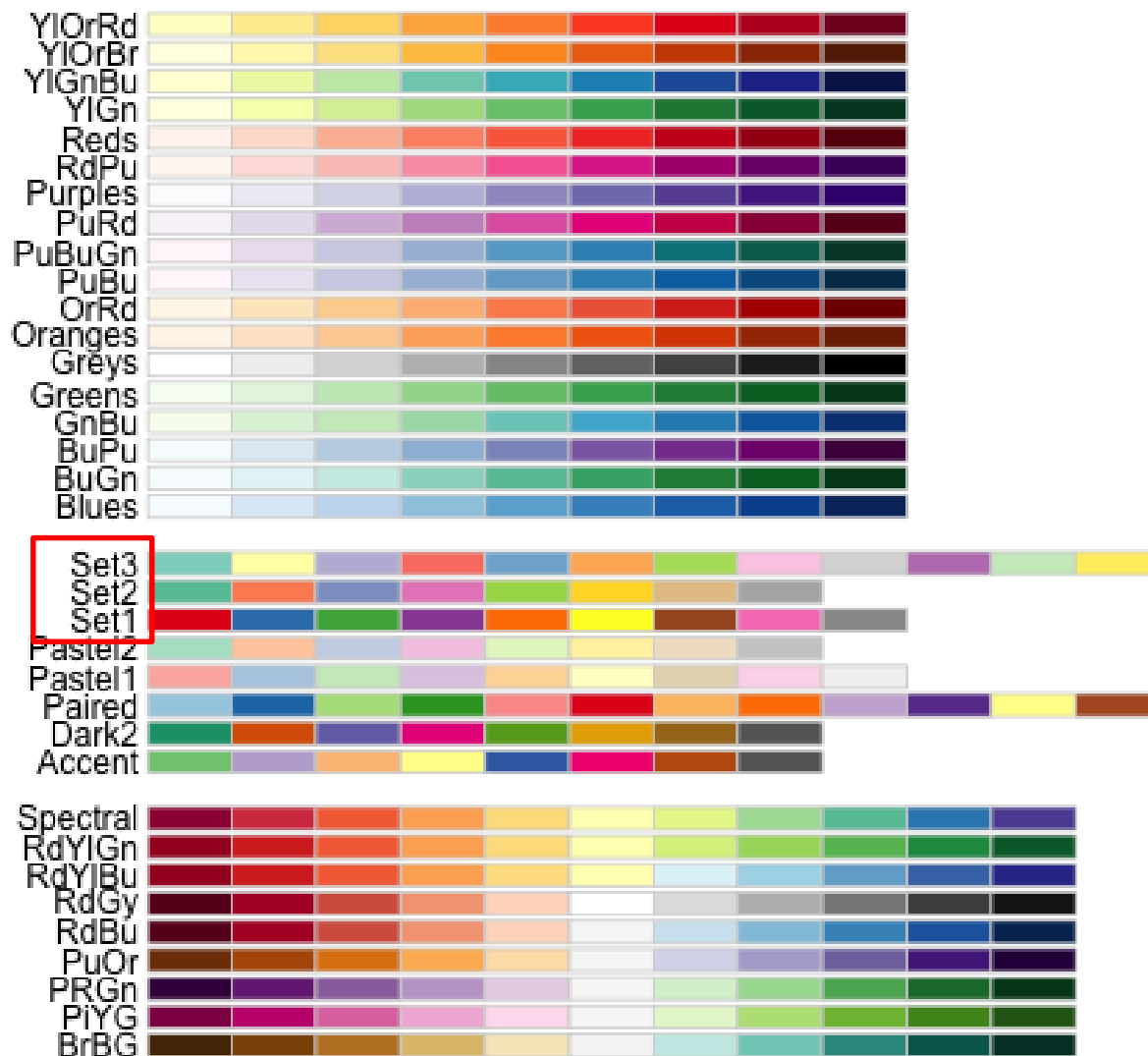
```
# ③ 'RColorBrewer' 패키지를 이용한 바 도표 색상 적용
library(RColorBrewer)
pal1 <- brewer.pal(5,"Set2") # RColorBrewer 색상표 참고
barplot(prop.table(table(spssdata$q50w1)), col=pal1,
        names.arg=c("전혀 만족하지 못한다", "만족하지 못하는 편이다",
                    "보통이다", "만족하는 편이다", "매우 만족한다"),
        space=1.5, border=NA, cex.names=0.5, ylim=c(0, 1.0))
```

