Statistical Modelling - Series 3

Konstantinos Papadakis (DSML 03400149)

13 January 2022

I) asfalies.txt

(i)

We start by loading the appropriate libraries and the dataset.

```
library(ggplot2)
library(car)
library(hnp)
library(pROC)

insurances <- read.table("./data/asfalies.txt", header = TRUE)
insurances$cartype <- as.factor(insurances$cartype)
```

We then fit our model. In its summary we can see the Wald tests P(>|z|) of the coefficients and the AIC value. Observe that according to the Wald tests, all variables have extremely low p-value which allows us to conclude that they are significant without doubt. The AIC value is quite high, which implies that our model is far from perfect.

```
mod1 <- glm(
  y ~ agecat + cartype + district, offset = log(n),
  family = poisson, data = insurances
)
summary(mod1)  # Includes Wald and AIC
```

```
##
## Call:
## glm(formula = y ~ agecat + cartype + district, family = poisson,
##     data = insurances, offset = log(n))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8590  -0.7506  -0.1297   0.6511   3.2310
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.93522    0.05525 -35.030  < 2e-16 ***
## agecat      -0.37628    0.04451  -8.453  < 2e-16 ***
## cartype2     0.16223    0.05048   3.214 0.001309 **
## cartype3     0.39535    0.05491   7.200 6.03e-13 ***
```

```
## cartype4      0.56543    0.07215   7.836 4.64e-15 ***
## district      0.21661    0.05853   3.701 0.000215 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 207.833  on 31  degrees of freedom
## Residual deviance:  41.789  on 26  degrees of freedom
## AIC: 222.15
##
## Number of Fisher Scoring iterations: 4
```

Observe that the p-value for the deviance is quite small, which means that our model is "far" from the saturated model. We concluded a similar result from the AIC value. On the other hand, our model is definitely better than a constant predictor, as the p-value for the delta-deviance is essentially 0.

```r
Dev <- function (mod) {
  # Don't use it for Bernoulli logistic
  pvalue <- pchisq(mod$deviance, mod$df.residual, lower.tail = FALSE)
  return(c(deviance = mod$deviance, pvalue = pvalue))
}

DeltaDev <- function (mod) {
  ddeviance <- mod$null.deviance - mod$deviance
  ddf <- mod$df.null - mod$df.residual
  pvalue <- pchisq(ddeviance, ddf, lower.tail = FALSE)
  return(c(ddeviance = ddeviance, pvalue = pvalue))
}
```

```r
print(Dev(mod1))
```

```
##    deviance      pvalue
## 41.78852567  0.02580847
```

```r
print(DeltaDev(mod1))
```

```
##    ddeviance       pvalue
## 1.660446e+02 5.091633e-34
```

(ii)

We can create approximate 95% confidence intervals for the coefficients using the Wald statistics.

```r
confint(mod1, level = 0.95)
```

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

```
##                   2.5 %     97.5 %
## (Intercept) -2.04472348 -1.8281432
```

```
## agecat      -0.46278476 -0.2882587
## cartype2     0.06402271  0.2619349
## cartype3     0.28814409  0.5034436
## cartype4     0.42299446  0.7059551
## district     0.10011971  0.3296247
```

These values can be interpreted as follows. Whenever the i-th covariate is increased by 1, the expected number of insurance claims is multiplied by e^{β_i}. Intervals for those multipliers are seen below.

```r
exp(confint(mod1, level = 0.95))
```

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

```
##                 2.5 %    97.5 %
## (Intercept) 0.1294160 0.1607117
## agecat      0.6295281 0.7495676
## cartype2    1.0661166 1.2994419
## cartype3    1.3339495 1.6544086
## cartype4    1.5265258 2.0257805
## district    1.1053032 1.3904462
```

Thus, if for example `agecat` changes from 0 to 1 (i.e. young to old), then we expect the insurance claims to drop by anywhere between 25% and 37%.

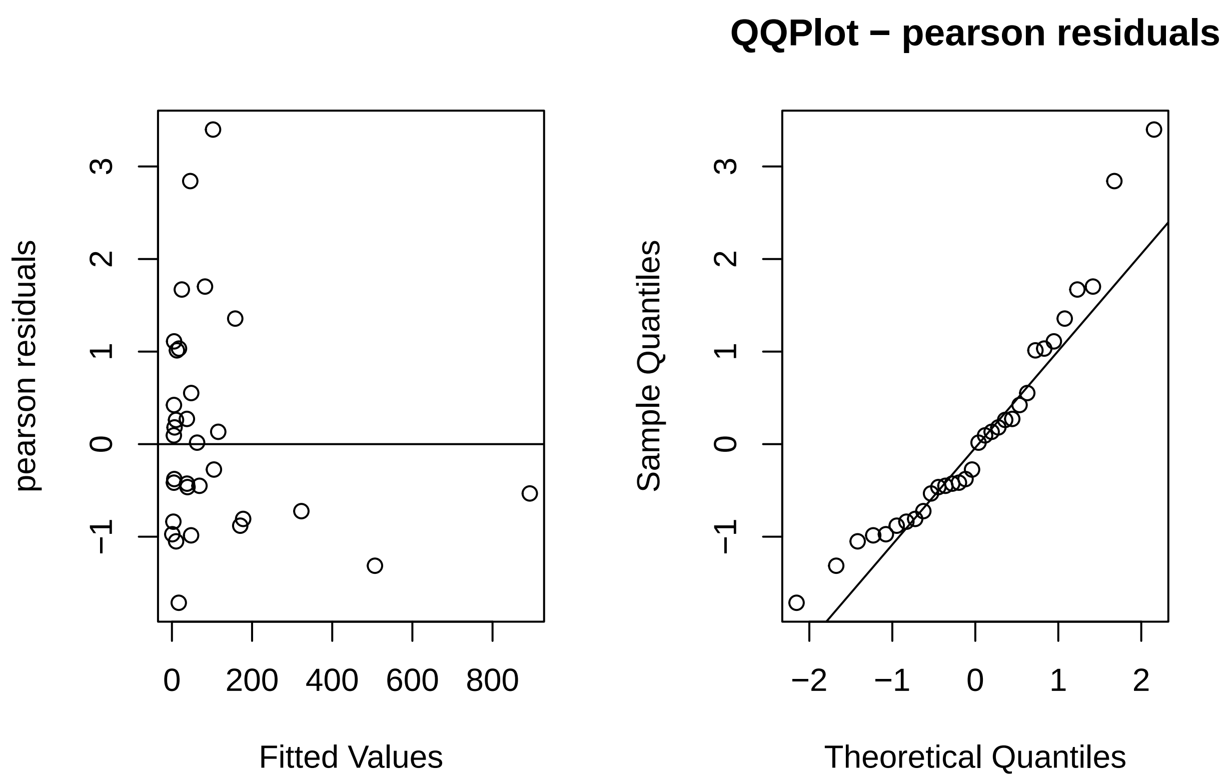
(iii)

