

분석 정보 시각표현하기 LAB01 ¶

정보 표현을 위한 그래픽 요소

시각화 도구를 이용하여 생성된 그래프나 차트는 인간의 눈을 통해 모양, 크기, 색, 위치 등의 패턴을 보게 되고 이를 뇌에서 해석한 뒤 학습

이 시지각 이론은 1900년대 초 독일의 심리학파인 게슈탈트 (Gestalt) 학파에 의해 활발하게 연구되었고, 시지각의 **게슈탈트 원리(The Gestalt Principles of Visual Perception)**로 정립되었다.

게슈탈트 원리

(1) 유사성의 법칙(Law of Similarity)

유사한 요소(형태, 색, 크기, 밝기)끼리 그룹 지어 하나의 패턴으로 보이게 함.

(2) 근접성의 법칙(Law of Proximity)

시공간적으로 서로 가까이 있는 것들을 함께 집단화하여 본다.

(3) 단순성의 법칙(Law of Simplicity)

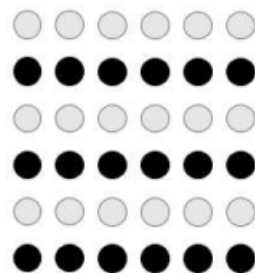
주어진 조건으로 최대한 가장 단순한 쪽으로 인식한다.

(4) 연속성의 법칙(Law of Continuity)

요소들이 부드러운 연속을 따라 함께 묶여 지각된다.

(5) 폐쇄성의 법칙(Law of Closure)

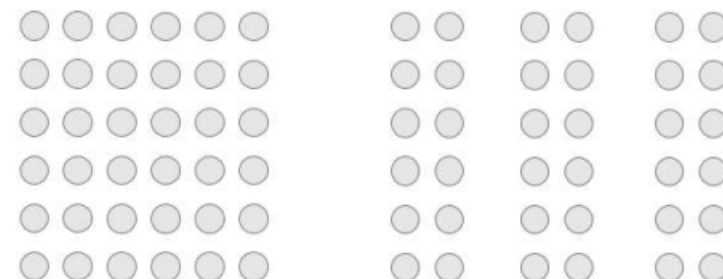
기존의 지식을 토대로 완성되지 않은 형태를 완성해 간다.



[그림 3-1] 유사성의 법칙



[그림 3-2] 폐쇄성의 법칙



[그림 3-3] 근접성의 법칙

(1) 유사성의 법칙(Law of Similarity)

유사한 요소(형태, 색, 크기, 밝기)끼리 그룹 지어 하나의 패턴으로 보이게 함.

색이 가장 좋다.

[Usage]

```
scale_color_brewer(type = 'seq', palette = 1)
```

type : 색상에 사용되는 변수의 클래스입니다.

Sequential (seq) | Qualitative (qual) | Diverging (div)

palette : 사용할 색상 조합. 정수 또는 팔레트 이름이 될 수 있습니다.

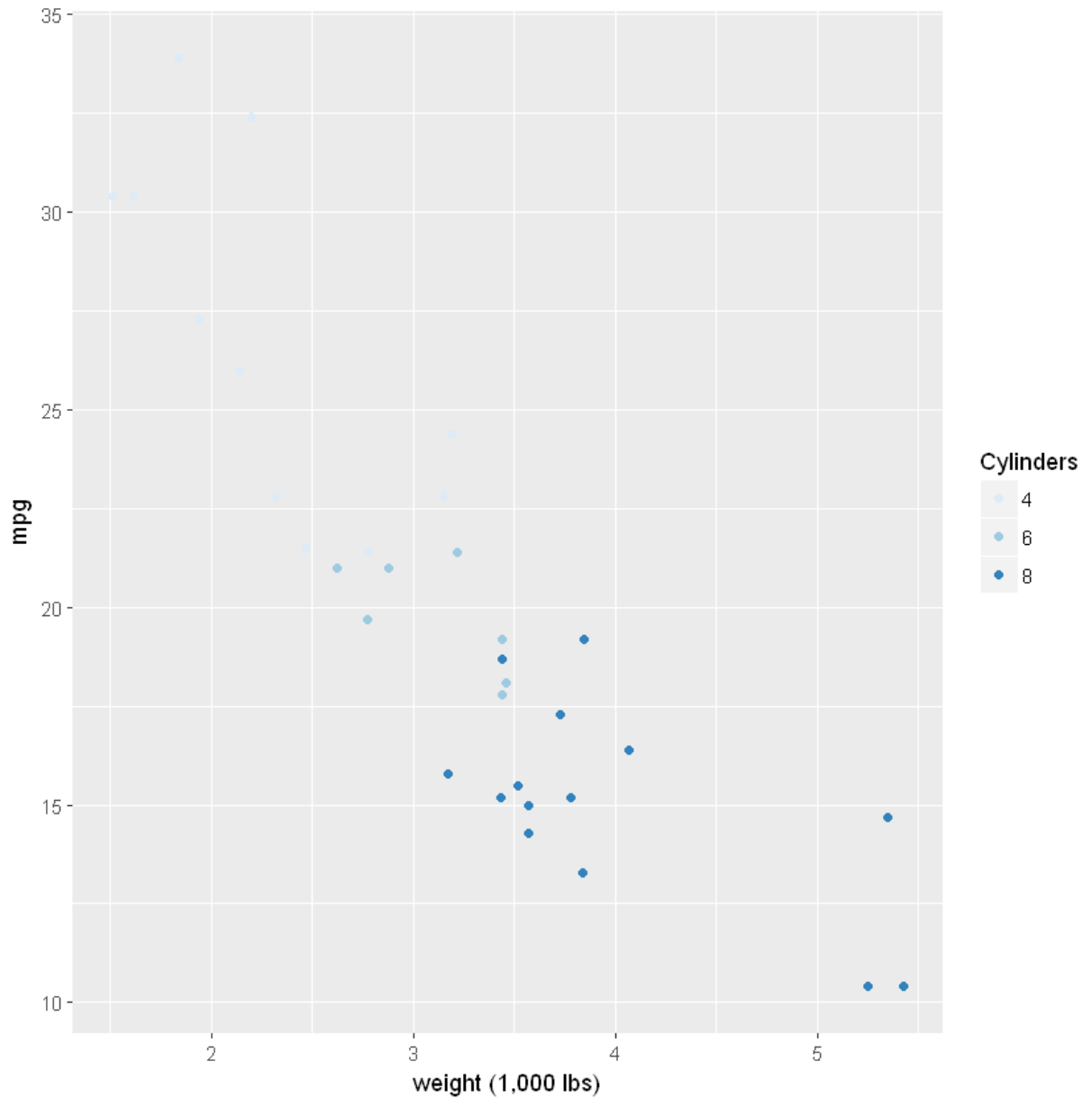
자세한 내용은 <http://colorbrewer2.org/>를 참조

다양한 scale_color_brewer

https://rstudio-pubs-static.s3.amazonaws.com/177286_826aed2f00794640b301aeb42533c6f1.html

```
In [16]: # Make number of cylinders a factor, so ggplot will choose a discrete scale.
library(ggplot2)
mtcars$cyl<-factor(mtcars$cyl)

ggplot(mtcars, aes(wt, mpg)) +
  geom_point(aes(color=cyl)) +
  xlab("weight (1,000 lbs)") +
  scale_color_brewer("Cylinders", type="seq", palette=1)
```



In []:

(2) 근접성의 법칙(Low of Proximity)

시공간적으로 서로 가까이 있는 것들을 함께 집단화하여 본다.

K-means 알고리즘에 의한 클러스터 분석을 수행해 보자.

군집분석(Cluster)

- (가) 군집분석은 비지도학습(Unsupervised learning) 분석 기법 중 하나.
(나) 사전 정보 없이 자료를 컴퓨터에게 주고 유사한 대상끼리 묶으라는 명령을 내리는 분석방법

K-means 군집 분석은? (거리를 이용)

- 01 분석자가 설정한 K개의 군집 중심점을 선정한다.
- 02 관측치가 거리상 가장 가까운 친구들을 군집화 시킨다.
- ...

