# Logistic Regression: Application in R

```
library(mosaic)
library(car)
library(ggplot2)
library(effects)
library(MASS)
```

## Illustration

Let's use the HELP dataset to examine whether `age` and `female` predict `homeless`, which is coded `1` if homeless and `0` if not.

```
helpdata <- read.csv("help.csv")
```

First let's look at the proportion of individuals in the sample who are homeless.

```
p <- mean(helpdata$homeless)
p
```

```
## [1] 0.4613687
```

You can also use the function `tally()` from the `mosaic` package to obtain cross-tabulations.

```
tally(~ homeless | female, margins=TRUE, data=helpdata)
```

```
##         female
## homeless   0   1
##    0     177  67
##    1     169  40
##    Total 346 107
```

Let's first fit a null model.

```
m0 <- glm(homeless ~ 1, family="binomial", data=helpdata)
summary(m0)
```

```
##
## Call:
## glm(formula = homeless ~ 1, family = "binomial", data = helpdata)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.112  -1.112  -1.112   1.244   1.244
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.15483    0.09425  -1.643      0.1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 625.28  on 452  degrees of freedom
```

```
## Residual deviance: 625.28  on 452  degrees of freedom
## AIC: 627.28
##
## Number of Fisher Scoring iterations: 3
```

If we exponentiate the intercept, $\beta_0$, we obtain the odds of being homeless in this sample.

```
exp(coef(m0))
```

```
## (Intercept)
##   0.8565574
```

```
plogis(coef(m0))
```

```
## (Intercept)
##   0.4613687
```

The `plogis()` function gives the probability (i.e., proportion) which because this is an intercept only model equals the proportion we calculated above.

Now let's include the predictors, `age` and `female`

```
m1 <- glm(homeless ~ age + female, family="binomial", data=helpdata)
summary(m1)
```

```
##
## Call:
## glm(formula = homeless ~ age + female, family = "binomial", data = helpdata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3600  -1.1231  -0.9185   1.2020   1.5466
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.89262    0.45366  -1.968   0.0491 *
## age          0.02386    0.01242   1.921   0.0548 .
## female      -0.49198    0.22822  -2.156   0.0311 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 625.28  on 452  degrees of freedom
## Residual deviance: 617.19  on 450  degrees of freedom
## AIC: 623.19
##
## Number of Fisher Scoring iterations: 4
```

The Wald statistic for each coefficient is given in the z value column, followed by the p value. The standard error for each coefficient is given in the Std. Error column. The estimate given is the log odds. To obtain the odds ratios, you have to exponentiate the coefficients.

```
exp(coef(m1))
```

```
## (Intercept)         age      female
##   0.4095802   1.0241460   0.6114159
```

So for a one-unit increase in age, the log-odds of being homeless increases by 0.02 and the odds of being

homeless increase by a factor of 1.02. Another way to interpret the odds is to subtract 1 from the odds and multiply by 100. Doing this results in 2, thus a one-unit increase in age results in a 2% increase in odds of being homeless. Being female results in a decrease in the log-odds of being homeless by -.49, which is a (1-.611)*100= 38.86% decrease in the odds of being homeless.

Profile likelihood confidence intervals can be obtained by using the function `confint()`.

```
confint(m1)
```

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

```
##                      2.5 %       97.5 %
## (Intercept) -1.7898020867 -0.008029871
## age         -0.0003570035  0.048446302
## female      -0.9453893343 -0.048920636
```

We see that the confidence intervals for the intercept and the coefficient for female, do not include 0. Thus, we would conclude that these are statistically significant. You can exponentiate these as we did with the coefficient estimates to obtain the confidence intervals for the odds ratios.

```
exp(confint(m1))
```

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

```
##                 2.5 %    97.5 %
## (Intercept) 0.1669932 0.9920023
## age         0.9996431 1.0496390
## female      0.3885283 0.9522567
```

Once exponentiated, the confidence intervals should not include 1 if they are statistically significant.

The `Anova` function from the `car` package gives likelihood ratio tests of the explanatory variables.

```
Anova(m1)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: homeless
##        LR Chisq Df Pr(>Chisq)
## age      3.7287  1    0.05349 .
## female   4.7443  1    0.02940 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Next, we can compare the full and null models using the likelihood ratio test.

