Marginal predictions

January 18, 2021

1 Introduction

In this task, we describe various R machineries for displaying predictions in both simple and complex (involving interaction terms) generalized linear models. We first consider existing approaches for *conditioning* predictions and then describe our proposed approach for *marginalized* predictions. These provide a unified and intuitive way of describing relationships from a fitted model, especially complex models involving interaction terms or some kind transformations on the dependent variables whose estimates are usually, but not always, a subject to less clarity of interpretation.

We can derive various quantities from a fitted regression model. The first and most obvious, is the model coefficient estimates. Others include predicted values of the outcome variable—1) predictions at a particular; and 2) mean or median value of the predictors. The first case simply involves evaluation of the fitted model function, say $\hat{f}(X = x)$, at some particular value of x. The second case, chooses the values of the predictor based on its distributional properties.

Most importantly, from these predicted values, we can also generate second class quantities of interest – *conditional* or *marginal* predictions. These quantities describes the change in the predicted value of the dependent variable after changing one independent variable – either a discrete change in the categorical variable(s) or an instantaneous change in continuous variables, while all other variable are held at specified values.

Suppose we are interested in the *conditional* predictions of a particular predictor (hence forth referred as *focal* predictor otherwise *non-focal*), x_f, in the set of predictors. To keep it simple, assume that the model has no interaction terms. Then the idea is to fix the values of *non-focal* predictor(s) at some typical values – typically determined by averaging in some meaningful way, for example, arithmetic mean and average over the levels of the factors of *non-focal* continuous and categorical predictors, respectively. This is achieved by averaging the columns of *model matrix*, $\mathbf{X}$, except for the column of focal predictor.

Consider a simple linear model with linear predictor $\eta = \mathbf{X}\beta$ and let $g(\mu) = \eta$ be an identity link function (in the case of simple linear model), where μ is the predicted (expected) value of outcome variable y. Let $\hat{\beta}$ be the estimate of β, together with the estimated covariance matrix $V(\hat{\beta})$ of $\hat{\beta}$. Let the entries of $\mathbf{X}^*$ include all conditioned (*non-focal*) and *focal* predictors. The model matrix, $\mathbf{X}^*$, inherits most of its key properties, for example transformation (e.g., scaling) on the predictors and interactions from the model matrix, $\mathbf{X}$. Then the predicted values $\hat{\eta}^* = \mathbf{X}^*\hat{\beta}$ represents the *conditional* predictions of the focal predictor. Alternatively, we can transform these predictions to response scale using $g^{-1}(\hat{\eta}^*)$.

Further, we can compute the standard errors (SEs) of the *conditional* predictions, $\hat{\eta}^*$, for constructing confidence intervals, as the $\text{sqrt}(\text{diag}(\mathbf{X}^* V(\hat{\beta})\mathbf{X}^{*T}))$. When computing *marginal* predictions, the *uncertainty* as a result of *non-focal* predictors are removed. This can be achieved in two ways:

- using variance-covariance matrix, $V(\hat{\beta})$
- using centered model matrix, $\mathbf{X}^*$

1.1 Variance-covariance

The computation of $\hat{\eta}^*$ remains the same as described above. However, to compute SEs $V(\hat{\beta})$ is modified by *zeroing-out* (the variance-covariance of all non-focal predictors are assigned zero) entries of *non-focal* terms in $V(\hat{\beta})$. This approach requires *centering* of the predictors in the model matrix, X. In other words, the fitted model should have *centered* predictors.

1.2 Centered model matrix

Suppose the *non-focal* entries in X^* are computed by some kind of averaging. Consider centered X^*, $X_c^* = (X^* - \bar{X}^*)$. It follows that the *non-focal* entries in X_c^* are all zero. As a result, the SEs are computed as $\text{sqrt}(\text{diag}(\mathbf{X}_c^* V(\hat{\beta})\mathbf{X}_c^{*T}))$, which actually *zeros-out*. More generally, *centering*, $\mathbf{X}_c^* = \mathbf{X}^* - k$ (for example $k = E(\mathbf{X}^*)$) impacts on the estimated value of the intercept and its associated variance. However, the slopes are not affect by this. The implication of this that since *non-focal* terms in $\mathbf{X}_c^*$ are zero, it doesn't matter what their corresponding values are in the variance-covariance matrix. Hence, we can compute *marginal* predictions from non-centered predictors (in other words, fitted models with predictors in their natural scales).

