1 Ordinary Least-squares Regression

1.1 Assumptions

Response=Systematic Component+Random Component. .ϵ is the Random Component under the following assumptions: .each Yi is also assumed to be independent and normally distributed .qq plot lying on the 45° degree dotted line. Standard Normal distribution .Histogram of residuals bell-shaped form as in the Normal distribution. .Homoscedasticity can be assessed via the diagnostic plot of residuals vs. fitted values. Funnel shapes indicates non-constant variance, i.e., heteroscedasticity. .This assumption commonly gets violated in multiple linear regression and is called heteroscedasticity: the variance of the ϵ_is is not constant. .OLS not suffice - Non-negative values. and Binary outcomes (Success or Failure). and Count data.

1.2 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) μ_i subject to k regressors $X_{i,j}$. modelling the mean μ_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.

2 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.

2.1 Estimation

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

```
library(glmbb)
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)
```

2.2 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.

2.3 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.

2.4 Predictions

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

3 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. Consequence of using these underestimated standard errors compared to the ones from negative_binomial_model are that more prone to committing Type I errors in our hypothesis testing, which are false positives (we would incorrectly conclude that there is a statistically significant association/causation between the response and regressors: rejecting H_0 when in fact it is true).

4 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

4.1 Estimation

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)
```

5 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 want to determine which Poisson regression model fits the data better: Model 1 or Model 2. We can compare the fits provided by these two models by the deviance. $D_k = kmodel/fullModel$ is formally called residual deviance, which is the test statistic. Large D_k values our given model fits the data poorly compared to the baseline model. Small D_k values our given model provides a good fit to the data 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)
```

We obtain a p-value $\leq .001$, which gives statistical evidence to state that our poisson_model_2 is not correctly specified when compared to the saturated model.

5.1 Analysis of Deviance for Nested Models

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

H_0 : Model 1 fits the data better than Model 2

H_A : Model 2 fits the data better than Model 1. We obtain a p-value $\leq .05$, column Pr(Chi), which gives us evidence to reject H_0 with $\alpha = 0.05$. Hence, we do have evidence to conclude that poisson_model_2 fits the data better than poisson_model. Therefore, in the context of model selection, we would choose poisson_model_2, that also includes the color of the prosoma.

5.2 Akaike Information Criterion

One of the drawbacks of the analysis of deviance is that it only allows to test nested regression models. Fortunately, we have alternatives for model selection. The 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.

```
glance(poisson_model) |> mutate_if(is.numeric, round, 3)
glance(poisson_model_2) |> mutate_if(is.numeric, round, 3)
```

5.3 Bayesian Information Criterion

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.

```
glance(poisson_model) |> mutate_if(is.numeric, round, 3)
glance(poisson_model_2) |> mutate_if(is.numeric, round, 3)
```

6 Multinomial Logistic Regression

Categorical Type Responses - more than two classes in the categorical response. Recall that Binary Logistic regression's link function (the logarithm of the odds or logit function) restricts the corresponding probability of success to a range between 0 and 1 while relating it to the systematic component.

```
training <- dataset |> select(genre, danceability, valence) |>
     mutate(genre = as.factor(genre))
levels(spotify_training$genre)
bin_spotify_training <- spotify_training |>
  filter(genre %in% c("edm", "latin")) |>
  mutate(genre = droplevels(genre))
spotify_bin_log_model <- glm(formula = genre ~ danceability +
     valence, data = bin_spotify_training, family = binomial) #
     with genre labels
library(broom)
tidy(spotify_bin_log_model, conf.int = TRUE, exponentiate = TRUE) |>
  mutate_if(is.numeric, round, 2)
```

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.

The Multinomial Logistic regression also models the logarithm of the odds. However, only one logarithm of the odds (or logit) will not be enough anymore. Recall we can capture the odds between two categories with a single logit function. What about adding some other ones?

6.1 Estimation

The estimates are obtained through maximum likelihood, where we assume a Multinomial joint probability mass function of the n responses Y_i. Final output is converted those scores into class probabilities using softmax functions

```
spotify_mult_log_model <- multinom(formula = genre ~ danceability +
     valence, data = spotify_training)
```

6.2 Inference

Provided the sample size n is large enough, it has an approximately Standard Normal distribution under H_o.

```
model <- multinom(formula = genre ~ danceability + valence, data =
     spotify_training)
mult_output <- tidy(model, conf.int = TRUE, exponentiate = TRUE) |>
     mutate_if(is.numeric, round, 3) |> dplyr::filter(p.value <
     0.05)
A tibble: 5 x 8
  y.level term estimate std.error statistic p.value
  <chr> <chr> <dbl> <dbl> <dbl> <dbl>
1 latin valence 1.05 0.013 4.01 0
2 r&b (Intercept) 14.5 0.811 3.30 0.001
3 r&b danceability 0.963 0.013 -3.02 0.003
4 rock (Intercept) 27.7 1.04 3.20 0.001
5 rock danceability 0.918 0.019 -4.51 0
```

Since our baseline response is edm, we can conclude the following with $\alpha = 0.05$ on the Spotify platform: There is a statistical difference in danceability in edm versus r&b and rock. There is a statistical difference in valence in edm versus latin.

6.3 Coefficient Interpretation

Let us interpret those significant regression coefficients from column estimate $\beta_1^{(latin,edm)}$: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. $\beta_1^{(r\&b,edm)}$:for each unit increase in the danceability score in the Spotify catalogue, the odds for a song for being r&b decrease by $(1 - 0.963)x100\% = 3.7\%$ compared to edm.

7

7.1 Predictions

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

8 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)
```

The categories 1,2,3,4,5 in the i-th response Y_i implicate an ordinal scale here,1¡2¡3¡4¡5. 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.

8.1 Estimation

```
ordinal_model <- polr(decision ~ parent_ed + GPA, data =
     college_data, Hess = TRUE)
```

All parameters in the Ordinal Logistic regression model are also unknown. Therefore, model estimates are obtained through maximum likelihood, where we also assume a Multinomial joint probability mass function of the n responses Y_i

8.2 Inference

```
library(broom)
summary_ordinal_model <- cbind(tidy(ordinal_model), p.value =
     pnorm(abs(tidy(ordinal_model)$statistic), lower.tail = FALSE)
```

```
    * 2) |>
  mutate_if(is.numeric, round, 2)
round(confint(ordinal_model), 2)
```

8.3 Inference

By using the column exp.estimate, along with the model equations on the original scale of the cumulative odds, we interpret the two regression coefficients above by each odds as follows:

```
tibble(summary_ordinal_model[1:2, 1:2], exp.estimate =
      round(exp(summary_ordinal_model[1:2, 2]), 2))
```

β_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)." β_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(β_2)=2.86 times (when compared to those respondents whose parents did not attend to graduate school and holding GPA constant).

8.4 Predictions

```
round(predict(ordinal_model, tibble(GPA = 3.5, parent_ed = "Yes"),
      type = "probs"), 2)
```

8.5 Brant-Wald Test

It is essential to remember that the Ordinal Logistic model 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)
```

The row Omnibus represents the global model, while the other two rows correspond to our two regressors: parent_ed and GPA. Note that with $\alpha = 0.05$, we are completely fulfilling the proportional odds assumption (the column probability delivers the corresponding p-values). Suppose that our example case does not fulfil the proportional odds assumption according to the Brant Wald test. It is possible to have a model under a non-proportional odds assumption. This is called a Generalized Ordinal Logistic regression model. This class of models can be fit via the function cumulative() from package VGAM.

9 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.

9.1 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-had side of the levels() output).

```
model_varying_intercept <- lm(
  formula = investment ~ market_value + capital + firm - 1,
  data = Grunfeld)
tidy(model_varying_intercept) |>
  mutate_if(is.numeric, round, 4) |>
  print(n = Inf)
glance(model_varying_intercept) |>
  mutate_if(is.numeric, round, 4)
```

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). This indicates that a model with estimated intercepts by firm fits the data better than a model without taking firm into account (at least by looking at the metrics!). Going back to model_varying_intercept and ordinary_model, we can test if there is a gain in considering a varying intercept versus fixed intercept. Hence, we will make a formal F-test to check whether the model_varying_intercept fits the data better than the ordinary_model.

$$\text{investment}_{i,j} = \beta_{0,j} + \beta_{1,j}\text{marketValue}_{i,j} + \beta_{2,j}\text{capital}_{i,j} + \varepsilon_{i,j}$$

$$\text{for } i = 1, \ldots, 20 \text{ and } j = 1, \ldots, 11.$$

```
anova(ordinary_model, model_varying_intercept) |>
  mutate_if(is.numeric, round, 4)
```

We obtain a p-value ¡0.001. Thus, with $\alpha = 0.05$, 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. Therefore, we lose another 10 degrees of freedom (column DF in the anova() output).

9.2 OLS Regression for Each Category

We can make the model more complex with two interactions (market_value * firm and capital * firm). 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) |>
  mutate_if(is.numeric, round, 2) |>
  print(n = Inf)
glance(model_by_firm) |>
  mutate_if(is.numeric, round, 2)
```

$$\text{investment}_{i,j} = \beta_{0,j} + \beta_{1,j}\text{marketValue}_{i,j} + \beta_{2,j}\text{capital}_{i,j} + \varepsilon_{i,j}$$

$$\text{for } i = 1, \ldots, 20 \text{ and } j = 1, \ldots, 11.$$

9.3 Interpret Coeffients

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`. We can double-check this by estimating an individual linear regression for `US Steel`:

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

9.4 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)
```