M7222 – 1. cvičení : **GLM01a** (*Toxic Chemical Production Data*)

Popis dat je v souboru `toxic.txt`, samotná data jsou uložena v souboru `toxic.dat`. Nejprve načteme popisný soubor pomocí příkazu `readLines()`. Protože je příkaz v závorkách, ihned se zobrazí obsah souboru.

```
> fileTxt <- paste(data.library, "toxic.txt", sep = "")
> con <- file(fileTxt)
> (popis <- readLines(con))
```

```
 [1] "Toxic Chemical Production Data"
 [2] "=============================="
 [3] "Artificial data record the volume of a toxic chemical "
 [4] "that is produced as a by-product in a certain industrial"
 [5] "manufacturing process."
 [6] ""
 [7] "The file toxic.dat contains these variables:"
 [8] ""
 [9] "VOL     The volume of toxic by-product produced (in litres)"
[10] "TEMP    The temperature of the manufacturing process (in^oC)"
[11] "CAT     The weight of catalyst (in kg)"
[12] "METHOD  The method used to produce the chemical (qualitative)"
```

```
> close(con)
```

Nyní načteme datový soubor pomocí příkazu `read.table()`. Příkazem `str()` vypíšeme strukturu datového rámce, příkazem `head()` se vypíše prvních šest řádků.

```
> fileDat <- paste(data.library, "toxic.dat", sep = "")
> data <- read.table(fileDat, header = TRUE)
> str(data)
```

```
,data.frame,:   8 obs. of  4 variables:
 $ VOL   : int  30 39 26 36 22 18 32 26
 $ TEMP  : int  90 85 70 80 80 85 90 85
 $ CAT   : num  1.5 1 1.5 2 1 2.5 1 2
 $ METHOD: Factor w/ 2 levels "A","B": 1 1 2 1 2 2 1 2
```

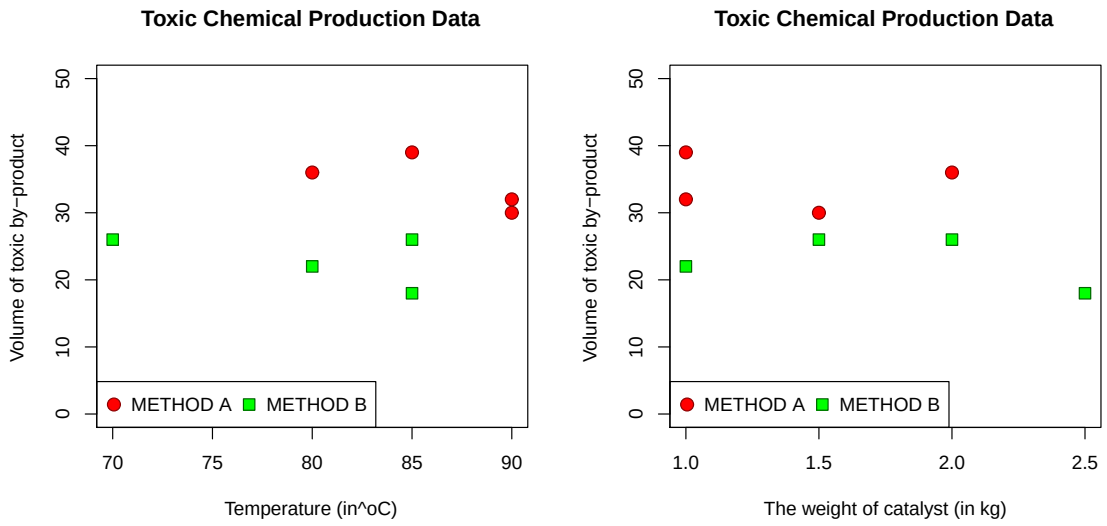
```
> head(data)
```

```
  VOL TEMP CAT METHOD
1  30   90 1.5      A
2  39   85 1.0      A
3  26   70 1.5      B
4  36   80 2.0      A
5  22   80 1.0      B
6  18   85 2.5      B
```

Datový soubor obsahuje 3 spojité proměnné (`VOL`, `TEMP` , `CAT`) a jednu kategoriální (`METHOD`). Jako závisle proměnnou budeme dále uvažovat proměnnou `VOL`. Na následujících obrázcích znázorníme vztah mezi závisle proměnnou a ostatními nezávisle proměnnými, přičemž graficky rozlišíme, o kterou metodu jde.

Máme celou řadu možností, jak to provést. Nejprve použijeme příkaz `plot()` a volbou parametrů `mfrow=c(1,2)` u příkazu `par()` ještě zařídíme, aby se oba dva grafy vykreslily vedle sebe.

```
> LA <- data$METHOD == "A"
> LB <- data$METHOD == "B"
> par(mfrow = c(1, 2))
> plot(data$VOL ~ data$TEMP, type = "n", ylim = c(0, 50),
     xlab = "Temperature (in^oC)", ylab = "Volume of toxic by-product",
     main = "Toxic Chemical Production Data")
> points(data$TEMP[LA], data$VOL[LA], pch = 21, col = "darkred",
     bg = "red", cex = 1.5)
> points(data$TEMP[LB], data$VOL[LB], pch = 22, col = "darkgreen",
     bg = "green", cex = 1.5)
> legend("bottomleft", paste("METHOD", c("A", "B")), col = c("darkred",
     "darkgreen"), pch = 21:22, ncol = 2, cex = 1, pt.bg = c("red",
     "green"), pt.cex = 1.5)
> plot(data$VOL ~ data$CAT, type = "n", ylim = c(0, 50),
     xlab = "The weight of catalyst (in kg)", ylab = "Volume of toxic by-product",
     main = popis[1])
> points(data$CAT[LA], data$VOL[LA], pch = 21, col = "darkred",
     bg = "red", cex = 1.5)
> points(data$CAT[LB], data$VOL[LB], pch = 22, col = "darkgreen",
     bg = "green", cex = 1.5)
> legend("bottomleft", paste("METHOD", c("A", "B")), col = c("darkred",
     "darkgreen"), pch = 21:22, ncol = 2, cex = 1, pt.bg = c("red",
     "green"), pt.cex = 1.5)
```

