Lesson 15

Christopher A. Swenson (chris@cswenson.com)

11/27/2021

Leukemia remission (logistic regression)

Load the leukemia data. Fit a logistic regression model of REMISS vs CELL + SMEAR + INFIL + LI + BLAST + TEMP. Calculate 95% confidence intervals for the regression parameters based on asymptotic normality and based on profiling the least-squares estimation surface. Fit a logistic regression model of REMISS vs LI. Create a sctterplot of REMISS vs LI and add a fitted line based on the logistic regression model. Calculate the odds ratio for LI and a 95% confidence interval. Conduct a likelihood ratio (or deviance) test for LI. Calculate the sum of squared deviance residuals and the sum of squared Pearson residuals. Use the hoslem.test function in the ResourceSelection package to conduct the Hosmer-Lemeshow goodness-of-fit test. Calculate a version of for logistic regression. Create residual plots using Pearson and deviance residuals. Calculate hat values (leverages), studentized residuals, and Cook's distances.

```
leukemia <- read.table("./Data/leukemia_remission.txt", header=T)
attach(leukemia)

model.1 <- glm(REMISS ~ CELL + SMEAR + INFIL + LI + BLAST + TEMP, family="binomial")
summary(model.1)
```

```
##
## Call:
## glm(formula = REMISS ~ CELL + SMEAR + INFIL + LI + BLAST + TEMP,
##     family = "binomial")
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.95404  -0.66259  -0.02516   0.78184   1.57465
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   64.25808   74.96480   0.857    0.391
## CELL          30.83006   52.13520   0.591    0.554
## SMEAR         24.68632   61.52601   0.401    0.688
## INFIL        -24.97447   65.28088  -0.383    0.702
## LI             4.36045    2.65798   1.641    0.101
## BLAST         -0.01153    2.26634  -0.005    0.996
## TEMP        -100.17340   77.75289  -1.288    0.198
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 34.372  on 26  degrees of freedom
## Residual deviance: 21.594  on 20  degrees of freedom
## AIC: 35.594
##
```

```
## Number of Fisher Scoring iterations: 8
```

```
#                Estimate Std. Error z value Pr(>|z|)
# (Intercept)   64.25808   74.96480   0.857    0.391
# CELL          30.83006   52.13520   0.591    0.554
# SMEAR         24.68632   61.52601   0.401    0.688
# INFIL        -24.97447   65.28088  -0.383    0.702
# LI             4.36045    2.65798   1.641    0.101
# BLAST         -0.01153    2.26634  -0.005    0.996
# TEMP        -100.17340   77.75289  -1.288    0.198
#
# (Dispersion parameter for binomial family taken to be 1)
#
# Null deviance: 34.372  on 26  degrees of freedom
# Residual deviance: 21.594  on 20  degrees of freedom
# AIC: 35.594

confint.default(model.1) # based on asymptotic normality
```

```
##                    2.5 %     97.5 %
## (Intercept)  -82.6702241 211.186392
## CELL         -71.3530610 133.013183
## SMEAR        -95.9024389 145.275071
## INFIL       -152.9226458 102.973698
## LI            -0.8490839   9.569992
## BLAST         -4.4534852   4.430423
## TEMP        -252.5662623  52.219462
```

```
confint(model.1) # based on profiling the least-squares estimation surface
```

```
## Waiting for profiling to be done...

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
##                    2.5 %     97.5 %
## (Intercept)  -68.6041096 241.051733
## CELL         -29.2192655 153.577371
## SMEAR        -62.1384936 163.961186
## INFIL       -171.6480455  70.015471
## LI             0.3514287  10.958965
## BLAST         -4.6844284   4.511037
## TEMP        -284.3010127  23.356799
```

```r
model.2 <- glm(REMISS ~ LI, family="binomial")
summary(model.2)
```

```
##
## Call:
## glm(formula = REMISS ~ LI, family = "binomial")
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -1.9448  -0.6465  -0.4947   0.6571   1.6971
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.777      1.379  -2.740  0.00615 **
## LI             2.897      1.187   2.441  0.01464 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 34.372  on 26  degrees of freedom
## Residual deviance: 26.073  on 25  degrees of freedom
## AIC: 30.073
##
## Number of Fisher Scoring iterations: 4
```

```
#             Estimate Std. Error z value Pr(>|z|)
# (Intercept)   -3.777      1.379  -2.740  0.00615 **
# LI             2.897      1.187   2.441  0.01464 *
# ---
# Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
#
# (Dispersion parameter for binomial family taken to be 1)
#
# Null deviance: 34.372  on 26  degrees of freedom
# Residual deviance: 26.073  on 25  degrees of freedom
# AIC: 30.073

plot(x=LI, y=REMISS,
     panel.last = lines(sort(LI), fitted(model.2)[order(LI)]))

exp(coef(model.2)[2]) # odds ratio = 18.12449
```

```
##       LI
## 18.12449
```

```
exp(confint.default(model.2)[2,]) # 95% CI = (1.770284, 185.561725)
```

```
##      2.5 %     97.5 %
##   1.770284 185.561725
```

```
anova(model.2, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: REMISS
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                    26     34.372
## LI     1   8.2988        25     26.073 0.003967 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#      Df Deviance Resid. Df Resid. Dev Pr(>Chi)
# NULL                    26     34.372
# LI    1   8.2988        25     26.073 0.003967 **

sum(residuals(model.2, type="deviance")^2) # 26.07296
```

```
## [1] 26.07296
```

```
model.2$deviance # 26.07296
```