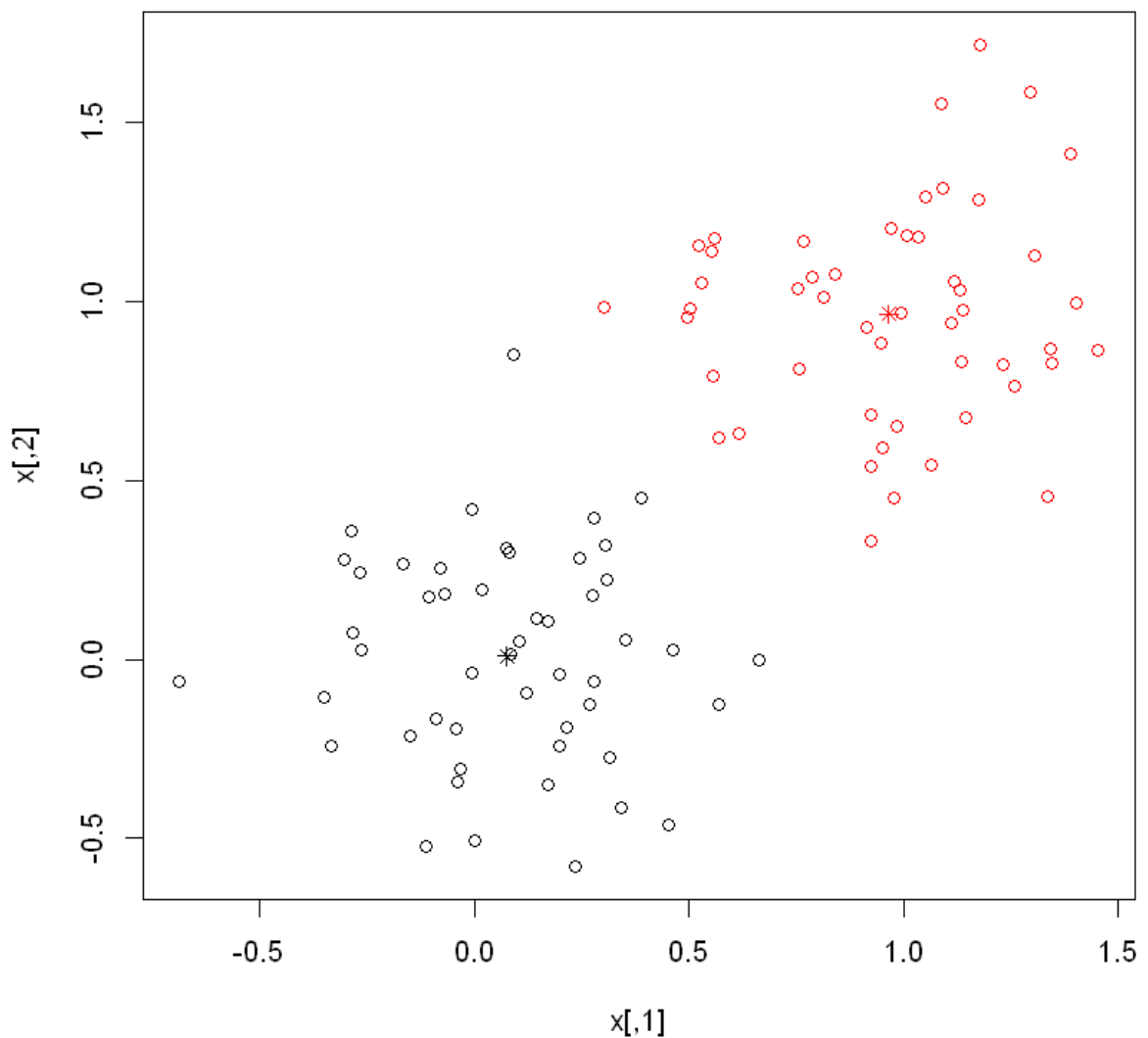
설명

- 01 정규 분포를 따르는 난수 발생
- 02 주어진 데이터를 이용, 유사한 친구끼리 묶기
- 03 그래프로 시각화 해보기

```
matrix(rnorm(100, sd=0.3), ncol=2) # 100개의 정규 분포 난수 발생(행렬 50,2로 만들기)
```

```
x : 데이터, centers = K개 그룹, iter.max : 반복 허용 횟수  
kmeans(x, centers = 2, iter.max = 20)
```

```
'cluster' 'centers' 'totss' 'withinss' 'tot.withinss' 'betweenss' 'size' 'iter' 'ifault'
```



5/21

크기(size),
모양(shape),
밝기(lightness/value),
색(color),
방향(orientation),
질감(texture),
위치(location)를 말한다.

(1) 크기(size)

기하적 요소들이 같은 크기를 가진 경우 상대적으로 하나만 작게 표시하면 작게 표현한 것만이 눈에 띄게, 그리고 또 클수록 시선이 주목된다.

(2) 모양(shape)

같은 모양 중 하나만 다른 모양이 있는 경우 다른 모양이 강조되어 눈에 띈다.

(3) 색(color)

같은 색을 이용한 것 중에서 다른 색을 사용한 요소가 더 잘 보인다.

(4) 밝기(명도 lightness/value)

동일한 색에 대하여 동일한 수준의 밝기(명도)를 가진 요소 중 일부만 다른 밝기를 줄때 눈에 띄게 된다. 밝기의 높고 낮음은 색보다 오히려 명시성에 영향을 준다.

(5) 방향(orientation)

하나만 다른 방향으로 기울이면 방향이 다른 요소에 시선이 끌리게 된다.

(6) 질감(texture)

같은 색상과 모양이라도 질감을 다르게 준 것만 강조된다.

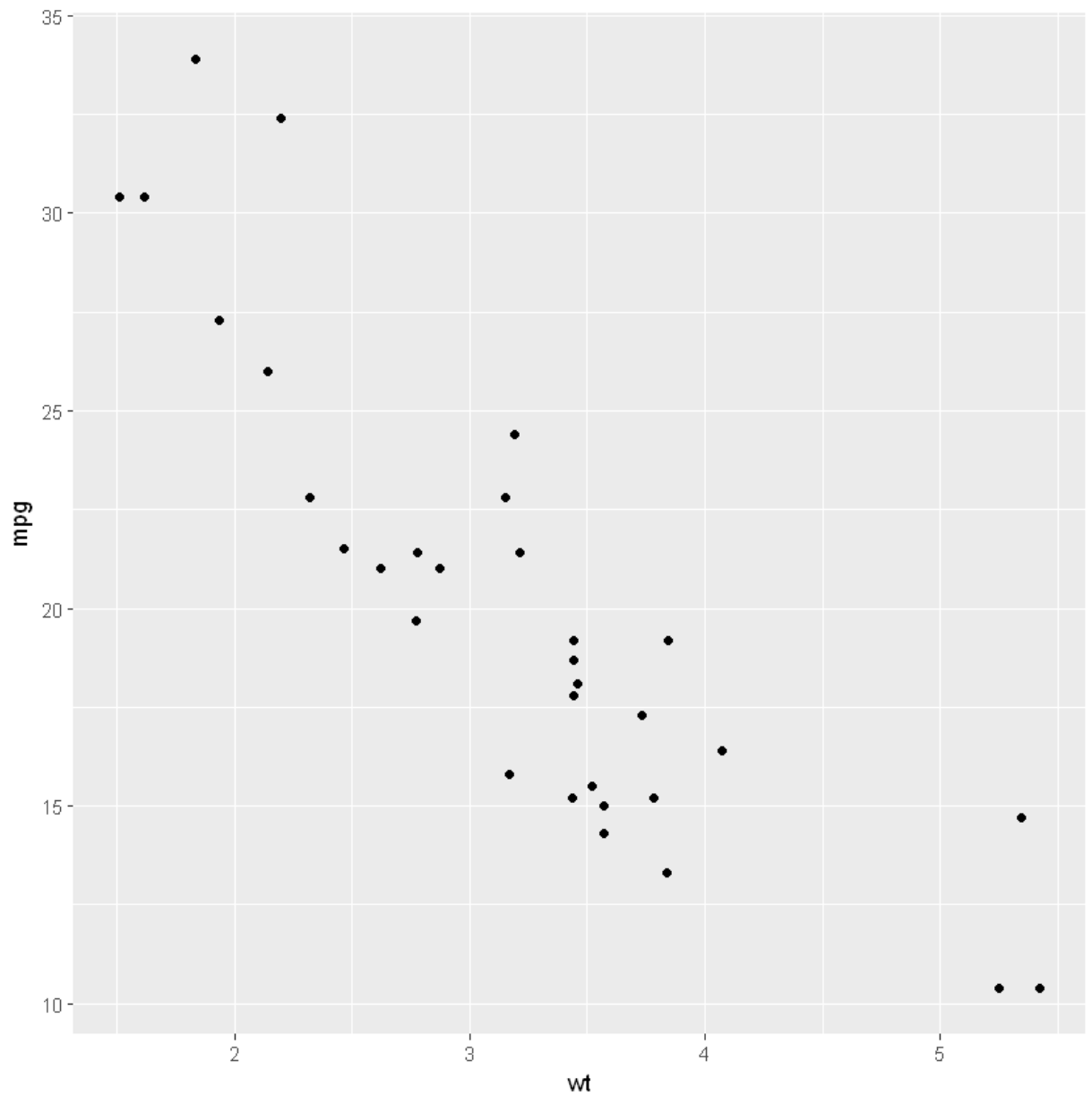
(7) 위치(location)