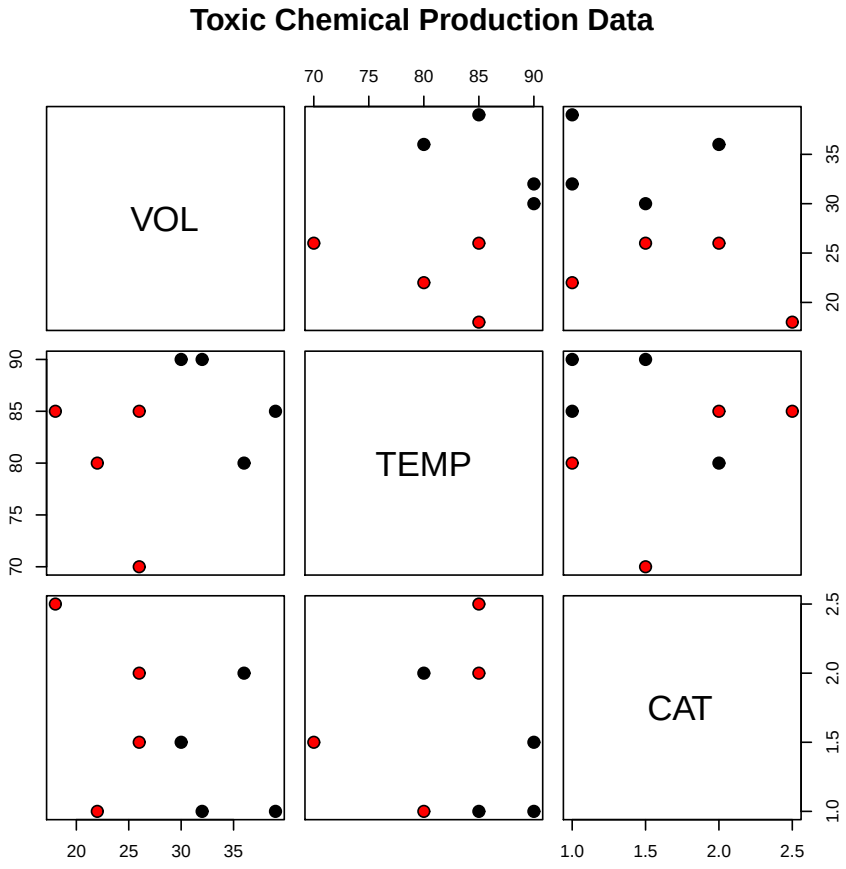


Obrázek 1: Dva bodové grafy vedle sebe pomocí příkazu `plot`.

S mnohem menším úsilím dosáhneme něčeho podobného tím, že zvolíme maticový bodový graf, který získáme pomocí příkazu `pairs()`.

Všimněme si dále, jak jednoduše lze díky volbě `pch=21` (vyplněný znak) a především s využitím volby `bg=as.numeric(data$METHOD)` barevně odlišit obě dvě metody.

```
> pairs(VOL ~ TEMP + CAT, data = data, pch = 21, bg = as.numeric(data$METHOD),
     cex = 1.5, main = popis[1])
```

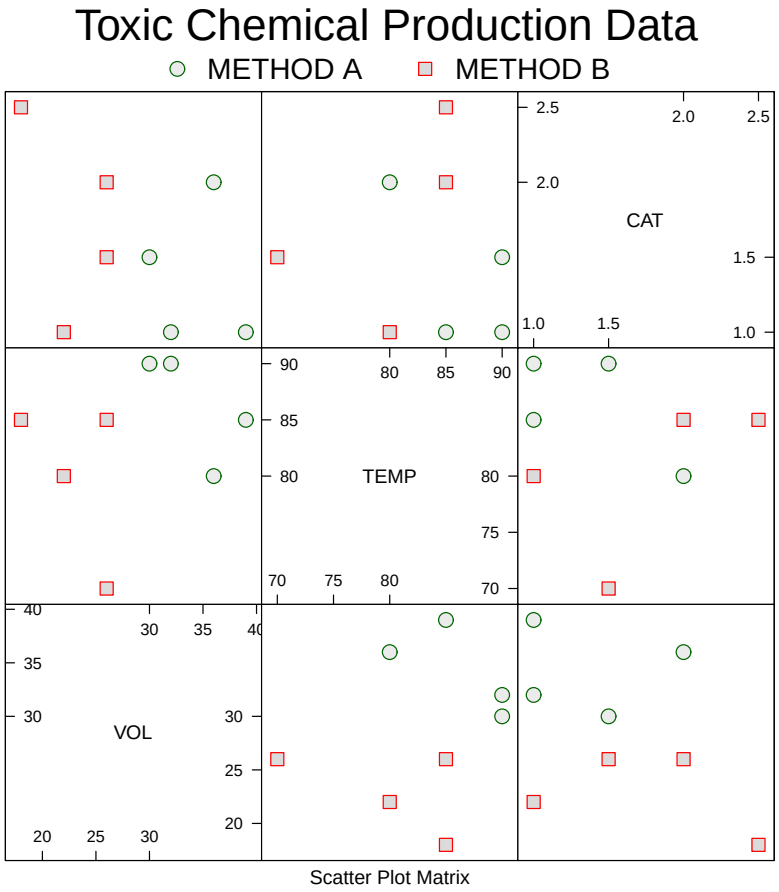


Obrázek 2: Maticové bodové grafy pomocí příkazu `pairs`.

