### 16.4.2 Fitted Model Values

Logistic regression involves two different kinds of fitted values: the intermediate "link" value $Y$ and the probability $P$. The `fitted` operator returns the probabilities:

```
> probs = fitted(mod)
```

There is one fitted probability value for each case.

To use a logistic regression model for estimation of a probability, you should turn the model into a function that accepts the explanatory variables as inputs. This is done with `makeFun()`.

```
> pf = makeFun(mod)
```

Calculating the model probability is a matter of providing inputs to the function. For instance, here is the model probability for a 70-year old smoker:

```
> pf( age=70, smoker="Yes" )

   1
0.78
```

As always, make sure to use the appropriate levels for categorical variables.

By default, `makeFun()` will produce a function that gives the model probabilities. If you want to calculate the link values, you need to tell `makeFun()` that you want this type of output from the function.

```
> linkf = makeFun(mod, type="link")
> linkf(age=70, smoker="Yes")

   1
1.26
```

Notice that the link values are not necessarily between zero and one, although the probability determined by the link value will always be.

### 16.4.3 Which Level is "Yes"?

In fitting a logistic model, it's crucial that the response variable be categorical, with two levels. It happens that in the `whickham` data, the `outcome` variable fits the bill: the levels are `Alive` and `Dead`.

The `glm` software will automatically recode the response variable as 0/1. The question is, which level gets mapped to 1? In some sense, it makes no difference since there are only two levels. But if you're talking about the probability of dying, it's nice not to mistake that for the probability of staying alive. So make sure that you know which level in the response variable corresponds to 1: it's the second level.