Quantitative Methods in Linguistics – Lecture 8

Adrian Brasoveanu*

April 12, 2014

Contents

1 Recap

Generating the 'dataset':

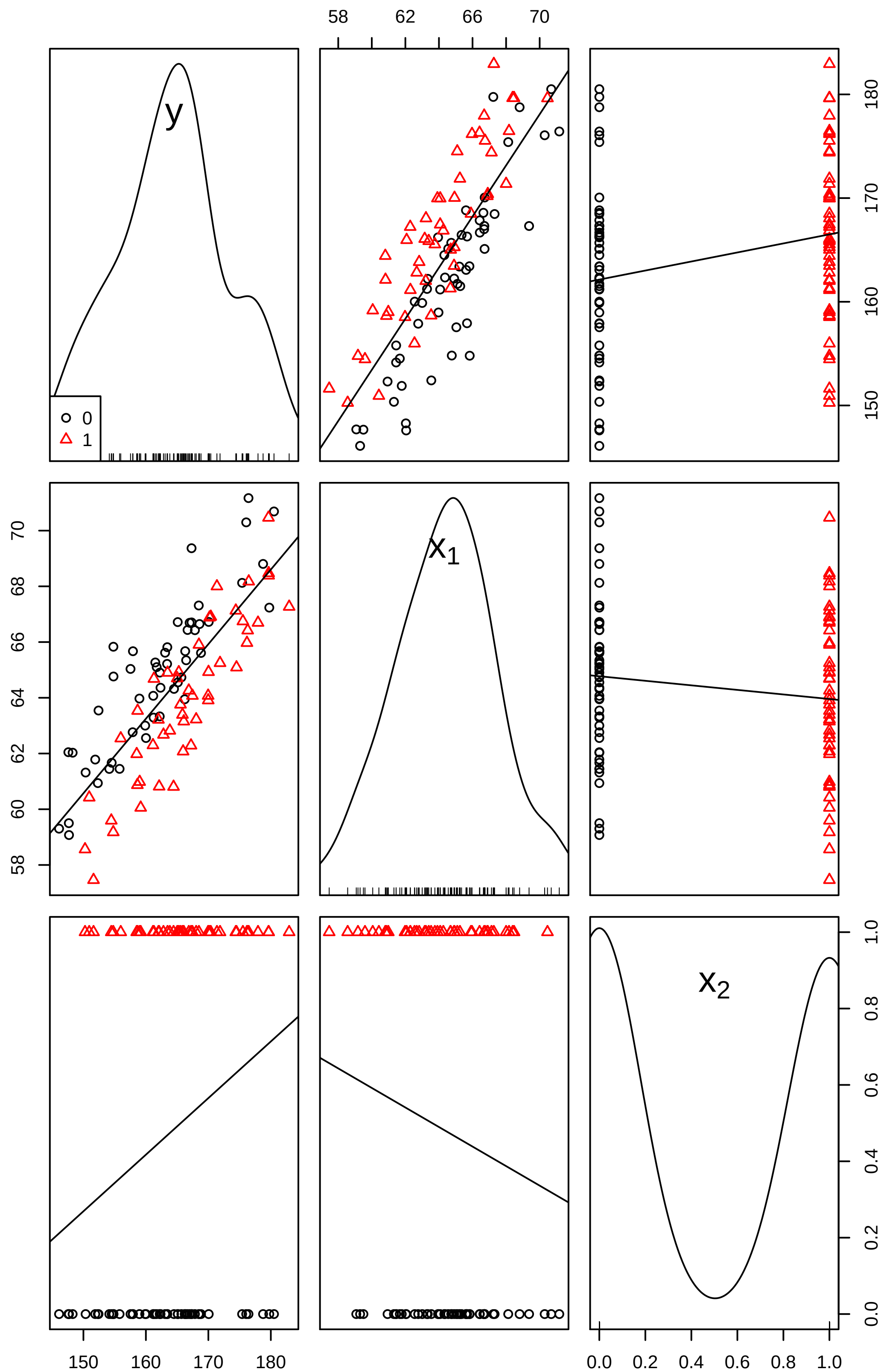
```
> x1 <- rnorm(100, 64, 3)
> x2 <- rbinom(100, 1, 0.5)
> y <- x1 * 2.5 + x2 * 6 + rnorm(100, 0, 4)
```

Scatter plot matrices (SPMs for short):

- smoothed histograms of the variables are displayed on the (upper-left / lower-right) diagonal
- the other panels display plots of all pairs of variables

```
> library("car")
> spm(~y + x1 + x2, smooth = FALSE, groups = as.factor(x2), var.labels = c(expression(y),
+     expression(x[1]), expression(x[2])))
```

*These notes have been generated with the 'knitr' package (Xie 2013) and are based on many sources, including but not limited to: Abelson (1995), Miles and Shevlin (2001), Faraway (2004), De Veaux et al. (2005), Braun and Murdoch (2007), Gelman and Hill (2007), Baayen (2008), Johnson (2008), Wright and London (2009), Gries (2009), Kruschke (2011), Diez et al. (2013), Gries (2013).



y
x1
x2
0
1
58
62
66
70
150
160
170
180
0.0
0.2
0.4
0.6
0.8
1.0

```
> reg0 <- lm(y ~ 1)
> summary(reg0)


Call:
lm(formula = y ~ 1)

Residuals:
    Min      1Q  Median      3Q     Max 
-18.130  -5.565   0.529   4.877  18.702 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  164.232      0.861     191   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.61 on 99 degrees of freedom

> reg2 <- lm(y ~ x2)
> summary(reg2)


Call:
lm(formula = y ~ x2)

Residuals:
    Min      1Q  Median      3Q     Max 
-16.240  -5.610   0.086   4.946  18.366 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   162.14       1.16   139.7   <2e-16 ***
x2              4.36       1.68     2.6    0.011 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.37 on 98 degrees of freedom
Multiple R-squared:  0.0646,Adjusted R-squared:  0.055 
F-statistic: 6.77 on 1 and 98 DF,  p-value: 0.0107

> anova(reg0, reg2)

Analysis of Variance Table

Model 1: y ~ 1
Model 2: y ~ x2
  Res.Df  RSS Df Sum of Sq    F Pr(>F)  
1     99 7341                           
2     98 6867  1       474 6.77  0.011 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> reg1 <- lm(y ~ x1)
> summary(reg1)
```

```
Call:
lm(formula = y ~ x1)

Residuals:
    Min      1Q  Median      3Q     Max
-13.011  -3.323   0.028   3.405  11.586

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    5.546     11.524    0.48     0.63
x1             2.465      0.179   13.78   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.05 on 98 degrees of freedom
Multiple R-squared:  0.66,Adjusted R-squared:  0.656
F-statistic:  190 on 1 and 98 DF,  p-value: <2e-16

> anova(reg0, reg1)

Analysis of Variance Table

Model 1: y ~ 1
Model 2: y ~ x1
  Res.Df  RSS Df Sum of Sq   F Pr(>F)
1     99 7341
2     98 2498  1      4843 190 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2 Fourth attempt: multiple linear regression

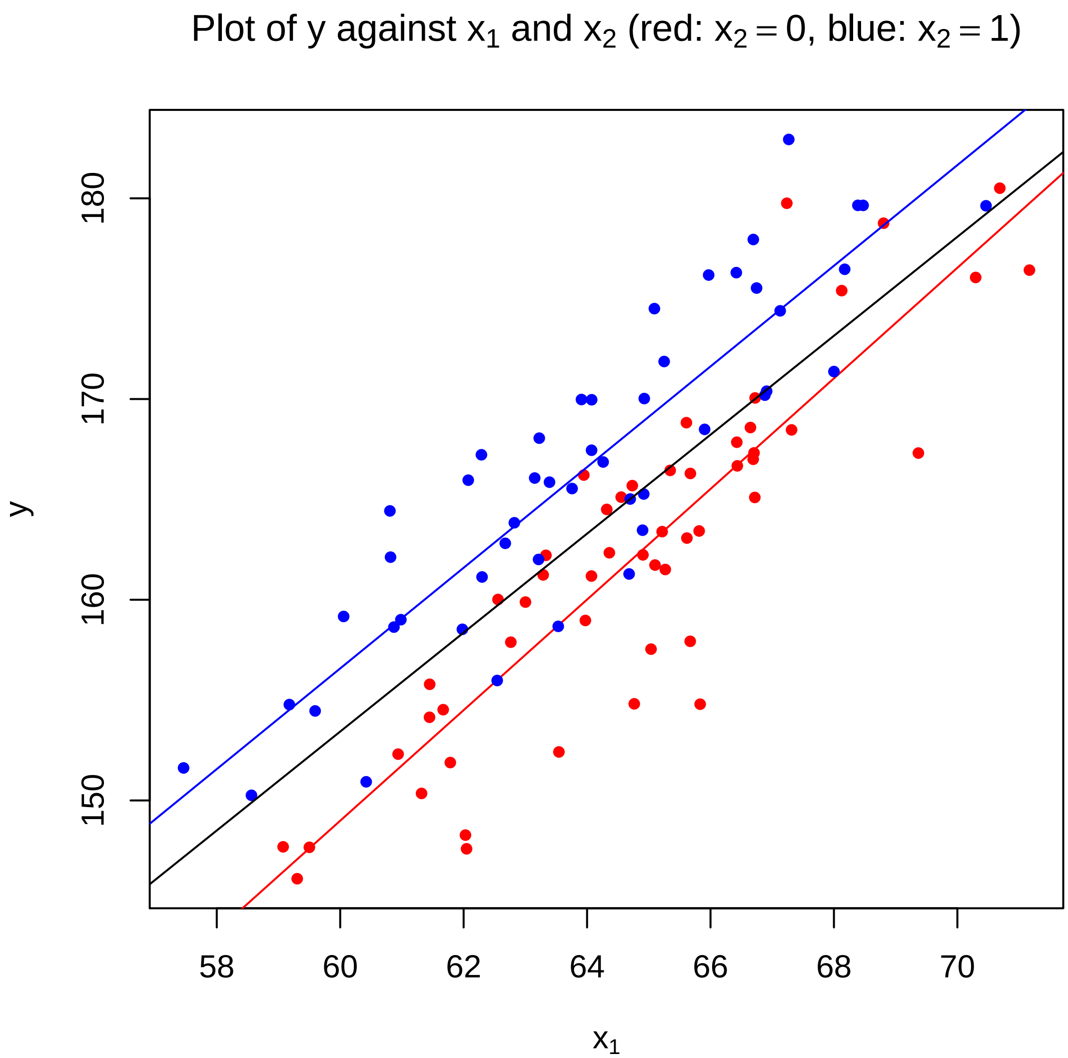
Multiple linear regression: predicting y values based on both x_1 values and x_2 values. This is actually the true population model, i.e., the model we used to generate the data.