Bodový graf můžeme získat také pomocí grafů z knihovny `lattice`, například díky příkazu `splom()`.

Chceme-li přidat legendu (volby parametrů u `key`), je třeba ošetřit, aby se tiskly stejné symboly ve stejných barvách jak v legendě tak v jednotlivých panelech.

```
> library(lattice)
> trellis.par.set(col.whitebg())
> super.sym <- trellis.par.get("superpose.symbol")
> super.sym$pch <- c(21, 22)
> super.sym$cex <- rep(1.5, length(super.sym$pch))
> graf <- splom(~data[1:3], groups = METHOD, data = data,
    pch = c(21, 22), cex = 1.5, bg = c(1, 2), panel = panel.superpose,
    as.table = T, key = list(title = popis[1], columns = 2,
        cex = 1.5, points = Rows(super.sym, 1:2), text = list(paste("METHOD",
            c("A", "B")))))
> print(graf)
```



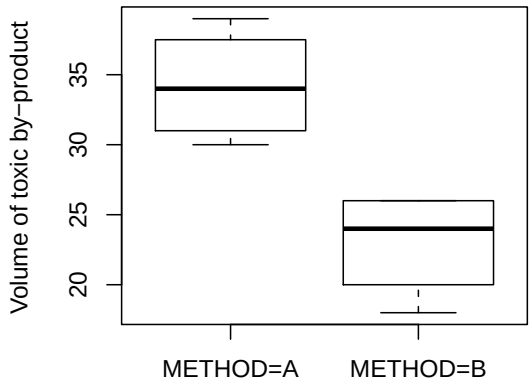
Obrázek 3: Maticové bodové grafy pomocí příkazu `splom` z knihovny `lattice`.

Analýza rozptylu proměnné VOL vzhledem k metodě A a B

Dříve než se pustíme do vytváření regresního modelu, je třeba ověřit homogenitu rozptylu u obou metod, což lze provést například pomocí příkazu `boxplot()` takto:

Opět máme celou řadu možností.

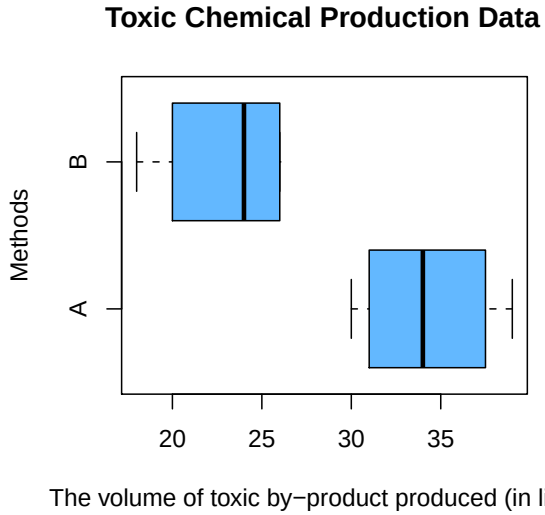
```
> boxplot(VOL ~ METHOD, data = data, ylab = "Volume of toxic by-product",
     names = paste("METHOD=", levels(data$METHOD), sep = ""))
```



Obrázek 4: Krabičkový graf pro `VOL` dle metod pomocí příkazu `boxplot`.

Další možností je použít jednoduchý příkaz `plot()`, který podle typu proměnných `VOL` (numerická proměnná) a `METHOD` (proměnná typu faktor) vytvoří krabičkový graf.

```
> plot(data$VOL ~ data$METHOD, main = popis[1], horizontal = TRUE,
     ylab = substr(popis[9], 9, nchar(popis[9])), xlab = "Methods",
     col = "steelblue1")
```



Obrázek 5: Krabičkový graf pro `VOL` dle metod pomocí příkazu `plot`.

Analýza rozptylu pomocí příkazu `lm()`

Uvažujme *ANOVA1* model $\boxed{Y_{jk} = \mu_j + \varepsilon_{jk} = \mu + \alpha_j + \varepsilon_{jk}}$, kde $j = 1, \ldots, a, \;\; k = 1, \ldots, n_j$. Díky příkazu

```
> summary(data)
```

```
      VOL             TEMP            CAT        METHOD
 Min.   :18.00   Min.   :70.00   Min.   :1.000   A:4
 1st Qu.:25.00   1st Qu.:80.00   1st Qu.:1.000   B:4
 Median :28.00   Median :85.00   Median :1.500
 Mean   :28.62   Mean   :83.12   Mean   :1.562
 3rd Qu.:33.00   3rd Qu.:86.25   3rd Qu.:2.000
 Max.   :39.00   Max.   :90.00   Max.   :2.500
```

vidíme, že jde o model s vyváženým plánem (*balanced design*), ve kterém je $a = 2, n_1 = n_2 = 4$.

