GLMs for binary outcomes

It is often difficult to understand how a binary response can give rise to a smooth, continuous relation between the predicted response, usually the probability of an event, and a continuous explanatory variable. Here, we illustrate two approaches which balance the tradeoff between ***exposure*** (showing the data) and ***summarization*** (compressing the data with a model summary).

The first method uses the ggplot2 package to plot the predicted response probability together with the discrete observations in what we call ***full model plots*** for the variables shown in a given plot. The second method uses the effects package to plot the high-order terms in a given model, providing a visual summary of all effects, but controlling for other variables in the model. This provides an instructive illustration of the difference between ***marginal plots*** and ***conditional plots***.

Example: Passengers on the Titanic– data plots [titanic-glm]

Data on 1309 passengers on the Titanic is recorded in the data frame `Titanicp` in the vcdExtra package. The goal is to understand how survival (*survived*) is related to the available explanatory variables. Here we use just passenger class (*pclass*), *age*, and *sex* as predictors. We load it into the R session using[1]

```
data(Titanicp, package="vcdExtra")
Titanicp <- Titanicp[!is.na(Titanicp$age),]
```

We will fit a logistic regression model for 'survived' using a generalized linear model, however before doing so formally it is useful to view the data together with some smoothed summary. For this purpose, the ggplot2 package is most convenient and flexible, because it offers a variety of smoothing methods and makes it easy to show confidence bands around the fitted curve.

The basic plot of *survived* vs. *age* is produced by `ggplot()` as shown below,[2] giving Figure 1. Using `color=sex` gives different point and line colors, but automatically also stratifies the plot by the levels of this variable. The default smoothing method for `stat_smooth()` is `loess`, producing a nonparametric smoothed curve. Adding `geom_point()` plots the binary observations, here jittered to reduce overplotting.

```
require(ggplot2)
ggplot(Titanicp, aes(age, as.numeric(survived)-1, color=sex)) +
  stat_smooth(method="loess", formula=y~x,
              alpha=0.2, size=2, aes(fill=sex)) +
  geom_point(position=position_jitter(height=0.03, width=0)) +
  xlab("Age") + ylab("Pr (survived)")
```

The points in Figure 1 show the data; the `loess` curves provide a minimal, but useful summary: among females, survival rises steadily with *age*, while for males, survival appears to drop precipitously agmong the young and then level off. The 95% confidence bands give a sense of relative precision, and are quite wide for those over 60 years.

Alternatively, one could use `stat_smooth(method="lm", ...)` to display the fitted values from a linear probability model, but here it is of more interest to show the fitted logistic model (separately for males and females), using `stat_smooth(method="glm", family=binomial ...)`, as shown in Figure 2.

```
ggplot(Titanicp, aes(age, as.numeric(survived)-1, color=sex)) +
  stat_smooth(method="glm", family=binomial, formula=y~x,
              alpha=0.2, size=2, aes(fill=sex)) +
  geom_point(position=position_jitter(height=0.03, width=0)) +
  xlab("Age") + ylab("Pr (survived)")
```

Showing these results on the scale of probabilities is easier to interpret, however the relationship between `Pr(survived)` and `age` is nonlinear, making it harder to understand interactions in the more complex models

[1] *age* contains 263 missing values. Removing these in the examples below does no harm, and avoids repeated warnings.

[2] *survived* is a factor, with levels `"died", "survived"`, represented as `1, 2` in the dataset. The expression `as.numeric(survived)-1` converts this to a 0/1 variable.

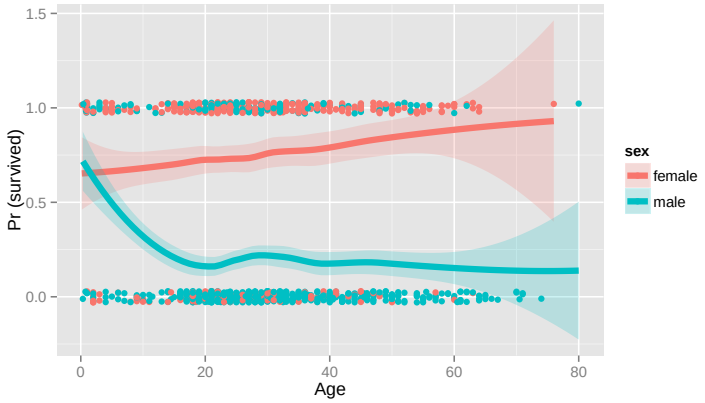


Figure 1: Survival on the Titanic, by age and sex, with a `loess` smooth and 95% confidence band