```
> y.x1.x2 <- data.frame(y, x1, x2)
> y.x1.x2[1:10, ]

       y    x1 x2
1  165.9 63.39  1
2  147.7 59.08  0
3  167.3 69.37  0
4  148.3 62.03  0
5  162.1 60.82  1
6  157.9 62.76  0
7  167.2 62.29  1
8  154.5 61.67  0
9  166.1 63.15  1
10 150.4 61.32  0

> sum(x2)

[1] 48
```

```
> split.y.x1.x2 <- split(y.x1.x2, x2)
> split.y.x1.x2$"0"

        y    x1 x2
2   147.7 59.08  0
3   167.3 69.37  0
4   148.3 62.03  0
6   157.9 62.76  0
8   154.5 61.67  0
10  150.4 61.32  0
11  151.9 61.78  0
12  170.1 66.72  0
14  168.6 66.65  0
17  152.4 63.54  0
19  167.8 66.43  0
20  147.7 59.50  0
23  163.1 65.62  0
25  178.8 68.80  0
28  167.0 66.69  0
29  163.4 65.22  0
30  147.6 62.05  0
31  166.2 63.95  0
32  179.8 67.24  0
34  166.4 65.35  0
37  168.5 67.31  0
40  155.8 61.45  0
43  159.0 63.97  0
47  159.9 63.00  0
48  176.4 71.17  0
50  166.3 65.67  0
51  165.1 66.72  0
52  154.8 65.83  0
55  157.5 65.04  0
57  162.2 64.90  0
59  152.3 60.94  0
60  162.2 63.33  0
64  164.5 64.32  0
69  163.4 65.82  0
70  168.8 65.61  0
72  160.0 62.56  0
74  161.7 65.10  0
75  162.3 64.36  0
76  165.7 64.73  0
77  180.5 70.69  0
81  175.4 68.13  0
84  166.7 66.43  0
85  176.1 70.30  0
88  146.1 59.30  0
90  167.3 66.71  0
92  161.5 65.27  0
93  154.8 64.76  0
94  161.2 64.07  0
96  161.2 63.29  0
97  157.9 65.67  0
```

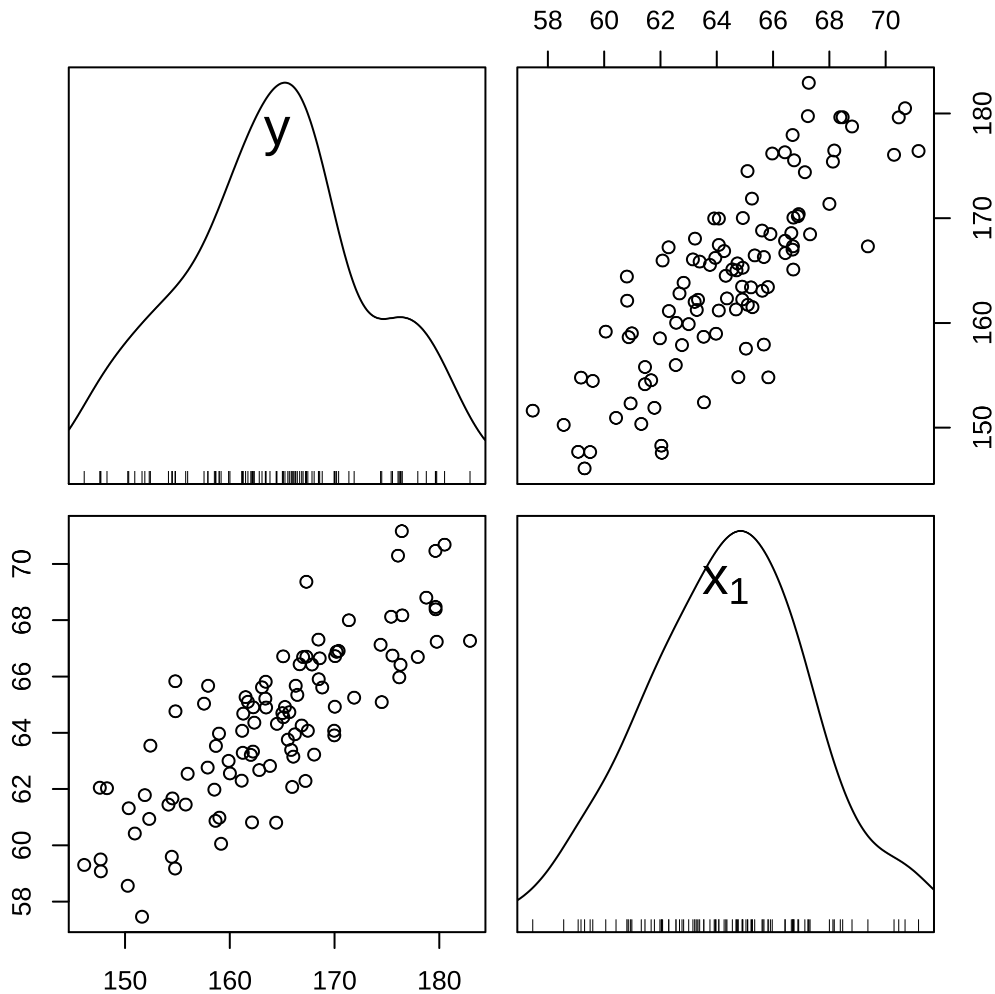
```
98  154.1 61.45  0
100 165.1 64.55  0

> split.y.x1.x2$"1"

       y    x1 x2
1  165.9 63.39  1
5  162.1 60.82  1
7  167.2 62.29  1
9  166.1 63.15  1
13 161.1 62.30  1
15 151.6 57.46  1
16 175.5 66.75  1
18 168.1 63.23  1
21 170.4 66.91  1
22 166.9 64.26  1
24 159.0 60.98  1
26 154.8 59.18  1
27 166.0 62.07  1
33 170.0 64.93  1
35 174.4 67.13  1
36 167.5 64.07  1
38 154.5 59.59  1
39 163.8 62.82  1
41 150.9 60.42  1
42 176.5 68.17  1
44 156.0 62.54  1
45 174.5 65.09  1
46 179.7 68.39  1
49 165.0 64.70  1
53 162.0 63.21  1
54 165.5 63.76  1
56 182.9 67.27  1
58 159.2 60.06  1
61 165.3 64.92  1
62 150.3 58.56  1
63 163.5 64.90  1
65 171.9 65.25  1
66 164.4 60.81  1
67 177.9 66.69  1
68 179.6 68.47  1
71 170.2 66.88  1
73 158.7 63.53  1
78 161.3 64.68  1
79 158.5 61.98  1
80 171.4 68.00  1
82 162.8 62.67  1
83 176.2 65.97  1
86 170.0 63.91  1
87 158.6 60.87  1
89 170.0 64.07  1
91 176.3 66.42  1
95 168.5 65.91  1
99 179.6 70.46  1
```

```
> nrow(split.y.x1.x2$"1")

[1] 48

> reg3.0 <- lm(y ~ x1, data = split.y.x1.x2$"0")
> summary(reg3.0)


Call:
lm(formula = y ~ x1, data = split.y.x1.x2$"0")

Residuals:
    Min      1Q  Median      3Q     Max
-10.259  -1.687   0.393   2.651  10.830

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -16.303     13.209   -1.23     0.22
x1             2.755      0.204   13.52   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.03 on 50 degrees of freedom
Multiple R-squared:  0.785,Adjusted R-squared:  0.781
F-statistic:  183 on 1 and 50 DF,  p-value: <2e-16

> reg3.1 <- lm(y ~ x1, data = split.y.x1.x2$"1")
> summary(reg3.1)


Call:
lm(formula = y ~ x1, data = split.y.x1.x2$"1")

Residuals:
   Min     1Q Median     3Q    Max
-7.032 -2.925  0.074  2.632  8.131

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    6.118     12.098    0.51     0.62
x1             2.508      0.189   13.27   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.73 on 46 degrees of freedom
Multiple R-squared:  0.793,Adjusted R-squared:  0.788
F-statistic:  176 on 1 and 46 DF,  p-value: <2e-16

> plot(split.y.x1.x2$"0"$x1, split.y.x1.x2$"0"$y, pch = 20, col = "red",
+     xlim = range(x1), ylim = range(y), xlab = expression(x[1]), ylab = "y",
+     main = expression(paste("Plot of y against ", x[1], " and ", x[2],
+         " (red: ", x[2] == 0, ", blue: ", x[2] == 1, ")")))
> abline(reg3.0, col = "red")
> points(split.y.x1.x2$"1"$x1, split.y.x1.x2$"1"$y, pch = 20, col = "blue")
> abline(reg3.1, col = "blue")
> abline(lm(y ~ x1), col = "black")
```



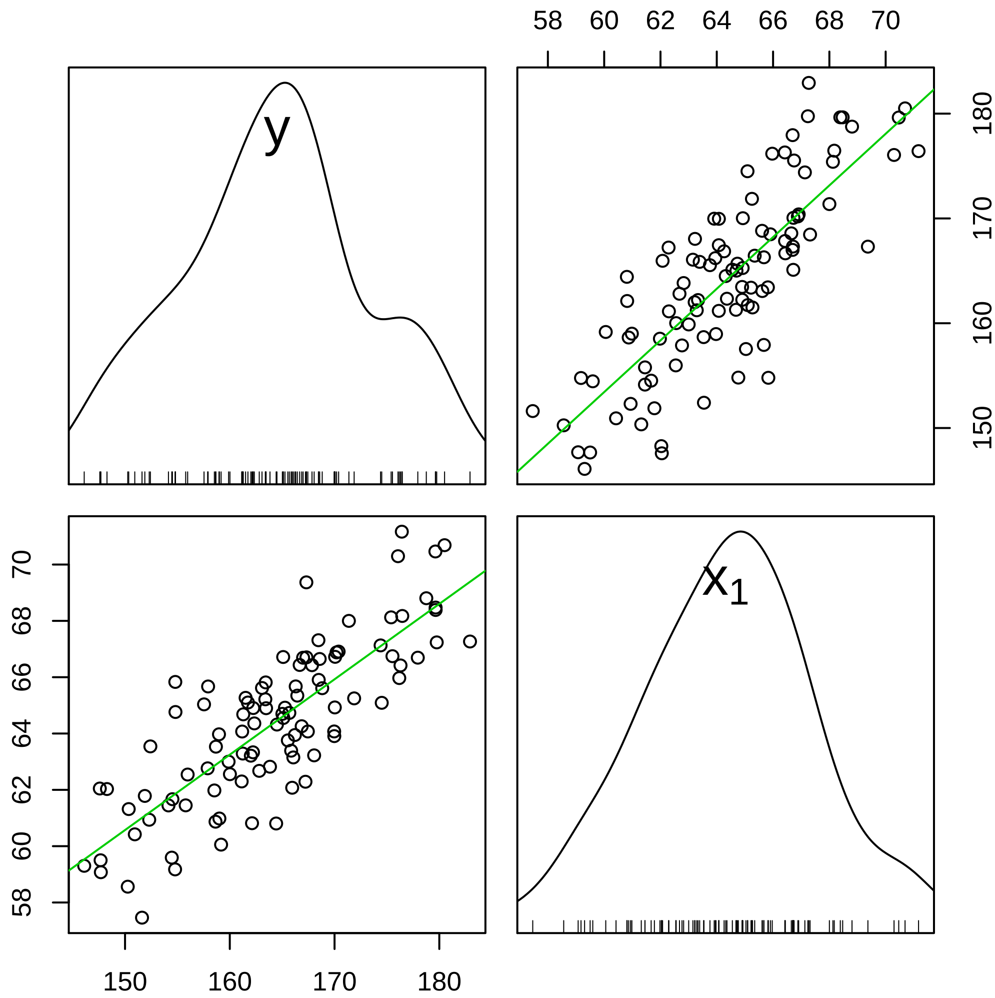
Scatterplot matrices that do the same kind of plots:

```
> scatterplotMatrix(~y + x1, smooth = FALSE, reg.line = FALSE, var.labels = c(expression(y),
+     expression(x[1])))
```



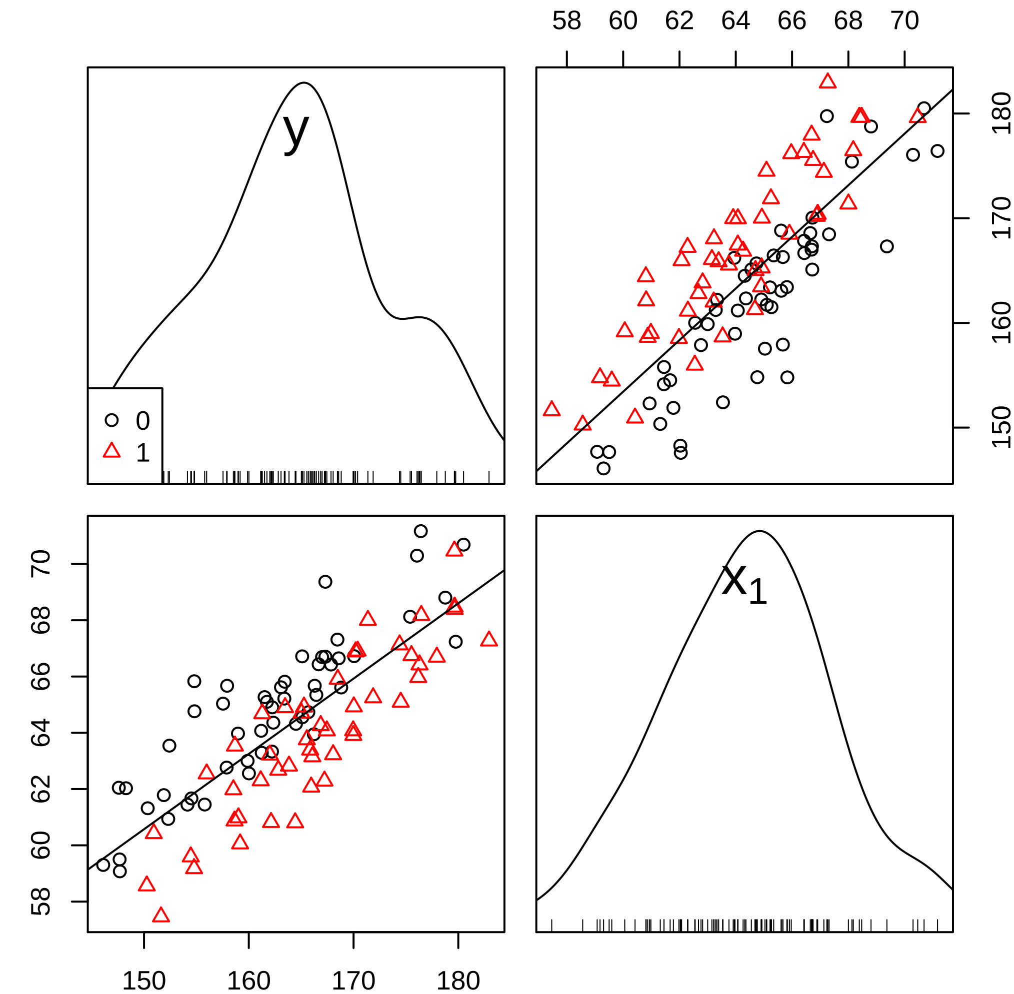
We can add regression lines for each plot

```
> scatterplotMatrix(~y + x1, smooth = FALSE, var.labels = c(expression(y),
+     expression(x[1])))
```



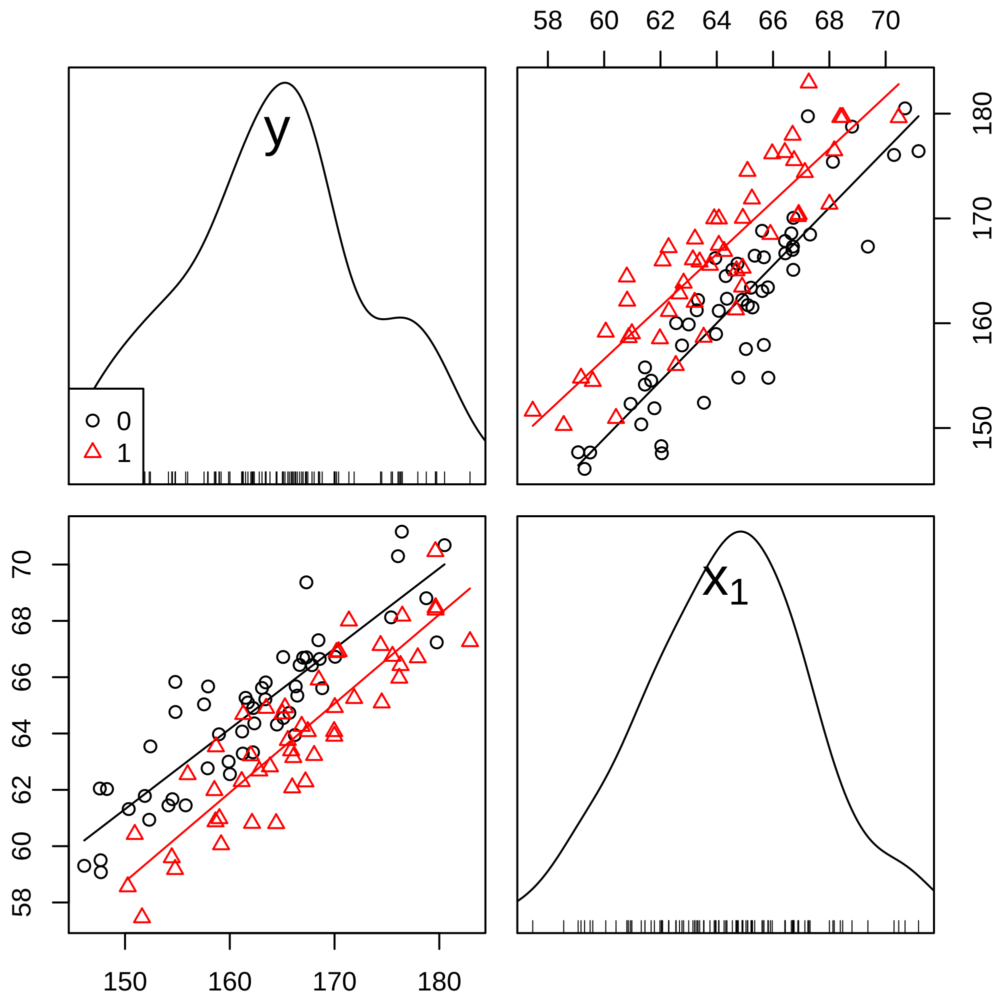
We can group the observations based on a factor:

```
> spm(~y + x1, smooth = FALSE, groups = as.factor(x2), by.groups = FALSE,
+     var.labels = c(expression(y), expression(x[1])))
```



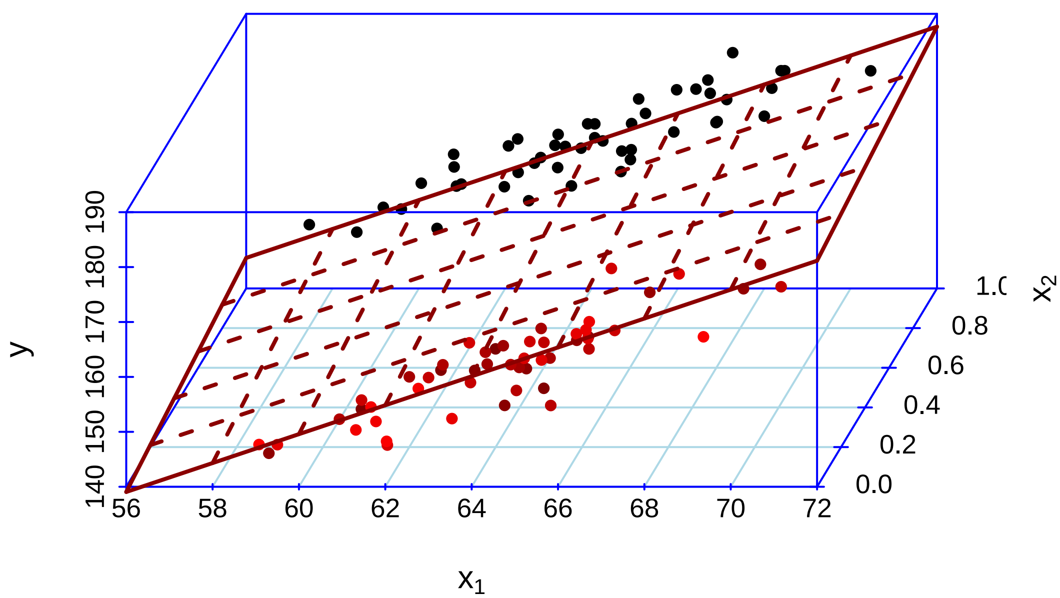
We can add regression lines for each group:

```
> spm(~y + x1, smooth = FALSE, groups = as.factor(x2), by.groups = TRUE,
+     var.labels = c(expression(y), expression(x[1])))
```

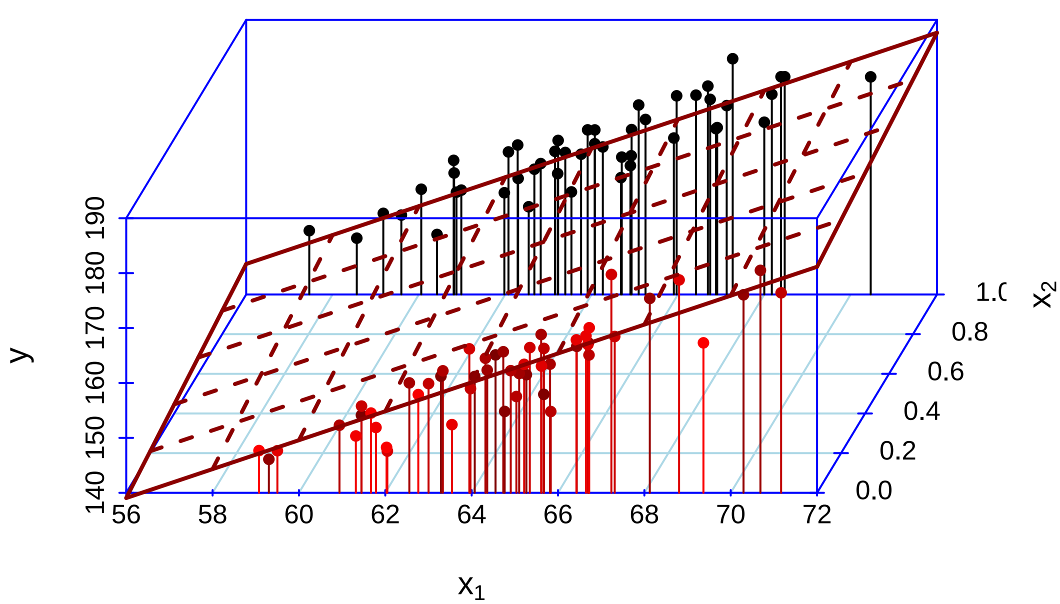


And here's the full regression with both predictors, and the corresponding 3D plots:

```
> reg3 <- lm(y ~ x1 + x2)
> par(mfrow = c(2, 1), mai = c(0.62, 0.62, 0.62, 0.22))
> library("scatterplot3d")
> s3d <- scatterplot3d(x1, x2, y, highlight.3d = TRUE, col.axis = "blue",
+     col.grid = "lightblue", pch = 20, xlab = expression(x[1]), ylab = expression(x[2]),
+     angle = 65, las = 2)
> s3d$plane3d(reg3, lty.box = "solid", col = "darkred", lwd = 2)
> s3d <- scatterplot3d(x1, x2, y, highlight.3d = TRUE, col.axis = "blue",
+     col.grid = "lightblue", pch = 20, type = "h", xlab = expression(x[1]),
+     ylab = expression(x[2]), angle = 65, las = 2)
> s3d$plane3d(reg3, lty.box = "solid", col = "darkred", lwd = 2)
```



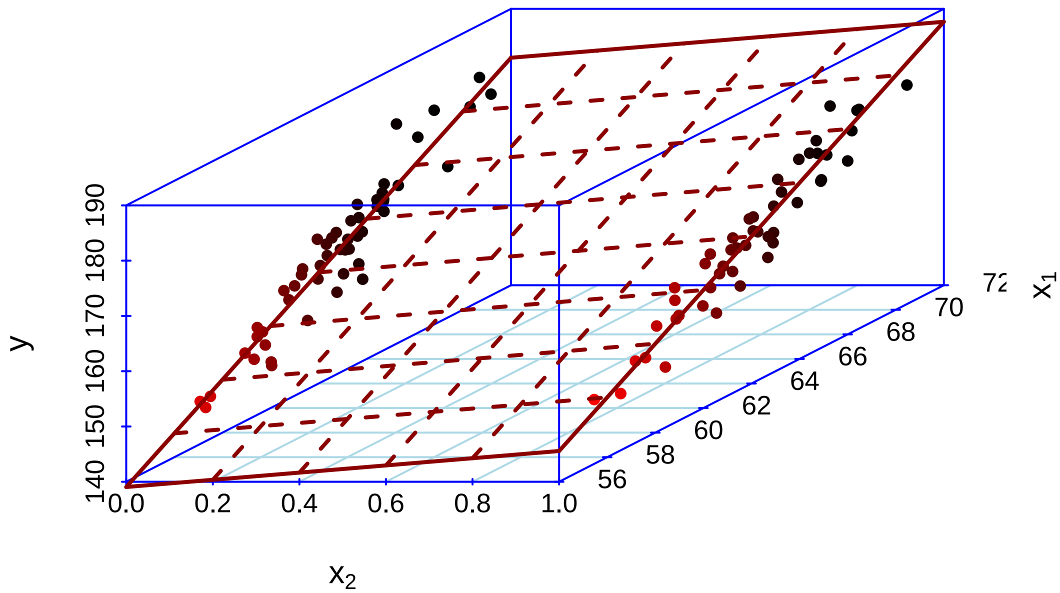
y
140
150
160
170
180
190
56
58
60
62
64
66
68
70
72
x_1
0.0
0.2
0.4
0.6
0.8
1.0
x_2



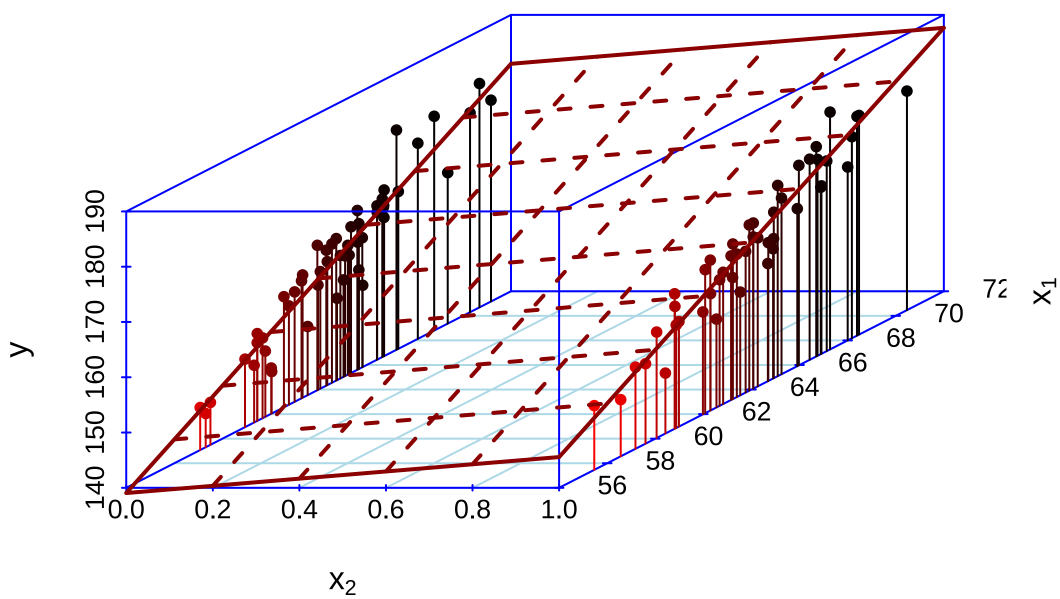
y
140
150
160
170
180
190
56
58
60
62
64
66
68
70
72
x_1
0.0
0.2
0.4
0.6
0.8
1.0
x_2