2 Available packages

The following R packages **effects**, **emmeans** and **margins** implement various schemes for constructing *conditional* predictions. However, currently, their ability to compute *marginal* prediction is limited to the use of variance-covariance approach which requires the fitted model to be centered. We propose **varpred** to overcome this limitation.

3 TODOs

1. Models with interactions
2. GLMs
3. LMEs

4 No interactions

4.1 Simulation

Consider a simple no-interaction terms simulation:

- $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$ s.t $\epsilon \sim N(0, 1)$ and $\{\beta_0 = -5, \beta_1 = 0.1, \beta_2 = -0.6\}$
  - $x_1 \sim unif(1, 9)$
  - x_2 – two level categorical variable

```
> set.seed(101)
> N <- 100
> x1_min <- 1
```

```
> x1_max <- 9
> b0 <- 0.3
> b1 <- 0.1
> b2 <- -0.6
> x2_levels <- factor(c("A", "B"))
> df <- expand.grid(x1u = runif(n=N, min=x1_min, max=x1_max)
+         , x2u = x2_levels
+ )
> X <- model.matrix(~x1u + x2u, df)
> betas <- c(b0, b1, b2)
> df$y <- rnorm(nrow(df), mean= X %*% betas, sd=1)
> df2 <- df
> df <- transform(df
+         , x1c = drop(scale(x1u, scale=FALSE))
+         , x1s = drop(scale(x1u, center=FALSE))
+         , x1sd = drop(scale(x1u))
+ )
> head(df)
```

```
       x1u x2u          y       x1c       x1s       x1sd
1 3.977587   A -0.4524966 -1.222599 0.6940038 -0.5134124
2 1.350599   A  0.1605887 -3.849588 0.2356505 -1.6165773
3 6.677472   A  1.5456482  1.477286 1.1650760  0.6203644
4 6.261523   A -0.4707503  1.061337 1.0925018  0.4456928
5 2.998846   A  1.3489423 -2.201340 0.5232344 -0.9244202
6 3.400439   A -0.4111428 -1.799747 0.5933038 -0.7557773
```

4.2 Model fitting

No scaling or centering covariates

```
> df_temp <- drop(df)          # margins doesn't work with transform
>                             # but we need df to extract attributes :(
> lm_u <- lm(y ~ x1u + x2u, data = df_temp)
```

Centered covariates $(x - \bar{x})$ model

```
> lm_c <- lm(y ~ x1c + x2u, data = df_temp)
```

Scaled covariates (x/sdx) model

```
> lm_s <- lm(y ~ x1s + x2u, data = df_temp)
```

Both scaled and centered covariates

```
> lm_sd <- lm(y ~ x1sd + x2u, data = df_temp)
```

Coefficients estimates

```
> coef_est <- modsummary(fun=coef)
> print(coef_est)
```

```
              lm_c   lm_s  lm_sd   lm_u
(Intercept)  0.808  0.229  0.808  0.229
x1c          0.111  0.638  0.265  0.111
x2uB        -0.535 -0.535 -0.535 -0.535
```

These four model are equivalent (they all have equal loglikehoods).

```
> ll_est <- modsummary(fun=logLik)
> print(ll_est)
```

```
        lm_c    lm_s   lm_sd    lm_u
[1,] -272.27 -272.27 -272.27 -272.27
```