- ✓ 'RColorBrewer' 패키지의 brewer.pal 함수를 이용하여 바의 색상을 지정
 - RColorBrewer 패키지를 설치하고 불러옴
 - 지정된 색상군과 색상의 수에 대한 정보는 pal1이라는 객체에 저장
 - barplot 함수에서 col 인자는 바 도표의 색상을 지정하기 위한 인자로 이 인자에 앞서 만들었던 pal1이라는 객체를 지정

3. 여러 형태의 도표 그리기



3. 여러 형태의 도표 그리기





3. 여러 형태의 도표 그리기

2) 'sjPlot' 패키지를 이용한 바 도표 출력

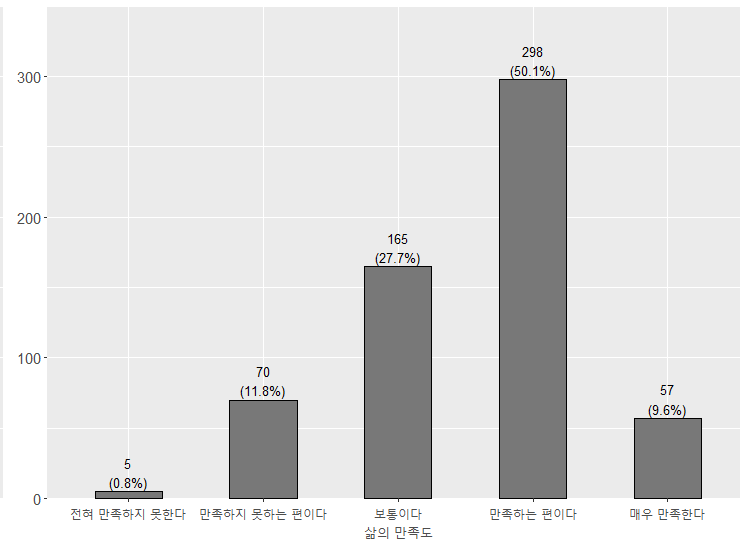
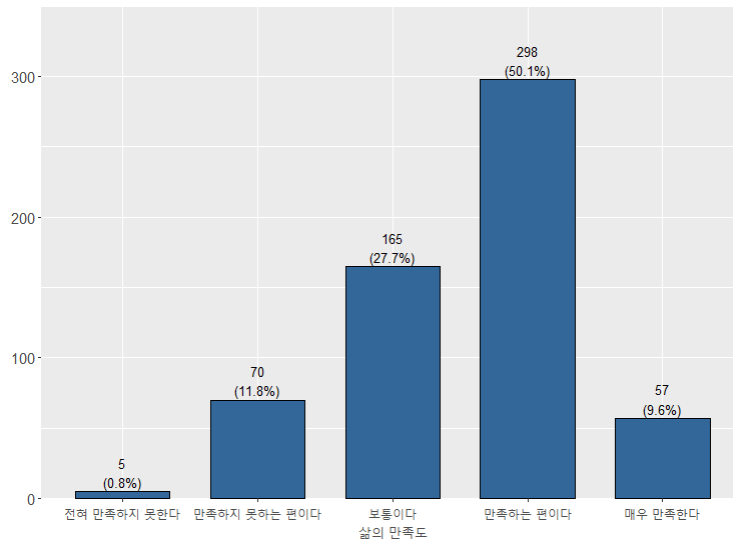
<분석 순서>

- 1) 'sjPlot' 패키지의 `sjp.frq` 함수를 이용하여 바 도표를 출력한다.
- 2) `set_theme` 함수를 이용하여 바 도표의 특성을 지정한다

```
# 'sjPlot' 패키지를 이용한 바 도표
library(sjPlot)
sjp.frq(spssdata$q50w1, axis.labels = c("전혀 만족하지 못한다", "만족
하지 못하는 편이다", "보통이다", "만족하는 편이다", "매우 만족한다"),
        axis.title = "삶의 만족도")
# 바 도표 꾸미기
set_theme(axis.textsize = 1.0, geom.label.size = 3.5)
sjp.frq(spssdata$q50w1, axis.labels = c("전혀 만족하지 못한다", "만족
하지 못하는 편이다", "보통이다", "만족하는 편이다", "매우 만족한다"),
        geom.size = 0.5, ylim = c(0, 350), geom.colors = "grey47",
        axis.title="삶의 만족도")
```