```
> par(mfrow = c(1, 1), mai = c(1.02, 0.82, 0.82, 0.42))
```

```
> par(mfrow = c(2, 1), mai = c(0.62, 0.62, 0.62, 0.22))
> library("scatterplot3d")
> s3d <- scatterplot3d(x2, x1, y, highlight.3d = TRUE, col.axis = "blue",
+     col.grid = "lightblue", pch = 20, xlab = expression(x[2]), ylab = expression(x[1]),
+     angle = 40, las = 2)
> s3d$plane3d(lm(y ~ x2 + x1), lty.box = "solid", col = "darkred", lwd = 2)
> s3d <- scatterplot3d(x2, x1, y, highlight.3d = TRUE, col.axis = "blue",
+     col.grid = "lightblue", pch = 20, type = "h", xlab = expression(x[2]),
+     ylab = expression(x[1]), angle = 40, las = 2)
> s3d$plane3d(lm(y ~ x2 + x1), lty.box = "solid", col = "darkred", lwd = 2)
```



y
190
180
170
160
150
140
0.0
0.2
0.4
0.6
0.8
1.0
x_2
56
58
60
62
64
66
68
70
72
x_1

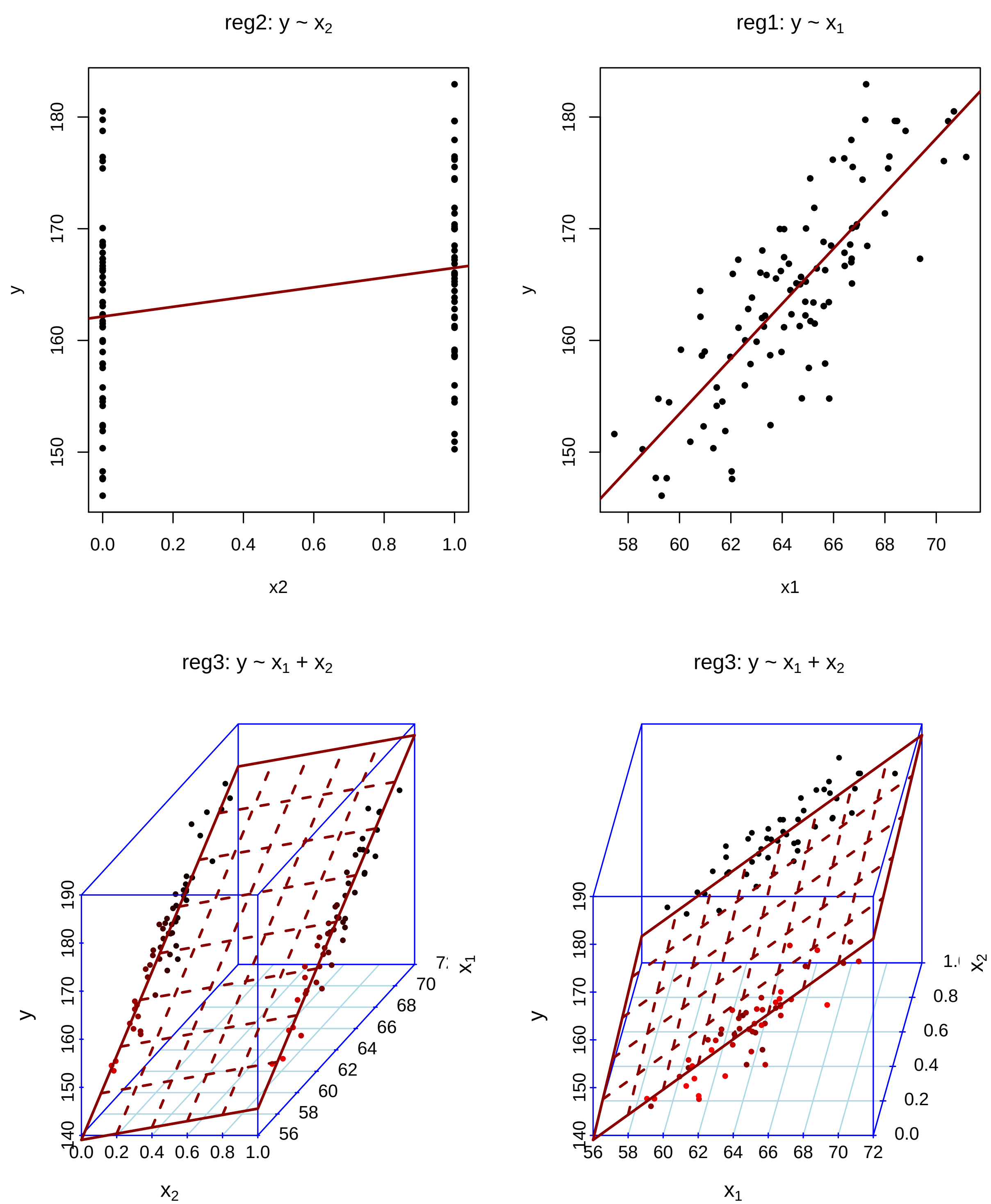


y
190
180
170
160
150
140
0.0
0.2
0.4
0.6
0.8
1.0
x_2
56
58
60
62
64
66
68
70
72
x_1

```
> par(mfrow = c(1, 1), mai = c(1.02, 0.82, 0.82, 0.42))
```

3 Graphical comparison of reg1, reg2 and reg3

```
> par(mfrow = c(2, 2))
> plot(x2, y, pch = 20, main = expression(paste("reg2: ", y, " ~ ",
+     x[2])))
> abline(reg2, lwd = 2, col = "darkred")
> plot(x1, y, pch = 20, main = expression(paste("reg1: ", y, " ~ ",
+     x[1])))
> abline(reg1, lwd = 2, col = "darkred")
> s3d <- scatterplot3d(x2, x1, y, highlight.3d = TRUE, col.axis = "blue",
+     col.grid = "lightblue", pch = 20, ylab = expression(x[1]), xlab = expression(x[2]),
+     main = expression(paste("reg3: ", y, " ~ ", x[1], " + ", x[2])))
> s3d$plane3d(lm(y ~ x2 + x1), lty.box = "solid", lwd = 2, col = "darkred")
> s3d <- scatterplot3d(x1, x2, y, highlight.3d = TRUE, col.axis = "blue",
+     col.grid = "lightblue", pch = 20, xlab = expression(x[1]), ylab = expression(x[2]),
+     main = expression(paste("reg3: ", y, " ~ ", x[1], " + ", x[2])),
+     angle = 65)
> s3d$plane3d(reg3, lty.box = "solid", lwd = 2, col = "darkred")
```




```
> par(mfrow = c(1, 1))
```

4 ANOVA and model selection

```
> anova(reg0, reg3)

Analysis of Variance Table

Model 1: y ~ 1
Model 2: y ~ x1 + x2
  Res.Df  RSS Df Sum of Sq   F Pr(>F)
1     99 7341
2     97 1463  2      5878 195 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> anova(reg0, reg2)

Analysis of Variance Table

Model 1: y ~ 1
Model 2: y ~ x2
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1     99 7341
2     98 6867  1       474 6.77  0.011 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> anova(reg2, reg3)

Analysis of Variance Table

Model 1: y ~ x2
Model 2: y ~ x1 + x2
  Res.Df  RSS Df Sum of Sq   F Pr(>F)
1     98 6867
2     97 1463  1      5404 358 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> anova(reg0, reg2, reg3)

Analysis of Variance Table

Model 1: y ~ 1
Model 2: y ~ x2
Model 3: y ~ x1 + x2
  Res.Df  RSS Df Sum of Sq     F  Pr(>F)
1     99 7341
2     98 6867  1       474  31.4 1.9e-07 ***
3     97 1463  1      5404 358.3 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> anova(reg0, reg1)
```

```
Analysis of Variance Table

Model 1: y ~ 1
Model 2: y ~ x1
  Res.Df  RSS Df Sum of Sq   F Pr(>F)
1     99 7341
2     98 2498  1      4843 190 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> anova(reg1, reg3)

Analysis of Variance Table

Model 1: y ~ x1
Model 2: y ~ x1 + x2
  Res.Df  RSS Df Sum of Sq    F  Pr(>F)
1     98 2498
2     97 1463  1      1035 68.6 6.6e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> anova(reg0, reg1, reg3)

Analysis of Variance Table

Model 1: y ~ 1
Model 2: y ~ x1
Model 3: y ~ x1 + x2
  Res.Df  RSS Df Sum of Sq     F  Pr(>F)
1     99 7341
2     98 2498  1      4843 321.1 < 2e-16 ***
3     97 1463  1      1035  68.6 6.6e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We test if we can remove the intercept, and we can – since we actually did not generate the data with an intercept:

```
> reg4 <- lm(y ~ -1 + x1 + x2)
> reg3 <- lm(y ~ x1 + x2)
> summary(reg4)$coef

   Estimate Std. Error t value   Pr(>|t|)
x1    2.504   0.008293 301.892 2.674e-147
x2    6.375   0.771391   8.264  6.920e-13

> summary(reg3)$coef

            Estimate Std. Error t value  Pr(>|t|)
(Intercept)   -8.298     9.0204 -0.9199 3.599e-01
x1             2.631     0.1390 18.9285 2.454e-34
x2             6.508     0.7856  8.2848 6.637e-13

> anova(reg4, reg3)
```

```
Analysis of Variance Table

Model 1: y ~ -1 + x1 + x2
Model 2: y ~ x1 + x2
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1     98 1476                         
2     97 1463  1      12.8 0.85   0.36
```

The reg3 model with an intercept accounts for more of the variation, but this difference is nonsignificant.

The difference between the residual sums of squares for the two models divided by the residual sum of squares of the first model gives us a frequently used effect size – partial eta-squared η^2:

```
> anova(reg4, reg3)$RSS

[1] 1476 1463

> anova.4.3 <- anova(reg4, reg3)
> (partial.eta.sq <- (anova.4.3$RSS[1] - anova.4.3$RSS[2])/anova.4.3$RSS[1])

[1] 0.008649
```

We write:

```
> text.1 <- paste("Including the intercept did not significantly improve the fit of the model:\nF(",
+     anova.4.3$Res.Df[1] - anova.4.3$Res.Df[2], ", ", anova.4.3$Res.Df[2],
+     ") = ", round(anova.4.3$F[2], 2), ", p = ", round(anova.4.3$"Pr(>F)"[2],
+         2), ", partial eta-squared = ", round(partial.eta.sq, 2),
+     ".", sep = "")
> cat(text.1)

Including the intercept did not significantly improve the fit of the model:
F(1, 97) = 0.85, p = 0.36, partial eta-squared = 0.01.
```

What if we actually have an intercept?

```
> y2 <- 25 + x1 * 2.5 + x2 * 6 + rnorm(100, 0, 4)
> reg4.y2 <- lm(y2 ~ -1 + x1 + x2)
> reg3.y2 <- lm(y2 ~ x1 + x2)
> summary(reg4.y2)$coef

   Estimate Std. Error t value   Pr(>|t|)
x1    2.888   0.009643 299.542 5.747e-147
x2    5.835   0.896924   6.505  3.293e-09

> summary(reg3.y2)$coef

            Estimate Std. Error t value  Pr(>|t|)
(Intercept)   30.915    10.0554   3.074 2.739e-03
x1             2.413     0.1550  15.570 3.940e-28
x2             5.336     0.8757   6.093 2.238e-08
```

The intercept is now significant:

```
> anova(reg4.y2, reg3.y2)
```

```
Analysis of Variance Table

Model 1: y2 ~ -1 + x1 + x2
Model 2: y2 ~ x1 + x2
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1     98 1995
2     97 1818  1       177 9.45 0.0027 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can examine the smaller model `reg1` to see if the intercept is significant there. It might be, but that would probably be a consequence of using the incorrect number of predictors:

```
> summary(update(reg1, . ~ . - 1))


Call:
lm(formula = y ~ x1 - 1)

Residuals:
    Min      1Q  Median      3Q     Max
-13.125  -3.202  -0.124   3.489  11.349

Coefficients:
   Estimate Std. Error t value Pr(>|t|)
x1   2.5508     0.0078     327   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.03 on 99 degrees of freedom
Multiple R-squared:  0.999,Adjusted R-squared:  0.999
F-statistic: 1.07e+05 on 1 and 99 DF,  p-value: <2e-16

> anova(reg1, update(reg1, . ~ . - 1))

