Anscombe's Quartet Revision

- ggplot2 패키지 설치 및 `library` 탑재

```
# install.packages("ggplot2", repos="http://cran.rstudio.com/")
library(ggplot2)
```

- ggplot2 패키지의 documentation 검색

```
help(package = ggplot2)
```

- anscombe quartet 자료가 들어있는 `datasets` 패키지의 자료 목록 검색

```
data(package = "datasets")
```

Anscombe 자료 가져다 붙이기

```
data(anscombe)
```

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

```
search()
```

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

- anscombe 자료의 구조 확인

```
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 ...
```

- 자료의 일부와 전체 출력

```
head(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
```

```
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
```

```
x1 <- anscombe$x1
x2 <- anscombe$x2
x3 <- anscombe$x3
x4 <- anscombe$x4
y1 <- anscombe$y1
y2 <- anscombe$y2
y3 <- anscombe$y3
y4 <- anscombe$y4
```

Ancombe 자료의 기초통계 요약

- 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
```

- `apply` 함수를 이용하여 `anscombe` data frame을 구성하는 각 벡터의 `sd` 계산.

  `old.par` 의 기능과 `options(digits = 3)` 를 하지 않았을 때 어떤 출력 결과물들이 나올지 상상.

```
old.par <- par(no.readonly = TRUE)
options(digits = 3)
apply(anscombe, MARGIN = 2, FUN = sd)
```

```
##   x1   x2   x3   x4   y1   y2   y3   y4
## 3.32 3.32 3.32 3.32 2.03 2.03 2.03 2.03
```

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

```
cor(anscombe)
```

```
##         x1     x2     x3     x4     y1     y2     y3     y4
## x1   1.000  1.000  1.000 -0.500  0.816  0.816  0.816 -0.314
## x2   1.000  1.000  1.000 -0.500  0.816  0.816  0.816 -0.314
## x3   1.000  1.000  1.000 -0.500  0.816  0.816  0.816 -0.314
## x4  -0.500 -0.500 -0.500  1.000 -0.529 -0.718 -0.345  0.817
## y1   0.816  0.816  0.816 -0.529  1.000  0.750  0.469 -0.489
## y2   0.816  0.816  0.816 -0.718  0.750  1.000  0.588 -0.478
## y3   0.816  0.816  0.816 -0.345  0.469  0.588  1.000 -0.155
## y4  -0.314 -0.314 -0.314  0.817 -0.489 -0.478 -0.155  1.000
```

- `(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     x4     y4
## x1   1.000  0.816  1.000  0.816  1.000  0.816 -0.500 -0.314
## y1   0.816  1.000  0.816  0.750  0.816  0.469 -0.529 -0.489
## x2   1.000  0.816  1.000  0.816  1.000  0.816 -0.500 -0.314
## y2   0.816  0.750  0.816  1.000  0.816  0.588 -0.718 -0.478
## x3   1.000  0.816  1.000  0.816  1.000  0.816 -0.500 -0.314
## y3   0.816  0.469  0.816  0.588  0.816  1.000 -0.345 -0.155
## x4  -0.500 -0.529 -0.500 -0.718 -0.500 -0.345  1.000  0.817
## y4  -0.314 -0.489 -0.314 -0.478 -0.314 -0.155  0.817  1.000
```

- 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
## 9.0 9.0 9.0 9.0 7.5 7.5 7.5 7.5
```

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

```
##   x1   x2   x3   x4   y1   y2   y3   y4
## 3.32 3.32 3.32 3.32 2.03 2.03 2.03 2.03
```

- 최소제곱법으로 추정한 회귀계수도 비교

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

```
## Intercept         X
##       3.0       0.5
```

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

```
## Intercept         X
##       3.0       0.5
```

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

```
## Intercept         X
##       3.0       0.5
```

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

```
## Intercept         X
##       3.0       0.5
```

- lm() 함수를 이용해서 선형모형으로 적합해도 같은 결과

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

```
## (Intercept)          x1
##         3.0         0.5
```

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

```
## (Intercept)          x2
##         3.0         0.5
```

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

```
## (Intercept)          x3
##         3.0         0.5
```

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