Variance covariance matrix. It is important to note that the slope variance for unscaled and centered models are the same. This makes it easier to center predictions from uncentered models.

```
> vcov_est <- modsummary(fun=function(x)vcov(x), simplify=TRUE
+         , combine="rbind", match_colnames = names(coef(lm_u)))
> print(vcov_est)
```

```
                  (Intercept)    x1u   x2uB model
lm_c.(Intercept)        0.009  0.000 -0.009  lm_c
lm_c.x1c                0.000  0.001  0.000  lm_c
lm_c.x2uB              -0.009  0.000  0.018  lm_c
lm_s.(Intercept)        0.031 -0.024 -0.009  lm_s
lm_s.x1s               -0.024  0.026  0.000  lm_s
```

```
lm_s.x2uB              -0.009  0.000  0.018  lm_s
lm_sd.(Intercept)       0.009  0.000 -0.009 lm_sd
lm_sd.x1sd              0.000  0.005  0.000 lm_sd
lm_sd.x2uB             -0.009  0.000  0.018 lm_sd
lm_u.(Intercept)        0.031 -0.004 -0.009  lm_u
lm_u.x1u               -0.004  0.001  0.000  lm_u
lm_u.x2uB              -0.009  0.000  0.018  lm_u
```

4.3 Conditional predictions

4.3.1 Continuous predictor

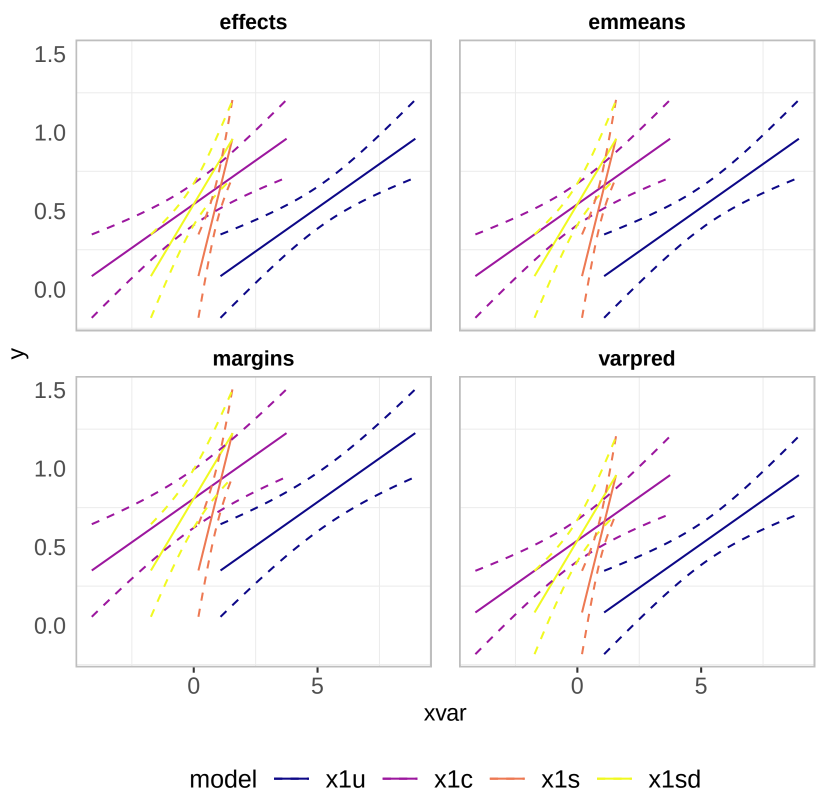


Figure 1: Conditional predictions on the variable-specific values.

Figure 1, displays the conditional predictions for of the models, i.e., variable-specific values (original (u), mean centered (c), divided by standard deviation (s), and both mean centered and scaled (sd)). The predictions looks different but we zoom-in in Figure 2 by transforming back all the scaled predictions to the original (unscaled) values – and actually, conditional predictions are the same for each of the methods. This implies that, if we know the scaling or centering parameter, we can always transform back predictions to the untransformed (unscaled) equivalent.