we show below. One simple way to view the results on the logit scale is to simply transform the Y axis using `coord_trans()`. Using this method does not allow the binary observations to be shown, however, because `logit(0)` and `logit(1)` are infinite.

```
logit <- function(x) log(x)/log(1-x)
ggplot(Titanicp, aes(age, as.numeric(survived)-1, color=sex)) +
  stat_smooth(method="glm", family=binomial, formula=y~x,
              alpha=0.2, size=2, aes(fill=sex)) +
  scale_y_continuous(breaks=c(.10, .25, .50, .75, .90)) +
  coord_trans(y="logit") + xlab("Age") + ylab("Pr (survived)")
```

These plots don't use passenger class, so this variable is ignored (or pooled, collapsed over) within each plot and fitted curve. Such marginal plots may be misleading if there are interactions of ***`pclass`*** with other variables. This turns out to be true here, as we will see in Example `[titanic-eff]`.

One of the strengths of ggplot2 is that it is simple to add faceting by one or two additional variables to show the same information (points, lines, curves) in different panels, broken down by those variables for easy comparison. This is done simply by adding (literally, + in R) `facet_grid(. ~ pclass)` to the plot in Figure 2, producing Figure 4.

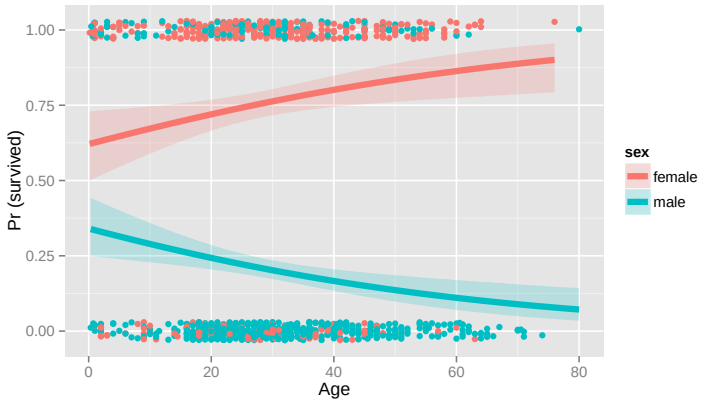


Figure 2: Survival on the Titanic, by age and sex, with a `glm` smooth and 95% confidence band

```
p <- ggplot(Titanicp, aes(age, as.numeric(survived)-1, color=sex)) +
  stat_smooth(method="glm", family=binomial, formula=y~x,
              alpha=0.2, size=2, aes(fill=sex)) +
        geom_point(position=position_jitter(height=0.03, width=0)) +
        xlab("Age") + ylab("Pr (survived)")
# facet by pclass
p + facet_grid(. ~ pclass)
```

For direct comparison with the marginal plot in Figure 4, it is also easy to add the plot collapsed over *pclass* as one more panel in this plot, by adding the option `margins=TRUE` in the last line in the code above (this plot is not shown).

```
# add plot collapsed over pclass
p + facet_grid(. ~ pclass, margins=TRUE)
```

The same type of plot as Figure 4 can be produced using facets for *sex*, with separate points and curves within each panel by interchanging the roles of *sex* and *pclass* in the code above, giving Figure 5.

```
# facet by sex, curves by class
p <- ggplot(Titanicp, aes(age, as.numeric(survived)-1, color=pclass)) +
  stat_smooth(method="glm", family=binomial, formula=y~x,
              alpha=0.2, size=2, aes(fill=pclass)) +
  geom_point(position=position_jitter(height=0.03, width=0)) +
        xlab("Age") + ylab("Pr (survived)")
# facet by sex
p + facet_grid(. ~ sex)
```

The plots above suggest that there are important interactions among the predictors *pclass*, *age*, and *sex* in their impact on *survival*. In particular, the interaction between sex and age appears to differ over passenger class (in Figure 4), and that between passenger class and age appears to differ between males and females (in Figure 5) suggesting a three-way interaction, but this suggestion turns out not to be supported by the data as we will see below.

