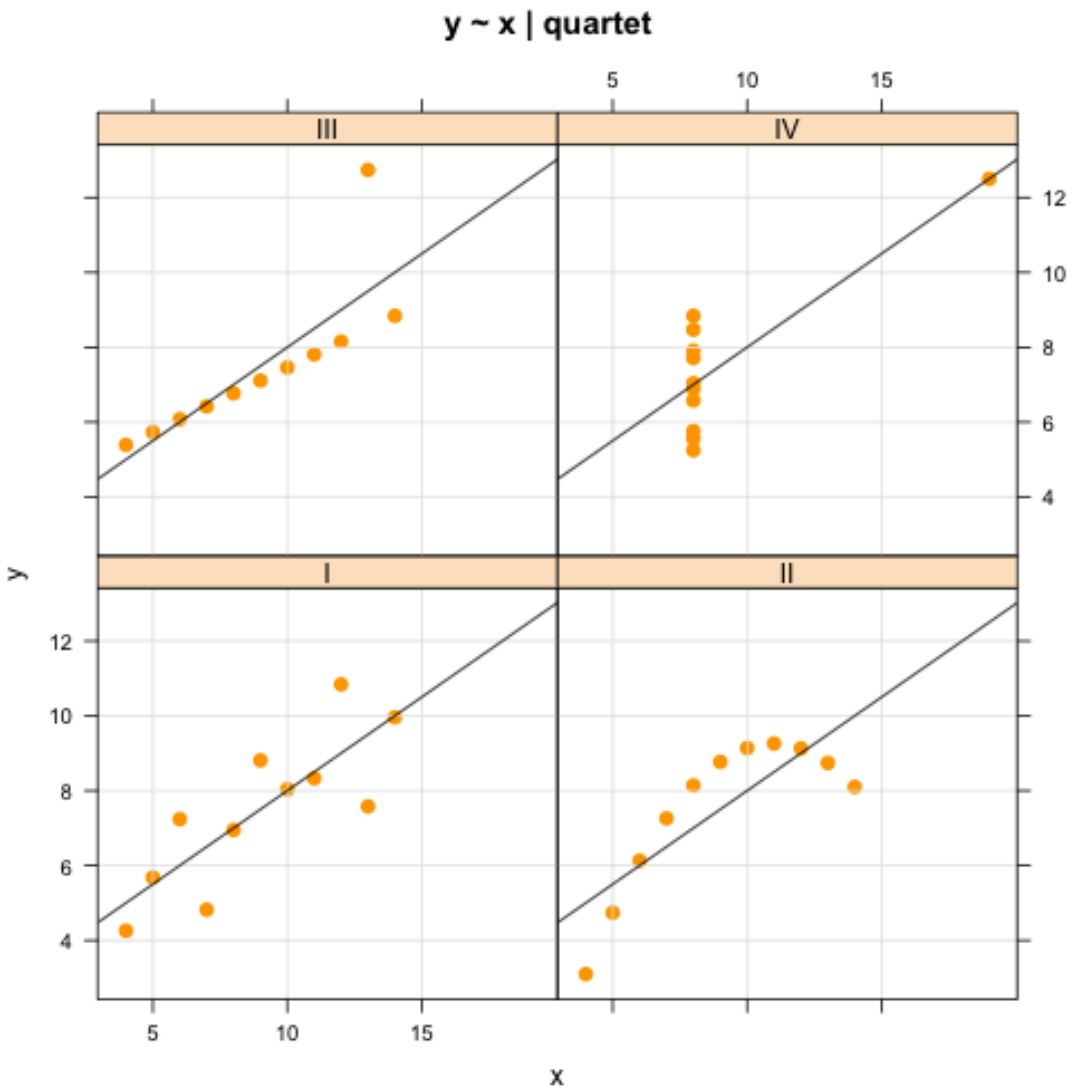
Visualization

Useful Plots

유 충현
블로그 모음 14탄(http://blog.naver.com/bdboys) • 2011년 8월 31일



Visualization

자료를 분석함에 있어서 가장 중요한 것 중에 하나는 자료의 특성을 파악하는 것이다. 자료의 특성을 파악하는 방법에 통계량을 구하여 파악하는 방법이 있겠으나 가장 효과적인 것은 그래프를 그려보는 것이다.