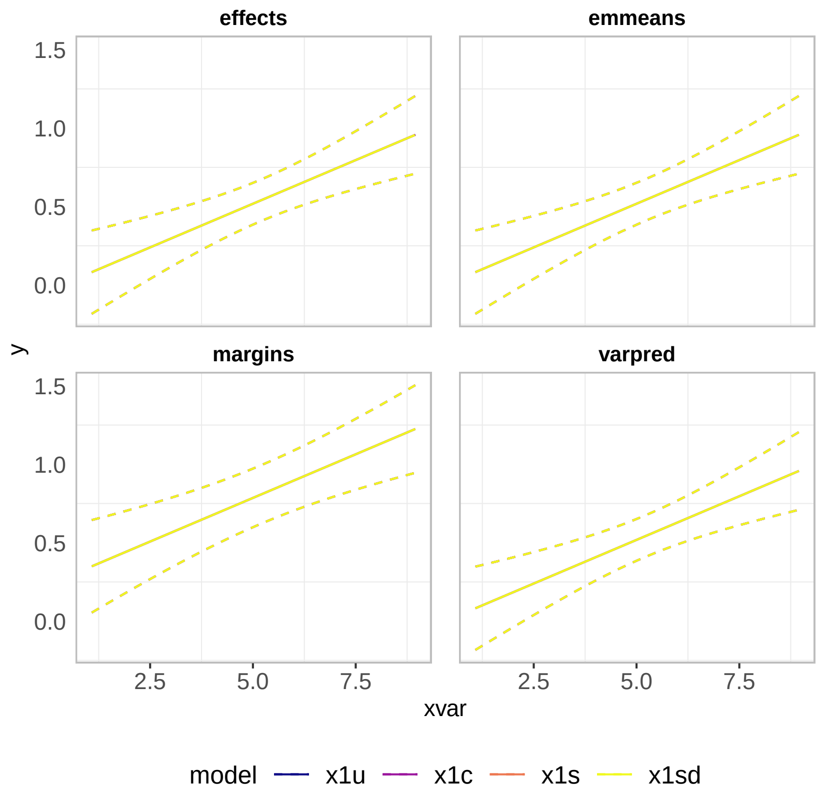


Figure 2: Back-transformed conditional predictions.

4.3.2 Categorical predictors

In this case, it doesn't matter which of the four models we choose since they all give the similar conditional predictions. Later, we'll show how we can estimate marginalized and/or centered (CI) for categorical predictors.

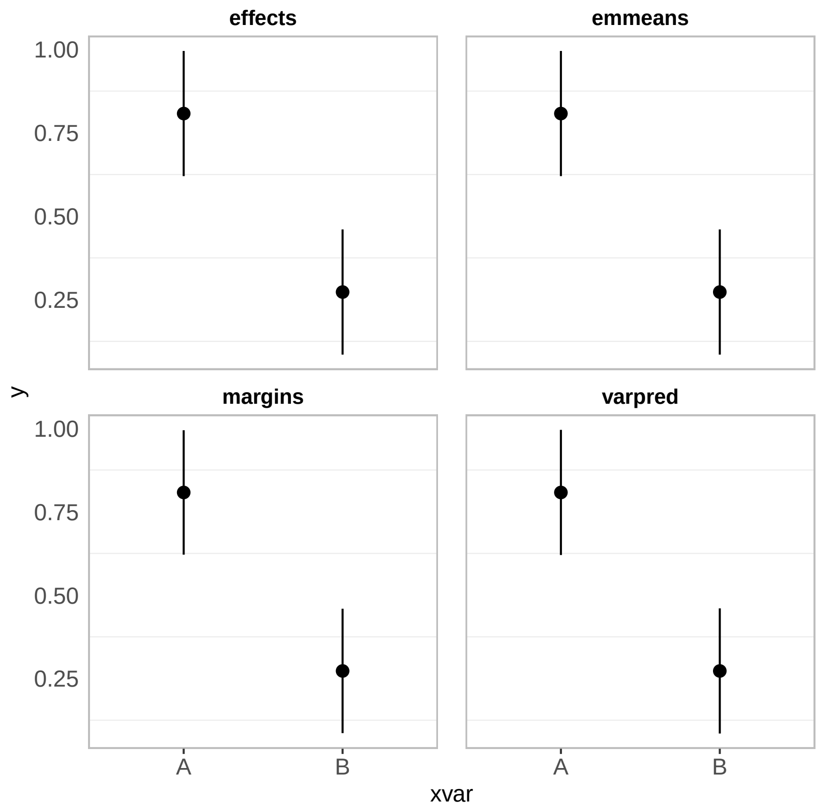


Figure 3: Conditional predictions.

4.4 Marginal predictions

We can implement marginalization by *zeroing-out* the variance of non-focal predictors (possible in all four methods) but this requires that continuous predictors are centered beforehand or appropriate contrast is applied in the case of categorical predictors. On the other hand, in **varpred** marginal predictions are computed by centering the model matrix (when computing SEs). This approach does not require centered predictors and also works with any contrast specification.

4.5 zeroed-out covariance matrix

The **zero_vcov** function in **jdeffects** *zeros-out* variance-covariances of non-focal terms of the predictor in question. For example:

- full variance-covariance matrix

```
              (Intercept)           x1u          x2uB
(Intercept)  0.030729586 -4.169417e-03 -9.047842e-03
x1u         -0.004169417  8.017823e-04 -3.588329e-19
x2uB        -0.009047842 -3.588329e-19  1.809568e-02
```

- *zeroed-out* variance-covariance matrix with `x1u` as the focal variable

```
            (Intercept)          x1u x2uB
(Intercept)           0 0.0000000000    0
x1u                   0 0.0008017823    0
x2uB                  0 0.0000000000    0
```

We repeat all the procedures in conditional predictions section but modify the methods to incorporate *zeroed-out* covariance matrix as one of the inputs.

4.5.1 Continuous predictor

The most modification in each of the methods is to either specify *zeroed-out* covariance matrix or a function to compute the variance-covariance matrix as one of the inputs. In particular, in **varpred**, **emmeans** and **effects** it is specified through **vcov.** and in **margins** through **vcov** argument. However, this seems not to work currently in **cplot** which computes the predictions in **margins**.

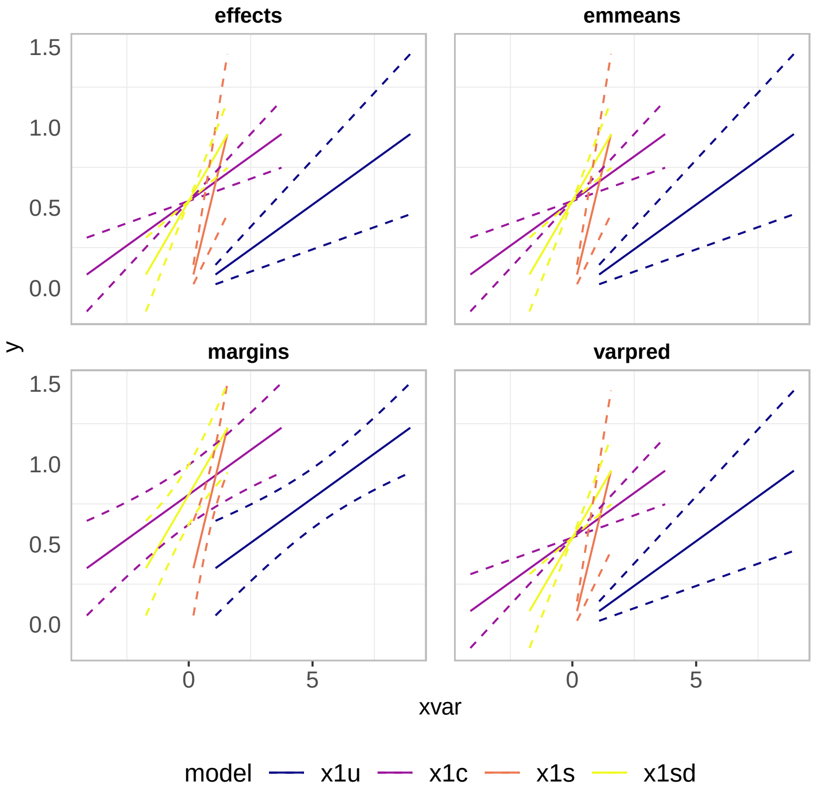
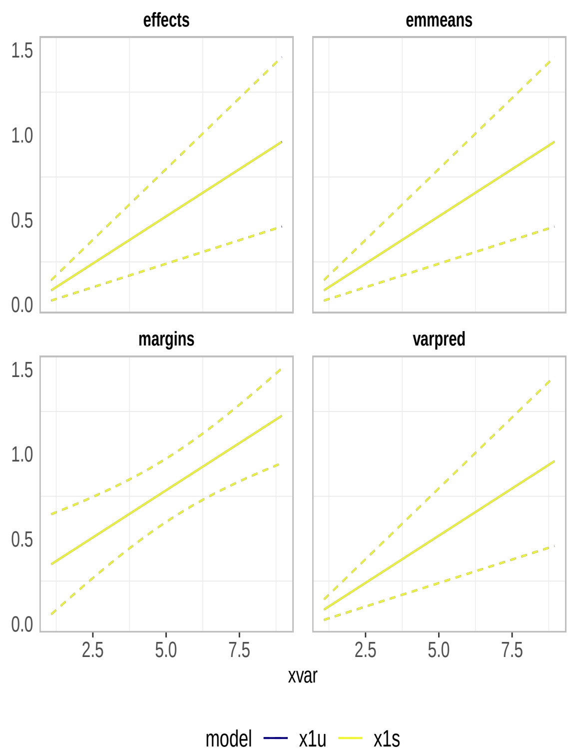


Figure 4: Marginal predictions with the zeroed-out covariances on the variable-specific scale.

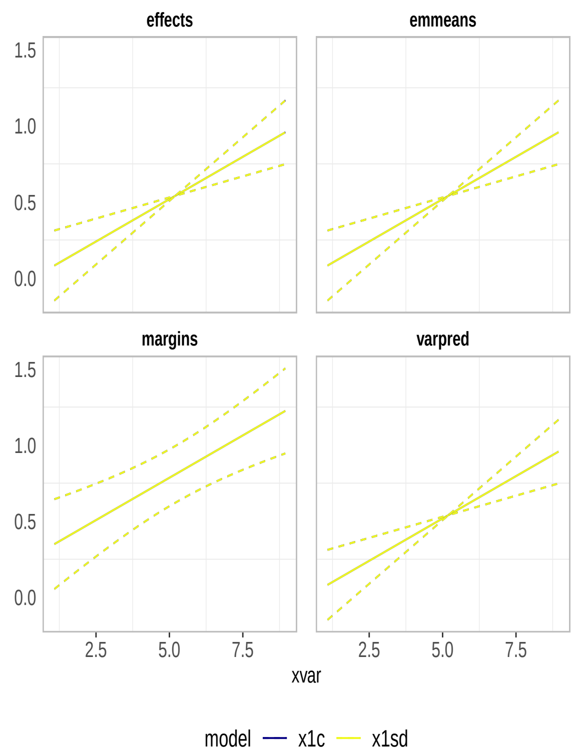
Also, we repeat above implementation but back-transform the predictions to original scale (unscale – uncenter and/or unscale) by modifying the previous code to include scaling parameters. To clearly distinguish between what happens when apply various scaling schemes, we separately do the plots (see Figure 5).

4.6 Centered model matrix

Marginal predictions in **effects** and **emmeans** require models fitted with transformed predictors. However, in **varpred** we can use model fitted using predictors on their original scale and estimate marginal predictions. In our example, we use the unscaled model `lm_u` and simply set `isolate=TRUE`. The predictions (see Figure 6) are similar to Figure 5 b (except for the **margins**).



(a) Either unscaled or devided by the *sd*.



(b) Centered or centered and/or devided by *sd*.

Figure 5: Back-transformed marginal predictions.

```
> ## Centered predictions from unscaled model
> vpred_c <- varpred(lm_u, focal = "x1u", isolate = TRUE)
> plot(vpred_c)
```