K ověření předpokladu homoskedasticity můžeme využít například Bartletův test.

```
> bartlett.test(VOL ~ METHOD, data = data)
```

```
    Bartlett test of homogeneity of variances

data:  VOL by METHOD
Bartlett,s K-squared = 0.0068, df = 1, p-value = 0.9345
```

To, co naznačovaly boxploty, ukázal i výsledek Bartletova testu. Protože p–hodnota není menší než 0.05, rozdílnost v rozptylech jednotlivých metod se nepotvrdila. Můžeme tedy uvažovat klasický regresní model.

Chceme–li vytvořit *ANOVA1* model a vypsat výsledky, v jazyce R píšeme

```
> m.lm <- lm(VOL ~ METHOD, data = data)
> summary(m.lm)
```

```
Call:
lm(formula = VOL ~ METHOD, data = data)

Residuals:
   Min     1Q Median     3Q    Max
-5.000 -2.750  0.375  3.000  4.750

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   34.250      1.966  17.422 2.29e-06 ***
METHODB      -11.250      2.780  -4.047  0.00675 **
---
Signif. codes:  0 ,***, 0.001 ,**, 0.01 ,*, 0.05 ,., 0.1 , , 1

Residual standard error: 3.932 on 6 degrees of freedom
Multiple R-squared: 0.7318, Adjusted R-squared: 0.6871
F-statistic: 16.37 on 1 and 6 DF,  p-value: 0.006752
```

Dříve než budeme interpretovat výsledky, pro správné pochopení kódování jednotlivých úrovní použijeme příkaz, pomocí kterého vypíšeme matici plánu

```
> model.matrix(m.lm)[, ]
```

```
  (Intercept) METHODB
1           1       0
2           1       0
3           1       1
4           1       0
5           1       1
6           1       1
7           1       0
8           1       1
```

Ještě vypíšeme vstupní data

```
> data
```

```
  VOL TEMP CAT METHOD
1  30   90 1.5      A
2  39   85 1.0      A
3  26   70 1.5      B
4  36   80 2.0      A
5  22   80 1.0      B
6  18   85 2.5      B
7  32   90 1.0      A
8  26   85 2.0      B
```

a vidíme, že matice plánu je plné hodnosti (model není přeparametrizovaný). Toho bylo dosaženo tím, že se položilo $\alpha_1 = 0$, takže hodnota `(Intercept)` je odhadem střední hodnoty pro metodu A a hodnota `METHODB` je rozdílem mezi střední hodnotou pro metodu A a střední hodnotou pro metodu B. Protože p–hodnota pro testování hypotézy $H_0 : \ \alpha_2 = 0$ je menší než 0.05, hodnoty proměnné `VOL` se pro obě dvě metody významně liší.

Klasickou *ANOVA* tabulku dostaneme pomocí příklazu

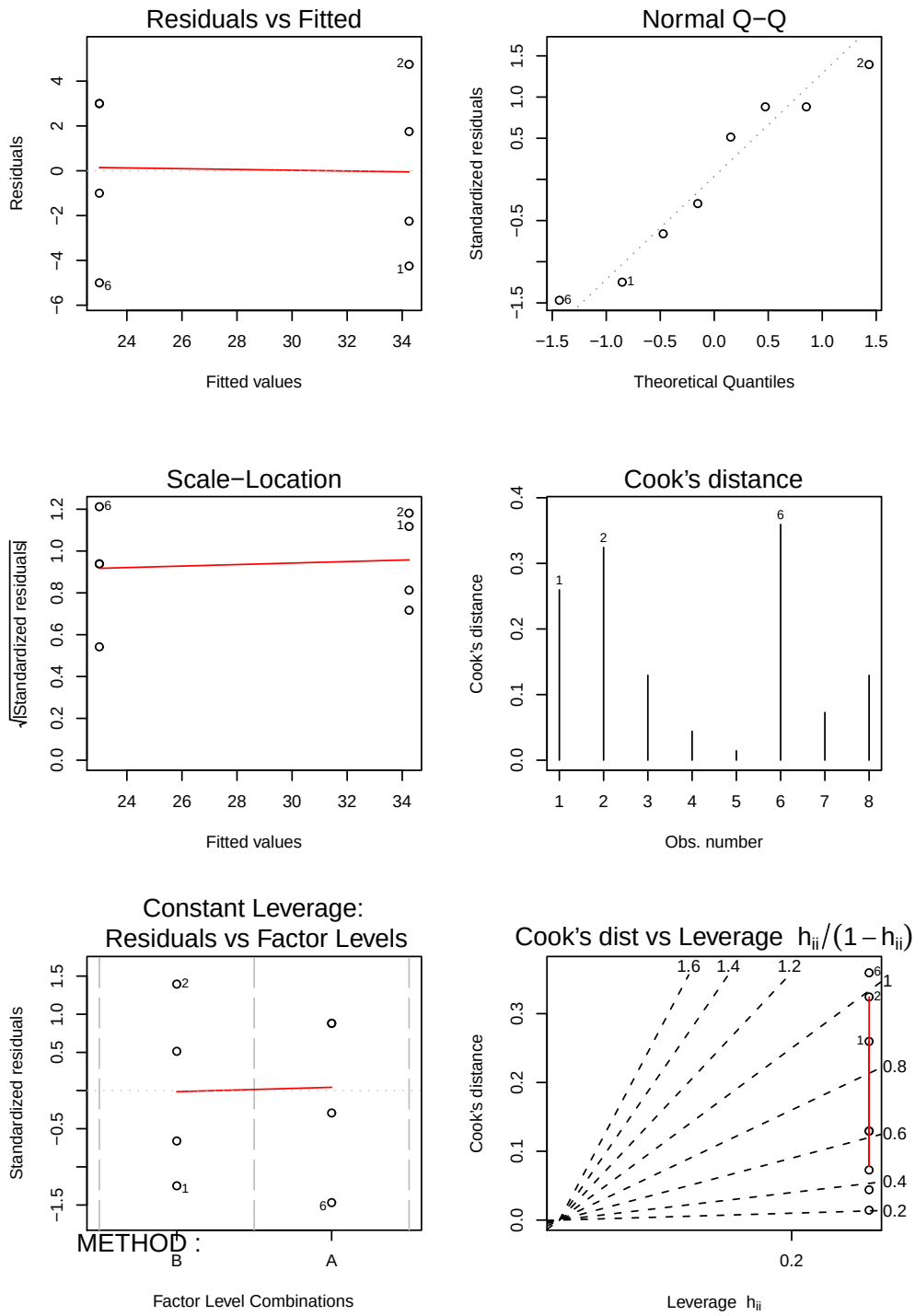
```
> anova(m.lm)
```

```
Analysis of Variance Table

Response: VOL
          Df Sum Sq Mean Sq F value   Pr(>F)
METHOD     1 253.12 253.125  16.375 0.006752 **
Residuals  6  92.75  15.458
---
Signif. codes:  0 ,***, 0.001 ,**, 0.01 ,*, 0.05 ,., 0.1 , , 1
```