위치를 다르게 함으로써 강조할 수 있다. 특정한 요소의 상대적인 위치를 확인하여 주변의 다른 요소와의 관계를 비교한다.

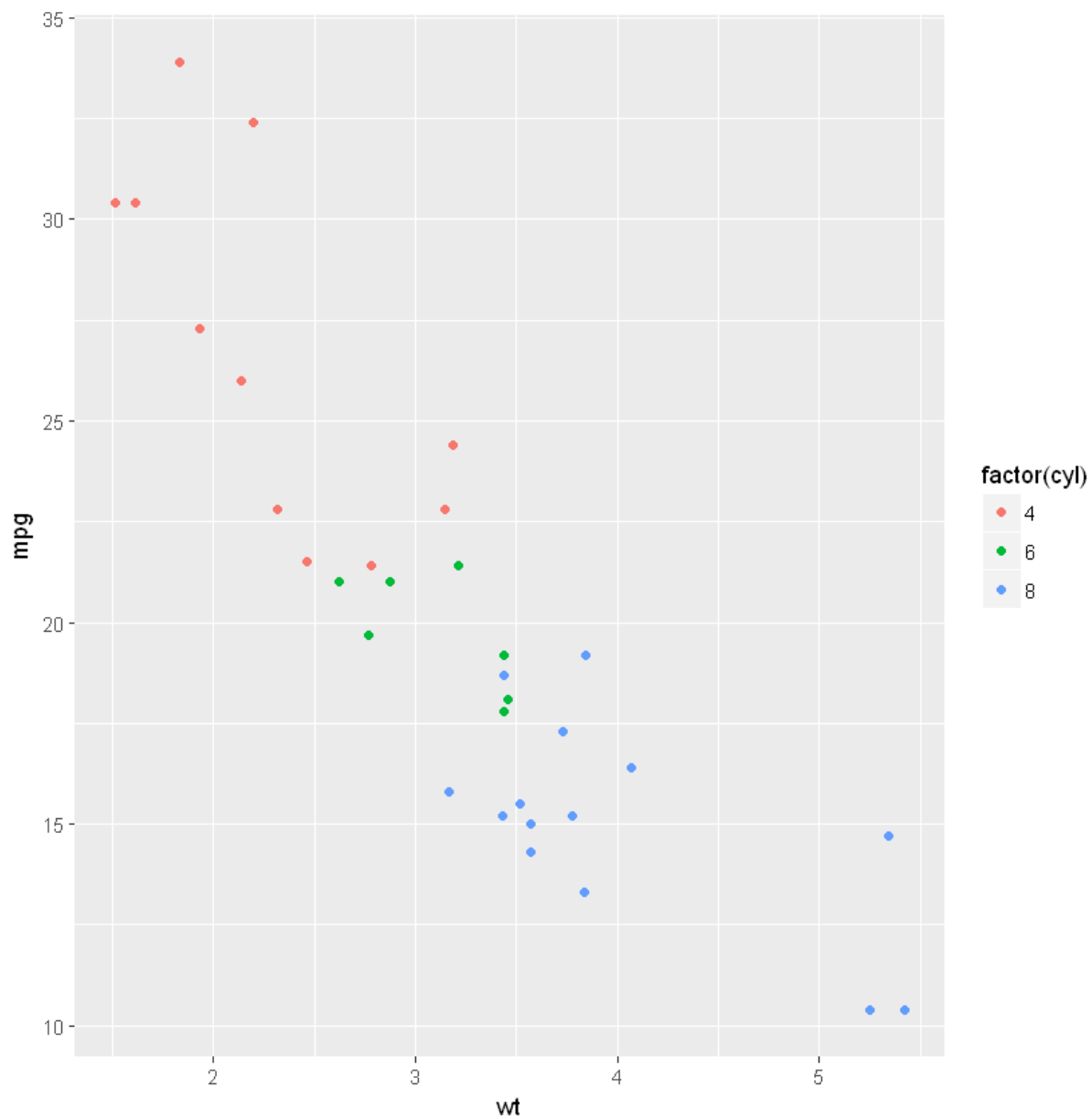
그래픽 7요소는 시각변수 7가지
크기(size),
모양(shape),
밝기(lightness/value),
색(color),
방향(orientation),

질감(texture),
위치(location)를 말한다.

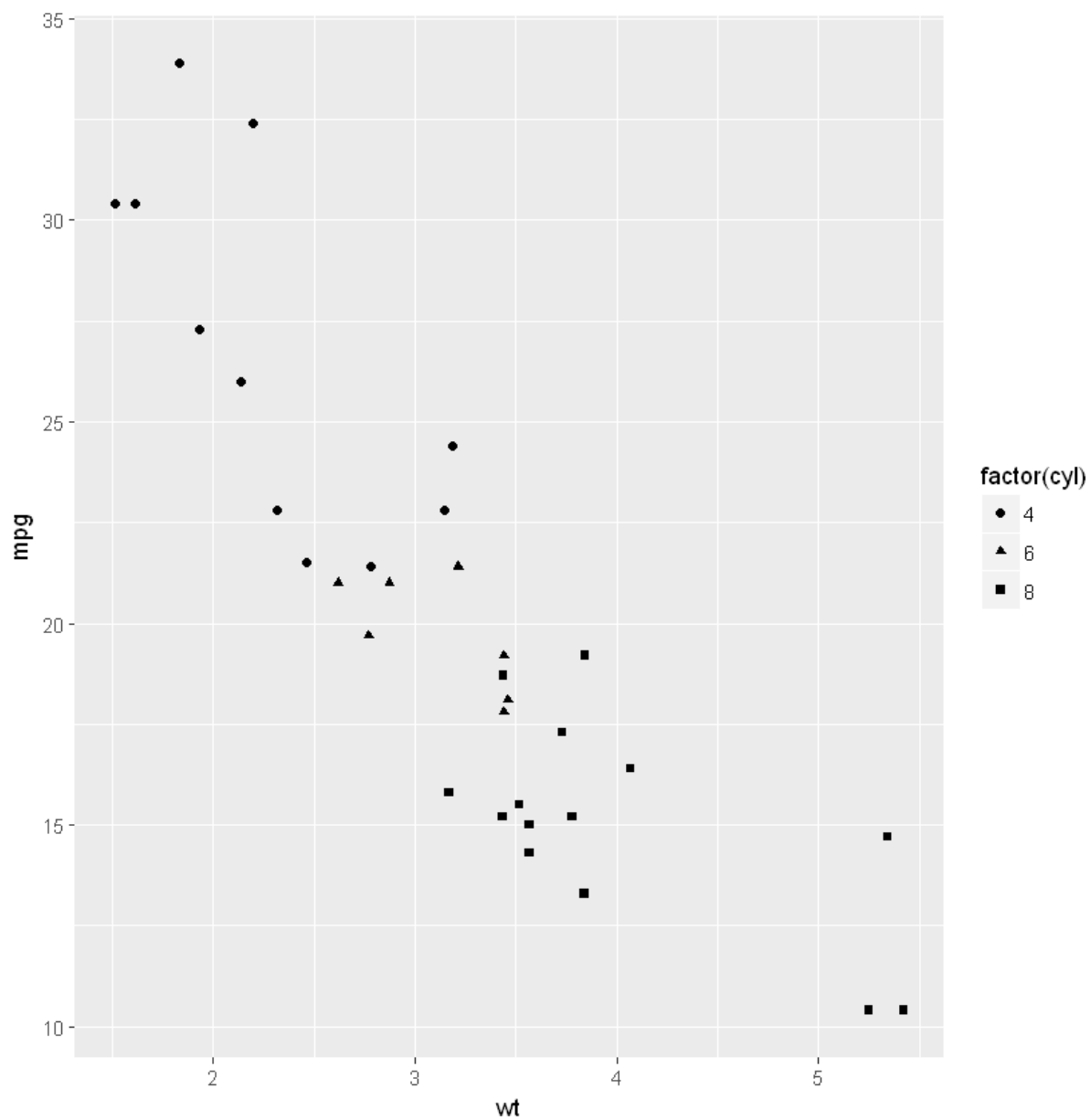
```
In [17]: p <- ggplot(mtcars, aes(wt, mpg))  
p + geom_point()
```