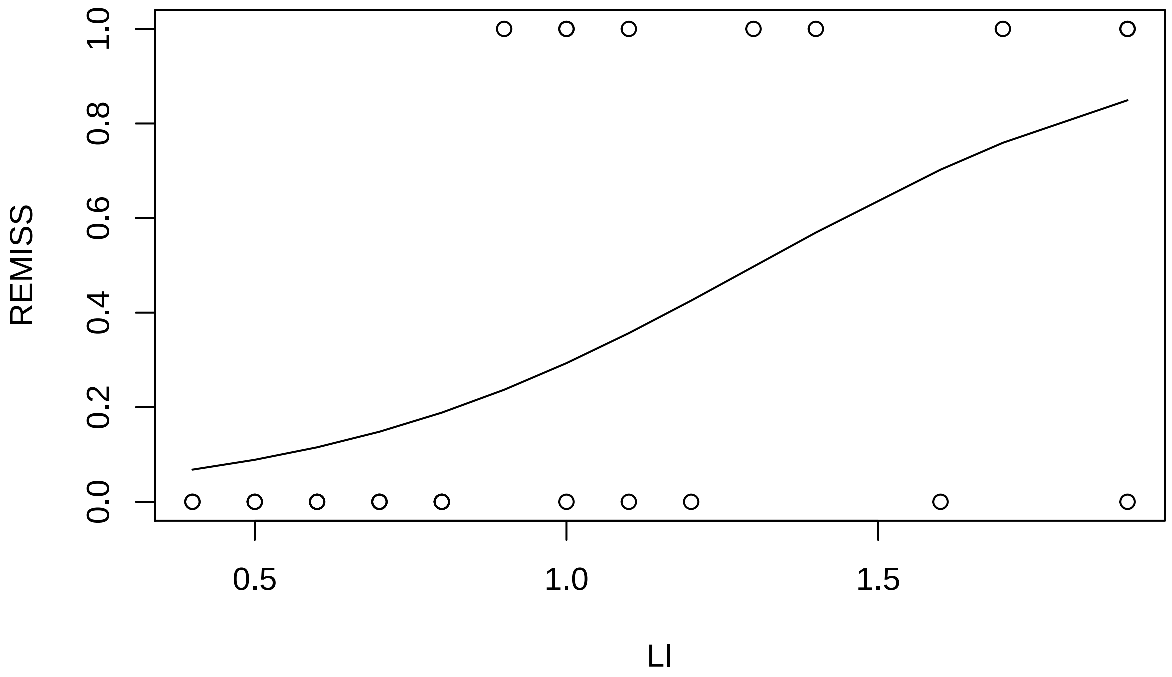
```
## [1] 26.07296
```

```
sum(residuals(model.2, type="pearson")^2) # 23.93298
```

```
## [1] 23.93298
```

```
library(ResourceSelection)
```

```
## ResourceSelection 0.3-5   2019-07-22
```



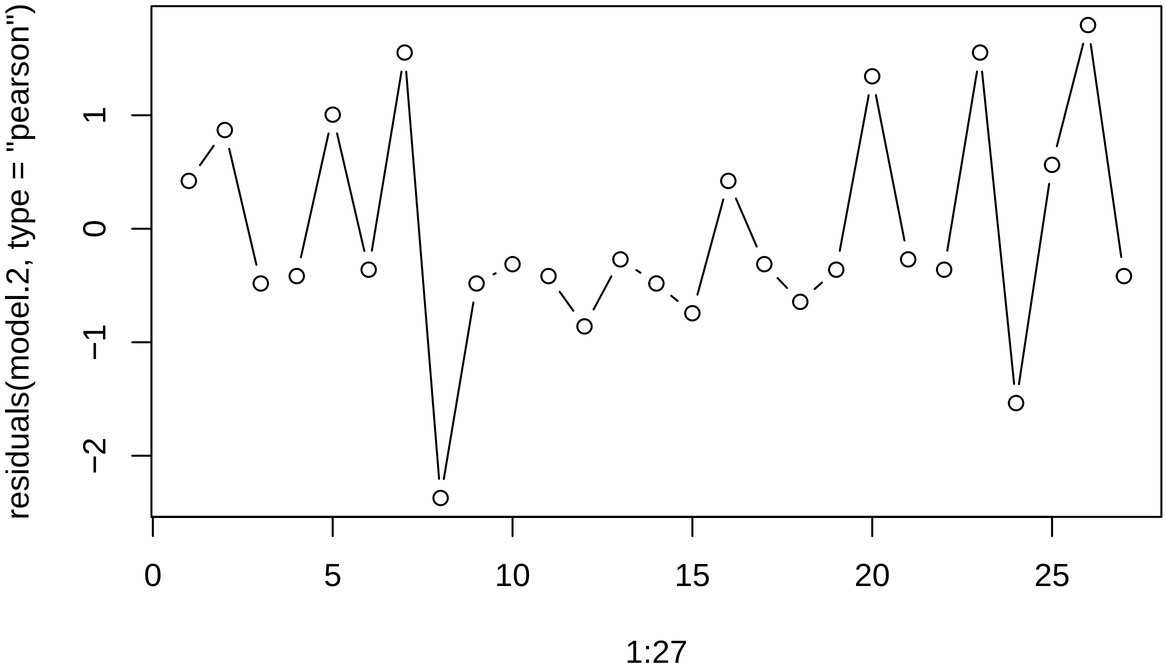
```
hoslem.test(model.2$y, fitted(model.2), g=9)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  model.2$y, fitted(model.2)
## X-squared = 7.3293, df = 7, p-value = 0.3954
```

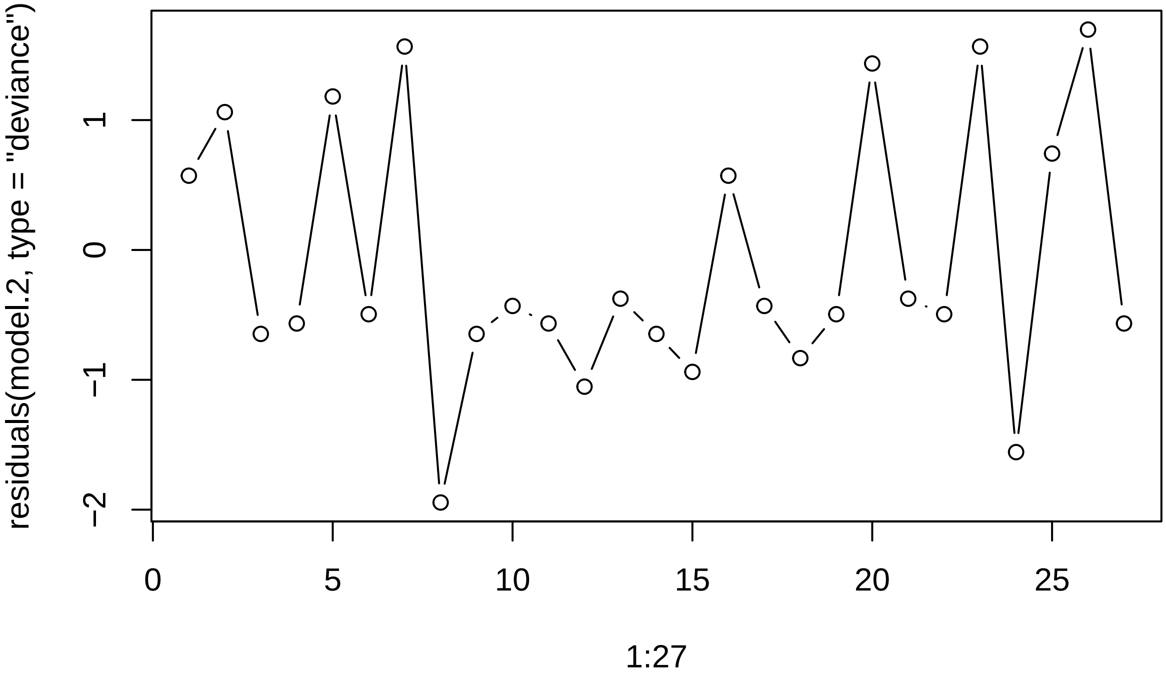
```
# Hosmer and Lemeshow goodness of fit (GOF) test
# data:  REMISS, fitted(model.2)
# X-squared = 7.3293, df = 7, p-value = 0.3954

1-model.2$deviance/model.2$null.deviance # "R-squared" = 0.2414424
```

```
## [1] 0.2414424
```

```
plot(1:27, residuals(model.2, type="pearson"), type="b")
```



```
plot(1:27, residuals(model.2, type="deviance"), type="b")
```



```
summary(influence.measures(model.2))
```

```
## Potentially influential observations of
##   glm(formula = REMISS ~ LI, family = "binomial") :
##
##   dfb.1_ dfb.LI dffit   cov.r cook.d hat
## 8  0.63  -0.83  -0.93_*  0.88  0.58   0.15
```

```
#  dfb.1_ dfb.LI dffit   cov.r cook.d hat
# 8  0.63  -0.83  -0.93_*  0.88  0.58   0.15

hatvalues(model.2)[8] # 0.1498395
```

```
##         8
## 0.1498395
```

```
residuals(model.2)[8] # -1.944852
```

```
##         8
## -1.944852
```

```
rstudent(model.2)[8] # -2.185013
```

```
##         8
## -2.185013
```

```
cooks.distance(model.2)[8] # 0.5833219
```

```
##         8
## 0.5833219
```

```
detach(leukemia)
```

