Anscombe's Quartet

작업환경 정리

- 현재 작업디렉토리 찾아보기

```
getwd()
```

```
## [1] "/Users/coop2711/Dropbox/works/class/Stat_Graphics/R.WD"
```

- 검색가능한 package 와 data 열거

```
search()
```

```
##  [1] ".GlobalEnv"        "package:knitr"     "package:stats"
##  [4] "package:graphics"  "package:grDevices" "package:utils"
##  [7] "package:datasets"  "KoreaEnv"          "package:methods"
## [10] "Autoloads"         "package:base"
```

- anscombe quartet 자료가 들어있는 ggplot2 패키지 설치

```
install.packages("ggplot2")
```

```
## Error in contrib.url(repos, "source"): trying to use CRAN without setting a mirror
```

- ggplot2 패키지를 search 가능한 library 에 탑재

```
search()
```

```
##  [1] ".GlobalEnv"        "package:knitr"     "package:stats"
##  [4] "package:graphics"  "package:grDevices" "package:utils"
##  [7] "package:datasets"  "KoreaEnv"          "package:methods"
## [10] "Autoloads"         "package:base"
```

- 다시 검색 목록에 올리고,

```
library(ggplot2)
search()
```

```
##  [1] ".GlobalEnv"        "package:ggplot2"   "package:knitr"
##  [4] "package:stats"     "package:graphics"  "package:grDevices"
##  [7] "package:utils"     "package:datasets"  "KoreaEnv"
## [10] "package:methods"   "Autoloads"         "package:base"
```

- 두번째 방법은 검색목록에 등장하는 위치(numbering)을 사용하는 것임.

```
detach(2)
search()
```

```
##  [1] ".GlobalEnv"        "package:knitr"     "package:stats"
##  [4] "package:graphics"  "package:grDevices" "package:utils"
##  [7] "package:datasets"  "KoreaEnv"          "package:methods"
## [10] "Autoloads"         "package:base"
```

- 특정 패키지의 help 문서를 살피기 위한 한두가지 시행 착오 …

```
help(package:methods)
```

```
## Error in help(package:methods): 'topic' should be a name, length-one character vector
or reserved word
```

```
help("package:methods")
```

```
## No documentation for 'package:methods' in specified packages and libraries:
## you could try '??package:methods'
```

- help 문서를 help 로 검색

```
?help
```

- 다음과 같은 문법을 적용해야 함을 알 수 있음.

```
help(package=methods)
```

- ggplot2 패키지에 포함되어 있는 anscombe 자료를 올리기 위하여 data() 함수 help 파일 확인.

```
?data
```

- ggplot2에 어떤 data set이 포함되어 있는지 확인하는 시행착오와 결과

```
data(package=ggplot2)
```

```
## Error in data(package = ggplot2): 객체 'ggplot2'를 찾을 수 없습니다
```

- package 이름은 character 처리하여야 함을 확인.

```
library("ggplot2")
```

- library 가 어떤 기능을 하는 함수인지 알아보려면? 특히 패키지 이름을 어떻게 표기하는지 유의

```
?library
```

- ggplot2 를 library 에서 내리려면 어떻게 해야 하는지 시행착오를 좀 거쳐 본다면 …

```
detach(ggplot2)
```

```
## Error in detach(ggplot2): invalid 'name' argument
```

```
detach("ggplot2")
```

```
## Error in detach("ggplot2"): invalid 'name' argument
```

- search()에 나오는 구조에 유의.

```
search()
```

```
##  [1] ".GlobalEnv"        "package:ggplot2"   "package:knitr"
##  [4] "package:stats"     "package:graphics"  "package:grDevices"
##  [7] "package:utils"     "package:datasets"  "KoreaEnv"
## [10] "package:methods"   "Autoloads"         "package:base"
```

```
detach("package:ggplot2")
```