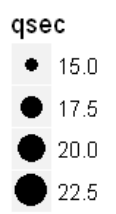
```
In [18]: #  $\Delta^H(color)$ ,  
p + geom_point(aes(colour = factor(cyl)))
```



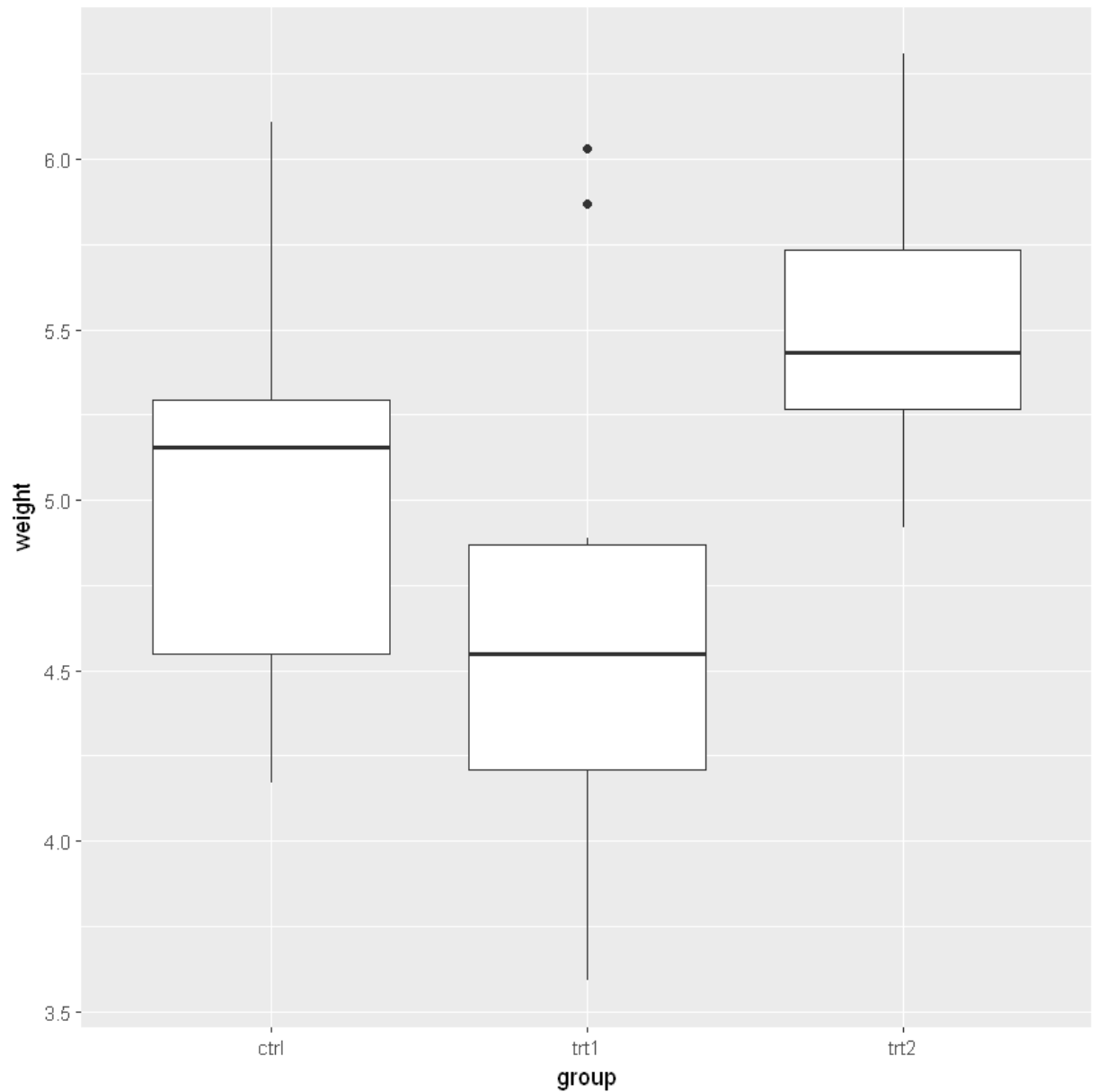

```
In [19]: # 모양(shape),  
p + geom_point(aes(shape = factor(cyl)))
```