```
anova(m0, m1)
```

```
## Analysis of Deviance Table
##
## Model 1: homeless ~ 1
## Model 2: homeless ~ age + female
##   Resid. Df Resid. Dev Df Deviance
## 1       452     625.28
## 2       450     617.19  2   8.0941
```

Thus, `age` and `female` results in a statistically significant improvement, $\chi^2(2) = 8.09, p = .018$

To obtain that p-value, I used the function `pchisq()` and you want to use `1-pchisq` as shown below.

```
1-pchisq(8.09, 2)
```

```
## [1] 0.0175097
```

Finally, McFadden's pseudo-$R^2$ can be calculated as

```
1-deviance(m1)/deviance(m0)
```

```
## [1] 0.01294459
```

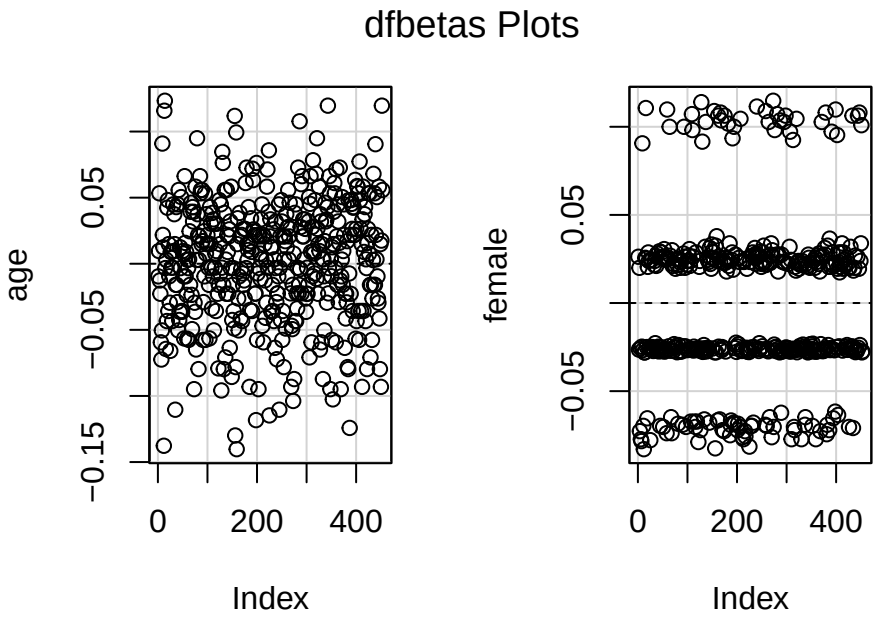
Thus, the predictors account for 1.3% of the variance in homelessness.

As with linear regression models, `hatvalues()` can be used to obtain leverage, `rstudent()` can be used to obtain the studentized residuals, `cooks.distance()` for the Cook's D, and `dfbetas()` for DFBETAS. In addition, the `outlierTest()` from the `car` package can be used to test for outliers.

```
outlierTest(m1)
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##     rstudent unadjusted p-value Bonferroni p
## 273 1.557504            0.11935           NA
```

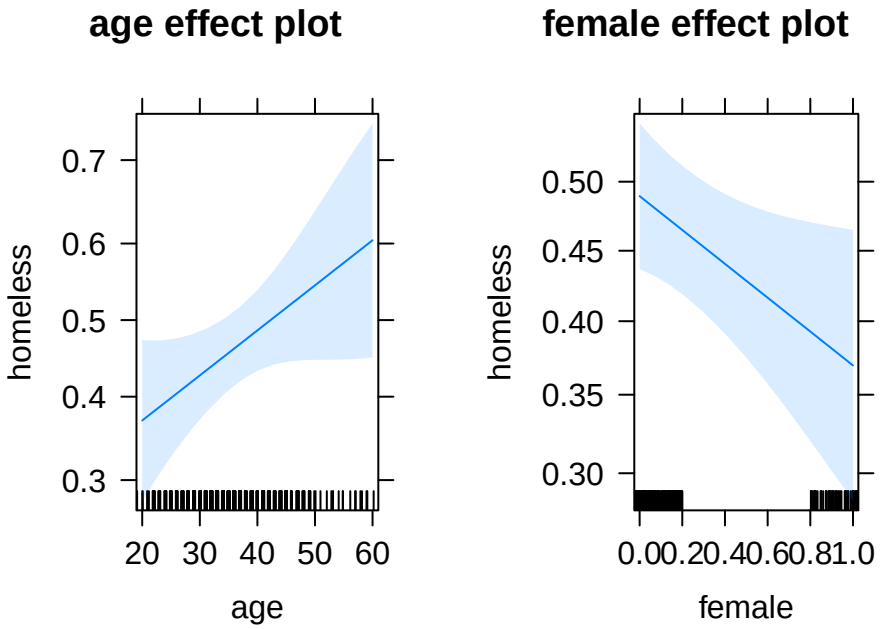
```
dfbetasPlots(m1)
```