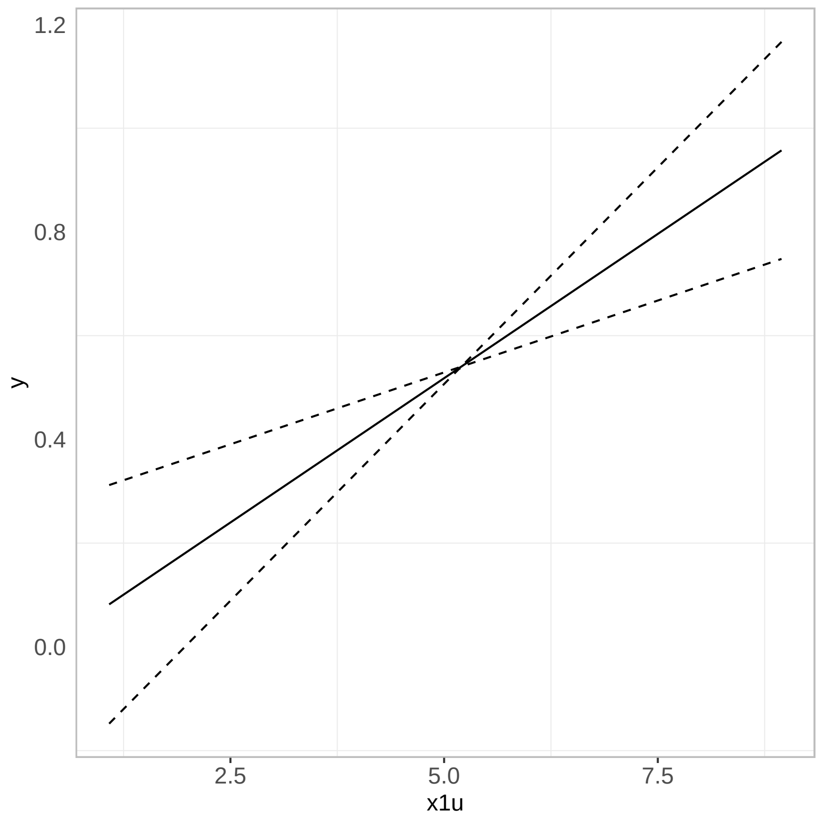


Figure 6: Using **varpred** to obtain marginal predictions from unscaled model.

4.6.1 Categorical predictors

Back to categorical predictor, `x2u`. The default contrast compares the other categories with no-variance reference category (base) and as a result, we can not compute the SEs using *zeroed-out* variance-covariance matrix unless appropriate contrast (`"contr.sum"`) was used during model specification. Model matrix centering provides a way to overcome this and computes centered independent of the contrasts. Currently, only **varpred** provide this functionality.

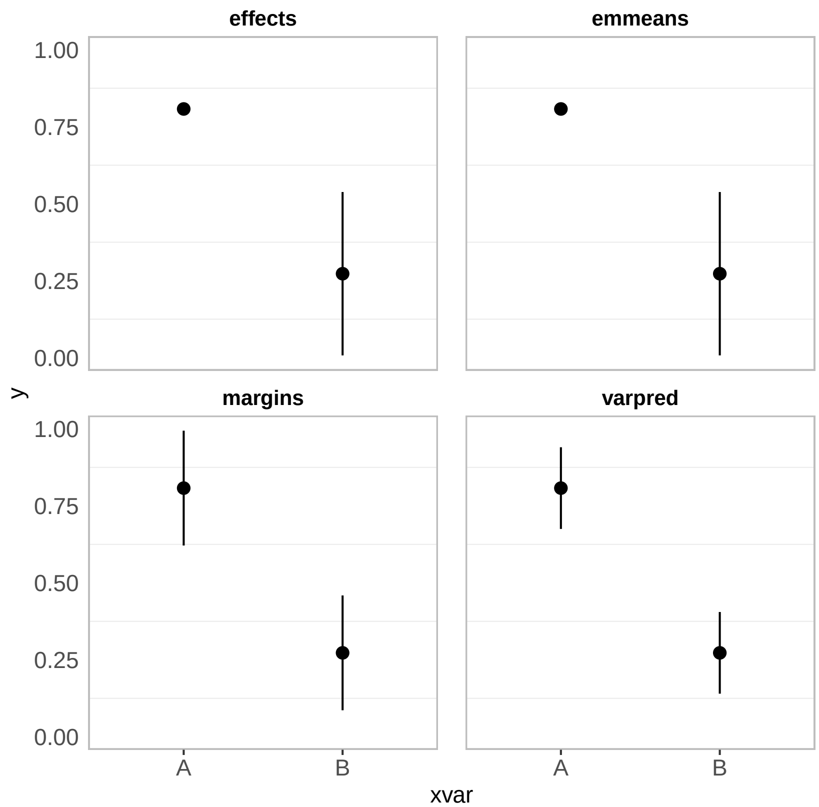


Figure 7: Conditional predictions.