One difficulty is that it is hard to determine what constitutes an interaction in plots on the scale of predicted probabilities, where the relations are non-linear. As well, such full model plots don't provide any way to tell which apparent interactions are important. △

Example: Passengers on the Titanic– effect plots [titanic-eff]

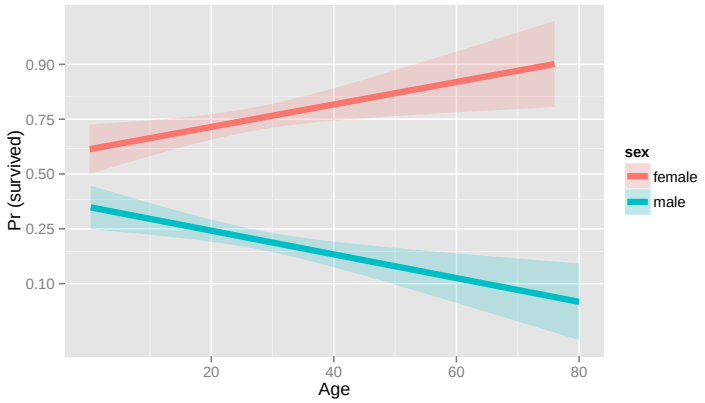


Figure 3: Survival on the Titanic, by age and sex plotted on the log odds scale where they are linear

The effects package provides a simple way to avoid these difficulties, by calculating and plotting the predicted effects for terms in a given model, where in any given term (e.g., `sex:age`) all low-order relatives (`sex`, `age`) are automatically included, and other variables (e.g., `pclass`) are averaged over in a sensible and flexible way. The function `allEffects()` in this package identifies all of the ***high-order terms*** in a given model, and provides plotting methods to visualize them.

As well, the plotting functions plot the response variable (survived) on the logit scale (by default), where relations are assumed to be linear, but labels the tick marks with their transformed probability values. Both of these features facilitate interpretation.

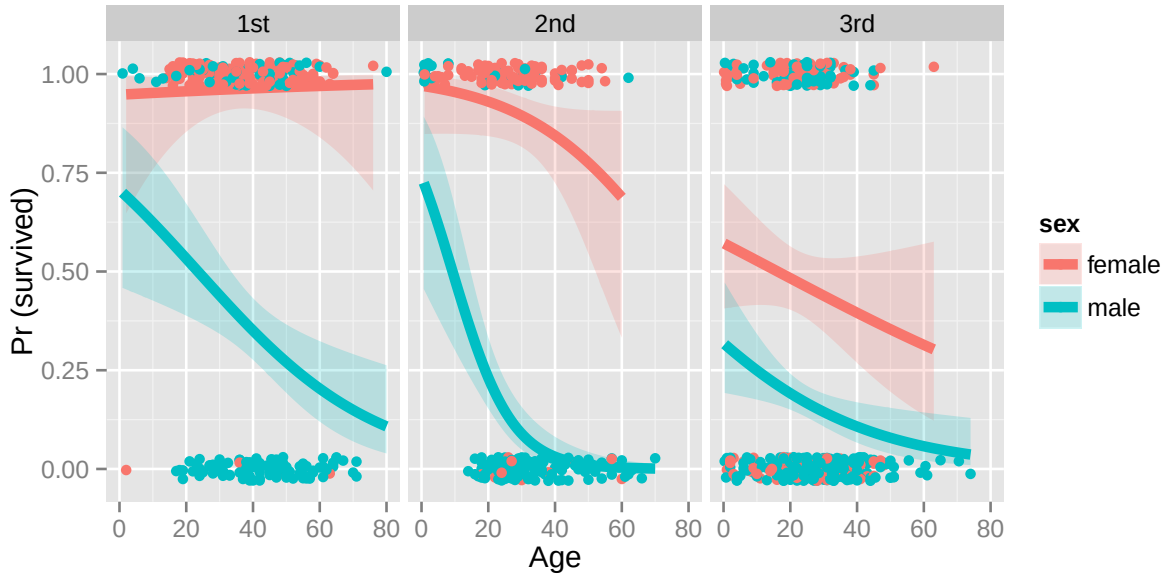


Figure 4: Survival on the Titanic, by age and sex, with panels for passenger class