Disease outbreak (logistic regression)

Load the disease outbreak data. Create interaction variables. Fit "full" logistic regression model of Disease vs four predictors and five interactions. Fit "reduced" logistic regression model of Disease vs four predictors. Conduct a likelihood ratio (or deviance) test for the five interactions. Display the analysis of deviance table with sequential deviances.

```
disease <- read.table("./Data/DiseaseOutbreak.txt", header=T)
attach(disease)

Age.Middle <- Age*Middle
Age.Lower <- Age*Lower
Age.Sector <- Age*Sector
Middle.Sector <- Middle*Sector
Lower.Sector <- Lower*Sector

model.1 <- glm(Disease ~ Age + Middle + Lower + Sector + Age.Middle + Age.Lower +
                 Age.Sector + Middle.Sector + Lower.Sector, family="binomial")
model.2 <- glm(Disease ~ Age + Middle + Lower + Sector, family="binomial")
anova(model.2, model.1, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: Disease ~ Age + Middle + Lower + Sector
## Model 2: Disease ~ Age + Middle + Lower + Sector + Age.Middle + Age.Lower +
##     Age.Sector + Middle.Sector + Lower.Sector
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        93    101.054
## 2        88     93.996  5   7.0583   0.2163
```

```
#   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
# 1        93    101.054
# 2        88     93.996  5   7.0583   0.2163

anova(model.1, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Disease
##
## Terms added sequentially (first to last)
##
##
##               Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                             97    122.318
## Age            1   7.4050        96    114.913 0.006504 **
## Middle         1   1.8040        95    113.109 0.179230
## Lower          1   1.6064        94    111.502 0.205003
## Sector         1  10.4481        93    101.054 0.001228 **
## Age.Middle     1   4.5697        92     96.484 0.032542 *
## Age.Lower      1   1.0152        91     95.469 0.313666
## Age.Sector     1   1.1202        90     94.349 0.289878
## Middle.Sector  1   0.0001        89     94.349 0.993427
## Lower.Sector   1   0.3531        88     93.996 0.552339
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#               Df Deviance Resid. Df Resid. Dev Pr(>Chi)
# NULL                             97    122.318
# Age            1   7.4050        96    114.913 0.006504 **
# Middle         1   1.8040        95    113.109 0.179230
# Lower          1   1.6064        94    111.502 0.205003
# Sector         1  10.4481        93    101.054 0.001228 **
# Age.Middle     1   4.5697        92     96.484 0.032542 *
# Age.Lower      1   1.0152        91     95.469 0.313666
# Age.Sector     1   1.1202        90     94.349 0.289878
# Middle.Sector  1   0.0001        89     94.349 0.993427
# Lower.Sector   1   0.3531        88     93.996 0.552339

detach(disease)
```

Toxicity and insects (logistic regression using event/trial data format)

Load the toxicity data. Create a Survivals variable and a matrix with Deaths in one column and Survivals in the other column. Fit a logistic regression model of Deaths vs Dose. Calculate 95% confidence intervals for the regression parameters based on asymptotic normality and based on profiling the least-squares estimation surface. Calculate the odds ratio for Dose and a 95% confidence interval. Display the observed and fitted probabilities. Create a sctterplot of observed probabilties vs Dose and add a fitted line based on the logistic regression model.

```
toxicity <- read.table("./Data/toxicity.txt", header=T)
attach(toxicity)

Survivals <- SampSize - Deaths
y <- cbind(Deaths, Survivals)
```

```
model.1 <- glm(y ~ Dose, family="binomial")
summary(model.1)
```

```
##
## Call:
## glm(formula = y ~ Dose, family = "binomial")
##
## Deviance Residuals:
##       1        2        3        4        5        6
## -0.5092  -0.1115   0.7461  -0.2869   0.4744  -0.5599
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.64367    0.15610  -16.93   <2e-16 ***
## Dose         0.67399    0.03911   17.23   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 383.0695  on 5  degrees of freedom
## Residual deviance:   1.4491  on 4  degrees of freedom
## AIC: 39.358
##
## Number of Fisher Scoring iterations: 3
```

```
#             Estimate Std. Error z value Pr(>|z|)
# (Intercept) -2.64367    0.15610  -16.93   <2e-16 ***
# Dose         0.67399    0.03911   17.23   <2e-16 ***
# ---
# Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
#
# (Dispersion parameter for binomial family taken to be 1)
#
# Null deviance: 383.0695  on 5  degrees of freedom
# Residual deviance:   1.4491  on 4  degrees of freedom
# AIC: 39.358

confint.default(model.1) # based on asymptotic normality
```

```
##                  2.5 %     97.5 %
## (Intercept) -2.9496351 -2.3377149
## Dose         0.5973404  0.7506451
```

```
confint(model.1) # based on profiling the least-squares estimation surface
```

```
## Waiting for profiling to be done...
```

```
##                  2.5 %     97.5 %
## (Intercept) -2.9554809 -2.3432165
## Dose         0.5985828  0.7519688
```

```
exp(coef(model.1)[2]) # odds ratio = 1.962056
```

```
##     Dose
## 1.962056
```