Analysis of Variance Table

Model 1: y ~ x1
Model 2: y ~ x1 - 1
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1     98 2498
2     99 2504 -1      -5.9 0.23   0.63
```

5 Adding interactions

We can add an interaction term to the `reg3` model:

```
> reg5 <- lm(y ~ x1 + x2 + x1:x2)
> summary(reg5)


Call:
lm(formula = y ~ x1 + x2 + x1:x2)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-10.259  -2.302   0.118   2.651  10.830

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -16.303     12.758   -1.28     0.20
x1             2.755      0.197   14.00   <2e-16 ***
x2            22.421     17.932    1.25     0.21
x1:x2         -0.247      0.278   -0.89     0.38
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.89 on 96 degrees of freedom
Multiple R-squared:  0.802,Adjusted R-squared:  0.796
F-statistic:  130 on 3 and 96 DF,  p-value: <2e-16
```

A shorter way of writing the same model formula:

```
> reg5 <- lm(y ~ x1 * x2)
> summary(reg5)


Call:
lm(formula = y ~ x1 * x2)

Residuals:
    Min      1Q  Median      3Q     Max
-10.259  -2.302   0.118   2.651  10.830

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -16.303     12.758   -1.28     0.20
x1             2.755      0.197   14.00   <2e-16 ***
x2            22.421     17.932    1.25     0.21
x1:x2         -0.247      0.278   -0.89     0.38
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.89 on 96 degrees of freedom
Multiple R-squared:  0.802,Adjusted R-squared:  0.796
F-statistic:  130 on 3 and 96 DF,  p-value: <2e-16
```

The interaction (product) term is not significant:

```
> anova(reg3, reg5)

Analysis of Variance Table

Model 1: y ~ x1 + x2
Model 2: y ~ x1 * x2
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1     97 1463
2     96 1451  1      11.9 0.79   0.38
```

In fact, the intercept and the interaction term together are not significant – as expected, given that we did not use either of them when we generated the data. We see here the advantage of using F-tests, which enables us to do arbitrary nested-model comparisons – we are not forced to compare models that only differ in one parameter, as we would be with t-tests:

```
> anova(reg4, reg5)

Analysis of Variance Table

Model 1: y ~ -1 + x1 + x2
Model 2: y ~ x1 * x2
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1     98 1476
2     96 1451  2      24.7 0.82   0.44
```

5.1 Interpreting interactions

What is the interpretation of the interaction term, i.e., what exactly is the difference between the `reg3` and `reg5` models?

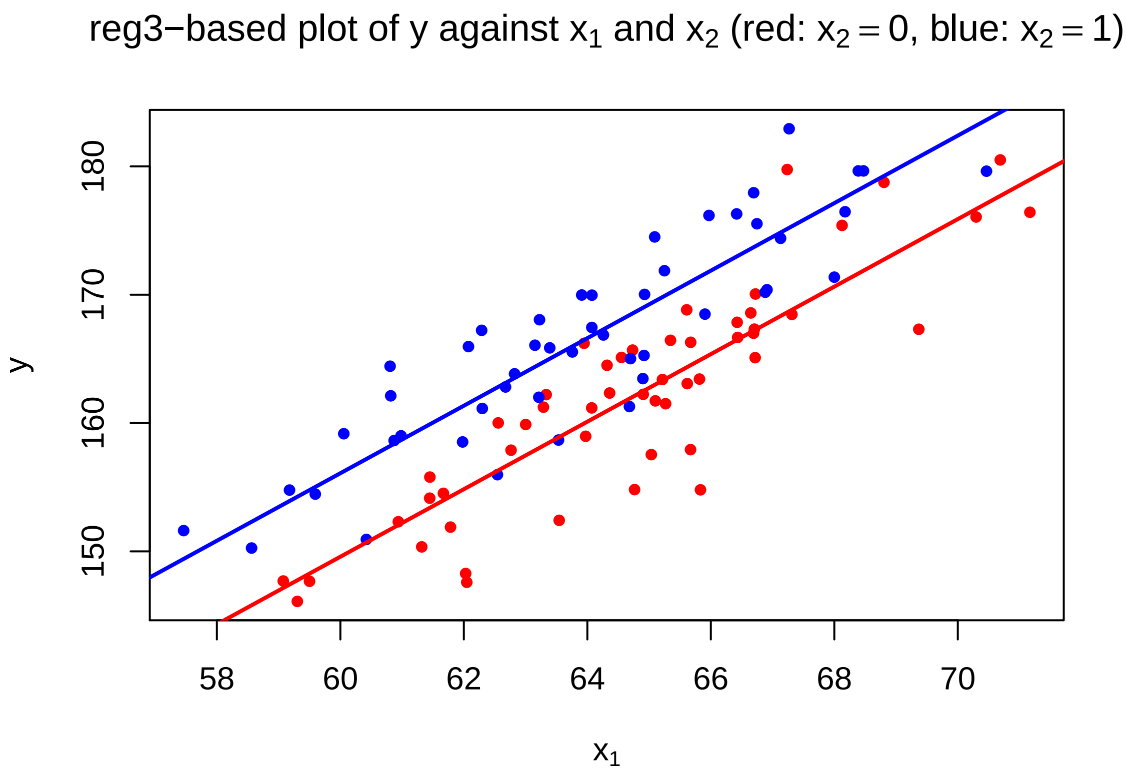
```
> reg3


Call:
lm(formula = y ~ x1 + x2)

Coefficients:
(Intercept)           x1           x2
      -8.30         2.63         6.51

> reg5


Call:
lm(formula = y ~ x1 * x2)

Coefficients:
(Intercept)           x1           x2        x1:x2
    -16.303        2.755       22.421       -0.247
```

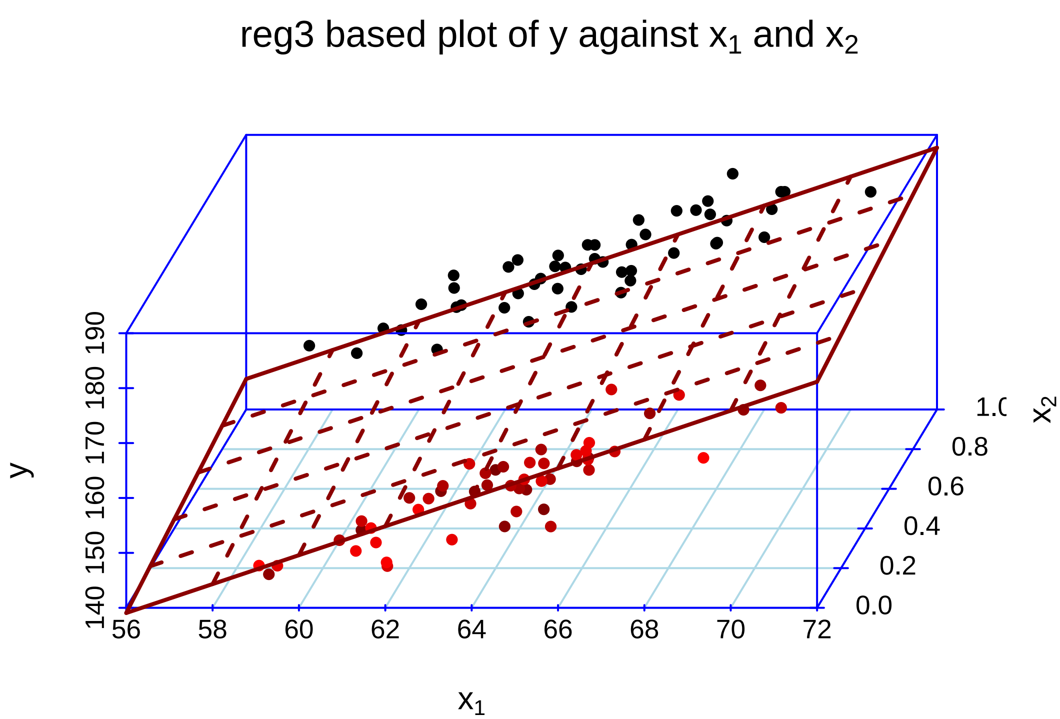
Both models fit two $y \sim x_1$ regression lines, one for each of the two x_2 groups. But in `reg3`, the two regression lines have *the same slope*, while in `reg5` they have *different slopes*. The difference between the two slopes in the `reg5` model is given by the interaction term.

(1) `reg3`: $y = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2$
   a. the first regression line (for the $x_2 = 0$ group): intercept = β_0, slope = β_1
   b. the second regression line (for the $x_2 = 1$ group): intercept = $\beta_0 + \beta_2$, slope = β_1

```
> par(mfrow = c(2, 1))
> plot(split.y.x1.x2$"0"$x1, split.y.x1.x2$"0"$y, pch = 20, col = "red",
+     xlim = range(x1), ylim = range(y), xlab = expression(x[1]), ylab = "y",
+     main = expression(paste("reg3-based plot of y against ", x[1],
+         " and ", x[2], " (red: ", x[2] == 0, ", blue: ", x[2] == 1,
```

```r
+         ")")))
> points(split.y.x1.x2$"1"$x1, split.y.x1.x2$"1"$y, pch = 20, col = "blue")
> abline(reg3$coef[1], reg3$coef[2], col = "red", lwd = 2)
> abline(reg3$coef[1] + reg3$coef[3], reg3$coef[2], col = "blue", lwd = 2)
> s3d <- scatterplot3d(x1, x2, y, highlight.3d = TRUE, col.axis = "blue",
+     col.grid = "lightblue", pch = 20, angle = 65, xlab = expression(x[1]),
+     ylab = expression(x[2]), main = expression(paste("reg3 based plot of y against ",
+         x[1], " and ", x[2])))
> s3d$plane3d(reg3$coef, lty.box = "solid", col = "darkred", lwd = 2)
```


reg3−based plot of y against x1 and x2 (red: x2 = 0, blue: x2 = 1)
y
150
160
170
180
58
60
62
64
66
68
70
x1


reg3 based plot of y against x1 and x2
y
140
150
160
170
180
190
56
58
60
62
64
66
68
70
72
x1
0.0
0.2
0.4
0.6
0.8
x2