```
## (Intercept)          x4
##         3.0         0.5
```

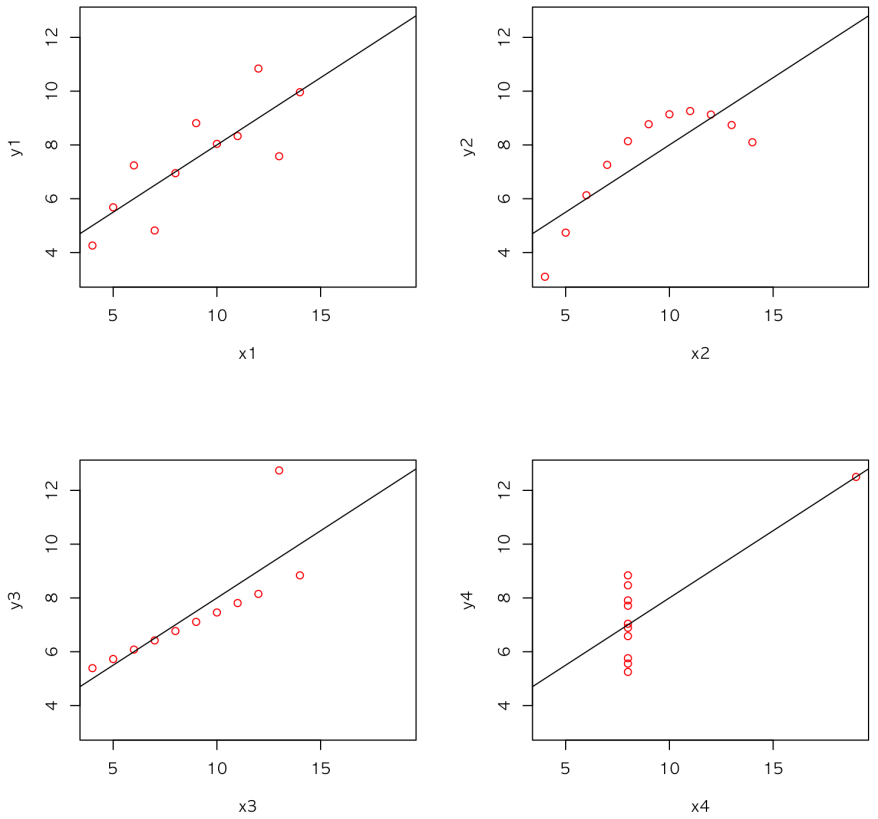
그러나 그림으로 비교하면?

산점도와 회귀선

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

- 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))
```



qplot()과 ggplot()을 이용한 그림 작성

- `anscombe` 을 long format 으로
- 각 그룹을 구분하는 `factor` 를 생성해야 함.

```
nrow(anscombe)
```

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

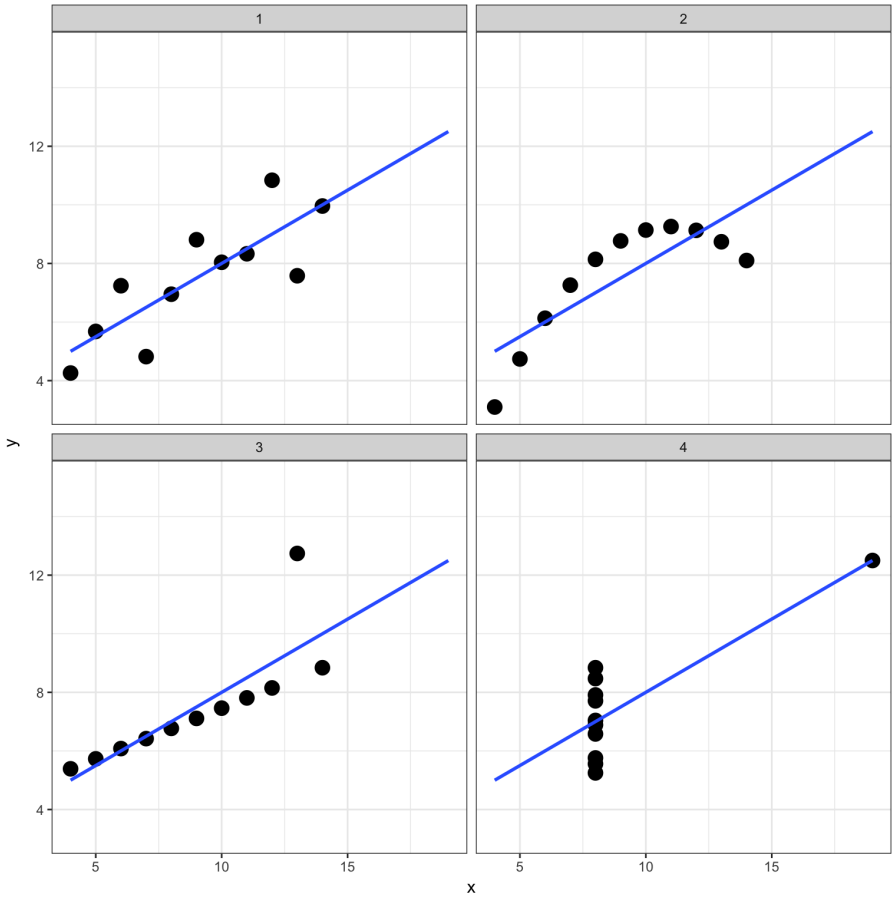
```
a_levels <- gl(4, nrow(anscombe))
a_levels
```