Pomocí příkazu `plot()` dostaneme implicitně 4 grafy, které se týkají analýzy rezidují. Budeme–li chtít všechny grafy, použijeme volbu `which`

```
> par(mfrow = c(3, 2))
> plot(m.lm, which = 1:6)
```



Obrázek 6: Analýza reziduí pomocí příkazu `plot`.

Analýza rozptylu pomocí příkazu `glm()`

Protože normální rozdělení je regulární rozdělení exponenciálního typu, podíváme se, jaké výsledky dostaneme, použijeme–li GLM model.

```
> m.glm <- glm(VOL ~ METHOD, data = data, family = gaussian)
> summary(m.glm)


Call:
glm(formula = VOL ~ METHOD, family = gaussian, data = data)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-5.000  -2.750   0.375   3.000   4.750

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   34.250      1.966  17.422 2.29e-06 ***
METHODB      -11.250      2.780  -4.047  0.00675 **
---
Signif. codes:  0 ,***, 0.001 ,**, 0.01 ,*, 0.05 ,., 0.1 , , 1

(Dispersion parameter for gaussian family taken to be 15.45833)

    Null deviance: 345.88  on 7  degrees of freedom
Residual deviance:  92.75  on 6  degrees of freedom
AIC: 48.307

Number of Fisher Scoring iterations: 2


> anova(m.glm, test = "F")


Analysis of Deviance Table

Model: gaussian, link: identity

Response: VOL

Terms added sequentially (first to last)


       Df Deviance Resid. Df Resid. Dev      F   Pr(>F)
NULL                       7     345.88
METHOD  1   253.12         6      92.75 16.375 0.006752 **
---
Signif. codes:  0 ,***, 0.001 ,**, 0.01 ,*, 0.05 ,., 0.1 , , 1
```

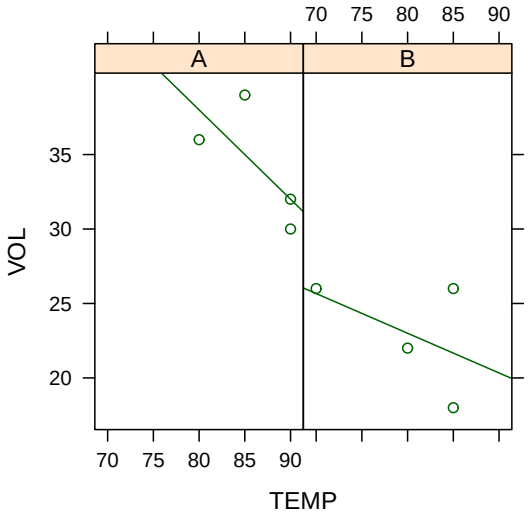
Ihned vidíme, že výsledky jsou stejné.

Analýza kovariance

Nyní se budeme snažit vyšetřit, zda na hodnoty proměnné `VOL` (*The volume of toxic by-product produced (in litres)*) nemají vliv kromě metody (což je proměnná `METDHOD` – typu faktor) také další kovariáty.

Uvažujme nejprve proměnnou `TEMP` (*The temperature of the manufacturing process (in°C)*). Pomocí příkazu `xyplot()` z knihovny `lattice` můžeme vytvořit velmi užitečný graf. Volbou `type=c("p","r")` dosáhneme toho, že v každém panelu je kromě bodů vykreslena také regresní přímka.

```
> print(xyplot(VOL ~ TEMP | METHOD, data, type = c("p", "r")))
```



Obrázek 7: Pro každou metodu zvlášť je vykreslen bodový graf (`VOL` vs `TEMP`) spolu s regresní přímkou pomocí příkazu `xyplot` z knihovny `lattice`.