- 검색 목록에서 빠져 나간 것 확인.

```
data(package="ggplot2")
```

- try() 함수의 용례

```
?try
data(package=ggplot2)
```

```
## Error in data(package = ggplot2): 객체 'ggplot2'를 찾을 수 없습니다
```

```
try(data(package=ggplot2))
```

Anscombe 자료의 기초통계 요약

- Anscombe 자료 가져다 붙이기

```
data(anscombe)
```

- 그러나 data() 함수로는 검색 목록에 올라가지 않는다는 것을 확인.

```
search()
```

```
##  [1] ".GlobalEnv"        "package:knitr"     "package:stats"
##  [4] "package:graphics"  "package:grDevices" "package:utils"
##  [7] "package:datasets"  "KoreaEnv"          "package:methods"
## [10] "Autoloads"         "package:base"
```

- anscombe 자료의 구조 확인 후 자료를 실제로 출력

```
str(anscombe)
```

```
## 'data.frame':    11 obs. of  8 variables:
##  $ x1: num  10 8 13 9 11 14 6 4 12 7 ...
##  $ x2: num  10 8 13 9 11 14 6 4 12 7 ...
##  $ x3: num  10 8 13 9 11 14 6 4 12 7 ...
##  $ x4: num  8 8 8 8 8 8 8 19 8 8 ...
##  $ y1: num  8.04 6.95 7.58 8.81 8.33 ...
##  $ y2: num  9.14 8.14 8.74 8.77 9.26 8.1 6.13 3.1 9.13 7.26 ...
##  $ y3: num  7.46 6.77 12.74 7.11 7.81 ...
##  $ y4: num  6.58 5.76 7.71 8.84 8.47 7.04 5.25 12.5 5.56 7.91 ...
```

```
anscombe
```

```
##    x1 x2 x3 x4    y1   y2    y3    y4
## 1  10 10 10  8  8.04 9.14  7.46  6.58
## 2   8  8  8  8  6.95 8.14  6.77  5.76
## 3  13 13 13  8  7.58 8.74 12.74  7.71
## 4   9  9  9  8  8.81 8.77  7.11  8.84
## 5  11 11 11  8  8.33 9.26  7.81  8.47
## 6  14 14 14  8  9.96 8.10  8.84  7.04
## 7   6  6  6  8  7.24 6.13  6.08  5.25
## 8   4  4  4 19  4.26 3.10  5.39 12.50
## 9  12 12 12  8 10.84 9.13  8.15  5.56
## 10  7  7  7  8  4.82 7.26  6.42  7.91
## 11  5  5  5  8  5.68 4.74  5.73  6.89
```

- anscombe 자료의 기초통계 요약. 분산이나 표준편차는 나오지 않음.

```
summary(anscombe)
```

```
##        x1              x2              x3              x4
##  Min.   : 4.0   Min.   : 4.0   Min.   : 4.0   Min.   : 8
##  1st Qu.: 6.5   1st Qu.: 6.5   1st Qu.: 6.5   1st Qu.: 8
##  Median : 9.0   Median : 9.0   Median : 9.0   Median : 8
##  Mean   : 9.0   Mean   : 9.0   Mean   : 9.0   Mean   : 9
##  3rd Qu.:11.5   3rd Qu.:11.5   3rd Qu.:11.5   3rd Qu.: 8
##  Max.   :14.0   Max.   :14.0   Max.   :14.0   Max.   :19
##        y1              y2              y3              y4
##  Min.   : 4.260   Min.   :3.100   Min.   : 5.39   Min.   : 5.250
##  1st Qu.: 6.315   1st Qu.:6.695   1st Qu.: 6.25   1st Qu.: 6.170
##  Median : 7.580   Median :8.140   Median : 7.11   Median : 7.040
##  Mean   : 7.501   Mean   :7.501   Mean   : 7.50   Mean   : 7.501
##  3rd Qu.: 8.570   3rd Qu.:8.950   3rd Qu.: 7.98   3rd Qu.: 8.190
##  Max.   :10.840   Max.   :9.260   Max.   :12.74   Max.   :12.500
```