- ✓ 'sjPlot' 패키지를 설치하고 불러옴
- ✓ `sjp.frq` 함수에 바 도표를 출력하고자 하는 대상 변수를 입력
- ✓ 바 도표의 특성을 지정하기 위해 `sjp.frq`, `set_theme` 함수를 이용함

3. 여러 형태의 도표 그리기





3. 여러 형태의 도표 그리기

2. 히스토그램 도표와 밀도 도표

1) 히스토그램 도표(Histogram) 출력

<분석 순서>

- 1) q33a01w1부터 q33a06w1까지 6개의 문항을 합하여 '부모에 대한 애착' 변수를 만든다.
- 2) hist 함수를 이용하여 히스토그램 도표를 출력한다.
- 3) 'RColorBrewer' 패키지를 이용하여 히스토그램 도표의 색상을 지정한다.

① 변수 계산 방법(1)

```
spssdata$attachment <- spssdata$q33a01w1+spssdata$q33a02w1+  
  spssdata$q33a03w1+spssdata$q33a04w1+spssdata$q33a05w1+  
  spssdata$q33a06w1
```

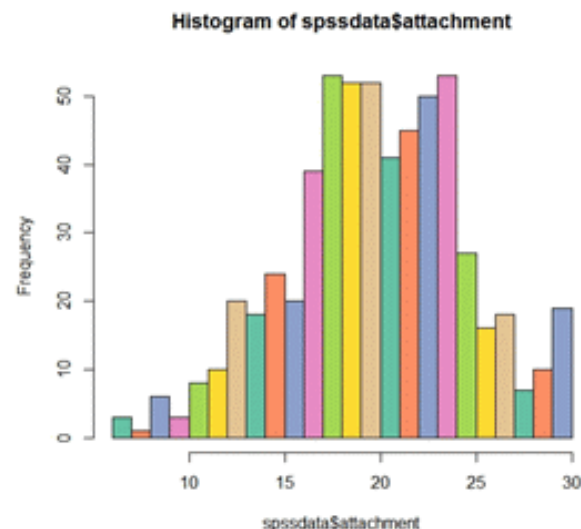
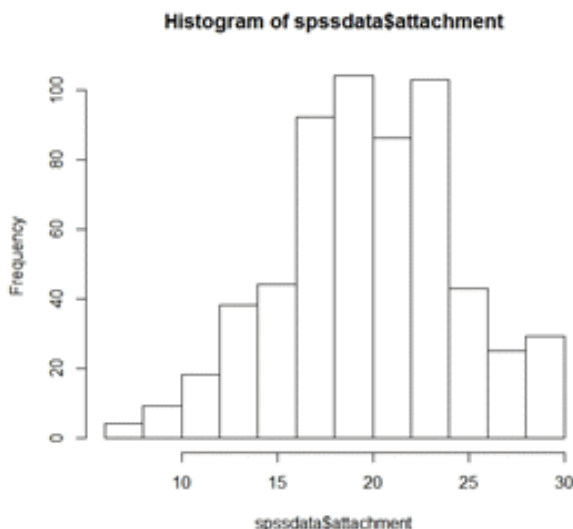
변수 계산 방법(2)

```
attach(spssdata)  
spssdata$attachment <- q33a01w1+q33a02w1+q33a03w1+q33a04w1+  
  q33a05w1+q33a06w1  
detach(spssdata)
```

3. 여러 형태의 도표 그리기

```
# ② 히스토그램 도표 만들기
hist(spssdata$attachment)
# ③ 'RColorBrewer' 패키지를 이용하여 히스토그램 도표의 색상을 지정
library(RColorBrewer)
pal1 <- brewer.pal(7,"set2")
hist(spssdata$attachment, breaks=20, col=pal1)
```

- ✓ hist 함수의 인자로 결과를 출력하고자 하는 해당 변수 (spssdata\$attachment)를 지정하여 히스토그램 도표를 만듦
- ✓ 히스토그램 도표에 색상을 지정하고, 계층의 수(breaks=20)를 지정