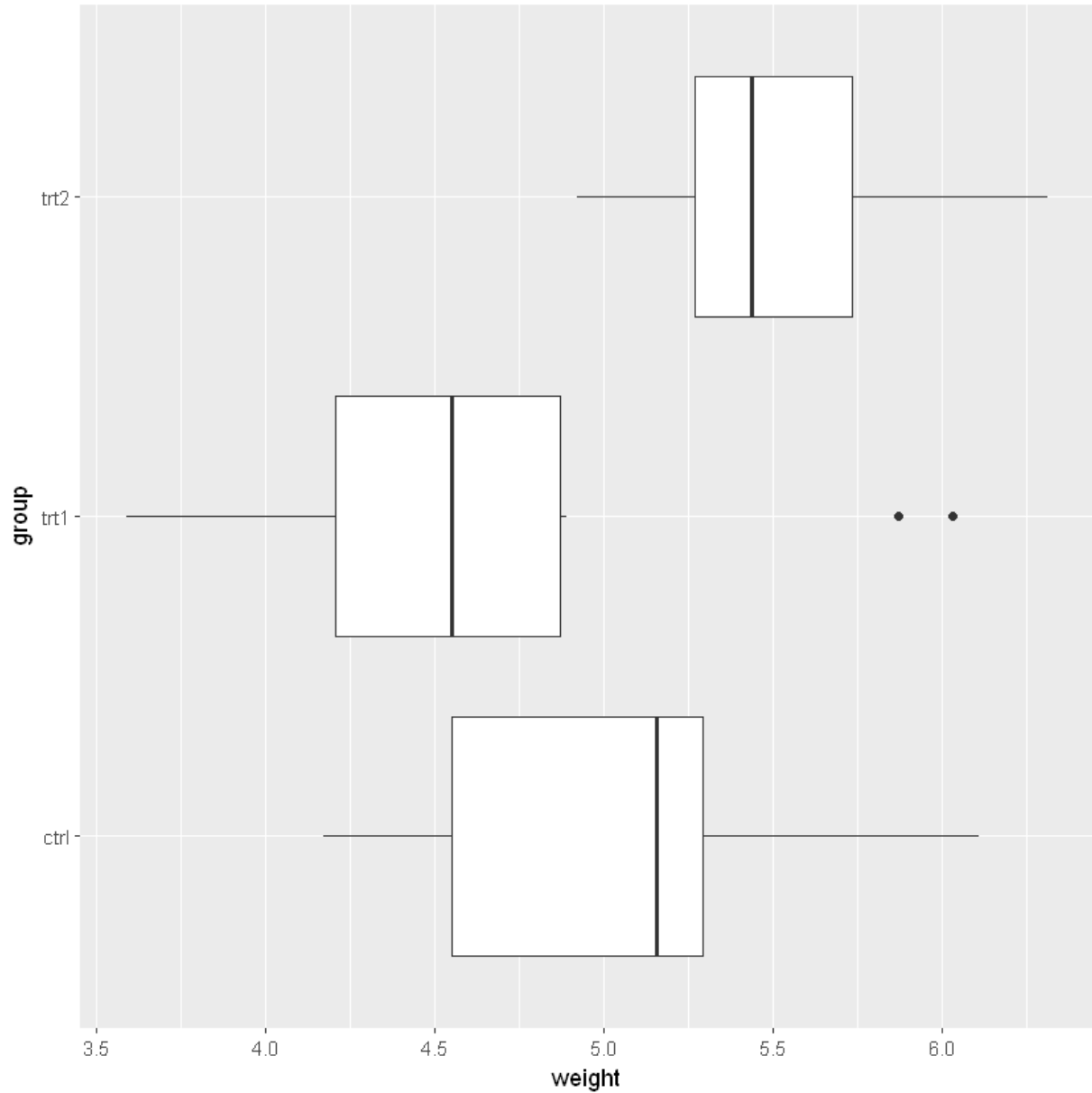


```
p + geom_point(aes(size = qsec))
```



```
In [29]: #방향(orientation),  
# coord_flip() : 수직이 수평, 수평이 수직이 되도록 좌표를 변경한다.  
  
library(ggplot2)  
# Basic box plot  
bp <- ggplot(PlantGrowth, aes(x=group, y=weight))+  
  geom_boxplot()  
bp  
# Horizontal box plot  
bp + coord_flip()
```





2 차원 커널 밀도 추정
지정된 범위에 걸쳐 커널 밀도 추정을 계산하고 래스터를 출력합니다.

```
In [40]: df <- data.frame(x=rnorm(100),y=rnorm(100))
k <- MASS::kde2d(df$x,df$y)
k
str(k)
```

\$x

```
-1.8787820099849 -1.69023447874199 -1.50168694749908 -1.31313941625617 -1.12459188501326
-0.936044353770353 -0.747496822527443 -0.558949291284534 -0.370401760041624 -0.181854228798714
0.00669330244419619 0.195240833687106 0.383788364930016 0.572335896172926 0.760883427415836
0.949430958658746 1.13797848990166 1.32652602114457 1.51507355238748 1.70362108363039
1.8921686148733 2.08071614611621 2.26926367735912 2.45781120860203 2.64635873984494
```

\$y

```
-2.23859607752579 -2.03632387029289 -1.83405166305998 -1.63177945582708 -1.42950724859417
-1.22723504136127 -1.02496283412836 -0.822690626895456 -0.62041841966255 -0.418146212429645
-0.21587400519674 -0.013601797963835 0.18867040926907 0.390942616501976 0.59321482373488
0.795487030967786 0.997759238200691 1.2000314454336 1.4023036526665 1.60457585989941
1.80684806713231 2.00912027436522 2.21139248159812 2.41366468883103 2.61593689606393
```

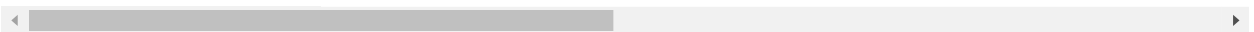
\$z

```
4.190153e-03 4.870558e-03 0.0049344534 0.0051438400 0.006546127 0.009703355 0.014119120 0.01828125 0.02072152 0.02127
6.902591e-03 8.233439e-03 0.0087525970 0.0096026358 0.012175072 0.016977520 0.023056081 0.02839995 0.03152064 0.03278
9.394385e-03 1.163271e-02 0.0132187642 0.0155833522 0.020154831 0.026995118 0.034465199 0.04023140 0.04334301 0.04523
1.079000e-02 1.404602e-02 0.0173255260 0.0221553109 0.029610658 0.038955602 0.047714247 0.05335898 0.05581546 0.05787
1.096841e-02 1.508414e-02 0.0202184525 0.0278226004 0.038429243 0.050507836 0.060859186 0.06675985 0.06876985 0.07068
1.075156e-02 1.528088e-02 0.0216375451 0.0311673209 0.044041320 0.058414639 0.070847293 0.07838497 0.08141473 0.08364
1.108604e-02 1.541902e-02 0.0217841142 0.0315666625 0.045023729 0.060700462 0.075574658 0.08649399 0.09270571 0.09643
1.201387e-02 1.562165e-02 0.0209054220 0.0295281772 0.042262641 0.058520342 0.076094480 0.09165131 0.10265687 0.10890
1.257350e-02 1.527587e-02 0.0192119733 0.0264873843 0.038542588 0.055613749 0.076150813 0.09651521 0.11234135 0.12103
1.177725e-02 1.386765e-02 0.0171786578 0.0241957315 0.036793561 0.055530817 0.078956410 0.10297873 0.12193466 0.13177
9.610826e-03 1.171795e-02 0.0156382446 0.0239252738 0.038348557 0.059131232 0.084434723 0.10984253 0.12931993 0.13854
7.008270e-03 9.757658e-03 0.0152864866 0.0257948795 0.042347395 0.064369307 0.089597184 0.11382009 0.13160200 0.13921
4.960111e-03 8.648264e-03 0.0160238922 0.0286032740 0.046395090 0.068008100 0.091198295 0.11264093 0.12806948 0.13451
3.798840e-03 8.272168e-03 0.0169201411 0.0304727891 0.047975321 0.067664845 0.087947680 0.10677150 0.12093651 0.12777
3.258272e-03 8.051693e-03 0.0169654447 0.0300820288 0.045954531 0.063092044 0.080935099 0.09850570 0.11305108 0.12165
2.941771e-03 7.552437e-03 0.0158227872 0.0274782836 0.041156291 0.056093815 0.072620564 0.09011691 0.10559387 0.11576
2.608758e-03 6.682092e-03 0.0137934956 0.0236297936 0.035331912 0.048932171 0.065076461 0.08265247 0.09804225 0.10794
2.172375e-03 5.506882e-03 0.0112675926 0.0193450972 0.029550653 0.042496411 0.058555087 0.07560611 0.08937646 0.09694
1.636693e-03 4.130008e-03 0.0084879079 0.0149058038 0.023814008 0.036136889 0.051738388 0.06756434 0.07893388 0.08333
1.078083e-03 2.735214e-03 0.0057381138 0.0105366834 0.017984538 0.029132829 0.043502159 0.05764257 0.06683892 0.06875
6.050100e-04 1.561685e-03 0.0034085887 0.0067068542 0.012481618 0.021809081 0.034204297 0.04645438 0.05416134 0.05491
2.852037e-04 7.609423e-04 0.0017728571 0.0038498511 0.007972667 0.015178724 0.025234672 0.03559241 0.04241734 0.04315
1.127412e-04 3.188113e-04 0.0008227023 0.0020392939 0.004778425 0.009981905 0.017738265 0.02631130 0.03257836 0.03394
```

```

3.787490e-05 1.183425e-04 0.0003548382 0.0010317720 0.002747021 0.006291271 0.011990919 0.01885799 0.02456038 0.02666
1.121164e-05 4.112733e-05 0.0001493054 0.0005102243 0.001521980 0.003782979 0.007707606 0.01287225 0.01772764 0.02023

```



List of 3

```

$ x: num [1:25] -1.88 -1.69 -1.5 -1.31 -1.12 ...
$ y: num [1:25] -2.24 -2.04 -1.83 -1.63 -1.43 ...
$ z: num [1:25, 1:25] 0.00419 0.0069 0.00939 0.01079 0.01097 ...