```
pearson <- residuals(m1, type="pearson") # obtains the Pearson residuals
devResid <- residuals(m1, type="deviance") # obtains the Deviance residuals
s.pearson <- rstandard(m1, type="pearson") # standardized Pearson residuals
s.devResid <- rstandard(m1, type="deviance") #standardized Deviance residuals
```

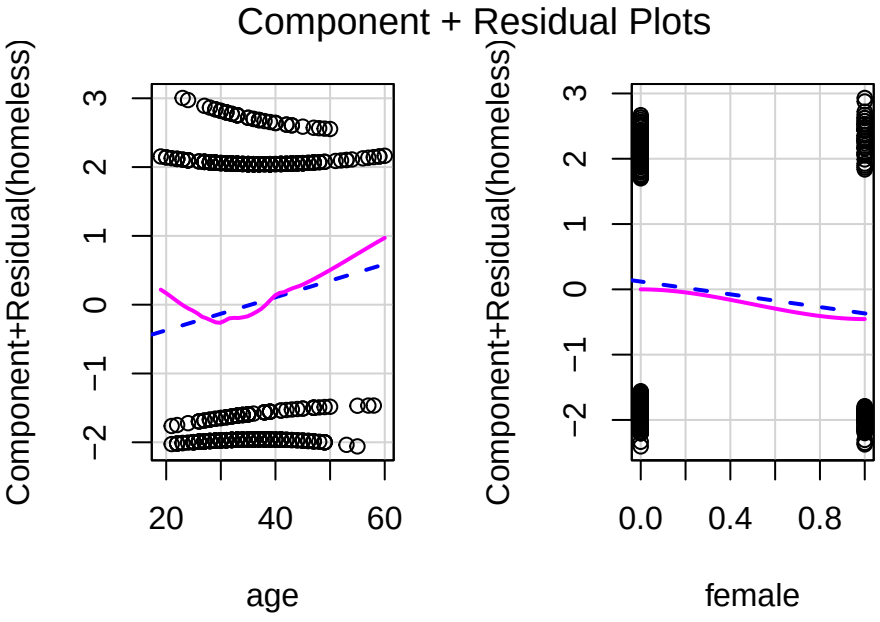
Using the `effects` package, we can create a graph of the effects on the probability scale.

```
plot(allEffects(m1))
```

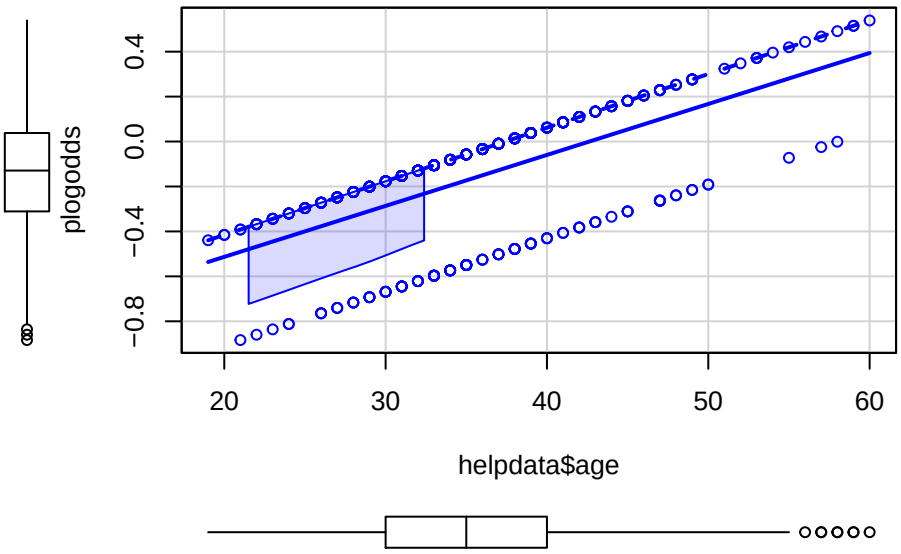



We can examine whether linearity on the logit scale holds by examining component+residual plots or plots of the predicted log-odds of the outcome against each predictor.