Z grafu je patrné, že přímky mají rozdílný průsečík i směrnici, proto uvažujme následující GLM model s formulí ve tvaru `VOL ~ METHOD * TEMP`. Opět vypíšeme matici plánu, výsledky a *ANOVA* tabulku deviancí.

```
> mTemp.glm <- glm(VOL ~ METHOD * TEMP, data, family = gaussian)
> model.matrix(mTemp.glm)[, ]
```

```
  (Intercept) METHODB TEMP METHODB:TEMP
1           1       0   90            0
2           1       0   85            0
3           1       1   70           70
4           1       0   80            0
5           1       1   80           80
6           1       1   85           85
7           1       0   90            0
8           1       1   85           85
```

```
> summary(mTemp.glm)

Call:
glm(formula = VOL ~ METHOD * TEMP, family = gaussian, data = data)

Deviance Residuals:
         1          2          3          4          5          6
-2.000e+00  4.000e+00  3.333e-01 -2.000e+00 -1.000e+00 -3.667e+00
         7          8
 2.132e-14  4.333e+00

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   86.0000    39.4273   2.181   0.0946 .
METHODB      -41.6667    46.5795  -0.895   0.4216
TEMP          -0.6000     0.4566  -1.314   0.2591
METHODB:TEMP   0.3333     0.5514   0.605   0.5781
---
Signif. codes:  0 ,***, 0.001 ,**, 0.01 ,*, 0.05 ,., 0.1 , , 1

(Dispersion parameter for gaussian family taken to be 14.33333)

    Null deviance: 345.875  on 7  degrees of freedom
Residual deviance:  57.333  on 4  degrees of freedom
AIC: 48.459

Number of Fisher Scoring iterations: 2
```

Tentokrát přestaly být významné všechny parametry (viz p–hodnoty `Pr(>|t|)`) pro jednotlivé koeficienty. Při interpretaci těchto výsledků, musíme mít stále na paměti, že máme velmi málo pozorování.

```
> anova(mTemp.glm, test = "F")

Analysis of Deviance Table

Model: gaussian, link: identity

Response: VOL

Terms added sequentially (first to last)


            Df Deviance Resid. Df Resid. Dev       F  Pr(>F)
NULL                            7     345.88
METHOD       1  253.125         6      92.75 17.6599 0.01367 *
TEMP         1   30.179         5      62.57  2.1055 0.22039
METHOD:TEMP  1    5.238         4      57.33  0.3654 0.57811
---
Signif. codes:  0 ,***, 0.001 ,**, 0.01 ,*, 0.05 ,., 0.1 , , 1
```

Když se díváme na tabulku *ANOVA*, která nabízí analýzu deviancí, vidíme, že podstatného snížení deviancí se dosáhlo přidáním proměnné `METHOD`, pak již snižování stagnuje. I když tomu bodový graf pomocí `xyplot()` nenasvědčuje, vyzkoušejme jednodušší model, který předpokládá stejnou směrnici regresní přímky pro obě metody.

```
> mTemp2.glm <- glm(VOL ~ METHOD + TEMP, data, family = gaussian)
> model.matrix(mTemp2.glm)[, ]
```

```
  (Intercept) METHODB TEMP
1           1       0   90
2           1       0   85
3           1       1   70
4           1       0   80
5           1       1   80
6           1       1   85
7           1       0   90
8           1       1   85
```

```
> summary(mTemp2.glm)
```

```
Call:
glm(formula = VOL ~ METHOD + TEMP, family = gaussian, data = data)

Deviance Residuals:
      1        2       3       4       5       6       7       8
-2.8571   4.2857  -0.7143  -0.5714  -1.0000  -3.1429  -0.8571   4.8571

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  66.2857    20.7052   3.201  0.02396 *
METHODB     -13.5714     2.9141  -4.657  0.00555 **
TEMP         -0.3714     0.2392  -1.553  0.18115
---
Signif. codes:  0 ,***, 0.001 ,**, 0.01 ,*, 0.05 ,., 0.1 , , 1

(Dispersion parameter for gaussian family taken to be 12.51429)

    Null deviance: 345.875  on 7  degrees of freedom
Residual deviance:  62.571  on 5  degrees of freedom
AIC: 47.158

Number of Fisher Scoring iterations: 2
```

Nyní se koeficient `METHOD` stává významným, ale významnost `TEMP` se neprokázala. Dále zjistěme, zda jednodušší model příliš nezvýšil devianci.

```
> anova(mTemp.glm, mTemp2.glm, test = "F")
```

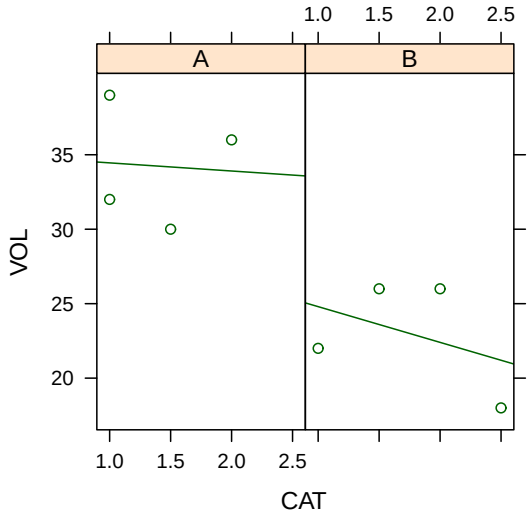
```
Analysis of Deviance Table

Model 1: VOL ~ METHOD * TEMP
Model 2: VOL ~ METHOD + TEMP
  Resid. Df Resid. Dev Df Deviance      F Pr(>F)
1         4     57.333
2         5     62.571 -1  -5.2381 0.3654 0.5781
```

Z výsledků je patrné, že významné zhoršení modelu se neprokázalo (viz p–hodnota `Pr(>F)`).