- 표준편차를 막무가내로 계산하라고 하면 오류 발생.

```
sd(anscombe)
```

```
## Error in is.data.frame(x): (리스트) 객체는 유형 'double'로 강제형변환 될 수 없습니다
```

- anscombe 자료의 구조로 인하여 apply() 함수 적용

```
apply(anscombe,2,sd)
```

```
##       x1       x2       x3       x4       y1       y2       y3       y4
## 3.316625 3.316625 3.316625 3.316625 2.031568 2.031657 2.030424 2.030579
```

- 피어슨 상관계수는 행렬구조(사실은 data.frame)에서 각 변수 간의 상관계수 계산에 적합

```
cor(anscombe)
```

```
##            x1         x2         x3         x4         y1         y2
## x1  1.0000000  1.0000000  1.0000000 -0.5000000  0.8164205  0.8162365
## x2  1.0000000  1.0000000  1.0000000 -0.5000000  0.8164205  0.8162365
## x3  1.0000000  1.0000000  1.0000000 -0.5000000  0.8164205  0.8162365
## x4 -0.5000000 -0.5000000 -0.5000000  1.0000000 -0.5290927 -0.7184365
## y1  0.8164205  0.8164205  0.8164205 -0.5290927  1.0000000  0.7500054
## y2  0.8162365  0.8162365  0.8162365 -0.7184365  0.7500054  1.0000000
## y3  0.8162867  0.8162867  0.8162867 -0.3446610  0.4687167  0.5879193
## y4 -0.3140467 -0.3140467 -0.3140467  0.8165214 -0.4891162 -0.4780949
##            y3         y4
## x1  0.8162867 -0.3140467
## x2  0.8162867 -0.3140467
## x3  0.8162867 -0.3140467
## x4 -0.3446610  0.8165214
## y1  0.4687167 -0.4891162
## y2  0.5879193 -0.4780949
## y3  1.0000000 -0.1554718
## y4 -0.1554718  1.0000000
```

- (x1, y1), (x2, y2), (x3, y3), (x4, y4) 간의 상관계수를 보기 쉽게 재배열. [,]의 용도에 유의

```
cor(anscombe[c(1,5,2,6,3,7,4,8)])
```

```
##           x1         y1         x2         y2         x3         y3
## x1  1.0000000  0.8164205  1.0000000  0.8162365  1.0000000  0.8162867
## y1  0.8164205  1.0000000  0.8164205  0.7500054  0.8164205  0.4687167
## x2  1.0000000  0.8164205  1.0000000  0.8162365  1.0000000  0.8162867
## y2  0.8162365  0.7500054  0.8162365  1.0000000  0.8162365  0.5879193
## x3  1.0000000  0.8164205  1.0000000  0.8162365  1.0000000  0.8162867
## y3  0.8162867  0.4687167  0.8162867  0.5879193  0.8162867  1.0000000
## x4 -0.5000000 -0.5290927 -0.5000000 -0.7184365 -0.5000000 -0.3446610
## y4 -0.3140467 -0.4891162 -0.3140467 -0.4780949 -0.3140467 -0.1554718
##           x4         y4
## x1 -0.5000000 -0.3140467
## y1 -0.5290927 -0.4891162
## x2 -0.5000000 -0.3140467
## y2 -0.7184365 -0.4780949
## x3 -0.5000000 -0.3140467
## y3 -0.3446610 -0.1554718
## x4  1.0000000  0.8165214
## y4  0.8165214  1.0000000
```

- 배열을 저장

```
a<-c(1,5,2,6,3,7,4,8)
```

- 평균과 표준편차 계산

```
apply(anscombe,2,mean)
```

```
##       x1       x2       x3       x4       y1       y2       y3       y4
## 9.000000 9.000000 9.000000 9.000000 7.500909 7.500909 7.500000 7.500909
```

```
apply(anscombe,2,sd)
```

- lm() 함수를 이용해서 계산해도 같은 결과

```
lm(y1~x1,data=anscombe)$coefficient
```

```
## (Intercept)          x1
##   3.0000909   0.5000909
```

```
lm(y2~x2,data=anscombe)$coefficient
```

```
## (Intercept)          x2
##    3.000909    0.500000
```

```
lm(y3~x3,data=anscombe)$coefficient
```

```
## (Intercept)          x3
##   3.0024545   0.4997273
```

```
lm(y4~x4,data=anscombe)$coefficient
```

```
## (Intercept)          x4
##   3.0017273   0.4999091
```

그러나 그림으로 비교하면?

- 산점도와 회귀선을 그려서 비교해 보자. 우선 모든 수직축과 수평축의 범위를 정하자.

```
##       x1       x2       x3       x4       y1       y2       y3       y4
## 3.316625 3.316625 3.316625 3.316625 2.031568 2.031657 2.030424 2.030579
```

- 번잡함을 덜기 이해 attach() 이용

```
attach(anscombe)
```

- 선형회귀계수도 비교

```
lsfit(x1,y1)$coefficient
```

```
## Intercept         X
## 3.0000909 0.5000909
```

```
lsfit(x2,y2)$coefficient
```

```
## Intercept         X
##  3.000909  0.500000
```

```
lsfit(x3,y3)$coefficient
```

```
## Intercept         X
## 3.0024545 0.4997273
```

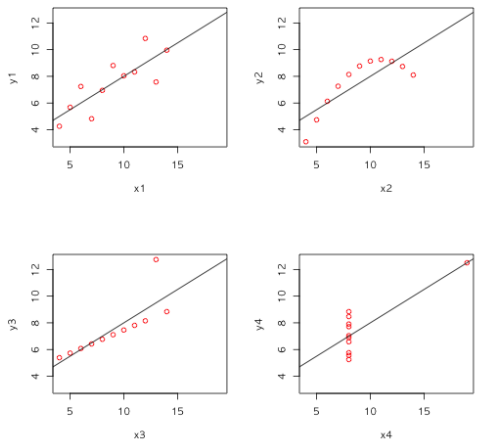
```
lsfit(x4,y4)$coefficient
```

```
## Intercept         X
## 3.0017273 0.4999091
```

```
x.min<-min(x1,x2,x3,x4)
y.min<-min(y1,y2,y3,y4)
y.max<-max(y1,y2,y3,y4)
x.max<-max(x1,x2,x3,x4)
```

- 한 장에 네개의 산점도를 그리기 위하여 par() 조정 후 작업. 점은 붉은 색으로, 회귀선은 최소제곱법 적용.

```
par(mfrow=c(2,2))
plot(x1,y1,xlim=c(x.min,x.max),ylim=c(y.min,y.max),col="red")
abline(lsfit(x1,y1))
plot(x2,y2,xlim=c(x.min,x.max),ylim=c(y.min,y.max),col="red")
abline(lsfit(x2,y2))
plot(x3,y3,xlim=c(x.min,x.max),ylim=c(y.min,y.max),col="red")
abline(lsfit(x3,y3))
plot(x4,y4,xlim=c(x.min,x.max),ylim=c(y.min,y.max),col="red")
abline(lsfit(x4,y4))
```



- png() 함수를 이용한 출력에는 다음 코드 필요

```
png(filename="anscombe.png",width=640,height=640)
dev.off()
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 4 4
## [36] 4 4 4 4 4 4 4 4 4 4
## Levels: 1 2 3 4
```

```
anscombe.long<-data.frame(x=c(x1,x2,x3,x4),y=c(y1,y2,y3,y4),group=a.levels)
anscombe.long
```

```
##     x     y group
## 1  10  8.04     1
## 2   8  6.95     1
## 3  13  7.58     1
## 4   9  8.81     1
## 5  11  8.33     1
## 6  14  9.96     1
## 7   6  7.24     1
## 8   4  4.26     1
## 9  12 10.84     1
## 10  7  4.82     1
## 11  5  5.68     1
## 12 10  9.14     2
## 13  8  8.14     2
## 14 13  8.74     2
## 15  9  8.77     2
## 16 11  9.26     2
## 17 14  8.10     2
## 18  6  6.13     2
## 19  4  3.10     2
## 20 12  9.13     2
## 21  7  7.26     2
## 22  5  4.74     2
## 23 10  7.46     3
## 24  8  6.77     3
## 25 13 12.74     3
## 26  9  7.11     3
## 27 11  7.81     3
## 28 14  8.84     3
## 29  6  6.08     3
```

```
## quartz_off_screen
##                 2
```

qplot()과 ggplot()을 이용한 그림 작성

- ggplot2 를 가져다 붙여놓기.

```
library("ggplot2")
search()
```

```
##  [1] ".GlobalEnv"        "package:ggplot2"   "anscombe"
##  [4] "package:knitr"     "package:stats"     "package:graphics"
##  [7] "package:grDevices" "package:utils"     "package:datasets"
## [10] "KoreaEnv"          "package:methods"   "Autoloads"
## [13] "package:base"
```

- qplot()이나 ggplot()을 이용하려면 anscombe 데이터프레임을 long format 으로 바꿔야 함.
- 바꿔주기 위해서는 각 그룹을 구분하는 factor를 생성해야 함.

```
nrow(anscombe)
```

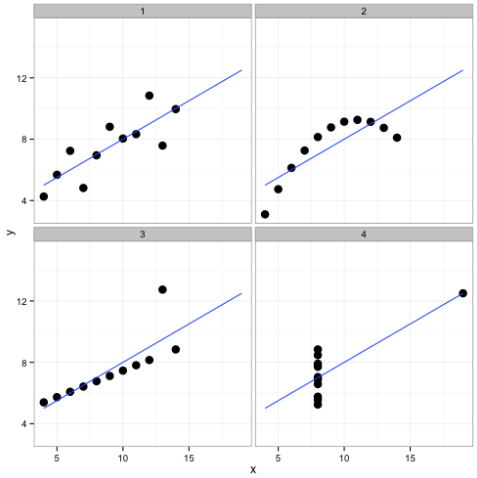
```
## [1] 11
```

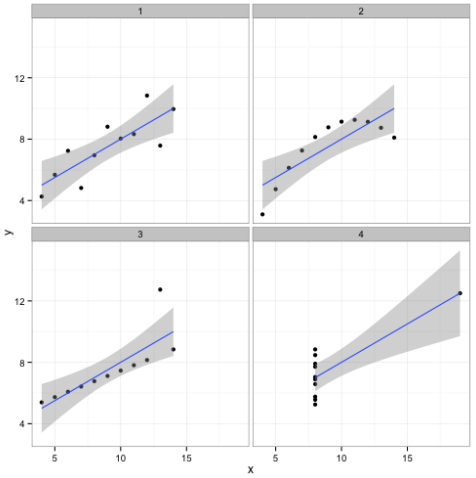
```
?gl
a.levels<-gl(4,nrow(anscombe))
a.levels
```

```
## 30  4  5.39     3
## 31 12  8.15     3
## 32  7  6.42     3
## 33  5  5.73     3
## 34  8  6.58     4
## 35  8  5.76     4
## 36  8  7.71     4
## 37  8  8.84     4
## 38  8  8.47     4
## 39  8  7.04     4
## 40  8  5.25     4
## 41 19 12.50     4
## 42  8  5.56     4
## 43  8  7.91     4
## 44  8  6.89     4
```

- ggplot() 으로 그리는 R 코드

```
theme_set(theme_bw())
ggplot(anscombe.long,aes(x,y))+
  geom_point(size=4)+
  geom_smooth(method="lm",fill=NA,fullrange=TRUE)+
  facet_wrap(~group)
```





- qplot() 으로 그리기. facet_wrap() 활용에 유의.

```
a1.qplot<-qplot(x,y, data=anscombe.long,geom=c("point","smooth"),method="lm")
a1.qplot+facet_wrap(~group,ncol=2)
```