```
crPlots(m1)
```

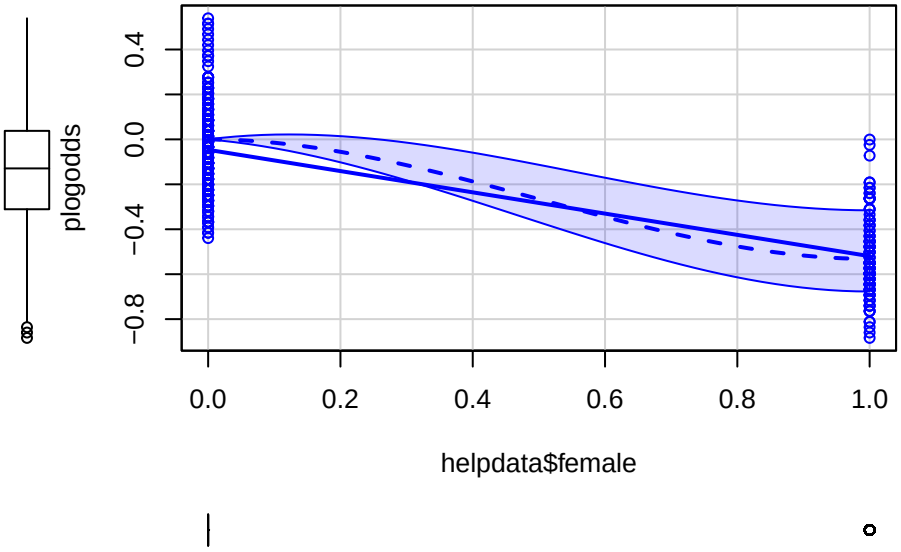



```
plogodds <- predict(m1)
scatterplot(helpdata$age, plogodds)
```






```
scatterplot(helpdata$female, plogodds)
```




## Multinomial and Ordinal Regression

Next we will model an outcome with more than two categories:

1. as a set of unordered categories, using a generalization of the binary logistic model; and
2. extending the binary logistic regression model to ordered categories

### Comparison

The two approaches to modeling response variables with more than two categories — the multinomial logistic model and the ordinal logistic model — address different sets of log-odds, corresponding to different dichotomies constructed from the multiple categories.

Consider, for example, the ordered categories {1, 2, 3, 4}:

Treating category 1 as the baseline, the coefficients of the multinomial logit model apply directly to the dichotomies {1, 2}, {1, 3}, and {1, 4}, and indirectly to any pair of categories.

The ordinal logit model applies to the dichotomies {1, 234}, {12, 34}, and {123, 4}, imposing the restriction that only the intercepts of the three regression equations differ.

Which of these models is most appropriate depends partly on the structure of the data and partly on our research question.

**Illustration: Ordinal Regression**

We will use data on mental impairment. Note that this data is different than the mental impairment data that you used for a homework a while ago. There are three variables:

Mental Impairment: 1=well, 2=mild symptoms, 3=moderate symptoms, and 4=impaired

Life events index: a composite measure of severity of important life events within the past three years. It ranges from 0 to 9.

SES: 0=low and 1=high

The data are available on Canvas in the Data Module. Download the data and save it in the same folder as this .Rmd file.

The function `polr()` from the `MASS` package requires that the outcome, `impair`, must be a factor, so we convert it to a factor after reading in the data and obtaining some summary statistics.

```
mental <- read.table("Mental.dat", header = TRUE)
summary(mental)
```

```
##      impair           ses            life
##  Min.   :1.000   Min.   :0.00   Min.   :0.000
##  1st Qu.:1.000   1st Qu.:0.00   1st Qu.:2.000
##  Median :2.000   Median :1.00   Median :4.000
##  Mean   :2.325   Mean   :0.55   Mean   :4.275
##  3rd Qu.:3.000   3rd Qu.:1.00   3rd Qu.:6.250
##  Max.   :4.000   Max.   :1.00   Max.   :9.000
```

```
str(mental)
```

```
## 'data.frame':    40 obs. of  3 variables:
##  $ impair: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ ses   : int  1 1 1 1 1 0 0 1 1 1 ...
##  $ life  : int  1 9 0 4 3 2 1 3 3 7 ...
```

```
mental$impair <- factor(mental$impair)
str(mental)
```

```
## 'data.frame':    40 obs. of  3 variables:
##  $ impair: Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ses   : int  1 1 1 1 1 0 0 1 1 1 ...
##  $ life  : int  1 9 0 4 3 2 1 3 3 7 ...
```

```
clm <- polr(impair ~ life + ses, method = "logistic", data=mental)
summary(clm)
```

```
##
## Re-fitting to get Hessian
```

```
## Call:
## polr(formula = impair ~ life + ses, data = mental, method = "logistic")
##
## Coefficients:
##        Value Std. Error t value
```

```
## life  0.3189     0.1210   2.635
## ses  -1.1112     0.6109  -1.819
##
## Intercepts:
##     Value   Std. Error t value
## 1|2 -0.2819  0.6423    -0.4389
## 2|3  1.2128  0.6607     1.8357
## 3|4  2.2094  0.7210     3.0644
##
## Residual Deviance: 99.0979
## AIC: 109.0979
```

The estimate for `ses`, in the column labeled `Value`, is -1.11 and suggests that mental impairment tends to decrease at the higher level of ses. Also, the column labeled `t value` is actually z statistics, so that means that $\pm 1.96$ is the cut-off for statistical significance at $p = .05$. Thus the estimate for ses is not statistically significant at an alpha level of .05.

You can use the predict function to obtain estimated probabilities at fixed values of explanatory variables to help interpret the model.

```
predict(clm, data.frame(ses=0, life=mean(mental$life)), type="probs")
```

```
##         1         2         3         4
## 0.1617811 0.3007047 0.2372921 0.3002222
```

```
predict(clm, data.frame(ses=1, life=mean(mental$life)), type="probs")
```

```
##         1         2         3         4
## 0.3696301 0.3536702 0.1529587 0.1237410
```

At the mean of life events, $\widehat{P}(y = 4) = \widehat{P}(impaired) = 0.300$ at low ses and 0.124 at high ses, while $\widehat{P}(y = 1) = \widehat{P}(well) = 0.162$ at low ses and 0.370 at high ses.

The estimate for `life` is 0.319 and suggests that mental impairment tends to be worse with more life events, and this estimate is statistically significant, $z = 2.635, p < .05$.

Again, we can obtain estimated probabilities at fixed values of the explanatory variables to aid in interpretation.

```
predict(clm, data.frame(ses=0, life=min(mental$life)), type="probs")
```

```
##          1          2          3          4
## 0.42998727 0.34080542 0.13029529 0.09891202
```

```
predict(clm, data.frame(ses=0, life=max(mental$life)), type="probs")
```

```
##          1          2          3          4
## 0.04102612 0.11914448 0.18048372 0.65934567
```

```
predict(clm, data.frame(ses=1, life=min(mental$life)), type="probs")
```

```
##          1          2          3          4
## 0.69621280 0.21463441 0.05428168 0.03487110
```

```
predict(clm, data.frame(ses=1, life=max(mental$life)), type="probs")
```

```
##         1         2         3         4
## 0.1150236 0.2518325 0.2439856 0.3891584
```

At low SES, $\widehat{P}(Y = 4) = \widehat{P}(impaired)$ changes from 0.099 to 0.659 as life events increases from its minimum to maximum values; at high SES, it changes from 0.035 to 0.389.

We can plot the effects using the `effects` package.

```
plot(allEffects(clm))
```

```
##
## Re-fitting to get Hessian
##
##
## Re-fitting to get Hessian
```

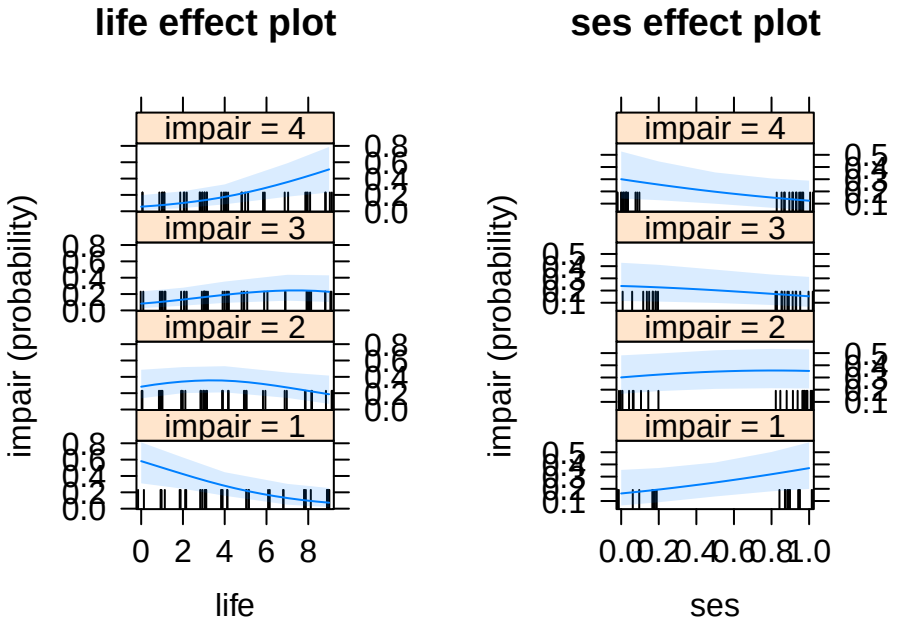



**Illustration: Multinomial Regression**

We can also use the multinomial model to fit these data. There are several functions in different packages that will fit the multinomial model but we will us the `multinom()` function in the `nnet` package.

```
library(nnet)
msat <- multinom(impair ~ life + ses, data=mental)
```

```
## # weights:  16 (9 variable)
## initial  value 55.451774
## iter  10 value 48.349823
## final  value 48.349131
## converged
```

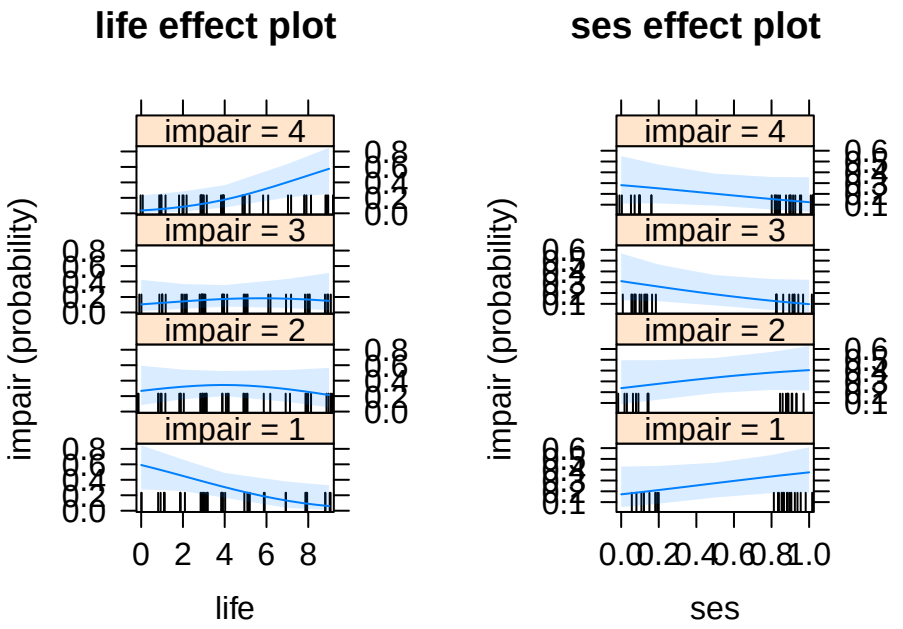
```
summary(msat)
```

```
## Call:
## multinom(formula = impair ~ life + ses, data = mental)
##
## Coefficients:
##   (Intercept)      life        ses
## 2  -0.6552117 0.2285365 -0.2487293
## 3  -0.6621855 0.2930040 -1.9437361
## 4  -1.9025935 0.5601587 -1.6062505
##
## Std. Errors:
##   (Intercept)      life       ses
```

```
## 2   0.8853634 0.1832107 0.9047728
## 3   0.9582681 0.2199351 1.1081671
## 4   1.1107458 0.2190388 1.0804573
##
## Residual Deviance: 96.69826
## AIC: 114.6983
```

By default, it uses the first category, which recall is 'well', as the reference.

```
plot(allEffects(msat))
```




It looks very much the same as the ordinal logistic model but they are not. You can use the `allEffects()` by itself to get the predicted probabilities that it uses for the plot. Comparing the two sets of predicted probabilities for the two models shows that they are not the same.

```
allEffects(clm)
```

```
##
## Re-fitting to get Hessian
##
##
## Re-fitting to get Hessian
##  model: impair ~ life + ses
##
## life effect (probability) for 1
## life
##         0         2         4         7         9
## 0.5815827 0.4234934 0.2796555 0.1297970 0.0730688
##
## life effect (probability) for 2
## life
##         0         2         4         7         9
## 0.2794591 0.3425757 0.3541377 0.2695805 0.1869671
##
## life effect (probability) for 3
## life
```

```
##          0          2          4          7          9
## 0.08273445 0.13262752 0.19041230 0.24364800 0.22766597
##
## life effect (probability) for 4
## life
##          0          2          4          7          9
## 0.05622379 0.10130330 0.17579454 0.35697451 0.51229817
##
## ses effect (probability) for 1
## ses
##         0       0.2       0.5       0.8         1
## 0.1617811 0.1942247 0.2517276 0.3195044 0.3696301
##
## ses effect (probability) for 2
## ses
##         0       0.2       0.5       0.8         1
## 0.3007047 0.3237442 0.3482270 0.3571952 0.3536702
##
## ses effect (probability) for 3
## ses
##         0       0.2       0.5       0.8         1
## 0.2372921 0.2263400 0.2025236 0.1733800 0.1529587
##
## ses effect (probability) for 4
## ses
##         0       0.2       0.5       0.8         1
## 0.3002222 0.2556911 0.1975218 0.1499205 0.1237410
```

```
allEffects(msat)
```

```
##  model: impair ~ life + ses
##
## life effect (probability) for 1
## life
##          0          2          4          7          9
## 0.59113452 0.44992618 0.30476942 0.12934774 0.06040317
##
## life effect (probability) for 2
## life
##         0         2         4         7         9
## 0.2677439 0.3218687 0.3443604 0.2901063 0.2139746
##
## life effect (probability) for 3
## life
##         0         2         4         7         9
## 0.1046690 0.1431440 0.1742222 0.1780906 0.1494314
##
## life effect (probability) for 4
## life
##          0          2          4          7          9
## 0.03645258 0.08506109 0.17664801 0.40245539 0.57619078
##
## ses effect (probability) for 1
## ses
##         0       0.2       0.5       0.8         1
```

```
## 0.1718204 0.2117585 0.2751940 0.3373452 0.3756073
##
## ses effect (probability) for 2
## ses
##         0       0.2       0.5       0.8         1
## 0.2370407 0.2779615 0.3352558 0.3814216 0.4040736
##
## ses effect (probability) for 3
## ses
##          0        0.2        0.5        0.8          1
## 0.31008817 0.25907199 0.18791977 0.12857657 0.09704889
##
## ses effect (probability) for 4
## ses
##         0       0.2       0.5       0.8         1
## 0.2810507 0.2512080 0.2016304 0.1526567 0.1232702
```

We can compare the multinomial model to the ordinal logit model, which has 4 fewer parameters and is therefore simpler.

```
x2 <- 2*(deviance(clm) - deviance(msat))
x2
```

```
## [1] 4.799268
```

```
pchisq(x2, 4, lower.tail=FALSE)
```

```
## [1] 0.3085208
```

We fail to reject the null hypothesis, which is that the models fit equally well, $\chi^2(4) = 4.799, p = 0.309$. Because they fit equally well, we will take the simpler ordinal logit model. Failing to reject this null hypothesis also implies that the proportional odds assumption holds. Of course, if our categories were not ordered, we would not be able to use the ordered logit model.

## Summary

It is problematic to apply least-squares linear regression to a dichotomous response variable:

- The errors cannot be normally distributed and cannot have constant variance.
- Even more fundamentally, the linear specification does not confine the probability for the response to the unit interval.

The logit model is simple to interpret, since it can be written as a linear model for the log-odds:

$$\log_e \frac{\pi_i}{1 - \pi_i} = \beta_0 + \beta_1 X_{i1} + \ldots + \beta_k X_{ik}$$

The model can be fit to data by the method of maximum likelihood.

Wald tests and likelihood-ratio tests for the coefficients of the model parallel $t$-tests and $F$-tests for the linear model.

The deviance for the model, defined as $G^2 = -2 \log_e L$, is analogous to the residual sum of squares for a linear model.

Several approaches can be taken to modeling response variables with more than two categories, including:

(a) modeling the categories using a logit model based on the multinomial distribution;

(b) fitting the ordinal logit model to a response variable with ordered categories and modeling the cumulative logits.