

How about more categories?

One-hot

- There are a few ways of modeling predictors with many categories:
 - **Contrasts:** Each category is compared to a **baseline**, and the coefficients are comparisons between baseline and levels
 - **One-hot:** coefficients are means of each level of the predictor
 - **Residuals:** an overall mean is measured, and coefficients are differences between each level and the global mean

```
> onehot = model.matrix(~0+x, data = df)
> fit_onehot = stan_glm(y ~ 0 + onehot, data = df, cores = 4)
> summary(fit_onehot)[1:3, 1:3]
```

	mean	mcse	sd
onehotxA	0.72	0	0.13
onehotxB	2.04	0	0.14
onehotxC	2.93	0	0.12

```
> fit_contrasts = stan_glm(y ~ x, data = df)
> summary(fit_contrasts)[1:3, 1:3]
```

	mean	mcse	sd
(Intercept)	0.72	0	0.13
xB	1.32	0	0.20
xC	2.20	0	0.186

How about more categories?

- There are a few ways of modeling predictors with many categories:
 - **Contrasts**: Each category is compared to a **baseline**, and the coefficients are comparisons between baseline and levels
 - **One-hot**: coefficients are means of each level of the predictor
 - **Residuals**: an overall mean is measured, and coefficients are differences between each level and the global mean

```
> fit_residual = stan_glmer(y ~ 1 + (1|x),
data = df, prior_intercept =
normal(mean(df$y), 0.1))
> summary(fit_residual)[1:4, 1:3]
```

	mean	mcse	sd
(Intercept)	1.90	0	0.10
b[(Intercept) x:A]	-1.16	0	0.16
b[(Intercept) x:B]	0.14	0	0.17
b[(Intercept) x:C]	1.01	0	0.15