3. 여러 형태의 도표 그리기

2) 밀도 도표(density chart) 출력

<분석 순서>

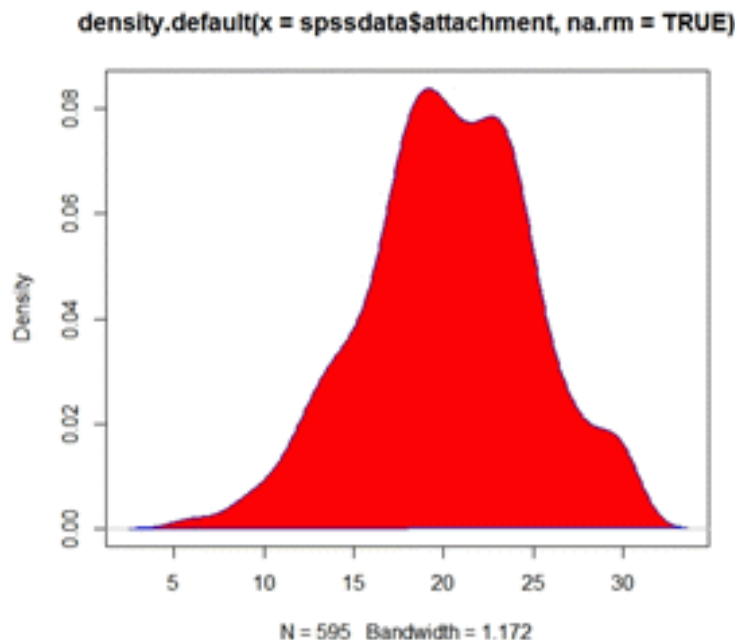
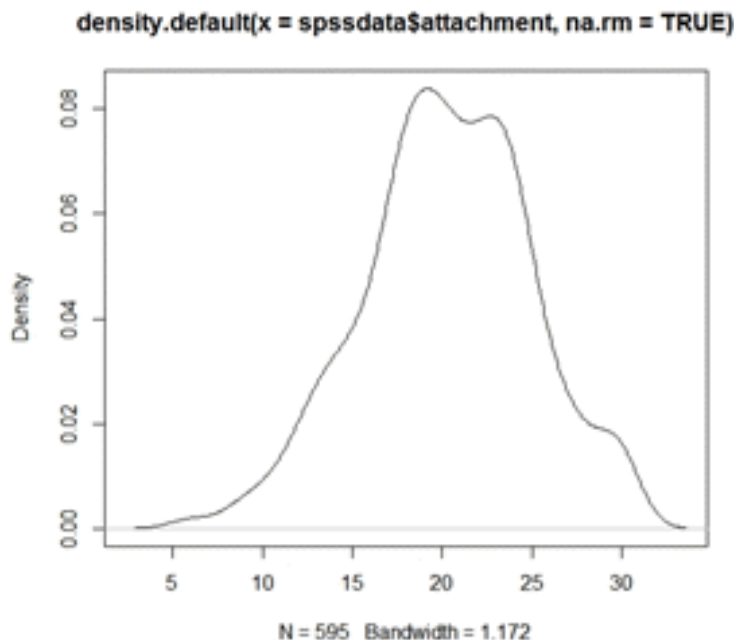
- 1) plot 함수와 density 함수로 밀도 도표를 만든다.
- 2) 밀도 도표에 색상을 추가한다.

```
# ① 밀도 도표 만들기(1)
plot(density(spssdata$attachment, na.rm=TRUE))
# 밀도 도표 만들기(2)
d <- density(spssdata$attachment, na.rm=TRUE)
plot(d)
# ② 밀도 도표에 색상 추가
polygon(d, col="red", border="blue")
```

- ✓ density 함수는 커널밀도추정을 통해 히스토그램 도표에서의 매끄럽지 못한 형태를 완만한 형태로 조절할 수 있는 함수
 - 해당 변수에 결측값이 있는 경우에는 na.rm 인자를 통해 결측값을 제외시키라고 지정해야 함
- ✓ density 함수로 만든 밀도 도표를 plot 함수를 이용하여 출력

3. 여러 형태의 도표 그리기

- ✓ 밀도 도표에 색상을 추가하고 싶다면 `polygon` 함수를 이용해 다각형의 면적에 색상을 추가할 수 있음
 - `polygon` 함수에서 테두리 선의 색은 `border` 인자로 지정할 수 있고, 테두리 내의 색상은 `col` 인자로 지정할 수 있음