Nyní se budeme věnovat další proměnné, a to `CAT` (*The weight of catalyst (in kg)*) Opět pomocí příkazu `xyplot()` z knihovny `lattice` vykreslíme regresní přímky pro jednotlivé metody.

```
> print(xyplot(VOL ~ CAT | METHOD, data, type = c("p", "r")))
```



Obrázek 8: Pro každou metodu zvlášť je vykreslen bodový graf (`VOL` vs CAT) spolu s regresní přímkou pomocí příkazu `xyplot` z knihovny `lattice`.

Vidíme, že přímky mají rozdílný průsečík i směrnici, proto vytvořme GLM model s formulí `VOL ~ METHOD * CAT`.

```
> mCat.glm <- glm(VOL ~ METHOD * CAT, data, family = gaussian)
> model.matrix(mCat.glm)[, ]
```

```
  (Intercept) METHODB CAT METHODB:CAT
1           1       0 1.5         0.0
2           1       0 1.0         0.0
3           1       1 1.5         1.5
4           1       0 2.0         0.0
5           1       1 1.0         1.0
6           1       1 2.5         2.5
7           1       0 1.0         0.0
8           1       1 2.0         2.0
```

```
> summary(mCat.glm)
```

```
Call:
glm(formula = VOL ~ METHOD * CAT, family = gaussian, data = data)

Deviance Residuals:
     1       2       3       4       5       6       7       8
-4.182   4.545   2.400   2.091  -2.800  -3.200  -2.455   3.600
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  35.0000     8.0006   4.375   0.0119 *
METHODB      -7.8000    11.0280  -0.707   0.5184
CAT          -0.5455     5.5709  -0.098   0.9267
METHODB:CAT  -1.8545     6.9357  -0.267   0.8024
---
Signif. codes:  0 ,***, 0.001 ,**, 0.01 ,*, 0.05 ,., 0.1 , , 1

(Dispersion parameter for gaussian family taken to be 21.33636)

    Null deviance: 345.875  on 7  degrees of freedom
Residual deviance:  85.345  on 4  degrees of freedom
AIC: 51.641

Number of Fisher Scoring iterations: 2
```

Tentokrát kromě průsečíku přestaly být významné všechny parametry (viz p–hodnoty `Pr(>|t|)`) pro jednotlivé koeficienty.

```
> anova(mCat.glm, test = "F")

Analysis of Deviance Table

Model: gaussian, link: identity

Response: VOL

Terms added sequentially (first to last)


           Df Deviance Resid. Df Resid. Dev       F  Pr(>F)
NULL                          7     345.88
METHOD      1  253.125         6      92.75 11.8635 0.02619 *
CAT         1    5.879         5      86.87  0.2755 0.62739
METHOD:CAT  1    1.526         4      85.35  0.0715 0.80239
---
Signif. codes:  0 ,***, 0.001 ,**, 0.01 ,*, 0.05 ,., 0.1 , , 1
```

Z analýzy deviancí vyplývá, že pouze přidáním proměnné `METHOD` došlo k podstatnému snížení deviancí. Proto opět uvažujme jednodušší model se stejnou směrnicí obou regresních přímek.

```
> mCat2.glm <- glm(VOL ~ METHOD + CAT, data, family = gaussian)
> model.matrix(mCat2.glm)[, ]

  (Intercept) METHODB CAT
1           1       0 1.5
2           1       0 1.0
3           1       1 1.5
4           1       0 2.0
5           1       1 1.0
6           1       1 2.5
7           1       0 1.0
8           1       1 2.0
```

```
> summary(mCat2.glm)


Call:
glm(formula = VOL ~ METHOD + CAT, family = gaussian, data = data)

Deviance Residuals:
     1       2       3       4       5       6       7       8
-4.032   4.097   2.565   2.839  -2.306  -3.694  -2.903   3.435

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   36.645      4.615   7.941  0.00051 ***
METHODB      -10.597      3.154  -3.360  0.02011 *
CAT           -1.742      2.995  -0.582  0.58601
---
Signif. codes:  0 ,***, 0.001 ,**, 0.01 ,*, 0.05 ,., 0.1 , , 1

(Dispersion parameter for gaussian family taken to be 17.37419)

    Null deviance: 345.875  on 7  degrees of freedom
Residual deviance:  86.871  on 5  degrees of freedom
AIC: 49.783

Number of Fisher Scoring iterations: 2
```

Teď se koeficient `METHOD` stává významným, ale významnost `CAT` se neprokázala. Ještě zjistěme, zda jednodušší model příliš nezvýšil devianci.

```
> anova(mCat.glm, mCat2.glm, test = "F")


Analysis of Deviance Table

Model 1: VOL ~ METHOD * CAT
Model 2: VOL ~ METHOD + CAT
  Resid. Df Resid. Dev Df Deviance      F Pr(>F)
1         4     85.345
2         5     86.871 -1  -1.5255 0.0715 0.8024
```

Z výsledků je patrné, že významné zhoršení modelu se neprokázalo (viz p–hodnota `Pr(>F)`).