```
> par(mfrow = c(1, 1))
```

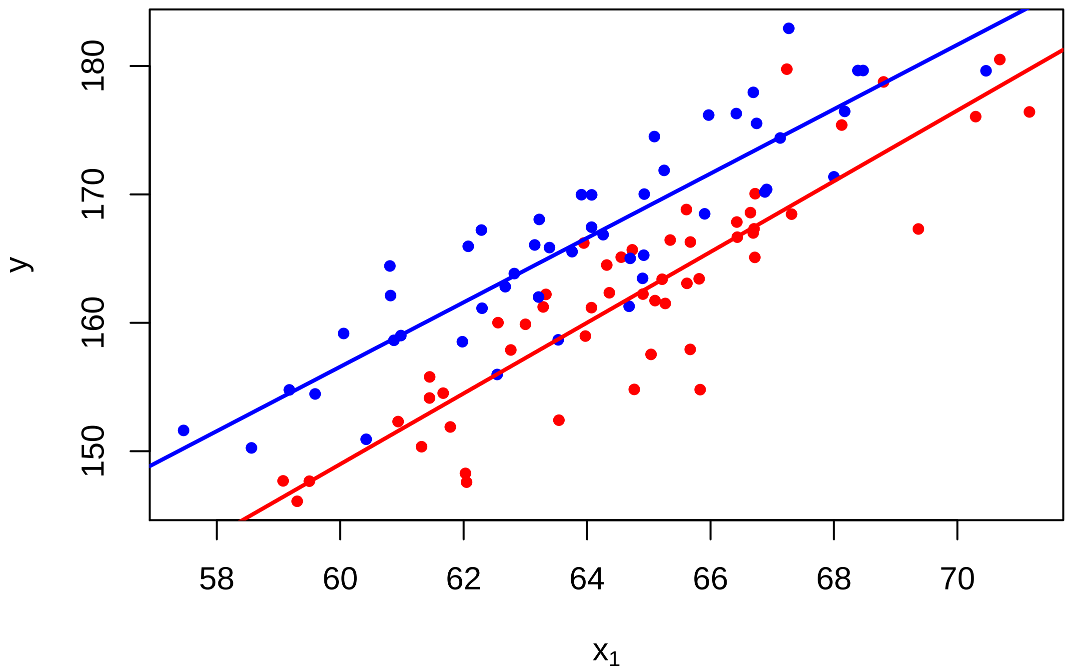
(2) `reg5`: $y = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \beta_3 \cdot x_1 \cdot x_2$,
equivalently: $y = \beta_0 + (\beta_1 + \beta_3 \cdot x_2) \cdot x_1 + \beta_2 \cdot x_2$

a. the first regression line (for the $x_2 = 0$ group): intercept = β_0, slope = β_1

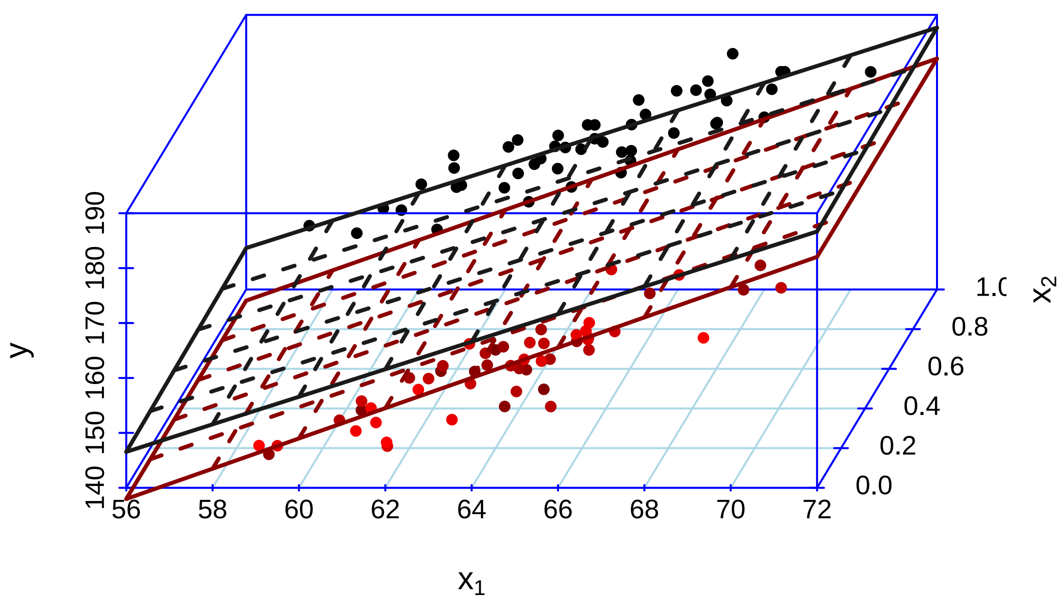
b. the second regression line (for the $x_2 = 1$ group): intercept = $\beta_0 + \beta_2$, slope = $\beta_1 + \beta_3$

```
> par(mfrow = c(2, 1))
> plot(split.y.x1.x2$"0"$x1, split.y.x1.x2$"0"$y, pch = 20, col = "red",
+     xlim = range(x1), ylim = range(y), xlab = expression(x[1]), ylab = "y",
+     main = expression(paste("reg5-based plot of y against ", x[1],
+         " and ", x[2], " (red: ", x[2] == 0, ", blue: ", x[2] == 1,
+         ")")))
> points(split.y.x1.x2$"1"$x1, split.y.x1.x2$"1"$y, pch = 20, col = "blue")
> abline(reg5$coef[1], reg5$coef[2], col = "red", lwd = 2)
> abline(reg5$coef[1] + reg5$coef[3], reg5$coef[2] + reg5$coef[4], col = "blue",
+     lwd = 2)
> s3d <- scatterplot3d(x1, x2, y, highlight.3d = TRUE, col.axis = "blue",
+     col.grid = "lightblue", pch = 20, angle = 65, xlab = expression(x[1]),
+     ylab = expression(x[2]), main = expression(paste("reg5 based plot of y against ",
+         x[1], " and ", x[2])))
> s3d$plane3d(c(reg5$coef[1], reg5$coef[2], 0), lty.box = "solid", col = "darkred",
+     lwd = 2)
> s3d$plane3d(c(reg5$coef[1] + reg5$coef[3], reg5$coef[2] + reg5$coef[4],
+     1), lty.box = "solid", col = "gray10", lwd = 2)
```

reg5−based plot of y against x_1 and x_2 (red: $x_2 = 0$, blue: $x_2 = 1$)



reg5 based plot of y against x_1 and x_2



```
> par(mfrow = c(1, 1))
```

But in our case, allowing for different slopes for the two regression lines does not significantly reduce the error because we generated the data without an interaction term.

```
> anova(reg3, reg5)

Analysis of Variance Table

Model 1: y ~ x1 + x2
Model 2: y ~ x1 * x2
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1     97 1463
2     96 1451  1      11.9 0.79   0.38
```

6 More on interactions

6.1 Multicollinearity and variable centering

Multicollinearity: two or more predictor variables in a multiple regression model are highly correlated.

In this case, the coefficient estimates may change erratically in response to small changes in the model or the data.

Multicollinearity does not reduce the predictive power or reliability of the model as a whole; it only affects the estimates and SEs of individual predictors.

That is, a multiple regression model with correlated predictors can indicate how well the entire bundle of predictors predicts the response variable, but it may not give valid results about any individual predictor, or about which predictors are redundant.

Adding product terms $predictor_1 \cdot predictor_2$, $predictor_1 \cdot predictor_1$ etc. can induce multicollinearity.

For example:

- when all the $predictor_1$ and $predictor_2$ values are positive, high values produce high products $predictor_1 \cdot predictor_2$ and low values produce low products $predictor_1 \cdot predictor_2$
- hence, the product variable is highly correlated with (at least one of) the component variables

```
> predictor1 <- c(2, 4, 5, 6, 7, 7, 8, 9, 9, 11)
> predictor2 <- c(13, 10, 8, 9, 7, 6, 3, 4, 10, 13)
> predictor1_predictor2 <- predictor1 * predictor2
> (predictor1.predictor2 <- data.frame(predictor1, predictor2, predictor1_predictor2,
+     row.names = letters[1:10]))

  predictor1 predictor2 predictor1_predictor2
a          2         13                    26
b          4         10                    40
c          5          8                    40
d          6          9                    54
e          7          7                    49
f          7          6                    42
g          8          3                    24
h          9          4                    36
i          9         10                    90
j         11         13                   143
```

```
> cor(predictor1, predictor2)

[1] -0.2507
```

Here are the correlations with higher-order (interaction) terms:

```
> cor(predictor1, predictor1_predictor2)

[1] 0.6616

> cor(predictor2, predictor1_predictor2)

[1] 0.5406

> cor(predictor1, predictor1^2)

[1] 0.976

> cor(predictor2, predictor2^2)

[1] 0.9814
```

Centering the variable remedies this because the low end of both scales now has large absolute values, so the product becomes large at the low end of the scale.

```
> predictor1c <- predictor1 - mean(predictor1)
> predictor2c <- predictor2 - mean(predictor2)
> predictor1c_predictor2c <- predictor1c * predictor2c
> (predictor1c.predictor2c <- data.frame(predictor1c, predictor2c, predictor1c_predictor2c,
+     row.names = letters[1:10]))

  predictor1c predictor2c predictor1c_predictor2c
a        -4.8         4.7                  -22.56
b        -2.8         1.7                   -4.76
c        -1.8        -0.3                    0.54
d        -0.8         0.7                   -0.56
e         0.2        -1.3                   -0.26
f         0.2        -2.3                   -0.46
g         1.2        -5.3                   -6.36
h         2.2        -4.3                   -9.46
i         2.2         1.7                    3.74
j         4.2         4.7                   19.74

> cor(predictor1c, predictor2c)

[1] -0.2507
```

And here are the correlations with the higher-order terms:

```
> cor(predictor1c, predictor1c_predictor2c)  # oops, centering doesn't always work

[1] 0.7194

> cor(predictor2c, predictor1c_predictor2c)  # desired effect

[1] 0.1845
```

```
> cor(predictor1c, predictor1c^2)  # desired effect

[1] -0.2215

> cor(predictor2c, predictor2c^2)  # desired effect

[1] -0.07475
```

6.2 Another example of regression with interaction terms

We discuss now another example of multiple regression with interaction terms.[1]

```
> icecream <- read.csv(paste("http://dl.dropbox.com/u/10246536/Web/RTutorialSeries/",
+     "dataset_multipleRegression_interactions.csv", sep = ""))
> head(icecream)