In the following, nothing is out of the ordinary. The Pearson and Deviance residuals are distributed "nicely" around 0, the Hat values and Cook's distances show that 4 data points are relatively influential. Finally, the likehood residuals are "nicely" distributed around 0 as well.

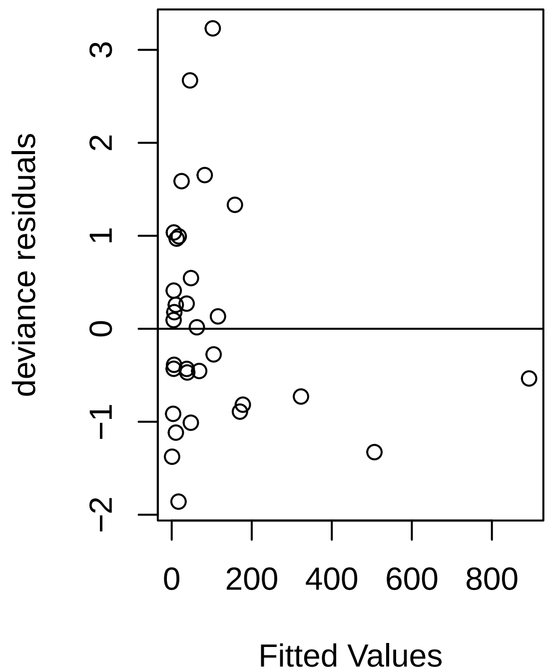
```r
PlotResiduals <- function(mod, type) {
  oldparams <- par(mfrow = c(1, 2))
  r <- residuals(mod, type = type)
  plot(mod$fitted.values, r,
     xlab = "Fitted Values", ylab = sprintf("%s residuals", type))
  abline(h = 0)
  qqnorm(r, main = sprintf("QQPlot - %s residuals", type))
  qqline(r)
  par(oldparams)
}


PlotHatvalues <- function(mod) {
  p = length(mod$coefficients)
  n = length(mod$fitted.values)
  plot(hatvalues(mod), ylab = "Hat values")
  abline(h = 2*p/n)
}


PlotCooks <- function(mod) {
  plot(cooks.distance(mod), ylab = "Cook's Distance")
```

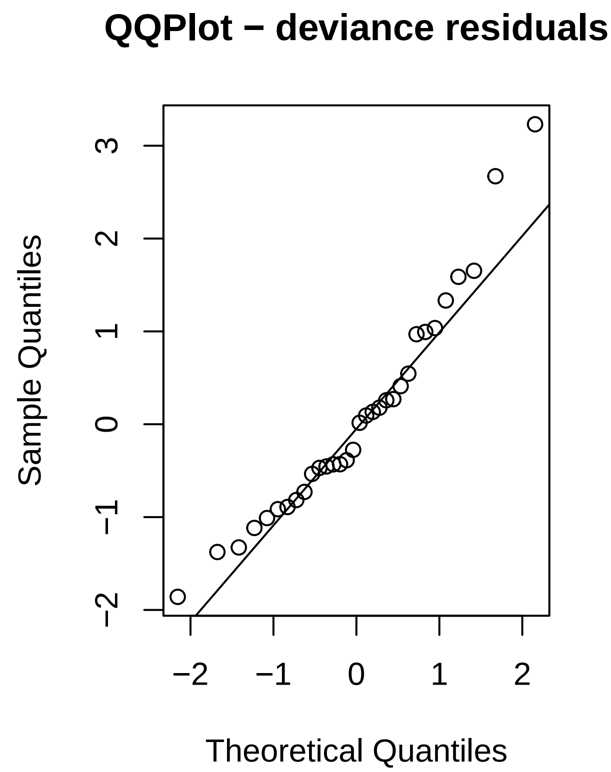
```
  abline(h = c(0.5, 1), lty = 2)
}


PlotLikelihoodResiduals <- function(mod) {
  oldparams <- par(mfrow = c(1, 2))
  plot(rstudent(mod), ylab = "Likelihood Residuals")
  abline(h = 0)
  plot(hatvalues(mod), rstudent(mod),
       xlab = "Hat values", ylab = "Likelihood Residuals")
  abline(h = 0)
  par(oldparams)
}
```

```
PlotResiduals(mod1, "pearson")
```

