R instructions for the 12th seminar

Logistic Regression where F is a binary factor and x1-x3 are continuous predictors fit ¡- glm(F x1+x2+x3,data=mydata,family=binomial()) summary(fit) display results confint(fit) 95exp(coef(fit)) exponentiated coefficients exp(confint(fit)) 95predict(fit, type="response") predicted values residuals(fit, type="deviance") residuals

```
load("GermanCredit.RData")
dim(GermanCredit)
library(DescTools)
WhichFactors(GermanCredit)
WhichNumerics(GermanCredit)
```

R Instructions for the problem 1:

1.
2.
3.
4. For our data "not paying back the credit" is a success. (We are modeling "not paying back"). In R philosophy the first level in any factor variable is treated as "failure". So firstly look into data set at first three variables and check the levels of `ffClass` and `fClass`.

   `levels(GermanCredit$fClass)` and `levels(GermanCredit$fClass)`

   (`ffClass` is for our problem appropriate.)

   Create the model itself:

```
model<-glm(ffClass~A00Amount100,data=GermanCredit,family=binomial(link="logit"))
```

   To obtain parameter estimates $\hat{\beta}_k$:

   `summary(model)`

   or simply:

   `coef(model)`

   $\hat{\beta}_0 = -1,23$; $\hat{\beta}_1 = 0,0112$

   To obtain parameter estimates $e^{\hat{\beta}_k}$:

   `exp(coef(model))`

5. To obtain confidence intervals for β_k:

   `confint(model,level = 0.95)`

   $\beta_0 \in (-1,44\ ;\ -1,02)$ $\beta_1 \in (0,0066\ ;\ 0,0158)$

6. To obtain confidence intervals for $e^{\hat{\beta}_k}$:

   `exp(confint(model,level = 0.95))`

7. p-values for testing parameter's significance via Wald statistics:

   `summary(model)`

   p-values are in the collumn "$Pr(> |z|)$".

8. `Null Deviance` $= D_0 = -2\log L_0$, where L_0 is a maximum likelihood of a "null" model with nothing but an intercept. (In our case $logit(p(x_1)) = \beta_0$)

   `Residual Deviance` $= D_1 = -2\log L_1$, where L_1 is a maximum likelihood of a "full" model with all predictors. (In our case $logit(p(x_1)) = \beta_0 + \beta_1 x_1$)

   Ratio $LR_{0,1} = D_0 - D_1 = -2\log\frac{L_0}{L_1} \approx \chi^2(df_0 - df_1)$.

Better model has smaller deviance. Significantly better full model then null model leads finally to large values of $LR_{0,1}$ which is considered to be a liklihood-ratio test statistic. Thus the concerned p-value is on the right tail of χ^2 distribution.

(For our problem the likelihood-ratio test is in fact a test about a significance of the parameter β_1. $LR_{0,1} = (1221.7 - 1199.1) = 22.6$; $LR_{0,1} \approx \chi^2(999 - 998)$ thus the $p = 0,000002$) and the full model is significantly better then the null model.

This p-value can be obtained also by:
`anova(model,test="Chi")`

`AIC`$= k - 2\log L_1 = k + D_1$, where $k = 2*$ number of parameters. Better model has smaller AIC. Compared with deviance, models are penalized for large number of parameters.

(For our problem the $AIC = 2 * 2 + 1199.1 = 1203.1$.)

9. Hosmer-Lemeshow I dont't have
   `residuals(model, type= "deviance")` (this deviance type is also in an output of summary)
   (other types: "deviance", "pearson", "working","response", "partial")
   `residuals(model, type= "response")`; (this response type means: observed minus probability of success)

10. Fitted values:
    logarithmic odds ratio $\log\left(\frac{p(\boldsymbol{x})}{1-p(\boldsymbol{x})}\right)$:
    `p1<-predict(model, type="link")`

    probability of success $p(\boldsymbol{x})$:
    `p2<-predict(model, type="response")`

overeni predpokladu linearity prave strany:

```
.........................................................................
gr<-rep(1:4,each=250)
dat<-data.frame(GermanCredit[,1:5])
dat<-dat[order(dat$A00Amount100),] #seradi tabulku vzestupne dle prom Amount100
dat<-data.frame(dat[,1:5],gr)
model1<-glm(ffClass~A00Amount100,data=dat[1:250,],family=binomial(link="logit"))
> model2<-glm(ffClass~A00Amount100,data=dat[250:500,],family=binomial(link="logit"))
> model3<-glm(ffClass~A00Amount100,data=dat[500:750,],family=binomial(link="logit"))
> model4<-glm(ffClass~A00Amount100,data=dat[750:1000,],family=binomial(link="logit"))$
> mean(pp1<-predict(model1, type="response"))
[1] 0.308
> mean(pp2<-predict(model2, type="response"))
[1] 0.247012
> mean(pp3<-predict(model3, type="response"))
[1] 0.2270916
> mean(pp4<-predict(model4, type="response"))
[1] 0.4183267
.........................................................................
```

Jelikoz prsti uspechu nejsou ve skupinach monotonni, nelze linearitu prave strany predpokladat