R이 통계분석 도구로서 장점 중에 하나는 다양한 그래프 출력이 가능한 것이다. 이것은 필자가 R을 즐겨 사용하는 이유 중에 하나다. "잘 그린 그래프 하나가 열 통계량 부럽지 않다." 라는 것도 필자의 신념이기도 하다.

R의 데이터 객체 중에 anscombe라는 이름의 데이터 프레임이 있다. 이 데이터는 Anscombe, Francis J.가 1973년 *American Statistician*에 실은 "Graphs in statistical analysis"라는 논문에서 사용된 데이터다. 이 데이터는 논문에서 단순선형회귀분석(Simple Linear Regression)에서 사용되는 네 쌍의 독립변수와 종속변수인 (x_i, y_i), 즉 $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$ 데이터를 포함하고 있다. 이 데이터는 가상의 데이터로 통계량과 회귀식은 동일하게 계산되지만 전혀 이질적인 데이터로 통계량보다는 그래프가 더 정확히 데이터의 특성을 설명할 수 있다는 것을 보여주는 사례라 할 수 있다.

이 데이터로 단순선형회귀분석을 수행해 보자.

1. 통계량 구하기

```
> dim(anscombe)
[1] 11  8
> names(anscombe)
[1] "x1" "x2" "x3" "x4" "y1" "y2" "y3" "y4"
> 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
>
> # (1) Statistic
> colMeans(anscombe)
      x1       x2       x3       x4       y1       y2
9.000000 9.000000 9.000000 9.000000 7.500909 7.500909
      y3       y4
7.500000 7.500909
```

```
> apply(anscombe, 2, var)
       x1        x2        x3        x4        y1
11.000000 11.000000 11.000000 11.000000  4.127269
       y2        y3        y4
 4.127629  4.122620  4.123249
>
> # (2) correlation
> attach(anscombe)
> apply(t(1:4), 2, function(x)
+  cor(get(paste("x",x,sep="")),get(paste("y",x,sep=""))))
[1] 0.8164205 0.8162365 0.8162867 0.8165214
> detach(anscombe)
```

먼저 통계량을 구해 보았다. x1, x2, x3, x4의 평균과 분산은 각각 9와 11로, y1, y2, y3, y4의 평균과 분산은 각각 7.5와 4.12로 동일하거나 거의 동일하게 계산되어졌다. $(x_1,y_1),(x_2,y_2),(x_3,y_3),(x_4,y_4)$에 대한 상관계수도 0.816으로 거의 동일하게 계산되었다. 이것으로 (x_i,y_i)가 양의 상관관계가 있음을 알았다.

2. 데이터 변환

분석의 용이함을 위해서 데이터를 변경해 본다.

```
> tmp <- unlist(as(anscombe, "vector"))
> x <- tmp[1:(NROW(anscombe)*4)]
> y <- tmp[(NROW(anscombe)*4+1):(NROW(anscombe)*8)]
> quartet <- rep(c("I","II","III","IV"),
+  each=NROW(anscombe))
> anscombe.dat <- data.frame(x, y, quartet)
> head(anscombe.dat)
   x    y quartet
1 10 8.04       I
2  8 6.95       I
3 13 7.58       I
4  9 8.81       I
5 11 8.33       I
6 14 9.96       I
> tail(anscombe.dat)
    x     y quartet
39  8  7.04      IV
```

```
40  8  5.25      IV
41 19 12.50      IV
42  8  5.56      IV
43  8  7.91      IV
44  8  6.89      IV
```

3. 회귀분석의 수행

다음처럼 회귀분석을 수행한다.

```
> out <-
+ apply(t(c("I","II","III","IV")), 2,
+  function(key) {
+    lm.obj <- lm(y~x, subset=quartet==key,
+      data=anscombe.dat)
+    round(c(lm.obj$coef["(Intercept)"],
+    lm.obj$coef["x"],
+    anova(lm.obj)["x", "Sum Sq"],
+    anova(lm.obj)["Residuals", "Sum Sq"],
+    summary(lm.obj)$coefficients["x", "Std. Error"],
+    summary(lm.obj)$r.squared),2)})
> dimnames(out) <-
+  list(c("Coefficient Intercept","Coefficient x",
+  "Regression sum of squares","Residuals sum of squares",
+  "Estimated standard error of b1", "Multiple R-Square"),
+  c("I","II","III","IV"))
> out
                                    I    II   III    IV
Coefficient Intercept            3.00  3.00  3.00  3.00
Coefficient x                    0.50  0.50  0.50  0.50
Regression sum of squares       27.51 27.50 27.47 27.49
Residuals sum of squares        13.76 13.78 13.76 13.74
Estimated standard error of b1   0.12  0.12  0.12  0.12
Multiple R-Square                0.67  0.67  0.67  0.67
>
```

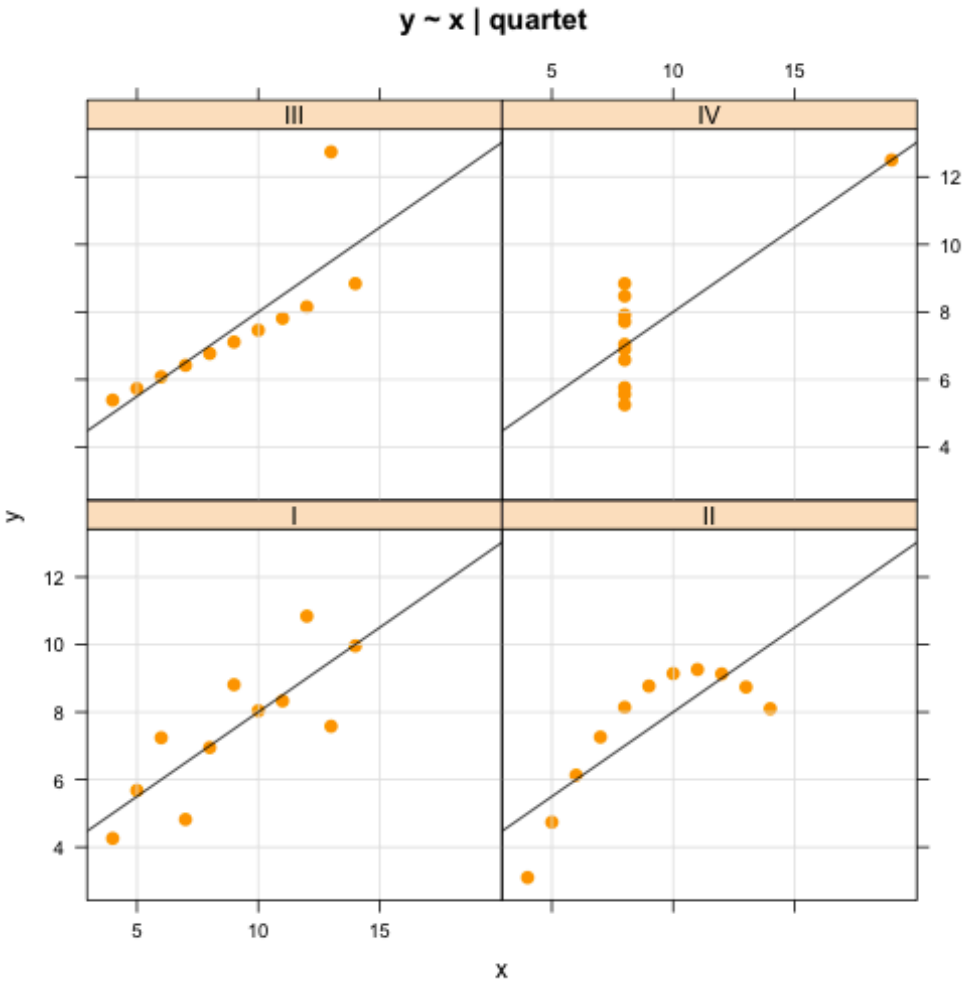
회귀분석의 결과를 보면 네 가지 데이터셋에 대한 단순선형회귀분석의 회귀계수가 동일함을 알 수 있다. 즉, $\hat{y} = 0.5x + 3$의 회귀식이 구해졌고 결정계수도 0.67로 동일하다.

이상의 결과를 보자면 네 가지 데이터는 회귀식도 동일하고 통계량도 거의 동일함을 알 수 있다. 과연 그럴까?

4. 그래프 그리기

이번에는 네 쌍의 데이터의 산점도와 회귀직선을 같은 좌표 상에 그려 보자. 여기서는 R의 lattice 라이브러리를 이용해서 Trellis 그래프로 그려 보기로 한다.

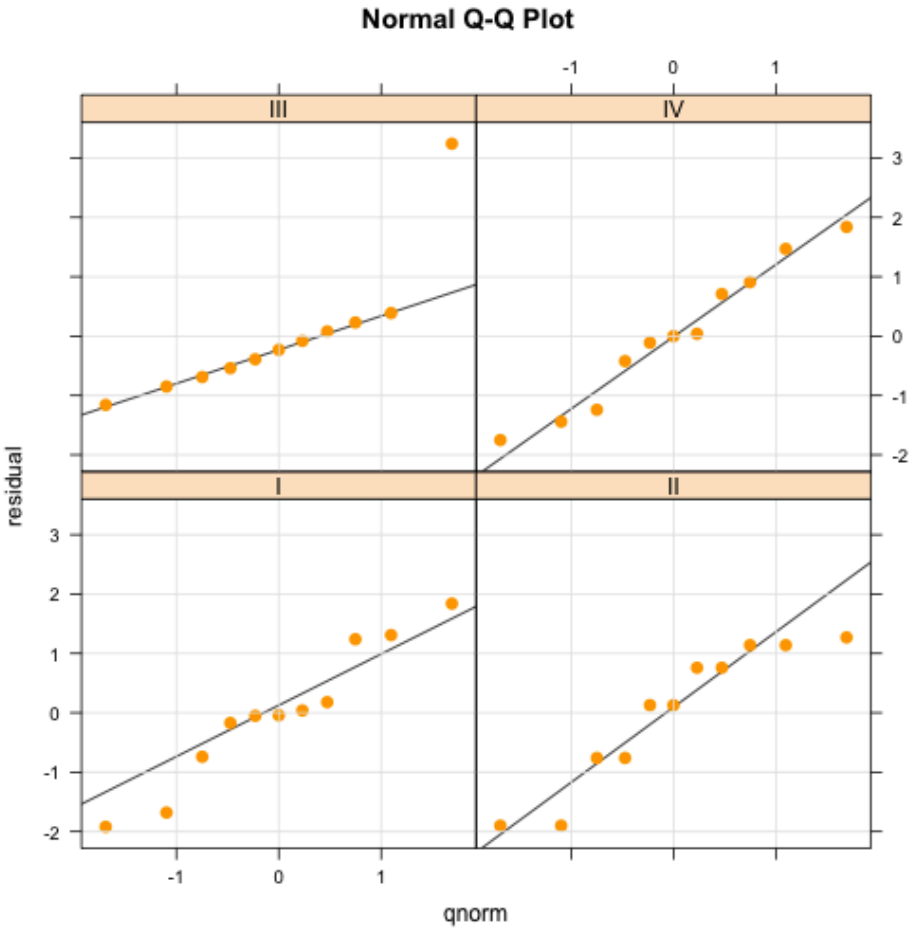
```
> # (4) Chart
> library(lattice)
> xyplot(y ~ x | quartet, data=anscombe.dat,
+  panel = function(x, y, ...) {
+    panel.xyplot(x, y, col="orange", pch=16, cex=1.1)
+    panel.xyplot(x, y, type="g")
+    panel.lmline(x, y, ...)
+  },
+  main="y ~ x | quartet")

>
```



통계량도 동일하고 계산된 회귀식도 동일하지만 출력된 그래프의 결과를 보면 전혀 다른 유형의 데이터임을 알 수 있다. 이것은 그래프를 그려 보지 않으면 네 쌍의 데이터에 대해서 동일한 해석을 할 수도 있음을 의미한다.

이번에는 네 개의 회귀모형에 대해서 회귀분석의 오차의 정규성의 가정의 잔차의 정규확률 그래프를 통해서 타당성 여부를 판단해 본다.

```
> # (5) Residuals
> resi <- apply(t(c("I","II","III","IV")), 2,
+  function(key) lm(y~x, subset=quartet==key,
+    data=anscombe.dat)$resi)
> anscombe.dat <- data.frame(anscombe.dat,
+ residual=as.vector(resi))
> qqmath(~ residual | quartet, data=anscombe.dat,
+       panel = function(x, ...) {
+          panel.qqmathline(x, ...)
+          panel.qqmath(x, col="orange", pch=16, cex=1.1)
+          panel.qqmath(x, type="g")
+       },
+  main="Normal Q-Q Plot")
>
```



회귀모형의 진단을 위한 등분산성의 가정도 이와 같은 Trellis 그래프로 확인할 수도 있으며 회귀모형에 대한 R의 lm 객체를 plot() 함수를 이용해서 검증할 수도 있다. 물론 이 경우에 R은 내부적으로 plot.lm() 함수를 사용할 것이다.