  DATE CONSUME PRICE INC TEMP
1    1   0.386 0.270  78   41
2    2   0.374 0.282  79   56
3    3   0.393 0.277  81   63
4    4   0.425 0.280  80   68
5    5   0.406 0.272  76   69
6    6   0.344 0.262  78   65
```

This dataset contains the following variables related to ice cream consumption:

- DATE: time period (1-30)
- CONSUME: ice cream consumption in pints per capita
- PRICE: per-pint price of ice cream in dollars
- INC: weekly family income in dollars
- TEMP: mean temperature in degrees F

Task: determine how much of the variance in ice cream consumption can be predicted by:

- per-pint price
- weekly family income
- mean temperature
- the interaction PRICE · INC between per-pint price and weekly family income

For example, the extent to which increasing the price decreases the ice cream consumption might be moderated by income: the higher the income, the smaller the consumption decrease *for the same price increase*.

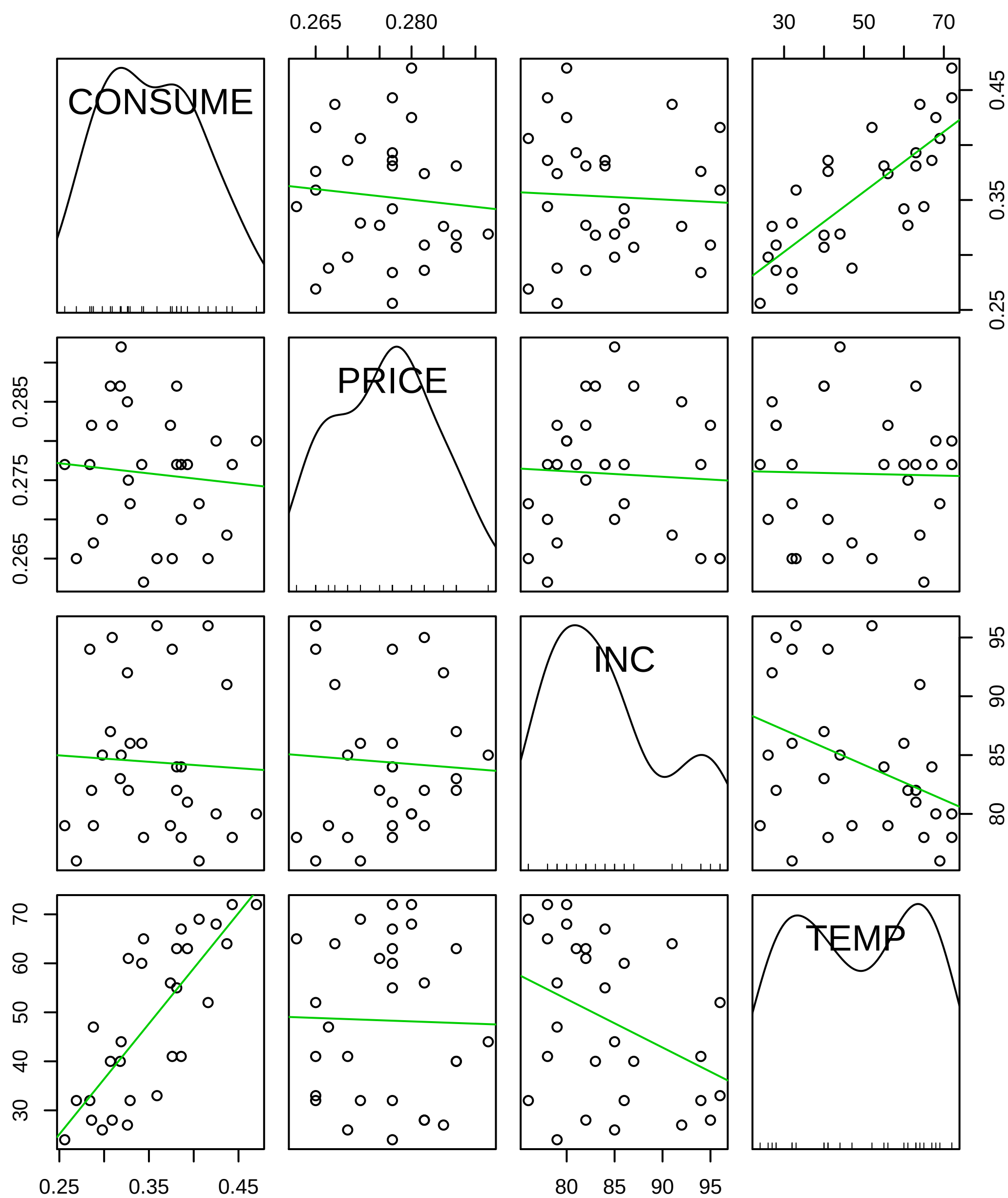
That is, the extent to which PRICE affects CONSUME is a function of INC. In its simplest form, the effect of PRICE is a linear function of INC – which is why modeling interactions is tantamount to adding product terms:

(3) The effect of PRICE on CONSUME is a linear function of INC:
$\text{CONSUME} = \beta_0 + (\beta_1 + \beta_2 \cdot \text{INC}) \cdot \text{PRICE} + \beta_3 \cdot \text{INC} + \beta_4 \cdot \text{TEMP}$

[1] Based on http://www.r-bloggers.com/r-tutorial-series-regression-with-interaction-variables/; http://rtutorialseries.blogspot.com/2010/01/r-tutorial-series-regression-with.html.

(4) The same regression model reexpressed with explicit interaction terms:
$\text{CONSUME} = \beta_0 + \beta_1 \cdot \text{PRICE} + \beta_3 \cdot \text{INC} + \beta_4 \cdot \text{TEMP} + \beta_2 \cdot \text{INC} \cdot \text{PRICE})$

```
> attach(icecream)
> library("car")
> spm(~CONSUME + PRICE + INC + TEMP, smooth = FALSE)
```



We can add the PRICE · INC interaction manually by creating the product vector and adding it as an additional predictor:

```
> PRICE_INCi <- PRICE * INC
> interactionModel <- lm(CONSUME ~ PRICE + INC + TEMP + PRICE_INCi)
> summary(interactionModel)


Call:
lm(formula = CONSUME ~ PRICE + INC + TEMP + PRICE_INCi)

Residuals:
     Min       1Q   Median       3Q      Max
-0.05753 -0.01636 -0.00085  0.01687  0.07189

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -6.329813   3.105350   -2.04    0.053 .
PRICE       23.354020  11.479635    2.03    0.053 .
INC          0.078076   0.036427    2.14    0.042 *
TEMP         0.002823   0.000417    6.77  5.3e-07 ***
PRICE_INCi  -0.278600   0.134440   -2.07    0.049 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0309 on 24 degrees of freedom
Multiple R-squared:  0.741,Adjusted R-squared:  0.698
F-statistic: 17.2 on 4 and 24 DF,  p-value: 8.97e-07
```

Or we can let the `lm` function do it for us:

```
> summary(lm(CONSUME ~ PRICE + INC + TEMP + PRICE:INC))


Call:
lm(formula = CONSUME ~ PRICE + INC + TEMP + PRICE:INC)

Residuals:
     Min       1Q   Median       3Q      Max
-0.05753 -0.01636 -0.00085  0.01687  0.07189

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -6.329813   3.105350   -2.04    0.053 .
PRICE       23.354020  11.479635    2.03    0.053 .
INC          0.078076   0.036427    2.14    0.042 *
TEMP         0.002823   0.000417    6.77  5.3e-07 ***
PRICE:INC   -0.278600   0.134440   -2.07    0.049 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0309 on 24 degrees of freedom
Multiple R-squared:  0.741,Adjusted R-squared:  0.698
F-statistic: 17.2 on 4 and 24 DF,  p-value: 8.97e-07
```

We see that the interaction is significant:

```
> noInteractionModel <- lm(CONSUME ~ PRICE + INC + TEMP)
> summary(noInteractionModel)


Call:
lm(formula = CONSUME ~ PRICE + INC + TEMP)

Residuals:
     Min       1Q   Median       3Q      Max
-0.05940 -0.01567  0.00523  0.01716  0.07052

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.087744   0.244740    0.36    0.723
PRICE       -0.386358   0.783086   -0.49    0.626
INC          0.002618   0.001076    2.43    0.023 *
TEMP         0.003119   0.000417    7.48  7.8e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0329 on 25 degrees of freedom
Multiple R-squared:  0.695,Adjusted R-squared:  0.658
F-statistic:   19 on 3 and 25 DF,  p-value: 1.26e-06

> anova(noInteractionModel, interactionModel)

Analysis of Variance Table

Model 1: CONSUME ~ PRICE + INC + TEMP
Model 2: CONSUME ~ PRICE + INC + TEMP + PRICE_INCi
  Res.Df    RSS Df Sum of Sq    F Pr(>F)
1     25 0.0271
2     24 0.0230  1   0.00411 4.29  0.049 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> detach(icecream)
```

References

Abelson, R.P. (1995). *Statistics as Principled Argument*. L. Erlbaum Associates.

Baayen, R. Harald (2008). *Analyzing Linguistic Data: A Practical Introduction to Statistics Using R*. Cambridge University Press.

Braun, J. and D.J. Murdoch (2007). *A First Course in Statistical Programming with R*. Cambridge University Press.

De Veaux, R.D. et al. (2005). *Stats: Data and Models*. Pearson Education, Limited.

Diez, D. et al. (2013). *OpenIntro Statistics: Second Edition*. CreateSpace Independent Publishing Platform. URL: http://www.openintro.org/stat/textbook.php.

Faraway, J.J. (2004). *Linear Models With R*. Chapman & Hall Texts in Statistical Science Series. Chapman & Hall/CRC.

Gelman, A. and J. Hill (2007). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Analytical Methods for Social Research. Cambridge University Press.

Gries, S.T. (2009). *Quantitative Corpus Linguistics with R: A Practical Introduction*. Taylor & Francis.

Gries, S.T. (2013). *Statistics for Linguistics with R: A Practical Introduction, 2nd Edition*. Mouton De Gruyter.
Johnson, K. (2008). *Quantitative methods in linguistics*. Blackwell Pub.
Kruschke, John K. (2011). *Doing Bayesian Data Analysis: A Tutorial with R and BUGS*. Academic Press/Elsevier.
Miles, J. and M. Shevlin (2001). *Applying Regression and Correlation: A Guide for Students and Researchers*. SAGE Publications.
Wright, D.B. and K. London (2009). *Modern regression techniques using R: A practical guide for students and researchers*. SAGE.
Xie, Yihui (2013). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC.