

## 0.1 GLM - Nature of the Model Function

**Deterministic:** For each one of the values of the regressor X, there is a single value of Y. **Stochastic:** Each value of X has a probability distribution associated to Y. **Black-box Models:** is focused on optimizing predictions subject to a set of regressors with less attention on the internal model's process. **Link function:** OLS regression models a continuous response  $Y_i$  (a random variable) via its conditioned mean (or expected value)  $\mu_i$  subject to  $k$  regressors  $X_{i,j}$ , modelling the mean  $\mu_i$  of a discrete-type response (such as binary or a count) is not straightforward. Hence, we rely on a monotonic and differentiable function called the link function.

## 1 Poisson Regression

GLM to model count-type responses. Bar chart count is right skewed then poisson. The equality of the expected value and variance in a random variable is called equidispersion.

The estimates are obtained through maximum likelihood where we assume a Poisson joint probability mass function of the n responses Yi.

```
library(glm)
data(crabs)
crabs <- crabs |> rename(n_males = satell) |> dplyr::select(-y)
group_avg_width <- crabs |> mutate(intervals = cut(crabs$width,
  breaks = 10)) |> group_by(intervals) |> summarise(mean =
  mean(n_males), n = n())
poi_model <- glm(n_males ~ width, family = poisson, data = crabs)
```

**Inference:** The fitted regression model will be used to identify the relationship between the logarithm of the response's mean and regressors. To determine the statistical significance of in this model, we also use the Wald statistic.

```
tidy(poi_model, conf.int = TRUE) |> mutate_if(is.numeric, round, 3)
```

Our sample gives us evidence to reject  $H_0$  ( $p$ -value < 0.001). So carapace width is statistically associated to the logarithm of the mean of n\_males.

### 1.1 Coefficient Interpretation

Moreover, it has a baseline: dark. We can check the baseline level, via levels().

```
poi_model_2 <- glm(n_males~width+color, family = poisson, data =)
tidy(poi_model_2, exponentiate = TRUE, conf.int = TRUE)
```

1.55 indicates that the mean count of male crabs (n\_males) around a female breeding nest increases by 55% when the female color of the prosoma changes from dark to light, while keeping the carapace female width constant.

### 1.2 Predictions

```
round(predict(poisson_model_2, newdata = tibble(width = 27.5, color
  = "light"), type = "response"), 2)
```

## 2 Overdispersion

When the variance is larger than the mean in a random variable, we have overdispersion. This matter will impact the standard error of our parameter estimates in a basic Poisson regression, as we will see further.

with the hypotheses  $H_0 : 1 + \gamma = 1$  |  $H_a : 1 + \gamma > 1$ . When there is evidence of overdispersion in our data, we will reject  $H_0$ .

```
dispersiontest(poisson_model_2) {AER}
```

With  $\alpha = 0.05$ , we reject  $H_0$  since the  $p$ -value < .001. Hence, the poisson\_model\_2 has overdispersion. Using a Negative Binomial model when overdispersion is present helps control your Type I error rate. Hence lower standard error.

## 3 Negative Binomial Regression

A Negative Binomial random variable depicts the number of  $y_i$  failed independent Bernoulli trials before experiencing  $m$  successes with a probability of success  $p_i$

The estimates are obtained through maximum likelihood where we assume a Poisson joint probability mass function of the n responses Yi.

```
library(MASS)
negative_bin_model <- glm.nb(n_males ~ width + color, data = crabs)
summary(negative_bin_model)
```

### 4 Model Selection

```
poi_model<-glm(n_males~ width,family = poisson,data = crabs)
poi_model_2<-glm(n_males~width+color,family = poisson,data = crabs)
library(broom)
summary_poisson_model_2 <- glance(poisson_model_2)
```

We can compare the fits provided by these two models by the de-

viance.  $D_k = k\text{model}/\text{fullModel}$  is formally called residual deviance, which is the test statistic. Large  $D_k$  means model fits poorly compared to the baseline model. Small  $D_k$  means model fits good compared to the baseline model. We cannot use anova() to perform this hypothesis testing. We will have to do it manually via glance().

```
pchisq(summary_poisson_model_2$deviance, #p-value for this test
  df = summary_poisson_model_2$df.residual, lower.tail = FALSE)
```

In Poisson regression, the pchisq test on your model's deviance is a Goodness-of-Fit (GoF) test.  $p$ -value  $\leq .001$ : Reject  $H_0$ : strong evidence that your model does not fit the data.

### 4.1 Analysis of Deviance for Nested Models

```
round(anova(poisson_model, poisson_model_2, test = "Chi"), 4)
```

$H_0$  : The simpler model (Model 1) is sufficient.

$H_a$  : The more complex model (Model 2) fits significantly better. With a  $p$ -value  $\leq 0.05$  (Pr(Chi)), we reject  $H_0$ . Significant evidence that poisson\_model\_2 fits the data better than the simpler poisson\_model. Therefore, we select poisson\_model\_2, which includes the additional predictor for the color of the prosoma.

### 4.2 Akaike Information Criterion-glance()

Drawbacks of the analysis of deviance - tests only nested regression models. AIC makes possible to compare models that are either nested or not.  $AIC_k$  favours models with small values of  $D_k$ . Models with smaller values of  $AIC_k$  are preferred because  $AIC_k = D_k + 2k$ . It also penalizes for including more regressors in the model. Hence, it discourages overfitting.

### 4.3 Bayesian Information Criterion-glance()

An alternative to AIC. The BIC also makes possible to compare models that are either nested or not. For a model with  $k$  regressors,  $n$  observations used for training, and a deviance  $D_k$ ; it is defined as:  $BIC_k = D_k + k * \log(n)$ . Models with smaller values of BIC are preferred.

## 5 Multinomial Logistic Regression

Categorical Type Responses - more than two classes in the categorical response. Models the logarithm of odds.  $\text{logit}(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 X_{i,\text{dance}} + \beta_2 X_{i,\text{val}}$

```
log_model <- glm(formula = genre ~ danceability + valence, data =
  data, family = binomial)
library(broom) tidy(log_model, conf.int = TRUE, exponentiate = TRUE)
```

To fit the model with the package nnet, we use the function multinom(), which obtains the corresponding estimates. Final output is converted those scores into class probabilities using softmax functions

**Inference:** Provided the sample size n is large enough, it has an approximately Standard Normal distribution under  $H_0$ .

```
model <- multinom(formula = genre ~ danceability + valence, data =
  spotify_training)
mult_output <- tidy(model, conf.int = TRUE, exponentiate = TRUE) |>
  dplyr::filter(p.value < 0.05)
```

Baseline response is edm, we can conclude There is a statistical difference in danceability in edm versus r&b and rock.

### 5.1 Coefficient Interpretation

$\beta_1^{(\text{latin,edm})}$ : for each unit increase in the valence score in the Spotify catalogue, the song is 1.05 times more probable to be latin than edm.

### 5.2 Predictions

```
pred_probs <- round(predict(model, tibble(danceability = 27.5,
  valence = 30), type = "probs"), 2)
```

## 6 Ordinal Logistic Regression

```
college_data$decision <- as.ordered(college_data$decision)
college_data$decision <- fct_relevel(college_data$decision,
  c("unlikely", "somewhat likely", "very likely"))
levels(college_data$decision)
```

$P(Y_i \leq 4 | X_{i,1}) = \frac{\exp(\beta_0^{(4)} - \beta_1 X_{i,1})}{1 + \exp(\beta_0^{(4)} - \beta_1 X_{i,1})}$ . The categories 1,2,3,4,5 in the i-th response  $Y_i$  implicate an ordinal scale. Hence, our Ordinal Logistic regression model will indicate how each one of the 4 regressors that affects the cumulative logarithm of the odds in the ordinal response. we would need four link functions, four intercepts, and four coefficients.

```
ordinal_model <- polr(decision ~ parent_ed + GPA, data = data, Hess
  = TRUE)
library(broom)
summary_ordinal_model <- cbind(tidy(ordinal_model), p.value =
  pvalue_schid(ordinal_model)$statistic, lower.tail = FALSE)
```

```
* 2) |> mutate_if(is.numeric, round, 2)
round(confint(ordinal_model), 2)
tibble(summary_ordinal_model[1:2, 1:2], exp.estimate =
  round(exp(summary_ordinal_model[1:2, 2]), 2))
```

**Inference:**  $\beta_1$  : "for each one-unit increase in the GPA, the odds that the student is very likely versus somewhat likely or unlikely to apply to graduate school increase by  $\_\_\_$  times (while holding parent\_ed constant)."  $\beta_2$  : "for those respondents whose parents attended to graduate school, the odds that the student is very likely versus somewhat likely or unlikely to apply to graduate school increase by  $\exp(\beta_2) = 2.86$  times.

## 6.1 Brant-Wald Test

OLM under the proportional odds assumption is the first step when performing Regression Analysis on an ordinal response. Is it possible to assess whether it fulfils this strong assumption statistically.

```
library(brant) brant(ordinal_model)
```

row Omnibus represents the global model. Note that with  $\alpha = 0.05$ , we are completely fulfilling the proportional odds assumption. Suppose that does not fulfil the proportional odds assumption, we can model under a non-proportional odds assumption. This is called a Generalized OLR model.

## 7 Linear Mixed-effects Models

A panel refers to a dataset in which each individual (e.g., a firm) is observed within a timeframe. Furthermore, the term balanced indicates that we have the same number of observations per individual. n rows in our training set will not be independent anymore.

### 7.1 OLS Regression with Varying Intercept

We will do this with the lm() function by adding -1 on the right-hand side of the argument formula. This -1 will allow the baseline firm to have its intercept (i.e., replacing the usual (Intercept) in column estimate with this specific baseline company's intercept). In this case, General Motors is the baseline company (as it appears on the left-hand side of the levels() output). Note that (1 — firm) allows the model to have a random intercept by firm. Note that (market\_value + capital — firm) allows the model to have a random intercept and slopes by firm. Restricted maximum likelihood is an estimation approach for Mixed-effects regression. 1. Estimate the variance components. 2. Estimate the fixed-effects given those variance estimates

```
model_varying_intercept <- lm(formula = investment ~ market_value +
  capital + firm - 1, data = Grunfeld)
tidy(model_varying_intercept) glance(model_varying_intercept)
```

By checking the adj.r.squared, we see that model\_varying\_intercept has a larger value (0.959) than ordinary\_model (0.816) (i.e., the first fitted model without firm as a regressor).

```
investment.i.i = \beta_{0,i} + \beta_1 marketValue_{i,i} + \beta_2 capital_{i,i} + \varepsilon_{i,i}
anova(ordinary_model, model_varying_intercept)
```

With  $pvalue \leq \alpha = 0.05$ , we have evidence to conclude that model\_varying\_intercept fits the data better than the ordinary\_model. However, this costs us one extra degree of freedom per firm except for the baseline. Lose 10 degrees of freedom (DF in the anova()).

### 7.2 OLS Regression for Each Category

We can make the model more complex with two interactions. This will estimate a linear regression by firm with its own slopes.

```
model_by_firm <- lm(investment ~ market_value * firm + capital *
  firm, data = Grunfeld)
tidy(model_by_firm) glance(model_by_firm)
```

```
investment.i.j = \beta_{0,j} + \beta_{1,j} marketValue_{i,j} + \beta_{2,j} capital_{i,j} + \varepsilon_{i,j}
```

for  $i = 1, \dots, 20$  and  $j = 1, \dots, 11$ .

**Inference:** Each regression coefficient is associated with a firm. For example, firmUS Steel:capital = 0.02 means that the variable capital has a slope of  $0.02 + 0.37 = 0.39$  for US Steel.

```
tidy(lm(investment ~ market_value + capital,
  data = Grunfeld) |> filter(firm == "US Steel")
  ) |> mutate_if(is.numeric, round, 2) |> print(n = Inf)
```

### 7.3 Generic Covariance Matrix D

This is a standard form in linear Mixed-effects modelling.

### 7.4 Full Mixed Models

```
full_mixed_model <- lmer( investment ~ market_value + capital +
  (market_value + capital | firm), data = Grunfeld)
library(lmerTest) summary(mixed_intercept_model)
coef(mixed_intercept_model)$firm
round(predict(full_mixed_model, newdata = tibble(firm = "General
  Motors", market_value = 2000, capital = 1000)), 2)
```