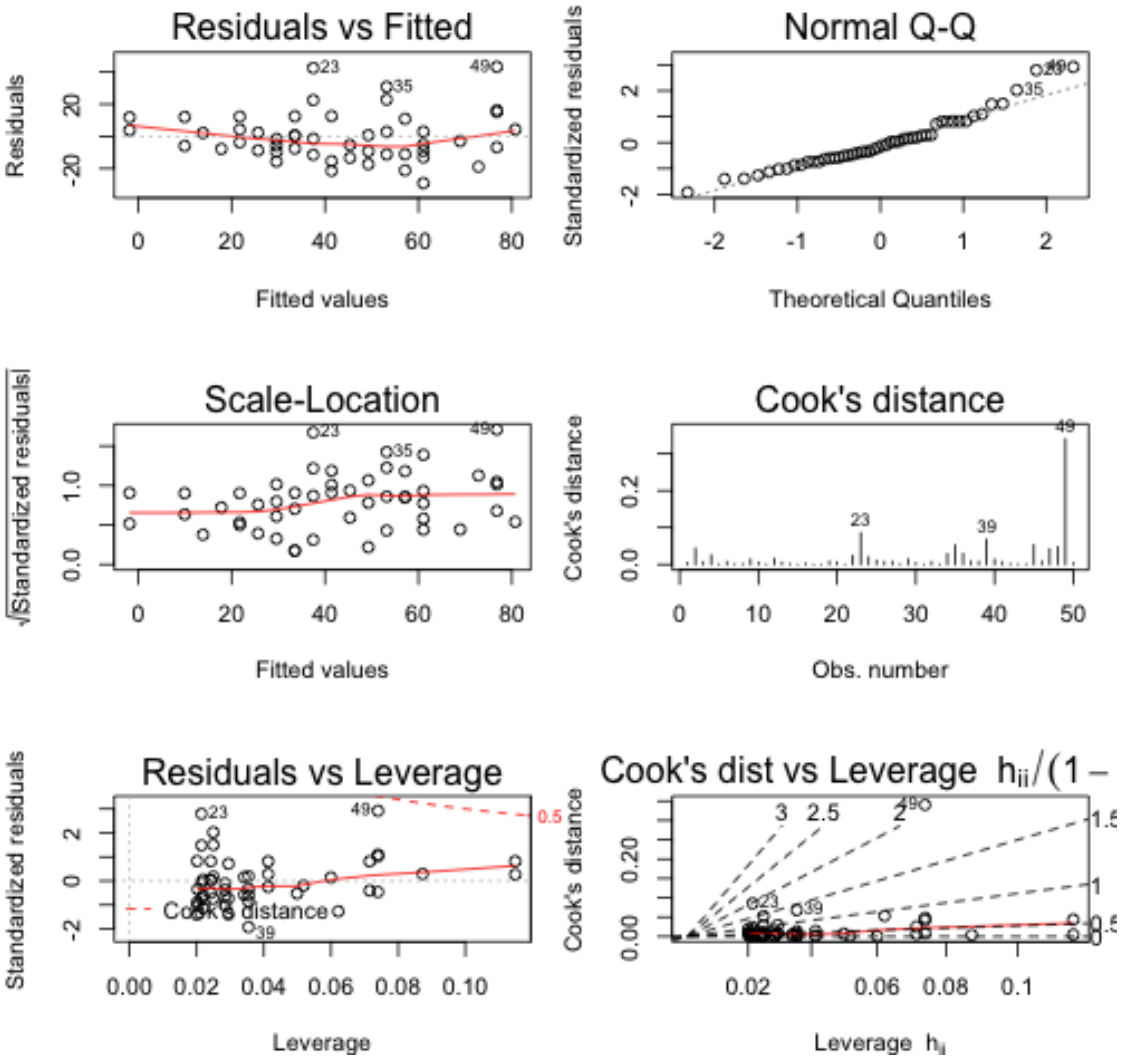
여기서는 cars라는 데이터 객체에 대해서 회귀분석을 수행하고 이를 그래프를 그려본다. cars라는 데이터 객체는 speed와 dist라는 변수를 가지고 있는데 각각 자동차의 속도와 제동거리를 의미한다. 독립변수를 속도, 종속변수를 제동거리로하여 회귀분석을 수행해 보고 회귀진단 그래프를 그려 보자.

```
> lm.cars <- lm(dist ~ speed, data=cars)
> summary(lm.cars)

Call:
lm(formula = dist ~ speed, data = cars)

Residuals:
    Min      1Q  Median      3Q     Max
-29.069  -9.525  -2.272   9.215  43.201

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -17.5791     6.7584  -2.601   0.0123 *
speed         3.9324     0.4155   9.464 1.49e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1

Residual standard error: 15.38 on 48 degrees of freedom
Multiple R-squared: 0.6511,   Adjusted R-squared: 0.6438
F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

회귀분석의 결과를 보면 dist=3.9324speed-17.5791라는 회귀식이 구해졌음을 알 수 있다.

또한 다음의 R 코드로 상기 회귀식에 대한 회귀진단 그래프를 plot() 함수로 그려보자. 여기서 which 인수를 사용하지 않으면 기본 값으로 4개의 그래프가 그려진다는 것을 알아야 한다.

```
> op <- par(no.readonly=TRUE)
> par(mfrow=c(3,2), mar=c(4, 4, 4, 1))
> plot(lm.cars, which=1:6)
> par(op)
```



자료의 시각화의 중요성을 역설하는 사례로서 anscombe라는 데이터 객체를 이용하다보니 내용이 회귀분석에 많이 치우친 것 같다. 그러나 다시 한번 강조하지만 데이터 분석의 시작은 데이터를 좌표상에 표현하는 Visualization으로 시작한다는 것을 잊지 말자는 것이다.