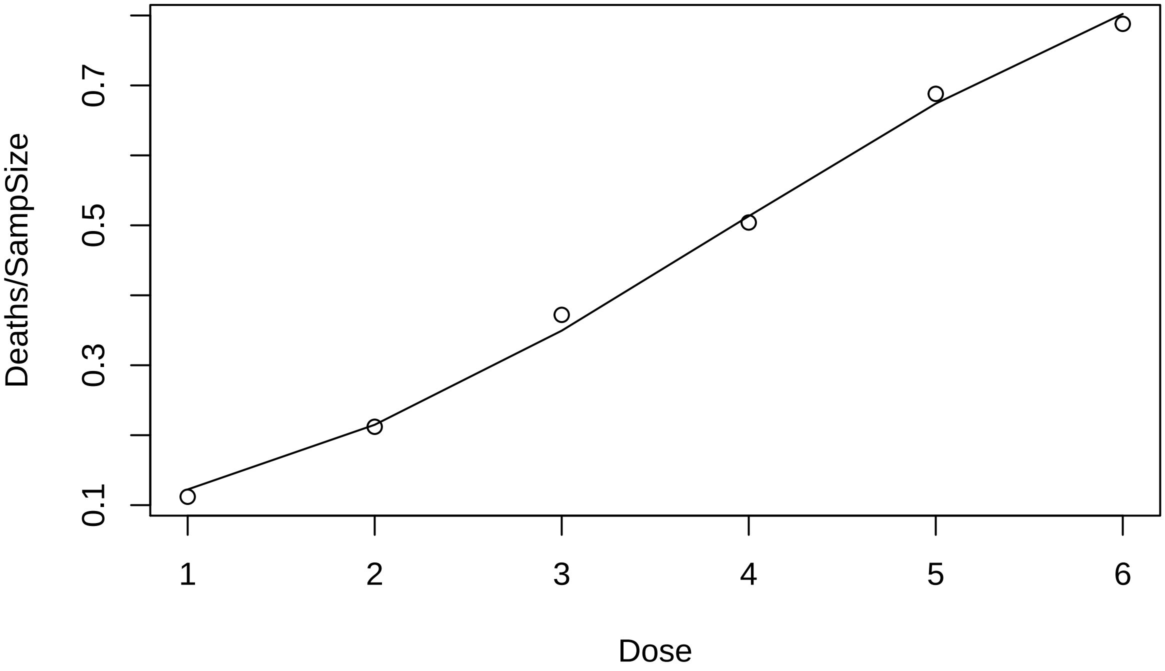
```
exp(confint.default(model.1)[2,]) # 95% CI = (1.817279, 2.118366)
```

```
##    2.5 %   97.5 %
## 1.817279 2.118366
```

```
cbind(Dose, SampSize, Deaths, Deaths/SampSize, fitted(model.1))
```

```
##   Dose SampSize Deaths
## 1    1      250     28 0.112 0.1224230
## 2    2      250     53 0.212 0.2148914
## 3    3      250     93 0.372 0.3493957
## 4    4      250    126 0.504 0.5130710
## 5    5      250    172 0.688 0.6739903
## 6    6      250    197 0.788 0.8022286
```

```
#   Dose SampSize Deaths
# 1    1      250     28 0.112 0.1224230
# 2    2      250     53 0.212 0.2148914
# 3    3      250     93 0.372 0.3493957
# 4    4      250    126 0.504 0.5130710
# 5    5      250    172 0.688 0.6739903
# 6    6      250    197 0.788 0.8022286

plot(x=Dose, y=Deaths/SampSize,
     panel.last = lines(sort(Dose), fitted(model.1)[order(Dose)]))
```

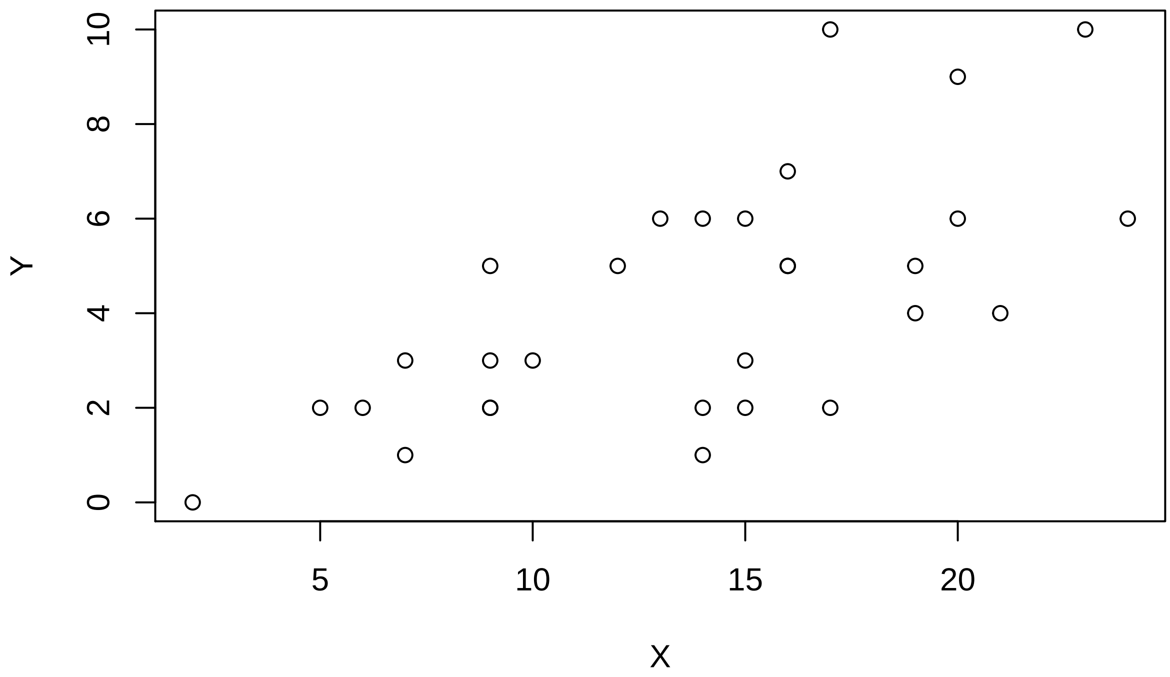


```
detach(toxicity)
```

Poisson example (Poisson regression)