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

```
anscombe_long <- data.frame(x = c(x1, x2, x3, x4),
                            y = c(y1, y2, y3, y4),
                            group = a_levels)
```

ggplot() 으로 그리는 R 코드

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

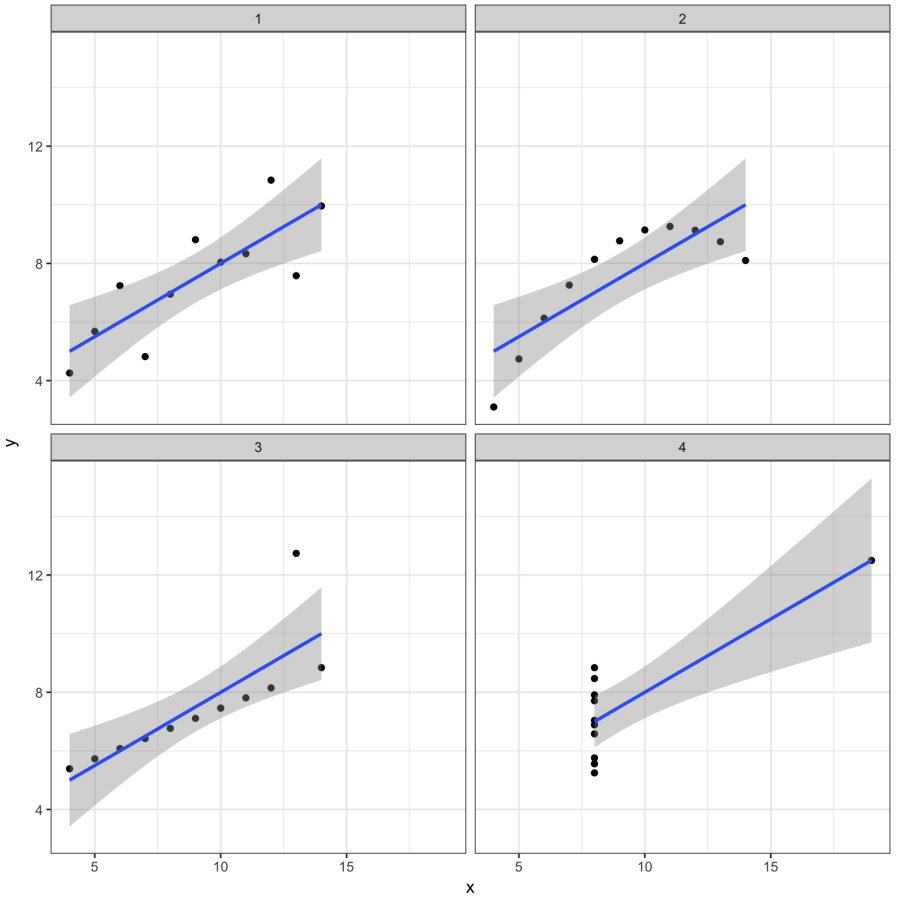


qplot() 으로 그리기. facet_wrap() 활용에 유의.

```
a1_qplot <- qplot(x, y,
                  data = anscombe_long,
                  geom = c("point", "smooth"),
                  method = "lm")
```

```
## Warning: Ignoring unknown parameters: method
```

```
a1_qplot + facet_wrap(~ group, ncol = 2)
```



Save

*작업 디렉토리에 생성된 오브젝트들의 이미지를 파일로 저장

```
par(old.par)
save.image(file = "Anscombe.RData")
```