3. 여러 형태의 도표 그리기

3. 파이 도표

1) 파이 도표(Pie chart) 만들기

<분석 순서>

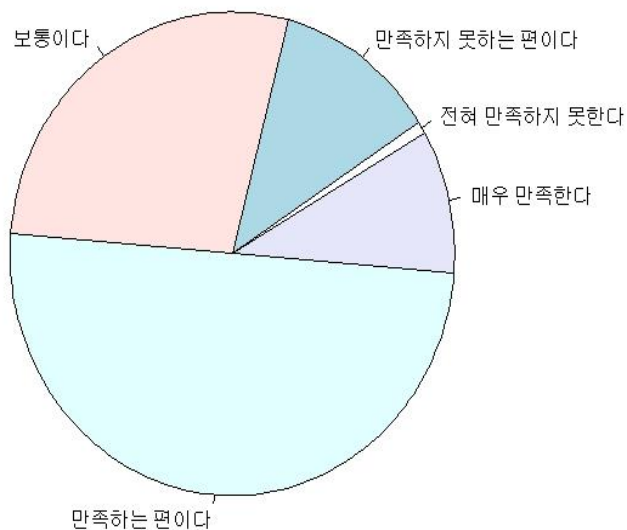
- 1) 파이 도표에 출력될 변수값 설명을 입력한다.
- 2) table 함수로 변수값의 사례수를 계산한다.
- 3) pie 함수로 파이 도표를 출력한다.

```
# 변수값 설명을 직접 입력하여 파이 도표 만들기
lbl <- c("전혀 만족하지 못한다", "만족하지 못하는 편이다", "보통이다",
        "만족하는 편이다", "매우 만족한다") # 변수값 설명 입력
pieg <- table(spssdata$q50w1)                # 각 변수값별 사례수 계산
pie(pieg, labels = lbl)
```

- ✓ pie 설명을 출력할 수 없으므로 직접 변수값 설명을 입력해야 함
 - 변수값 설명을 파이 도표에 적용하기 위해 우선 문자형 벡터의 형태로 변수값 설명을 직접 입력하고, 이 내용을 lbl이라는 객체에 저장

3. 여러 형태의 도표 그리기

- ✓ pie 함수는 앞서 살펴보았던 barplot 함수와 같이 지정한 변수에 대한 빈도나 비율을 **직접 계산하지 않음**
 - table 함수를 통해 각 변수값의 사례수를 계산하여 pieg라는 객체에 저장
- ✓ pie 함수에 변수값에 따른 사례수의 정보가 포함된 객체와 변수값 설명이 저장된 객체를 지정





3. 여러 형태의 도표 그리기

2) 변수값 설명과 백분율이 출력되는 파이 도표 만들기

<분석 순서>

- 1) 파이 도표에 출력될 변수값 설명을 입력한다.
- 2) `prop.table` 인자를 이용하여 변수값별 백분율을 계산한다.
- 3) 변수값 설명과 백분율을 붙인다.
- 4) 백분율에 '%' 기호를 붙인다.
- 5) `pie` 함수로 파이 도표를 출력한다.

```
# ① 변수값 설명을 문자형 벡터로 입력하여 lbl 객체에 저장
lbl <- c("전혀 만족하지 못한다", "만족하지 못하는 편이다", "보통이다",
"만족하는 편이다", "매우 만족한다")
# ② 변수값별 백분율 구하기
pct <- round(100*prop.table(table(spssdata$q50w1)), 1)
# ③ 변수값 설명과 백분율 붙이기
lbl_w_pct <- paste(lbl, pct)
# ④ 백분율 뒤에 '%' 기호 붙이기
lbl_w_pct <- paste0(lbl_w_pct, "%")
# ⑤ pie 함수를 이용하여 파이 도표 출력
pie(pct, labels = lbl_w_pct, col=rainbow(5),
main="학생이 느끼는 삶의 만족도", init.angle=0, radius=1.0)
```



3. 여러 형태의 도표 그리기

- ✓ 변수값 설명을 문자형 벡터로 입력하여 `lbl`이라는 객체에 저장
- ✓ 변수값별 백분율을 계산
 - `table` 함수로 대상 변수(`spssdata$q50w1`)의 변수값별 사례수를 계산
 - `prop.table` 함수로 `table` 인자에서 구한 변수값별 사례수를 변수값별 비율로 계산
 - `prop.table` 함수로 계산한 변수값별 비율에 100을 곱하여 백분율로 계산(`round` 함수를 이용하여 소수점 자리 지정)
 - 변수값별 백분율을 계산한 결과를 `pct`라는 객체에 저장
- ✓ **`paste` 함수를 사용해서** 변수값 설명과 백분율 값을 붙임
- ✓ 백분율 뒤에 '%' 기호 붙이기
- ✓ `pie` 함수에는 변수값별 비율이 저장된 `pct`라는 객체를 지정하여, 이 객체에 저장되어 있는 백분율에 따라 파이 도표에서 각 파이의 넓이를 결정하도록 함

3. 여러 형태의 도표 그리기





3. 여러 형태의 도표 그리기

3) 3D 파이 도표 만들기

<분석 순서>

- 1) 파이 도표에 출력될 변수값 설명을 입력한다.
- 2) `prop.table` 인자를 이용하여 변수값별 백분율을 계산한다.
- 3) 변수값 설명과 백분율을 붙인다.
- 4) 백분율에 '%' 기호를 붙인다.
- 5) 'plotrix' 패키지를 이용하여 3D 파이 도표를 출력한다.

```
# ① 변수값 설명을 문자형 벡터로 입력하여 lbl 객체에 저장
lbl <- c("전혀 만족하지 못한다", "만족하지 못하는 편이다", "보통이다",
"만족하는 편이다", "매우 만족한다")
# ② 변수값별 백분율 구하기
pct <- round(100*prop.table(table(spssdata$q50w1)), 1)
# ③ 변수값 설명과 백분율 붙이기
lbl_w_pct <- paste(lbl, pct)
# ④ 백분율 뒤에 '%' 기호 붙이기
lbl_w_pct <- paste0(lbl_w_pct, "%")
```

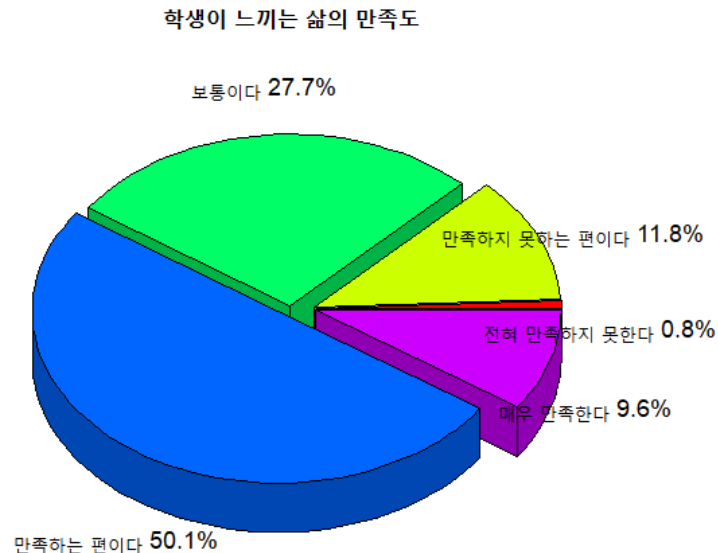
3. 여러 형태의 도표 그리기

```
# 3D 파이 도표 만들기
install.packages("plotrix")
library(plotrix)
pie3D(pct, labels = lbl_w_pct, col=rainbow(5),
main="학생이 느끼는 삶의 만족도",
labelcex=1.1, explode=0.1, theta=1.1, shade=0.3)
```

- ✓ ①부터 ④까지는 '변수값 설명과 백분율이 출력되는 파이 도표 만들기'의 내용과 동일
- ✓ 'plotrix' 패키지를 이용하여 3D 파이 도표를 만들기
 - 'plotrix' 패키지를 설치하고 불러온다.
 - pie3D 함수에는 변수값별 비율이 저장된 pct라는 객체를 지정한다. 이 객체에 저장되어 있는 백분율에 따라 3D 파이 도표에서 각 파이의 넓이를 결정
 - 각 변수값별 설명을 출력할 수 있는 labels 인자에는 변수값 설명, 변수값별 비율, 그리고 '%' 기호를 합친 문자열을 저장한 객체인 lbls를 지정
 - 파이 도표의 제목을 출력할 수 있는 main 인자에는 변수 설명을 입력

3. 여러 형태의 도표 그리기

- labelcex 인자로 변수값 설명의 글자 크기를 조정하고, explode 인자로 변수 속성에 따른 파이 조각들 간의 간격을 조정
- theta 인자를 통해 파이 도표의 기울기 각도를 조정
 - » theta 인자의 값을 크게 할수록 파이 도표는 평면적으로 나타남
- shade 인자를 통해서 변수 속성에 따른 파이 조각의 그림자 명암을 조정





Weekly Assignment #3

1. 남성, 여성 청소년 그룹 각각에 대해서 다음을 출력하는 코드를 작성하여 제출. 변수는 `spssdata$q50w1`)
 - 빈도표
 - 기술통계량
 - 바 차트
 - 파이 차트