```

레벨 (등고선) 플롯 : filled.contour

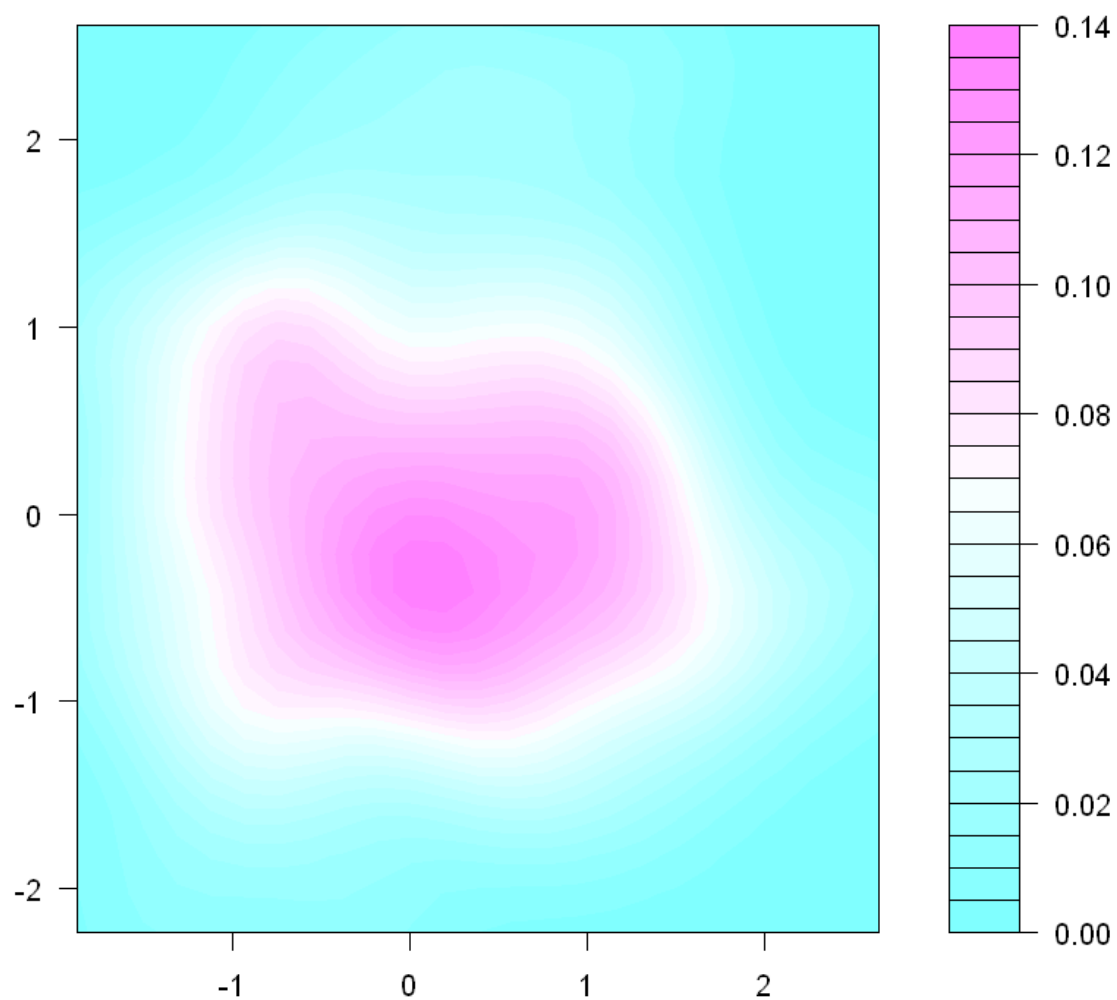
이 함수는 등고선 사이의 영역이 단색으로 채워진 등고선 플롯을 생성합니다

```

filled.contour(x = seq(0, 1, length.out = nrow(z)),
               y = seq(0, 1, length.out = ncol(z)),
               z,
               xlim = range(x, finite = TRUE),
               ylim = range(y, finite = TRUE),
               zlim = range(z, finite = TRUE),
               levels = pretty(zlim, nlevels), nlevels = 20,
               color.palette = cm.colors,
               col = color.palette(length(levels) - 1),
               plot.title, plot.axes, key.title, key.axes,
               asp = NA, xaxs = "i", yaxs = "i", las = 1,
               axes = TRUE, frame.plot = axes, ...)
filled.contour(x, y, z, levels, col)

```

```
In [41]: filled.contour(k)
```

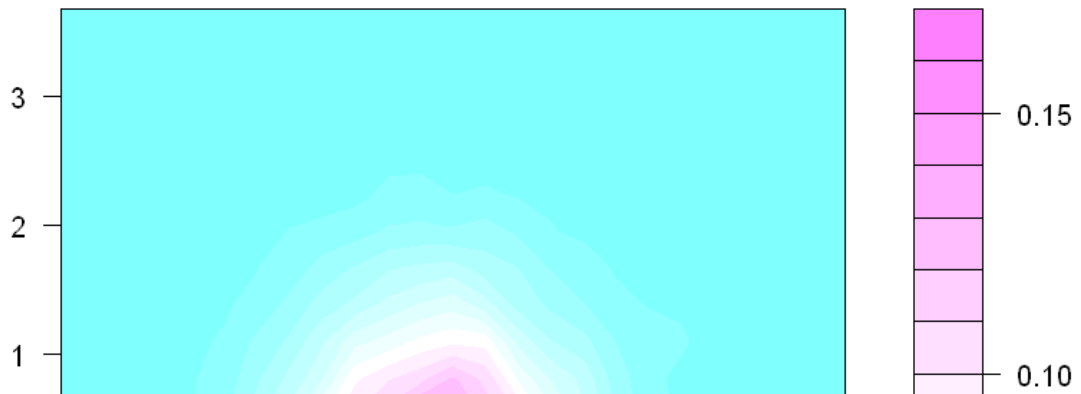


```
In [39]: #밝기(lightness/value)
df <- data.frame(x=rnorm(10000),y=rnorm(10000))
k <- MASS::kde2d(df$x,df$y)

filled.contour(k)

library(ggplot2)
ggplot(df,aes(x=x,y=y))+geom_density2d()

ggplot(df,aes(x=x,y=y))+
  stat_density2d(aes(alpha=..level..), geom="polygon") +
  scale_alpha_continuous(limits=c(0,0.2),breaks=seq(0,0.2,by=0.025))+
  geom_point(colour="red",alpha=0.02)+
  theme_bw()
```



In []:

그래픽 품질 향상을 위한 그래픽 디자인 원리

1. 타이포그래피(Typography)

2. 색상(Color)

3. 그리드(Grid)

4. 아이소타이프(ISOTYPE)

애드워드 튜페의 시각정보 디자인 7원칙

1. 시각적인 비교를 강화한다.
정보는 비교를 통해 사용자에게 더욱더 의미있게 된다.
관련된 변수와 추세를 비교할 수 있는 도구를 제시해야 한다.
2. 인과관계를 제시한다.
원인과 결과를 명확하게 제시해야 한다.
3. 다차원 변수를 표시한다.
관련 있는 여러개의 변수를 이용하여 정보를 표현한다.
4. 데이터, 텍스트, 그래픽을 한 화면에 효과적인 정보전달을 위해 어울리도록 배치한다.
범례와 레이블 이 포함된 경우 정보 전달이 효과적이다.
5. 콘텐츠의 질과 연관성, 진실성을 명확하게 한다.

시각화하려는 정보가 사용자의 목적을 달성하는데 유익한 것인가? 고민해야 한다.

6. 시간적인 순서가 아니라 공간적인 순서로 배치한다.
시간보다는 공간에 따라 배치해야 한다. 사용자의 이해가 쉬워진다.
7. 정량적 자료에 대한 양적 성질을 제거하지 말아야 한다.
그래프나 차트로 표현이 가능하다. 한눈에 양적인 정보를 파악할 수 있다.

수행 내용 : 그래픽 품질 향상하기

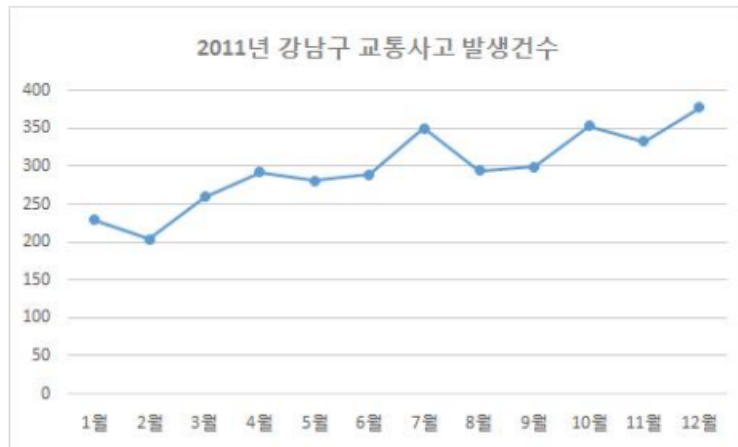
7가지 요소를 잘 표현하고 있는지 확인

1. 시각적 비교를 강화한다.
2. 인과관계를 제시한다.
3. 다차원 변수를 표시한다.
4. 텍스트, 그래픽, 데이터를 한 화면에 조화롭게 배치한다.
5. 콘텐츠의 질과 연관성, 진실성을 분명히 한다.
6. 시간 순서가 아닌 공간 순서로 나열한다.
7. 정량적 자료의 정량성을 제거하지 않는다.

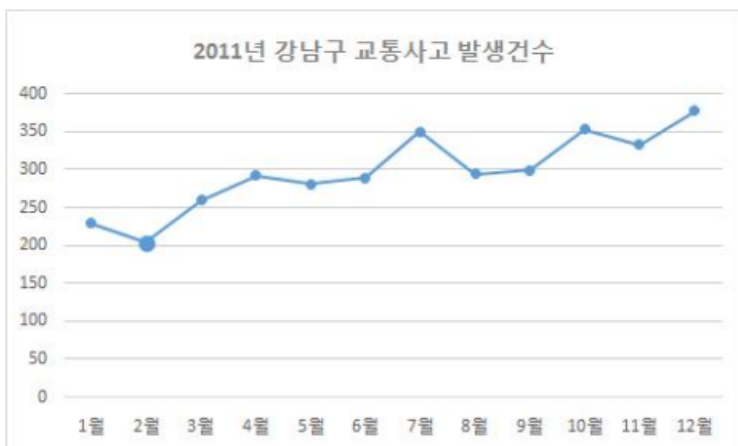
2. 그래프에서 정보 전달 효과를 높이기 위해 수정하거나 강조할 그래픽 요소를 선택하여 전달 요소를 강조한다.

• 그래픽 요소 : 크기, 모양, 색, 밝기(명도), 방향, 질감, 위치

다음 예는 2011년 강남구에서 매월 발생한 사고 건수를 그린 선 그래프이다.

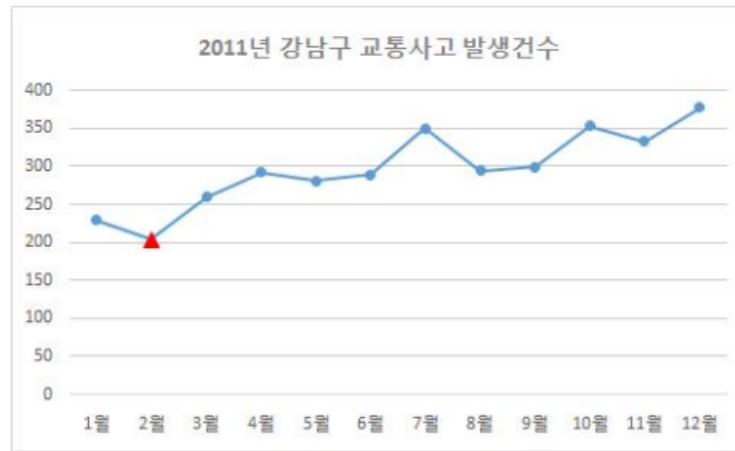


[그림 3-4] 2011년 강남구 교통사고 발생 건수에 대한 선 그래프



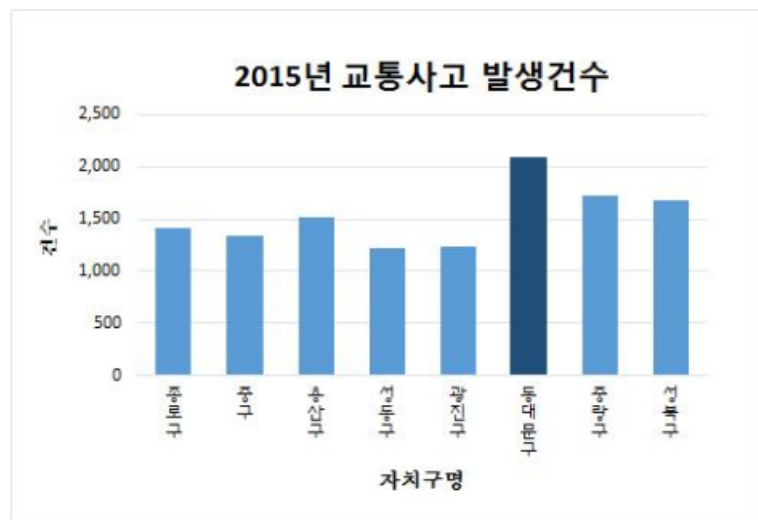
[그림 3-5] 사례 1- 2011년 2월 데이터 요소의 원을 다른 요소보다 크게 강조

3. 색과 모양을 이용하여 강조한다.



[그림 3-7] 사례 3-2011년 2월 데이터 요소 삼각형의 색상을 변경하여 강조

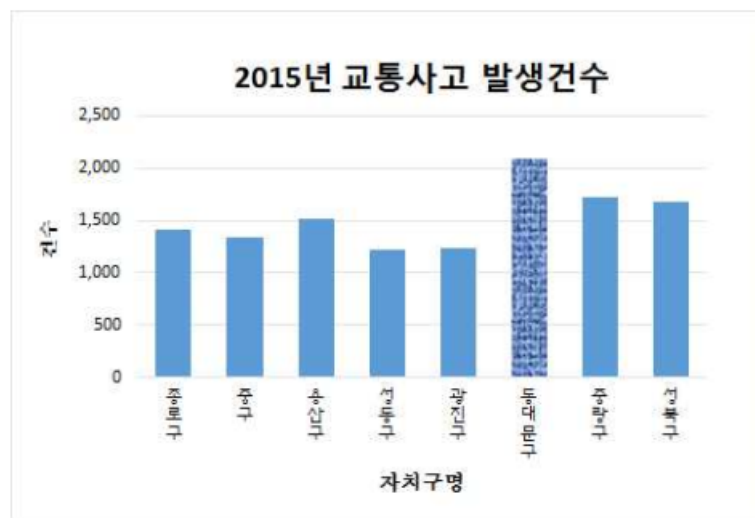
4. 밝기(명도 lightness/value)를 이용하여 전달 요소를 강조한다.



[그림 3-8] 사례 4-동대문구 막대 명도를 높여 강조

5. 질감을 이용하여 전달 요소를 강조한다.

사례 5. 2015년 7개 자치구 중 동대문구 막대만 다른 질감을 주어 강조하였다.