Na závěr ještě prozkoumejme, jak dopadne model, když budeme uvažovat obě kovariáty.

```
> mTempCat.glm <- glm(VOL ~ METHOD * (TEMP + CAT), data, family = gaussian)
> model.matrix(mTempCat.glm)[, ]


  (Intercept) METHODB TEMP CAT METHODB:TEMP METHODB:CAT
1           1       0   90 1.5            0         0.0
2           1       0   85 1.0            0         0.0
3           1       1   70 1.5           70         1.5
4           1       0   80 2.0            0         0.0
5           1       1   80 1.0           80         1.0
6           1       1   85 2.5           85         2.5
7           1       0   90 1.0            0         0.0
8           1       1   85 2.0           85         2.0
```

```
> summary(mTempCat.glm)


Call:
glm(formula = VOL ~ METHOD * (TEMP + CAT), family = gaussian,
    data = data)

Deviance Residuals:
      1       2       3       4       5       6       7       8
 0.6667  0.6667  0.6190 -0.3333 -1.8571 -3.0952 -1.0000  4.3333

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  136.3333    61.8279   2.205    0.158
METHODB      -94.5714    68.2295  -1.386    0.300
TEMP          -1.0667     0.6464  -1.650    0.241
CAT           -7.3333     6.4644  -1.134    0.374
METHODB:TEMP   0.8571     0.7620   1.125    0.378
METHODB:CAT    6.1905     7.8311   0.791    0.512

(Dispersion parameter for gaussian family taken to be 17.09524)

    Null deviance: 345.875  on 7  degrees of freedom
Residual deviance:  34.190  on 2  degrees of freedom
AIC: 48.323

Number of Fisher Scoring iterations: 2


> anova(mTempCat.glm, test = "F")


Analysis of Deviance Table

Model: gaussian, link: identity

Response: VOL

Terms added sequentially (first to last)


            Df Deviance Resid. Df Resid. Dev       F  Pr(>F)
NULL                            7     345.88
METHOD       1  253.125         6      92.75 14.8068 0.06138 .
TEMP         1   30.179         5      62.57  1.7653 0.31528
CAT          1    2.590         4      59.98  0.1515 0.73462
METHOD:TEMP  1   15.108         3      44.87  0.8838 0.44641
METHOD:CAT   1   10.683         2      34.19  0.6249 0.51208
---
Signif. codes:  0 ,***, 0.001 ,**, 0.01 ,*, 0.05 ,., 0.1 , , 1
```

Tentokrát přestaly být významné všechny parametry. Vzhledem k tomu, že máme 8 pozorování a 6 neznámých parametrů, hned si uvědomíme, že jsme to se složitostí modelu přehnali.

Z předchozí analýzy je vidět, jak mnoho kroků i pro tak jednoduchá data jsme museli provést při hledání optimálního modelu, a to nám pořád není jasné, který model by byl nejvhodnější.

V jazyku R je naštěstí k dispozici stepwise procedura (příkaz `step()`), pomocí které lze model nalézt automaticky. Děje se tak pomocí AIC kritéria (*Akaikeovo informační kritérium*), které je vypočteno na základě deviace a počtu regresorů. Čím je nižší hodnota AIC, tím je model vhodnější.

Nejprve vytvoříme nulový model pomocí formule `VOL ~ 1`.

```
> summary(mNULL <- glm(VOL ~ 1, data, family = gaussian))


Call:
glm(formula = VOL ~ 1, family = gaussian, data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-10.625   -3.625   -0.625    4.375   10.375

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   28.625      2.485   11.52 8.37e-06 ***
---
Signif. codes:  0 ,***, 0.001 ,**, 0.01 ,*, 0.05 ,., 0.1 , , 1

(Dispersion parameter for gaussian family taken to be 49.41071)

    Null deviance: 345.88  on 7  degrees of freedom
Residual deviance: 345.88  on 7  degrees of freedom
AIC: 56.836

Number of Fisher Scoring iterations: 2
```

V příkazu `step()` uvedeme vedle nulového modelu i nejbohatší model, který chceme uvažovat. Pokud neuvedeme jinak, stepwise procedura bude na základě AIC kritéria přidávat a ubírat proměnné tak dlouho, dokud další změna nepřinese zlepšení AIC kritéria. Každý krok je komentován.

```
> s0m0 <- step(mNULL, scope = ~METHOD * (TEMP + CAT))


Start:  AIC=56.84
VOL ~ 1

         Df Deviance    AIC
+ METHOD  1    92.75 48.307
<none>       345.88 56.836
+ CAT     1   282.99 57.231
+ TEMP    1   334.00 58.557

Step:  AIC=48.31
VOL ~ METHOD

         Df Deviance    AIC
+ TEMP    1    62.57 47.158
<none>         92.75 48.307
+ CAT     1    86.87 49.783
```

```
- METHOD  1   345.88 56.836

Step:  AIC=47.16
VOL ~ METHOD + TEMP

              Df Deviance    AIC
<none>              62.57 47.158
- TEMP         1    92.75 48.307
+ METHOD:TEMP  1    57.33 48.459
+ CAT          1    59.98 48.820
- METHOD       1   334.00 58.557
```

Všimněme si, že jednou z možných změn je také žádná změna, označeno `<none>`, to znamená, že jde o porovnání se současným stavem.

V prvním kroku lze proměnné pouze přidávat a je vidět, že pouze přidáním proměnné `METHOD` se kritérium AIC zlepšilo. Dostali jsme model s formulí `VOL ~ METHOD`.

V druhém kroku lze přidávat i odebírat proměnné (tedy jednu proměnnou `METHOD`). Ukazuje se, že pouze přidáním proměnné `TEMP` se AIC kritérium zlepší. Výsledkem tohoto kroku je model s formulí `VOL ~ METHOD + TEMP`.

Ve třetím kroku opět lze ubírat a přidávat proměnné. Ale ukáže se, že žádná změna nepřináší zlepšení AIC, proto procedura končí a jako optimální je vybrán model s formulí `VOL ~ METHOD + TEMP`.