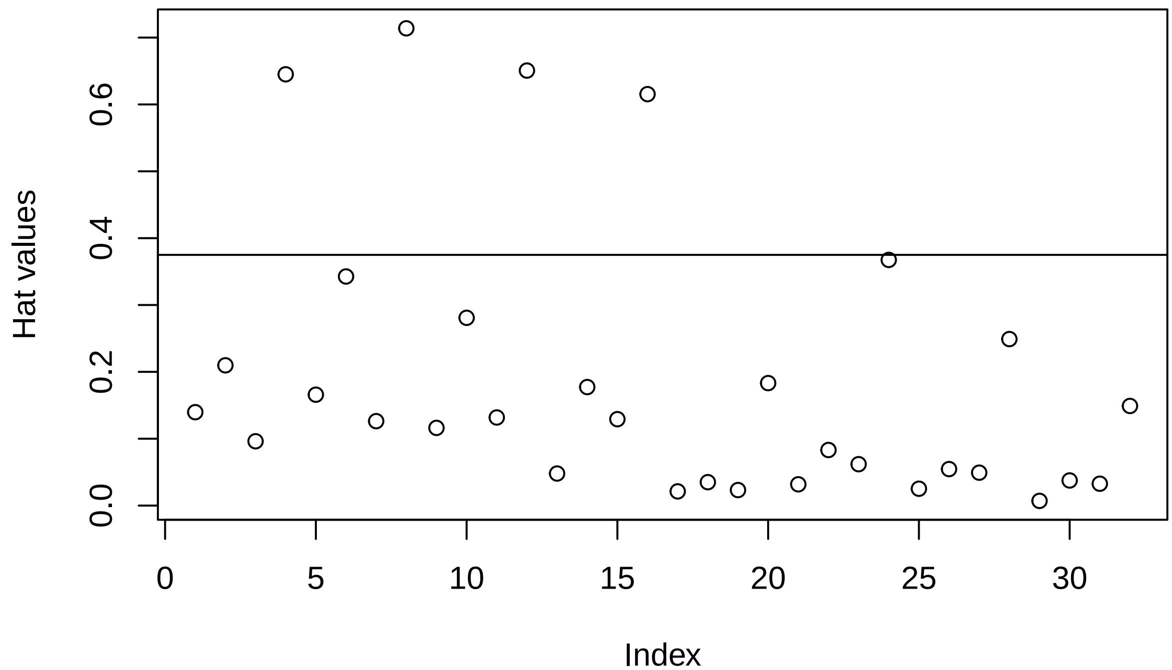



```
PlotResiduals(mod1, "deviance")
```

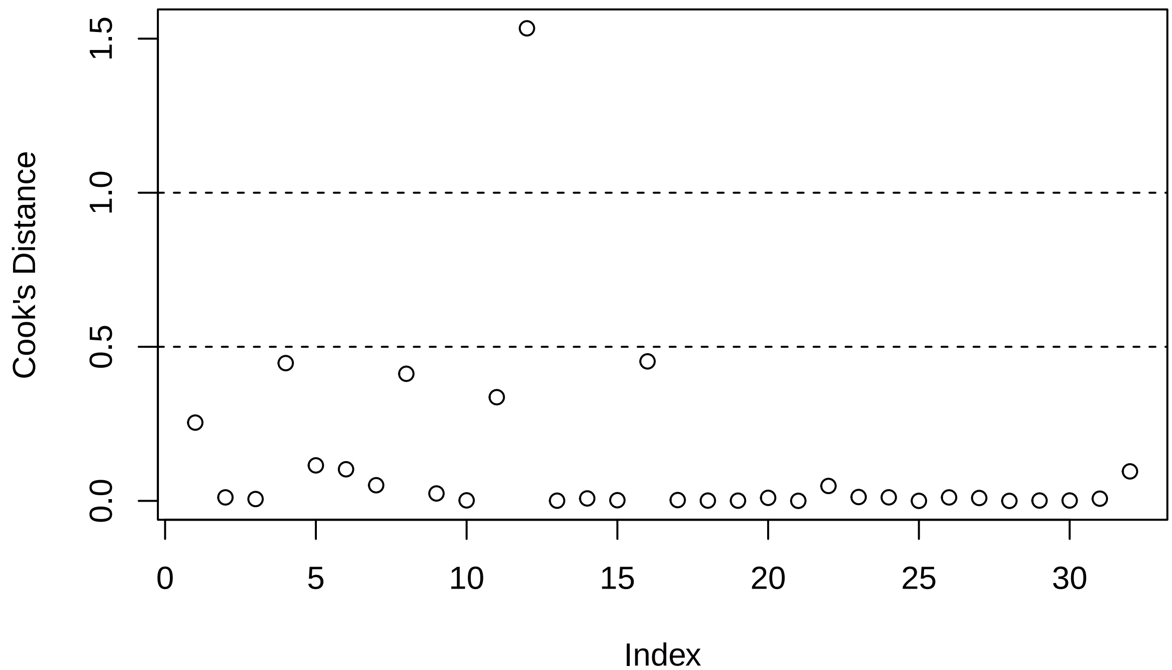




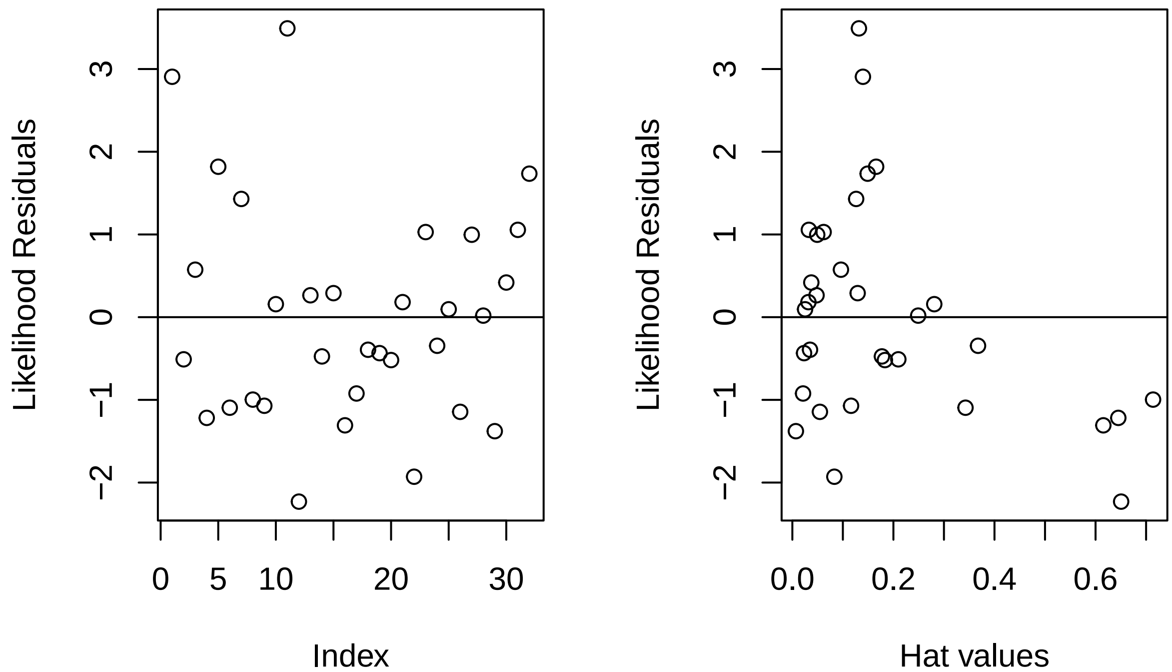

```
PlotHatvalues(mod1)
```




```
PlotCooks(mod1)
```




```
PlotLikelihoodResiduals(mod1)
```



(iv)

After trying all recommended combinations, `cartype:district` was the only one which created a variable with p-value < 0.05, so we select this one.

The new model's residual deviance dropped from 41.79 to 37.27. This decrease corresponds to a p-value equal to 0.21, which is not low enough to reject the original (simpler) model.

AIC increased from 222.15 to 223.63 (i.e. it worsened), which is due to `cartype:district` introducing (4-1) * (2-1) = 3 extra covariates.

```r
mod_alt <- glm(
  y ~ agecat + cartype + district + cartype:district, offset = log(n),
  family = poisson, data = insurances
)

summary(mod_alt)
```

```
##
## Call:
## glm(formula = y ~ agecat + cartype + district + cartype:district,
##     family = poisson, data = insurances, offset = log(n))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

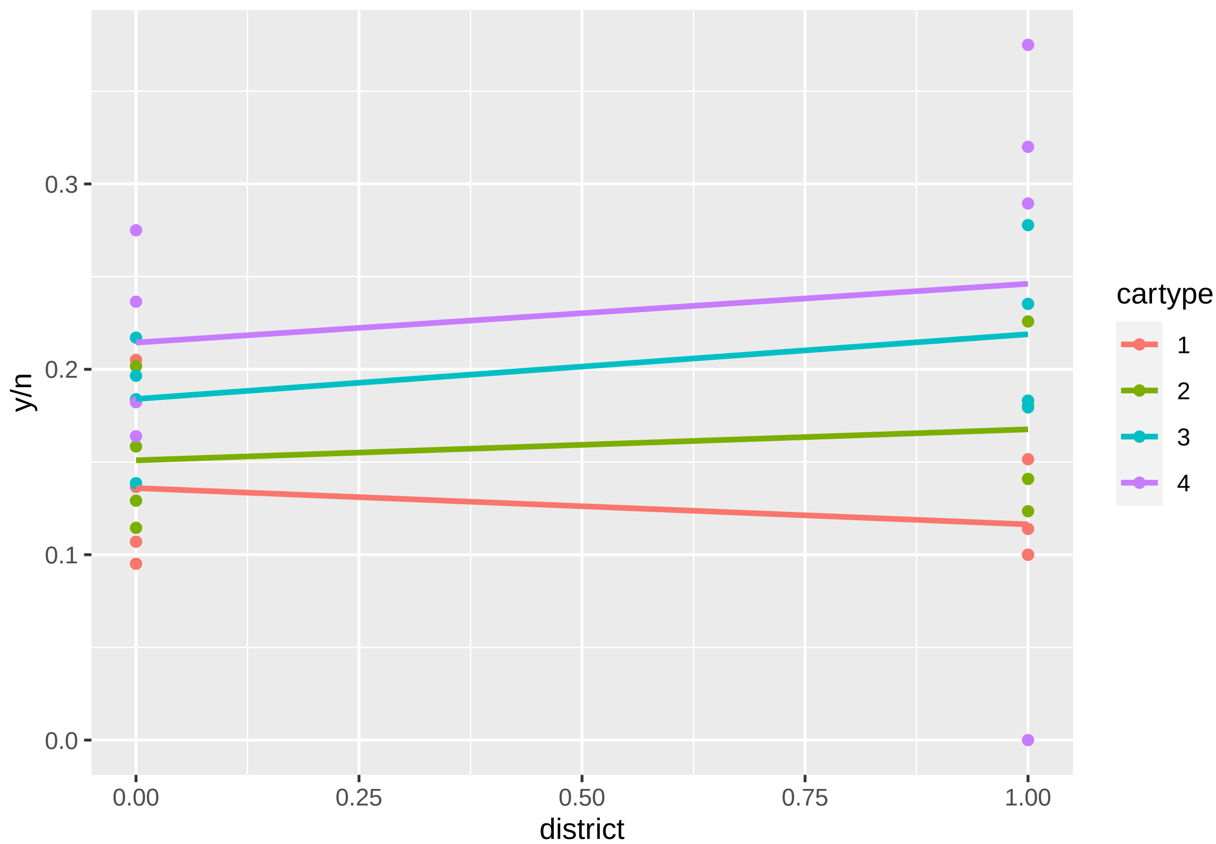
```
## -1.7226  -0.6658   0.0260   0.4098   3.2367
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -1.92280    0.05648 -34.042  < 2e-16 ***
## agecat             -0.37562    0.04452  -8.438  < 2e-16 ***
## cartype2            0.15317    0.05291   2.895   0.0038 **
## cartype3            0.38172    0.05770   6.616 3.69e-11 ***
## cartype4            0.51016    0.07750   6.583 4.62e-11 ***
## district            0.07745    0.15269   0.507   0.6120
## cartype2:district  0.09978    0.17654   0.565   0.5719
## cartype3:district  0.14557    0.18866   0.772   0.4404
## cartype4:district  0.44498    0.22036   2.019   0.0434 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 207.83  on 31  degrees of freedom
## Residual deviance:  37.27  on 23  degrees of freedom
## AIC: 223.63
##
## Number of Fisher Scoring iterations: 4
```

```
anova(mod1, mod_alt, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ agecat + cartype + district
## Model 2: y ~ agecat + cartype + district + cartype:district
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        26     41.789
## 2        23     37.270  3   4.5184   0.2107
```

From the plot below we observe there is a trend to have more insurances claims per capita in Athens compared to other cities, except when the car is of type 4, in which case the trend is reversed (Athens has fewer claims per capita compared to other cities). This is in agreement with the small p-value for the covariate `cartype4:district`.

```
ggplot(insurances, aes(district, y/n, color = cartype)) +
  geom_point() +
  geom_smooth(formula = y~x, method = "lm", se = FALSE)
```



II) leukaemia.txt

(i)

In the same fashion as part I, we fit our model and observe the Wald tests P(>|z|) of the coefficients and the AIC value in the model's summary. According to the Wald statistics, only `age`, `index` and `temperature` appear to be relevant. The AIC value is quite low, which implies that our model is performing relatively well (especially if we take into account that we have "redundant" variables).

```r
leuk <- read.table("./data/leukaemia.txt", header = TRUE)
mod2 <- glm(
  response ~ .,
  family = binomial, data = leuk
)
summary(mod2)
```

```
##
## Call:
## glm(formula = response ~ ., family = binomial, data = leuk)
##
## Deviance Residuals:
```

```
##      Min        1Q    Median        3Q       Max
## -1.73878  -0.58099  -0.05505   0.62618   2.28425
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 98.52361   40.85385   2.412  0.01588 *
## age         -0.06029    0.02729  -2.210  0.02714 *
## smear       -0.00480    0.04108  -0.117  0.90698
## infiltrate   0.03621    0.03934   0.921  0.35728
## index        0.39845    0.13278   3.001  0.00269 **
## blasts       0.01343    0.05782   0.232  0.81627
## temperature -0.10223    0.04181  -2.445  0.01448 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 70.524  on 50  degrees of freedom
## Residual deviance: 40.060  on 44  degrees of freedom
## AIC: 54.06
##
## Number of Fisher Scoring iterations: 6
```

Observe that the p-value for the deviance is high, which means that our model performs well even when compared to the saturated model (possibly equally well). We concluded a similar result from the AIC value. The p-value for the delta-deviance is very small, which means that we definitely prefer our model over the constant predictor.

```
print(Dev(mod2))
```

```
##   deviance     pvalue
## 40.0599149  0.6411612
```

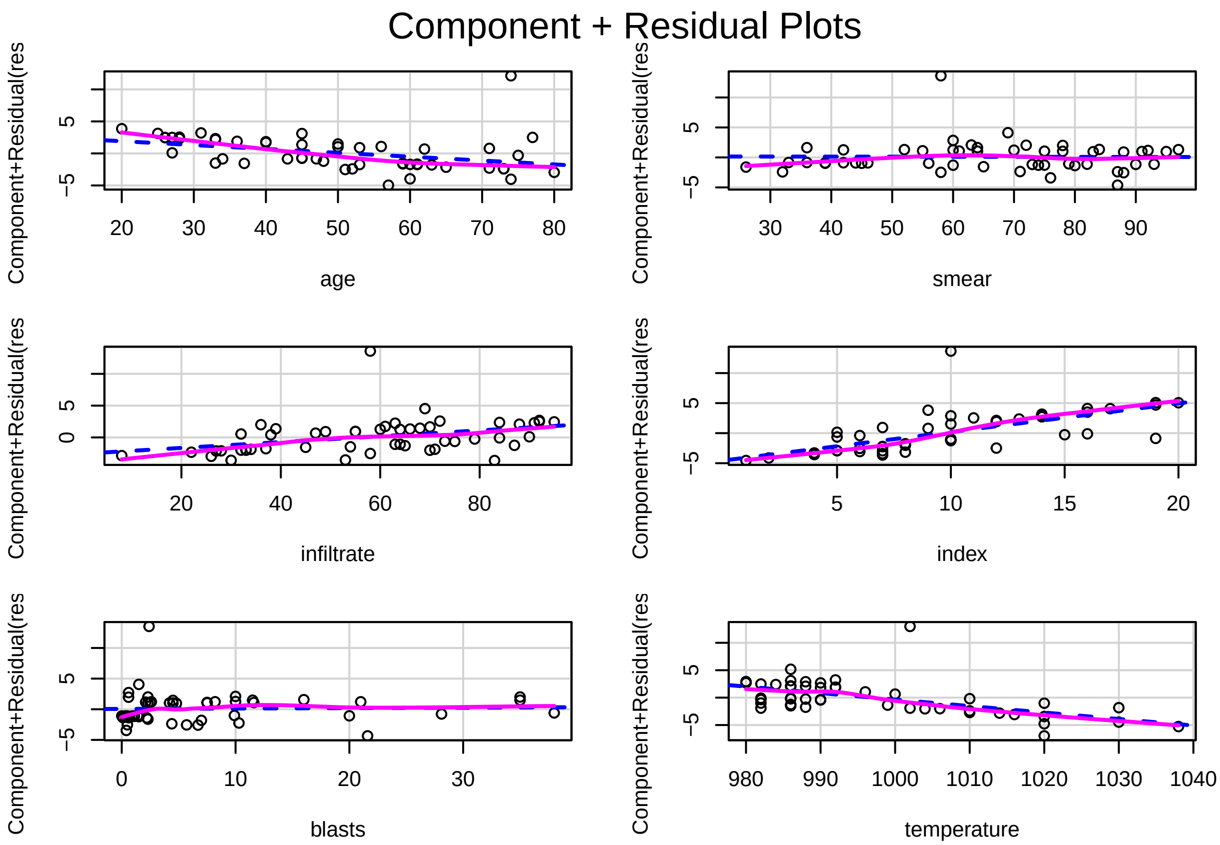
```
print(DeltaDev(mod2))
```

```
##    ddeviance       pvalue
## 3.046452e+01 3.206902e-05
```

(ii)

In the Partial Residual Plots, we see that the fitted curves are close to the expected lines, which is an indication that our covariates don't require any further transformation.

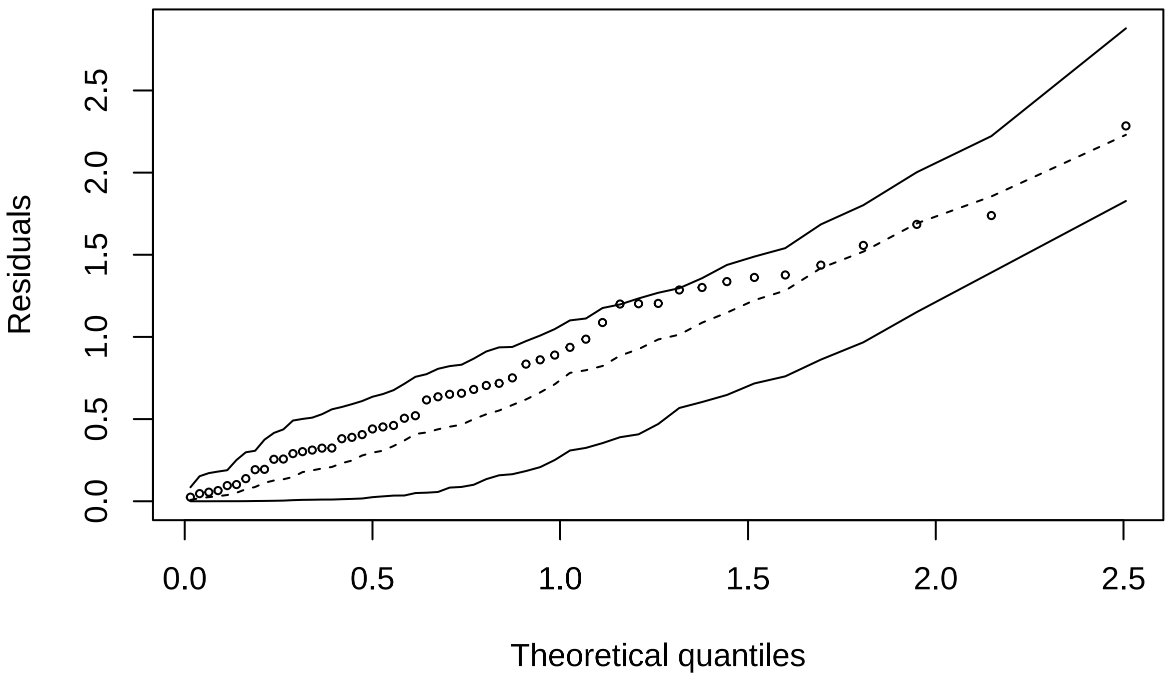
```
crPlots(mod2)
```



The residuals are well within the simulated envelope, so everything looks good here as well.

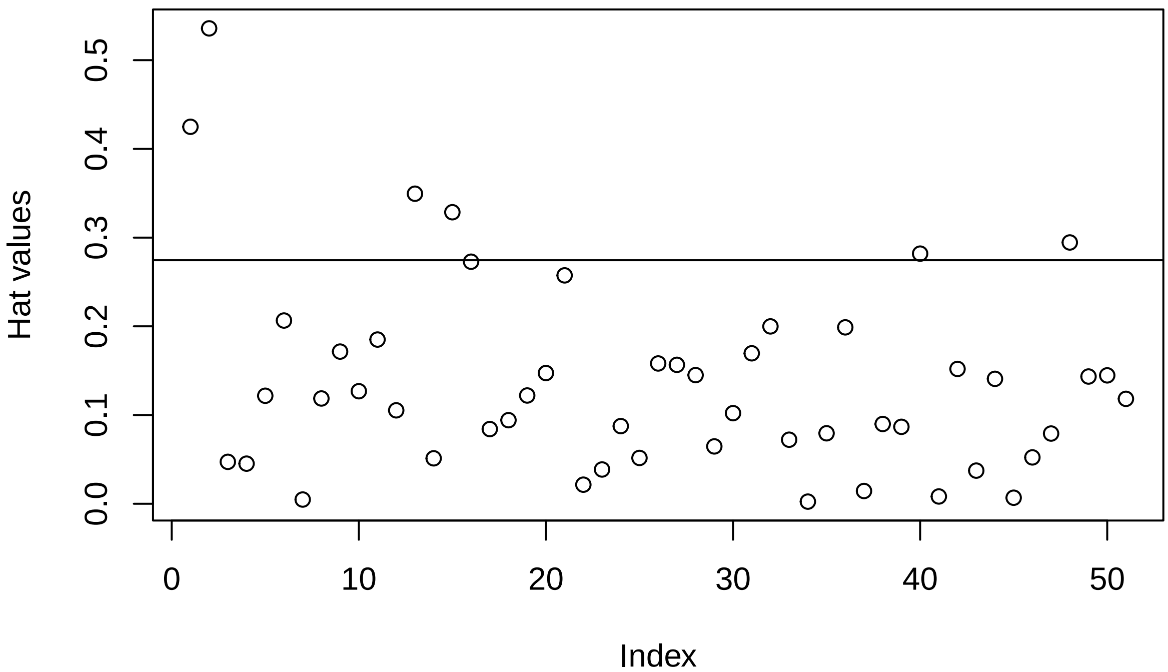
```
hnp(mod2)
```

```
## Binomial model
```

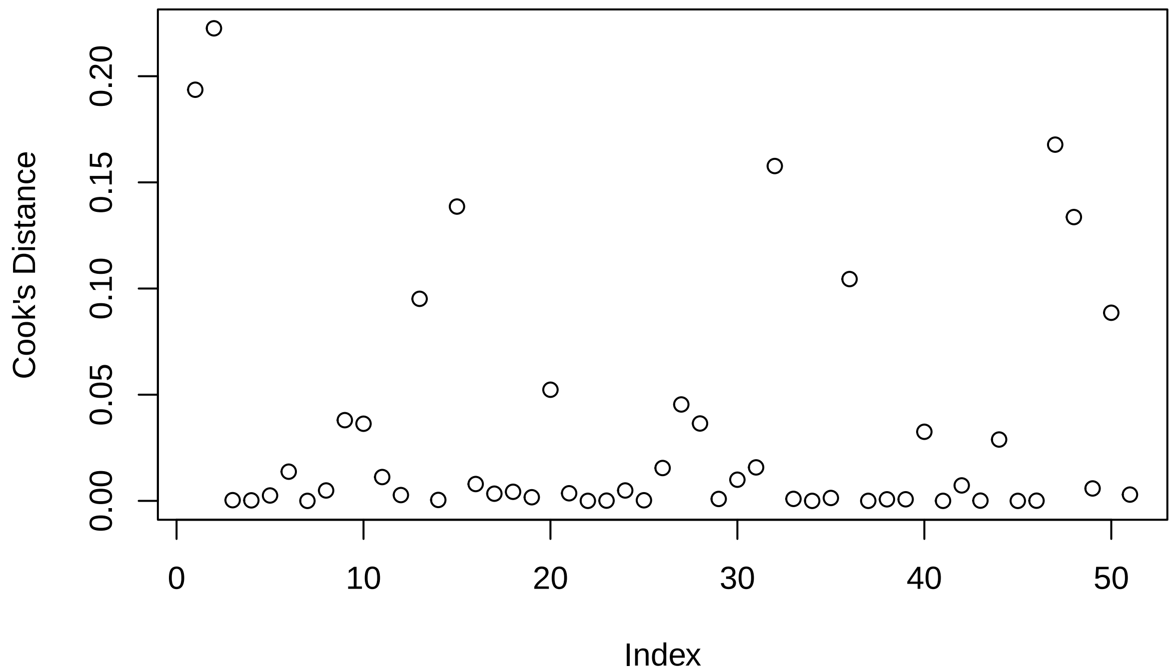


Only a few data stand out when it comes to importance.

```
PlotHatvalues(mod2)
```

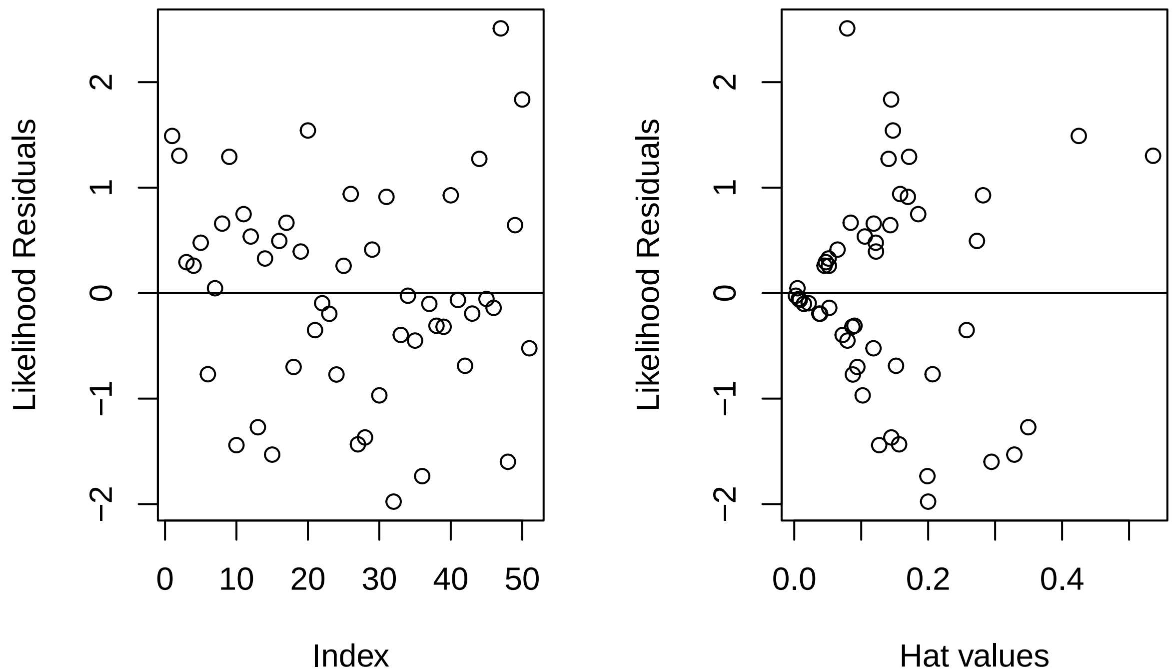



```
PlotCooks(mod2)
```



It seems that the leverage and the likelihood-residual variance are positively related.

```
PlotLikelihoodResiduals(mod2)
```



(iii)

In the same way as part I, we can create 95% confidence intervals for the coefficients using the Wald statistics.

```
confint(mod2, level = 0.95)
```

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

```
##                   2.5 %       97.5 %
## (Intercept) 26.64202264 190.29749077
## age         -0.12190343  -0.01163601
## smear       -0.08979333   0.07514375
## infiltrate  -0.03313436   0.12429855
## index        0.17586891   0.70761406
## blasts      -0.09662909   0.13388017
## temperature -0.19644228  -0.02902827
```

These values can be interpreted as follows. Whenever the i-th covariate is increased by 1, the odds of a positive response are multiplied by e^{β_i}. Intervals for those multipliers are seen below.

```
exp(confint(mod2, level = 0.95))
```

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

```
##                    2.5 %       97.5 %
## (Intercept) 3.719490e+11 4.417232e+82
## age         8.852339e-01 9.884314e-01
## smear       9.141201e-01 1.078039e+00
## infiltrate  9.674086e-01 1.132354e+00
## index       1.192282e+00 2.029144e+00
## blasts      9.078927e-01 1.143256e+00
## temperature 8.216488e-01 9.713890e-01
```

Thus, if for example `index` (leukaemia cells) is increased by 1, the odds of a positive response $\frac{P(\text{positive})}{P(\text{negative})}$ will likely increase by a factor between 1.19 and 2.03.

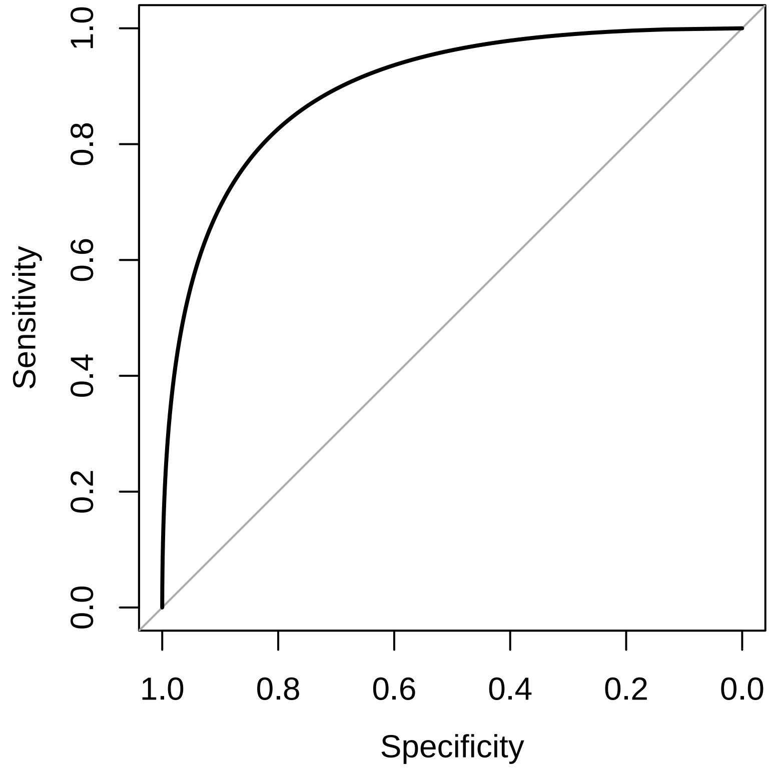
(iv)

We see that the ROC curve is heavily "pointing" towards (specificity, sensitivity) = (1, 1), and the Area Under the Curve (AUC) is remarkably high (0.8962), which is something that we expected from the values of the AIC and the Deviance.

```r
PlotRoc <- function(mod) {
  oldparams <- par(pty = "s")
  roc(mod$y, mod$fitted.values, smooth=TRUE, plot=TRUE)
  par(oldparams)
}

PlotRoc(mod2)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```


1.0
0.8
0.6
0.4
0.2
0.0
Sensitivity
1.0
0.8
0.6
0.4
0.2
0.0
Specificity