Load the poisson data. Create a scatterplot of the data. Fit a Poisson regression model of y vs x. Calculate 95% confidence intervals for the regression parameters based on asymptotic normality and based on profiling the least-squares estimation surface. Create a sctterplot of y vs x and add a fitted line based on the Poisson regression model. Conduct a likelihood ratio (or deviance) test for x. Calculate the sum of squared deviance residuals and the sum of squared Pearson residuals and calculate p-values based on chi-squared goodness-of-fit tests. Calculate pseudo for Poisson regression. Create residual plots using Pearson and deviance residuals. Calculate hat values (leverages) and studentized residuals.

```
poisson <- read.table("./Data/poisson_simulated.txt", header=T)
attach(poisson)

plot(x=X, y=Y)
```



```
model.1 <- glm(Y ~ X, family="poisson")
summary(model.1)
```

```
##
## Call:
## glm(formula = Y ~ X, family = "poisson")
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.78079  -0.87855  -0.04969   0.77007   1.97433
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.30787    0.28943   1.064    0.287
## X            0.07636    0.01730   4.413 1.02e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 48.310  on 29  degrees of freedom
## Residual deviance: 27.842  on 28  degrees of freedom
## AIC: 124.5
##
## Number of Fisher Scoring iterations: 4
```

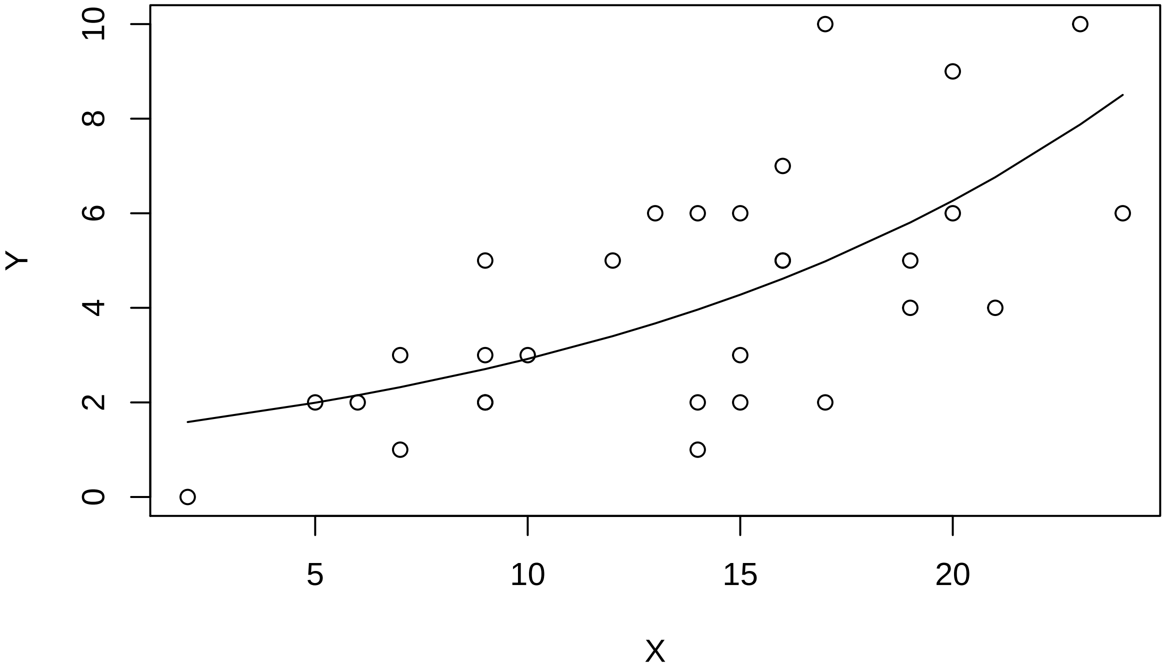
```
#              Estimate Std. Error z value Pr(>|z|)
# (Intercept)  0.30787    0.28943   1.064    0.287
# x            0.07636    0.01730   4.413 1.02e-05 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# (Dispersion parameter for poisson family taken to be 1)
#
# Null deviance: 48.310  on 29  degrees of freedom
# Residual deviance: 27.842  on 28  degrees of freedom
# AIC: 124.5

confint.default(model.1) # based on asymptotic normality
```

```
##                   2.5 %    97.5 %
## (Intercept) -0.25941568 0.8751485
## X            0.04244234 0.1102723
```

```
confint(model.1) # based on profiling the least-squares estimation surface
```

```
## Waiting for profiling to be done...

##                   2.5 %    97.5 %
## (Intercept) -0.28030073 0.8552531
## X            0.04284331 0.1107317
```

```
plot(x=X, y=Y,
     panel.last = lines(sort(X), fitted(model.1)[order(X)]))
```



```
anova(model.1, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
```

```
##
## Response: Y
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                   29     48.310
## X     1   20.468        28     27.842 6.065e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#      Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
# NULL                   29     48.310
# x     1   20.468        28     27.842 6.065e-06 ***

sum(residuals(model.1, type="deviance")^2) # 27.84209
```

```
## [1] 27.84209
```

```
model.1$deviance # 27.84209
```

```
## [1] 27.84209
```

```
pchisq(model.1$deviance, 28, lower.tail=F) # p-value = 0.4728389
```

```
## [1] 0.4728389
```

```
sum(residuals(model.1, type="pearson")^2) # 26.09324
```

```
## [1] 26.09324
```

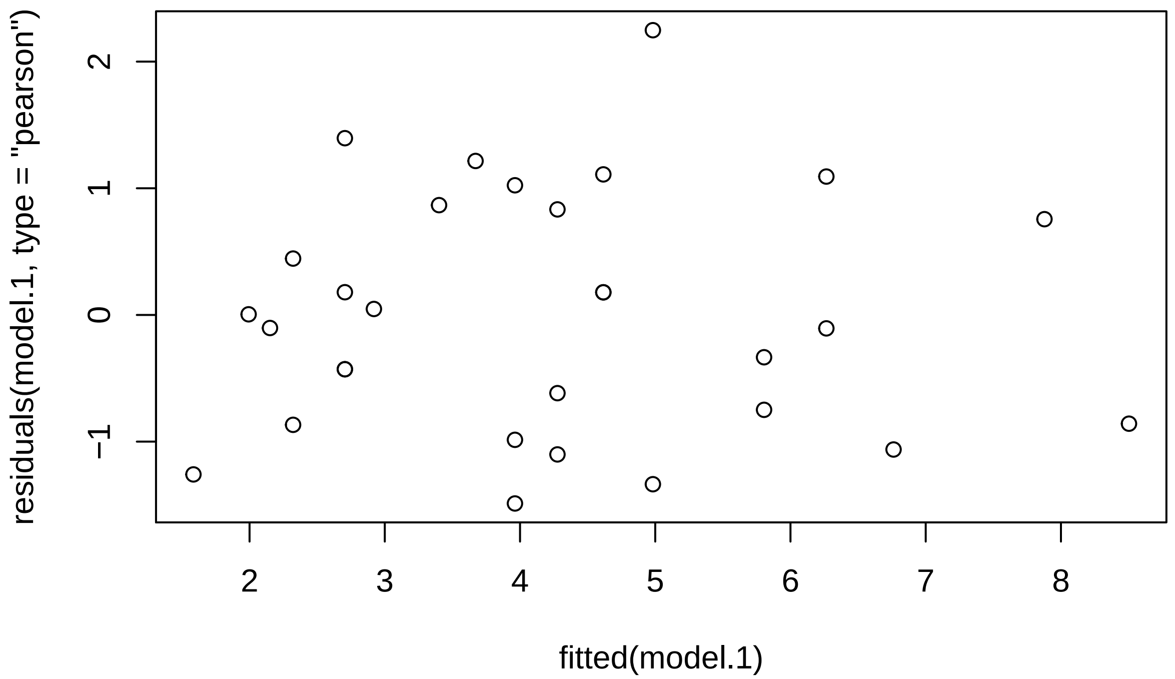
```
pchisq(sum(residuals(model.1, type="pearson")^2), 28, lower.tail=F) # p-value = 0.5679192
```

```
## [1] 0.5679192
```

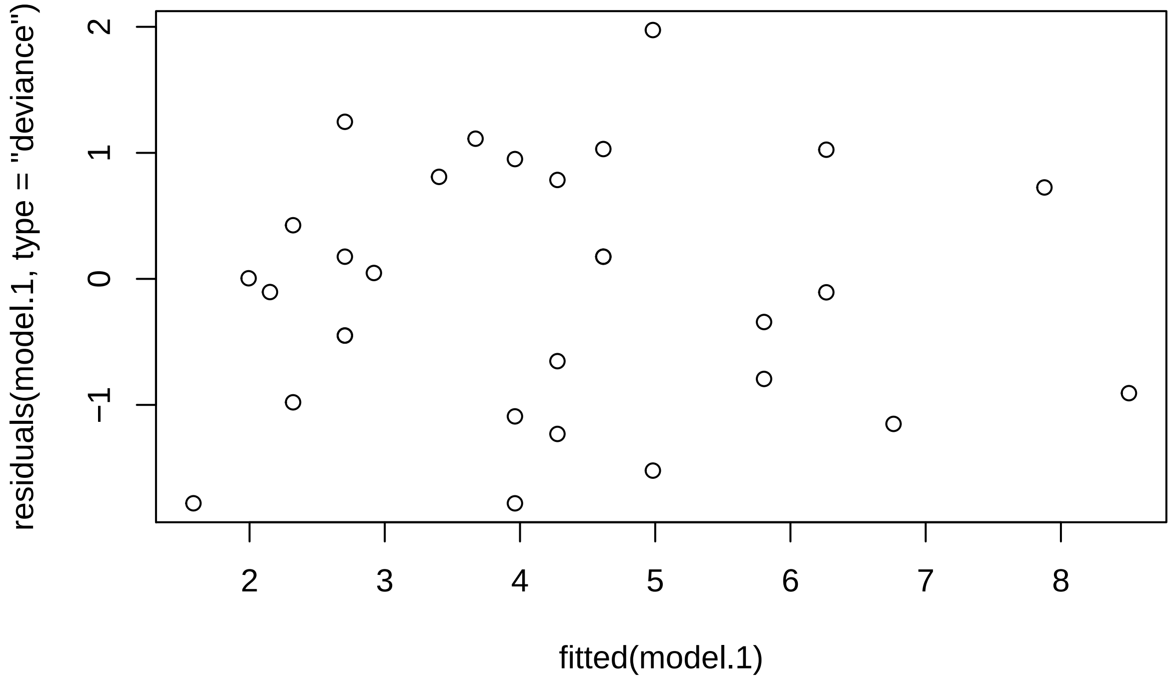
```
1-model.1$deviance/model.1$null.deviance # Pseudo R-squared = 0.423676
```

```
## [1] 0.423676
```

```
plot(fitted(model.1), residuals(model.1, type="pearson"))
```



```
plot(fitted(model.1), residuals(model.1, type="deviance"))
```



```
summary(influence.measures(model.1))
```

```
## Potentially influential observations of
##   glm(formula = Y ~ X, family = "poisson") :
##
##    dfb.1_ dfb.X dffit cov.r   cook.d hat
## 10 -0.22   0.30  0.37  1.25_*  0.08   0.18
## 21  0.37  -0.48 -0.57  1.30_*  0.15   0.23_*
```

```
#    dfb.1_ dfb.x dffit cov.r   cook.d hat
# 10 -0.22   0.30  0.37  1.25_*  0.08   0.18
# 21  0.37  -0.48 -0.57  1.30_*  0.15   0.23_*

residuals(model.1)[8] # 1.974329
```

```
##        8
## 1.974329
```

```
rstudent(model.1)[8] # 2.028255
```

```
##        8
## 2.028255
```

```
detach(poisson)
```

Hospital recovery (exponential regression)

Load the recovery data. Create log(prog) variable. Obtain starting values for nonlinear model parameters from fitting a simple linear regression model of log(prog) vs days. Fit nonlinear regression model to data using these starting values. Create a scatterplot of prog vs days and add a fitted line based on the nonlinear regression model.

```
recovery <- read.table("./Data/recovery.txt", header=T)
attach(recovery)

logprog <- log(prog)
summary(lm(logprog ~ days))
```

```
##
## Call:
## lm(formula = logprog ~ days)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37241 -0.07073  0.02777  0.05982  0.33539
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.037159   0.084103   48.00 5.08e-16 ***
## days        -0.037974   0.002284  -16.62 3.86e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1794 on 13 degrees of freedom
## Multiple R-squared:  0.9551, Adjusted R-squared:  0.9516
## F-statistic: 276.4 on 1 and 13 DF,  p-value: 3.858e-10
```

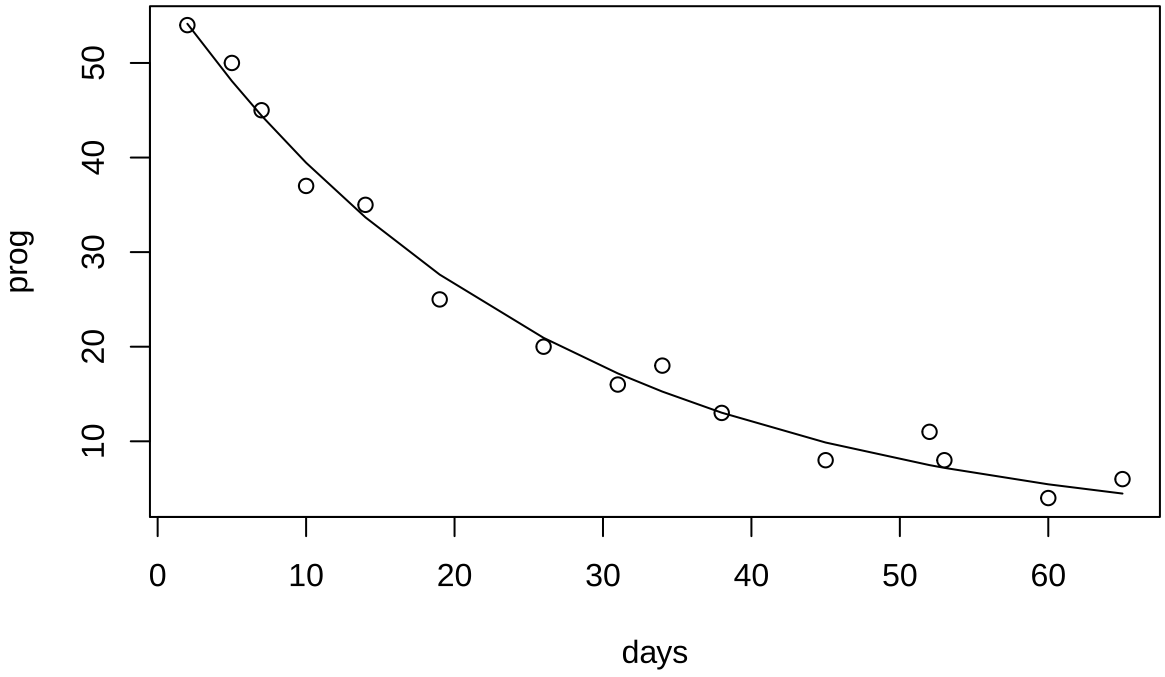
```
#              Estimate Std. Error t value Pr(>|t|)
# (Intercept)  4.037159   0.084103   48.00 5.08e-16 ***
# days        -0.037974   0.002284  -16.62 3.86e-10 ***

exp(4.037159) # 56.66513
```

```
## [1] 56.66513
```

```
model.1 <- nls(prog ~ theta1 * exp(theta2 * days),
               start=list(theta1=56.7, theta2=-0.038))
summary(model.1)
```

```
##
## Formula: prog ~ theta1 * exp(theta2 * days)
##
## Parameters:
##         Estimate Std. Error t value Pr(>|t|)
## theta1 58.606532   1.472159   39.81 5.70e-15 ***
## theta2 -0.039586   0.001711  -23.13 6.01e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.951 on 13 degrees of freedom
##
## Number of iterations to convergence: 3
## Achieved convergence tolerance: 8.837e-06
```

```
#          Estimate Std. Error t value Pr(>|t|)
# theta1 58.606532   1.472159   39.81 5.70e-15 ***
# theta2 -0.039586   0.001711  -23.13 6.01e-12 ***
# ---
# Residual standard error: 1.951 on 13 degrees of freedom

plot(x=days, y=prog,
     panel.last = lines(sort(days), fitted(model.1)[order(days)]))
```

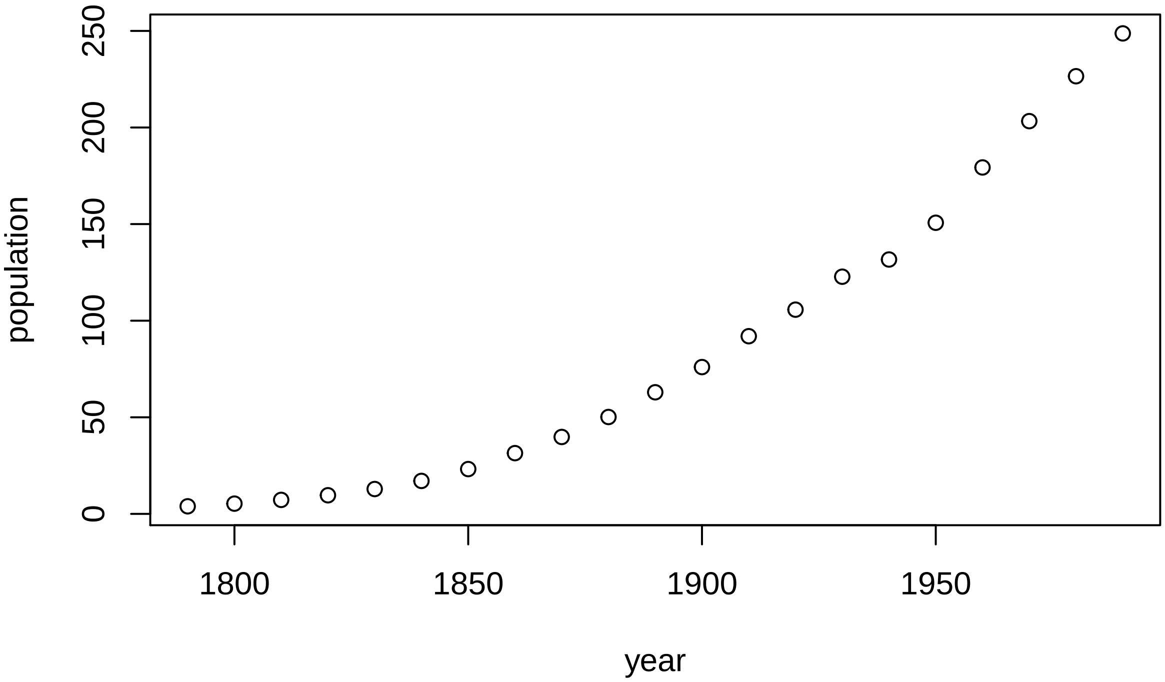


```
detach(recovery)
```

U.S. census population (population growth nonlinear regression)

Load the census data. Obtain starting values for nonlinear model parameters from observing features of a scatterplot of population vs year. Fit nonlinear regression model to data using these starting values. Create a scatterplot of population vs year and add a fitted line based on the nonlinear regression model. Create a residual plot.

```
census <- read.table("./Data/us_census.txt", header=T)
attach(census)

plot(x=year, y=population)
```



```
log(350/3.929-1) # 4.478259
```

```
## [1] 4.478259
```

```
log(350/5.308-1) - log(350/3.929-1) # -0.3048229
```