[그림 3-9] 사례 5-동대문구 막대를 질감을 주어 강조함

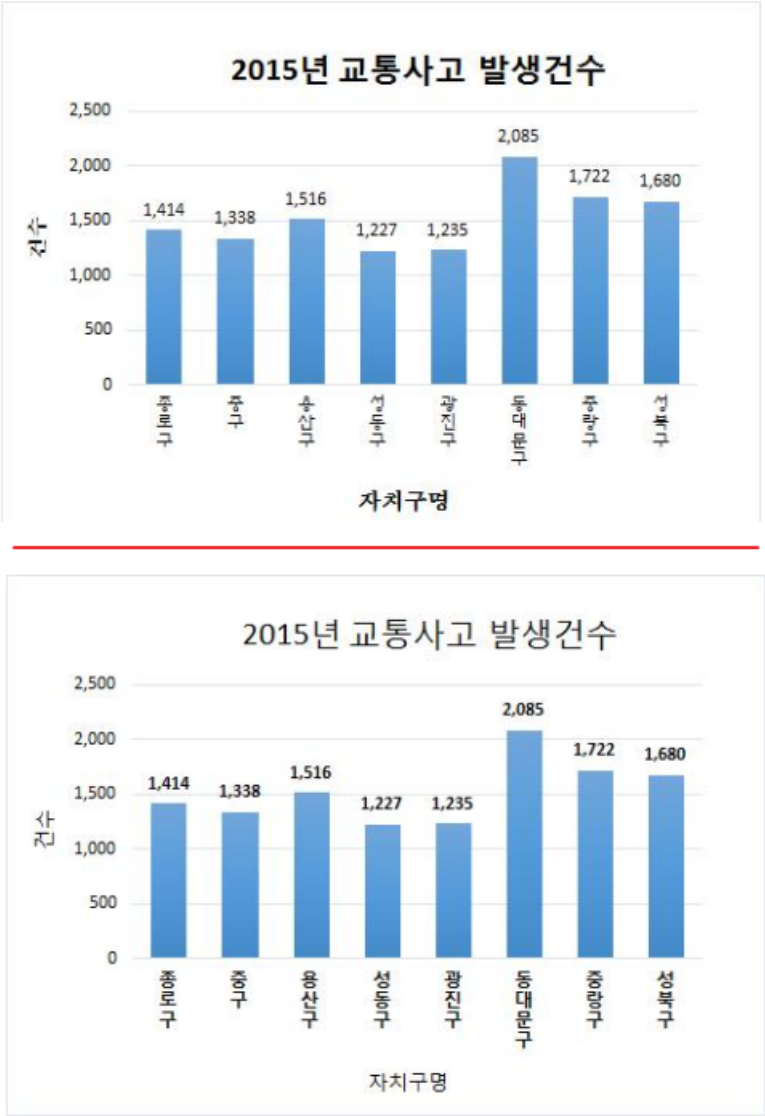
[3] 타이포그래피, 그리드, 색상, 아이소타이프 등의 그래픽 디자인 원리 적용

1. 타이포그래피 통해 그래프 품질 향상

이 가이드 요소에 대한 서체, 글자의 무게,
크기, 색채, 스타일, 간격 등을 조정하여 그래프의 품질을 향상

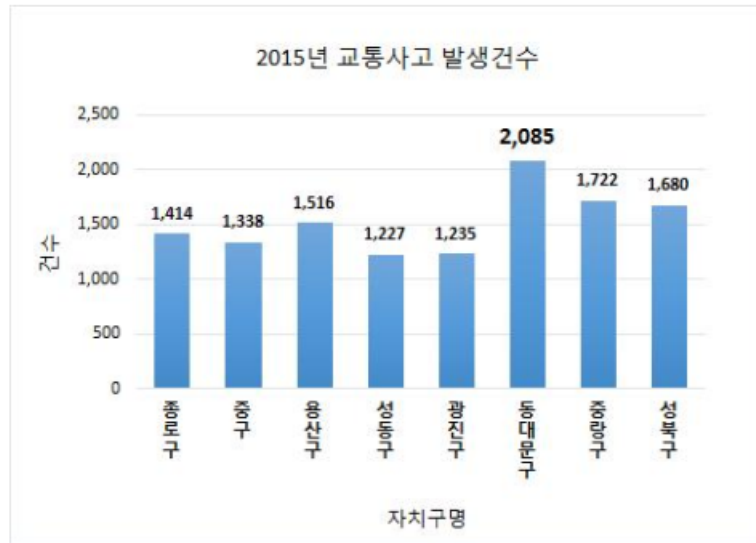
이 막대 그래프에서 X축과, Y축의 축 제목, 데이터 값에 대한 레이블, 그래프
제목 등의 가이드 요소별로 서체와 글자 크기, 굵기가 다르다

서체와 무게

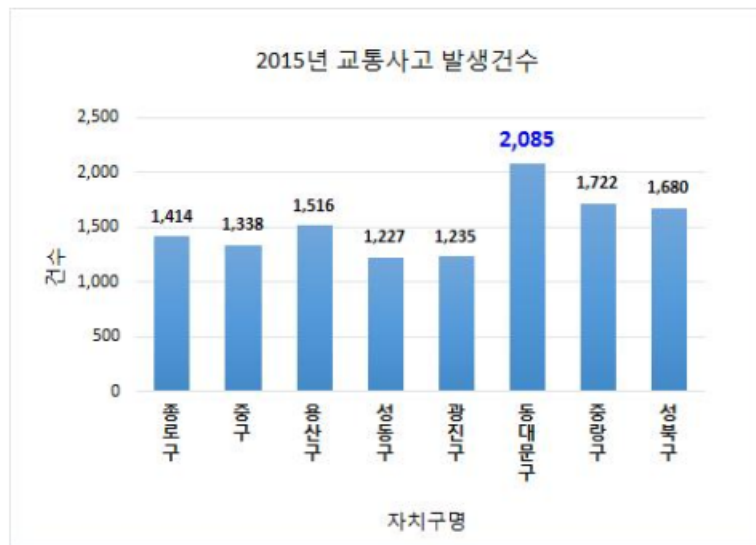


[그림 3-12] 사례 7-축 제목과 차트 제목의 글자 굵기는 낮게, X축 항목과 데이터레이블은 굵게

크기와 색채



[그림 3-13] 사례 8-그래프 제목과 데이터 레이블의 글자의 크기 변경



[그림 3-14] 사례 9-감조하려는 데이터의 레이블 글자 색상 변경

스타일과 간격

2. 색상을 통해 그래프 품질을 향상한다

(1) 구분 표현

정보를 분류하거나 묶기 위해 구분이 색상들을 선택하여 표현한다.

(2) 순서 표현

정보의 순서 등 위계를 표현하는 경우 색의 채도나 색의 스펙트럼 변화를 이용한다.

색상 팔레트를 활용하면 좀 더 쉽게 색상을 통한 순서상 표현을 할 수 있다,

(3) 비율 표현

비율은 연속된 크기순으로 표시하되 0을 중심으로 명도를 단계별로 변화를 주어 표시하고, 위와 아래의 작은 수치들은 보색 대비가 되는 두 가지 색을 선택하여 사용한다. 만약 회색을 기준 색으로 하여 0으로 두고 빨간색이 늘어나면 부정적 수치 증가하는 것이고, 녹색이 늘어나면 긍정적 수치가 증가하는 것으로 표현할 수 있다

<https://www.theguardian.com/news/datablog/2010/may/17/uk-public-spending-departments-money-cuts>
 (https://www.theguardian.com/news/datablog/2010/may/17/uk-public-spending-departments-money-cuts)

2008년과 2009년 공공 부문에 대한 영국 정부 비용을 부처별로 쪼개어 보여준 것이다.
 즉, 정량적인 의미나 범주 구분을 위해 색상을 사용하지 않았다.

장식적인 기능으로 색상을 사용하여 매력적인 시각화로 만듦.

3. 그리드를 통해 그래프 품질을 향상한다.

<참조-Reference>

https://bioboot.github.io/bgggn213_f17/class-material/08_draw_lecture_plots2/ (https://bioboot.github.io/bgggn213_f17/class-material/08_draw_lecture_plots2/)