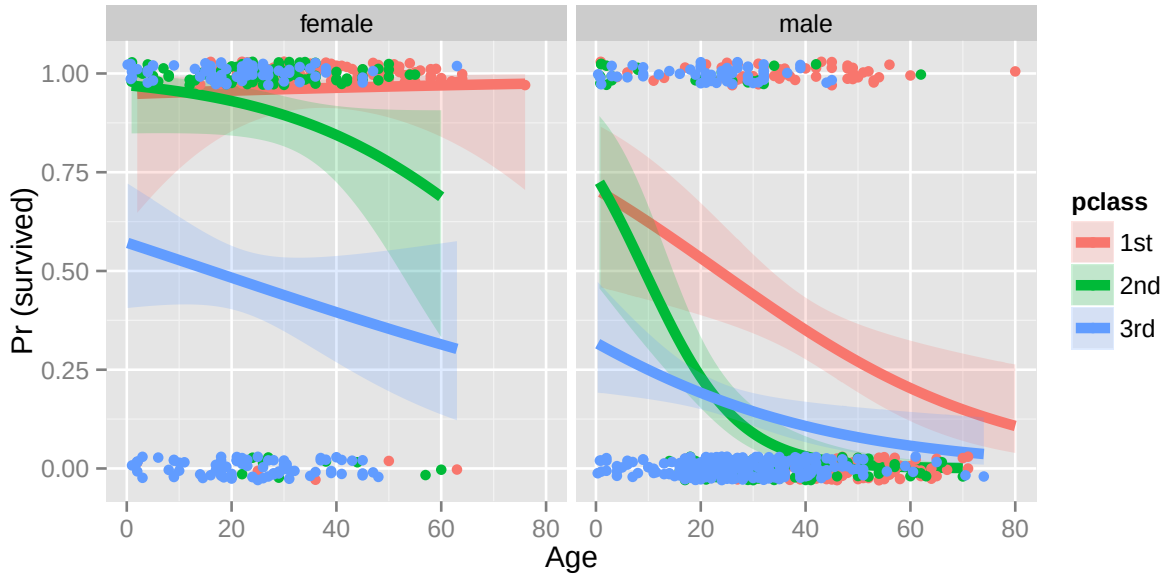


Figure 5: Survival on the Titanic, by age and passenger class, with panels for passenger sex

But first, we need to determine a reasonable model.

A simple way to proceed is to fit some initial screening models, using only main effects, then all two-way terms, ..., up to the full model containing all n-way effects for n predictors. The `anova()` method gives a compact summary of the differences among these.

```
titanic.glm1 <- glm(survived ~ pclass + sex + age, data=Titanicp, family=binomial)
titanic.glm2 <- glm(survived ~ (pclass + sex + age)^2, data=Titanicp, family=binomial)
titanic.glm3 <- glm(survived ~ (pclass + sex + age)^3, data=Titanicp, family=binomial)
anova(titanic.glm1, titanic.glm2, titanic.glm3, test="Chisq")

## Analysis of Deviance Table
##
## Model 1: survived ~ pclass + sex + age
## Model 2: survived ~ (pclass + sex + age)^2
## Model 3: survived ~ (pclass + sex + age)^3
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      1041        982
## 2      1036        918  5     64.6  1.4e-12 ***
## 3      1034        916  2      1.9     0.39
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

An even more compact summary, including AIC and BIC statistics, may be obtained using `vcdExtra::sumarise()` with a "glmlist" object:

```
vcdExtra::summarise(glmlist(titanic.glm1, titanic.glm2, titanic.glm3))

## Model Summary:
##              LR Chisq   Df Pr(>Chisq)   AIC   BIC
## titanic.glm1      982 1041      0.902 -1100 -5323
## titanic.glm2      918 1036      0.996 -1154 -5357
## titanic.glm3      916 1034      0.996 -1152 -5347
```

From this, we see that the model `titanic.glm2` with all two-way terms is substantially better than the one-way model, and is not improved by adding the three-way interaction of `pclass:sex:age`. The `summary()` method for "glm" objects provides details of the goodness of fit of a given model, and the estimated coefficients, together with signifcance tests.

```
summary(titanic.glm2)

##
## Call:
## glm(formula = survived ~ (pclass + sex + age)^2, family = binomial,
##     data = Titanicp)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.614  -0.675  -0.443   0.377   3.245
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)       3.39036    0.80528    4.21  2.6e-05 ***
## pclass2nd         0.76467    0.95984    0.80  0.42565
## pclass3rd        -3.27000    0.76482   -4.28  1.9e-05 ***
## sexmale          -2.59223    0.75357   -3.44  0.00058 ***
```

```
## age               -0.00395    0.01757   -0.22  0.82216
## pclass2nd:sexmale -0.87884    0.68994   -1.27  0.20274
## pclass3rd:sexmale  1.88570    0.58463    3.23  0.00126 **
## pclass2nd:age     -0.06053    0.02142   -2.83  0.00471 **
## pclass3rd:age     -0.00624    0.01619   -0.39  0.69971
## sexmale:age       -0.03138    0.01512   -2.08  0.03793 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1414.62  on 1045  degrees of freedom
## Residual deviance:  917.84  on 1036  degrees of freedom
## AIC: 937.8
##
## Number of Fisher Scoring iterations: 5
```

In general, I don't find these numerical results particularly useful for interpretation. *pclass* and *age* are factors, and so the coefficients of terms involving them relate to the parameterization used by `glm()`, which here is the (default) `contr.treatment()`, where the first level of a factor is the baseline, against which the others are compared.[3]

```
contrasts(Titanicp$pclass)

##     2nd 3rd
## 1st   0   0
## 2nd   1   0
## 3rd   0   1
```

The `Anova()` function in the car package gives a more useful and compact summary here,

```
Anova(titanic.glm2)

## Analysis of Deviance Table (Type II tests)
##
## Response: survived
##            LR Chisq Df Pr(>Chisq)
## pclass        121.5  2    < 2e-16 ***
## sex           275.8  1    < 2e-16 ***
## age            34.9  1    3.4e-09 ***
## pclass:sex     38.2  2    5.2e-09 ***
## pclass:age     10.1  2     0.0065 **
## sex:age         4.3  1     0.0374 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From this, we can proceed to visualize the predicted effects for the three two-way interactions in the `titanic.glm2` model. `allEffects()` calculates the fitted values for each term and returns a "efflist" object, a list of "eff" objects for the terms.

[3] In this example, *pclass* should arguably be treated as an *ordered* factor, so that differences among the passenger classes are resolved into linear trends (slopes) and quadratic trends (curvature) on the logit scale. This can be done by re-assigning *pclass* globally as an ordered factor, `Titanicp$pclass <- ordered(Titanicp$pclass)`, or better yet, by using the option `contrasts=list(pclass=contr.poly)` in the call to `glm()` to use `contr.poly()` locally within this call. The parameterization used makes no difference in overall tests of model effects or in model-based plots, but does make a difference for interpretation of parameter values.

```
library(effects)
titanic.eff2 <- allEffects(titanic.glm2)
names(titanic.eff2)

## [1] "pclass:sex" "pclass:age" "sex:age"
```

`allEffects()` is very general in how variables not included in a given term are represented. By default, these other columns of the model matrix are averaged over (`typical=mean`) and factors such as ***`pclass`*** and ***`sex`*** are represented by their proportions in the data. These options can be changed as shown below, to use the median value, and to calculate predicted effects for equal proportions of class and sex.

```
titanic.eff2a <- allEffects(titanic.glm2,
  typical=median,
  given.values=c(pclass2nd=1/3, pclass3rd=1/3, sexmale=0.5)
        )
```

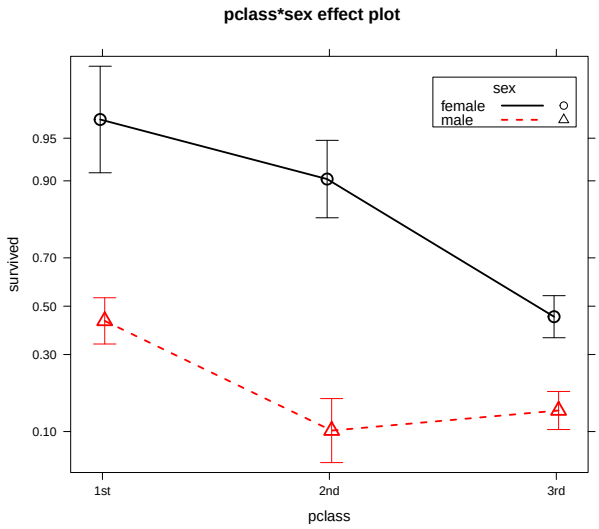
For interactive use, it is easy to plot various terms using a menu to select the high-order terms to be plotted; `...` stands for other arguments to control these plots.

```
plot(titanic.eff2, ask=TRUE, ...)
```

Below we plot these terms separately, using some options to control the plots. In particular, `multiline=TRUE` plots the levels of a factor within a single plot to make visual comparison easier, rather than using a multi-panel display. By default, this suppresses confidence intervals, so we turn that back on using `ci.style="bars"`.

The first term, showing the `pclass*sex` effect can be plotted as follows:

```
ticks <- list(at=c(.01, .05, seq(.1, .9, by=.2), .95, .99))
plot(titanic.eff2[1], ticks=ticks, multiline=TRUE, ci.style="bars", key=list(x=.7, y=.95))
```



It can be seen directly that although women were, overall, more likely to survive than men, the difference in survival between the sexes in 3^{rd} class were much smaller than in 1^{st} or 2^{nd} class, which explains the `pclass:sex` interaction.

The other two terms plotted in the same way:

```
plot(titanic.eff2[2], ticks=ticks, multiline=TRUE, ci.style="bars", key=list(x=.7, y=.95))
```

```
plot(titanic.eff2[3], ticks=ticks, multiline=TRUE, ci.style="bars", key=list(x=.7, y=.95))
```

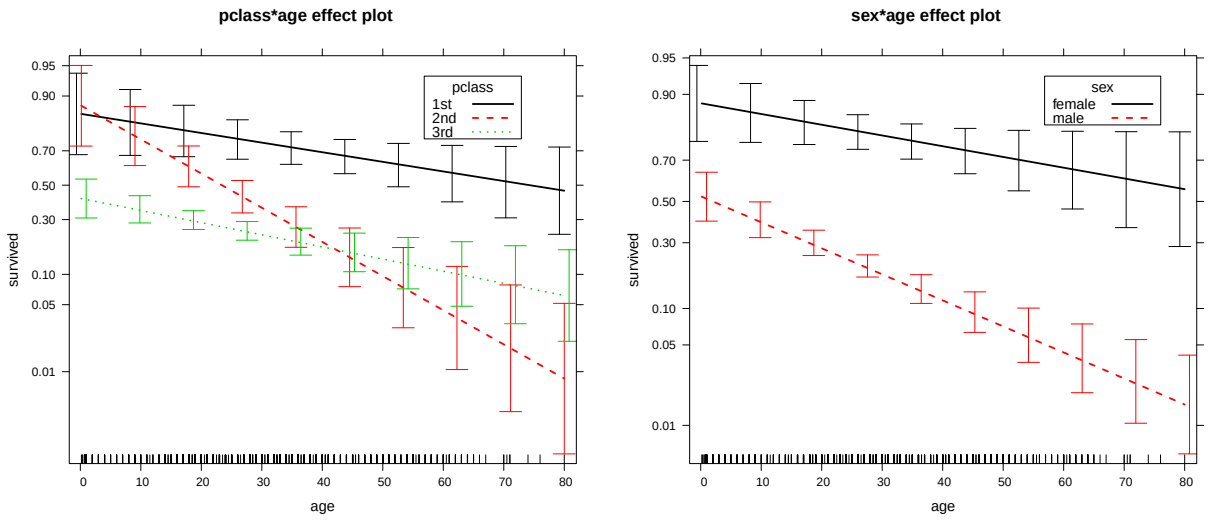


Figure 6: Effect plots for two-way interactions in the model `titanic.glm2`. These plots show the predicted value of the log odds of survival for the two variables shown in each plot, averaged over the remaining variable.

The interpretation of the results from these plots is much simpler than from a table of coefficients or even than from the full model plots shown earlier. In Figure 6 (left) it can be seen that while survival in all classes decreased with age, and survival decreased overall with lower class, the effect of age on survival was much greater in 2^{nd} class– a steeper negative slope.

The right panel of Figure 6 shows the `sex*age` effect that we have seen in earlier figures (Figure 2 and Figure 3), but with one crucial difference (that argues strongly for effect plots): In the earlier full model plots, passenger class was not controlled (or adjusted for), so the apparent tendency of older female passengers to be more likely to survive than younger ones is due in part to the different distributions of passenger class across sex and age. A correct interpretation of the `sex*age` effect is that controlling for passenger class, for both genders, survival decreased with age, but not as much for women as for men.

△