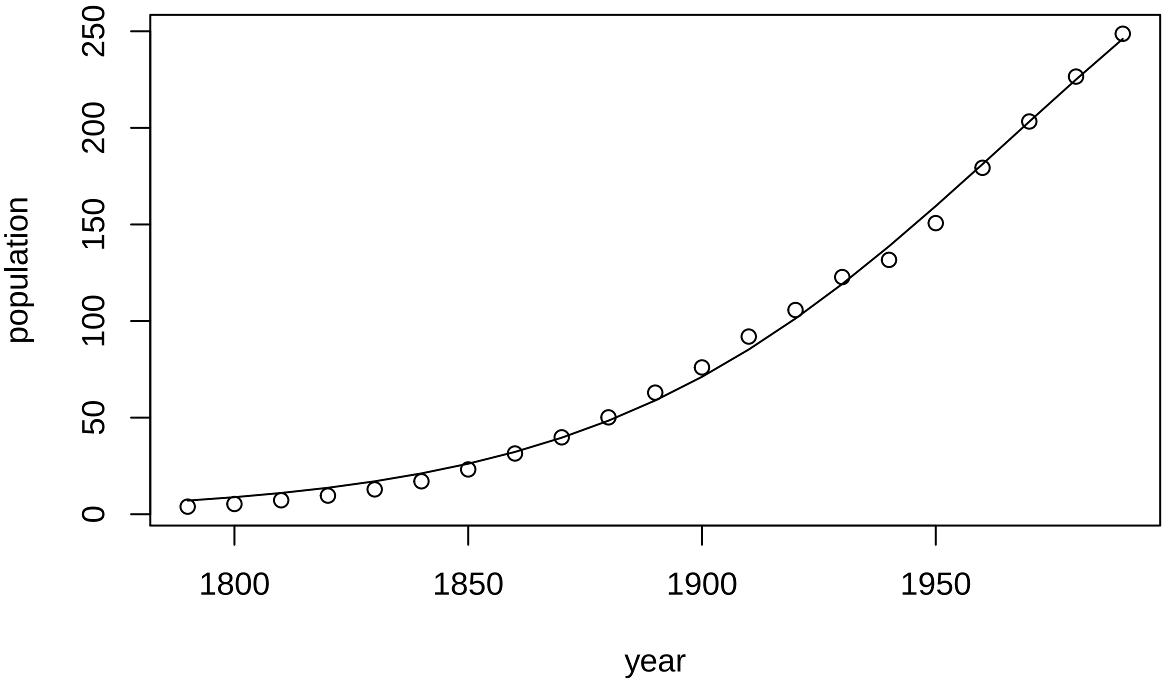
```
## [1] -0.3048229
```

```
model.1 <- nls(population ~ beta1 / (1 + exp(beta2 + beta3 * (year - 1790) / 10)),
               start=list(beta1=350, beta2=4.5, beta3=-0.3))
summary(model.1)
```

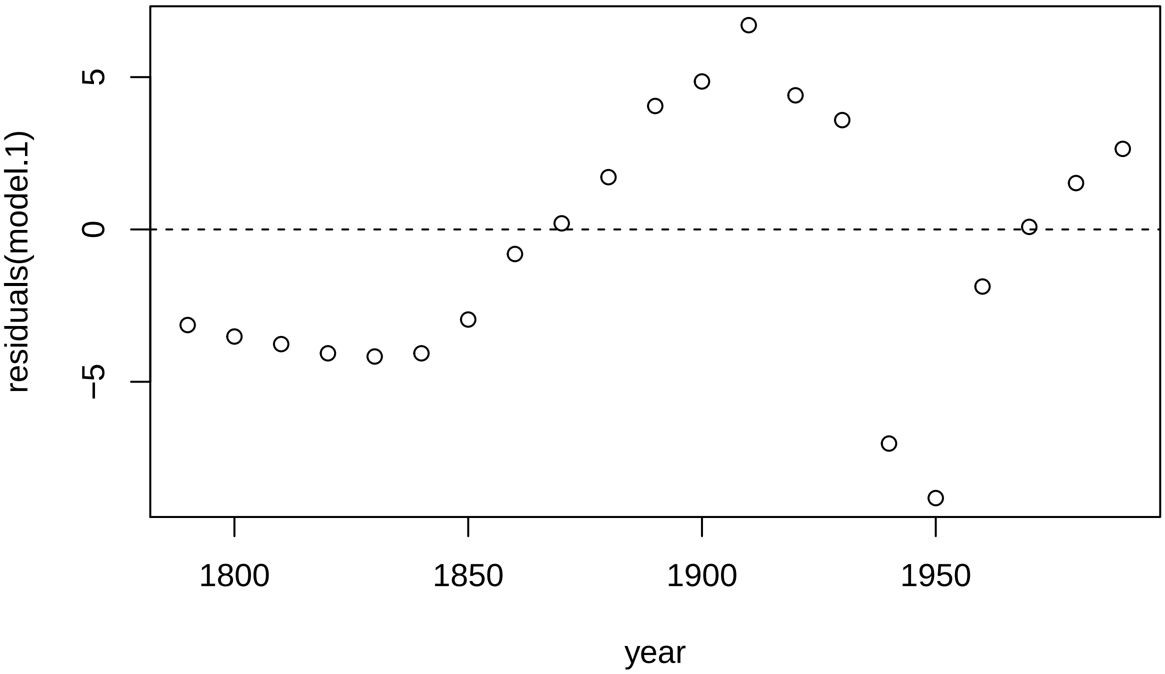
```
##
## Formula: population ~ beta1/(1 + exp(beta2 + beta3 * (year - 1790)/10))
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## beta1 389.16552   30.81197   12.63  2.2e-10 ***
## beta2   3.99035    0.07032   56.74  < 2e-16 ***
## beta3  -0.22662    0.01086  -20.87  4.6e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 4.45 on 18 degrees of freedom
##
## Number of iterations to convergence: 6
## Achieved convergence tolerance: 1.476e-06
```

```
#        Estimate Std. Error t value Pr(>|t|)
# beta1 389.16551   30.81196   12.63  2.2e-10 ***
# beta2   3.99035    0.07032   56.74  < 2e-16 ***
# beta3  -0.22662    0.01086  -20.87  4.6e-14 ***
# ---
# Residual standard error: 4.45 on 18 degrees of freedom

plot(x=year, y=population,
     panel.last = lines(sort(year), fitted(model.1)[order(year)]))
```



```
plot(x=year, y=residuals(model.1),
     panel.last = abline(h=0, lty=2))
```



```
detach(census)
```