Analysis of Layoff Data

Alex Kazachek 251063057
Ningsong Shen 251080589
Esther Yue 251082339

19 April 2023

1 Introduction

In recent years, the global technology industry has seen unprecedented growth as digitization and the Internet has fuelled demand for technology products and services by consumers and corporations alike. The industry further flourished through the COVID-19 pandemic as organizations across every industry were forced to accelerate their digital transformation initiatives, leading to record-breaking profits at many of the largest technology companies such as Microsoft, Google, and Amazon. Subsequently, headcount at major tech companies expanded substantially from 2019 to 2022; both Amazon and Meta doubled their corporate staff during this time period, while other companies such as Microsoft, Salesforce, and Snap also hired thousands of additional workers in order to meet this surging demand.

However, the current global economy is seeing higher interest rates, rising inflation, and the looming threat of a recession. These factors have caused the sector to decline considerably since the latter half of 2022 as softening consumer spending, diminishing demand for tech products and services, and the uncertain macroeconomic environment has dampened profit growth of tech companies. As these companies seek ways to remain competitive in a highly uncertain environment, mass layoffs have become an unfortunate reality for many employees as technology firms of all sizes seek to trim costs and increase efficiency in response to slowing industry growth. According to layoffs.fyi, more than 170,000 employees from nearly 600 tech companies have been laid off since the beginning of 2023.

The impacts layoffs can have can extend beyond immediate job loss for individuals. Mass layoffs can negatively impact the job market, as future workers may avoid seeking employment at companies that have laid off employees in the past. Companies that have previously gone through mass layoffs may also find it increasingly challenging to attract and retain talent. In today's economic environment, it is becoming increasingly critical to be able to predict when layoffs may occur, as being able to identify the signs that contribute to making a company more layoff-prone can provide them with the opportunity to take proactive measures to avoid or minimize mass layoffs, such as implementing alternative cost-cutting measures or diversifying revenue streams. Employees that can identify factors that potentially indicate layoffs in their company can better prepare for potential job losses, such as seeking other employment opportunities.

In this research report, we aim to analyze trends in layoffs using beta regression on a comprehensive dataset of layoffs during and after the COVID-19 pandemic. Overall, we hope to gain insights into the factors that contribute to the occurrence of layoffs so that

businesses and employees can better understand and prepare for the impact of mass layoffs in the future.

2 Data Set Description

We used the Layoffs Dataset sourced from Kaggle [Swa23]. The original data was compiled by Roger Lee from the Layoffs.fyi Tracker, which is a database tracking layoffs in the technology industry and those closely related to it [Lee23]. The database is a personal project of his and is based primarily on public news reports from Bloomberg, TechCrunch, the New York Times, or the San Francisco Business Times. Other tips and employee self-reporting supplement these primary sources. Although this does not represent a comprehensive picture of the layoff landscape, it is about the best data that we can get for this topic, and should be a fairly reliable and representative sample of activity at the largest technology companies. Despite the unofficial procurement of the data, it is considered an invaluable source by major news outlets such as Bloomberg and the New York Times, with accuracy and detail rivalling reports by governmental agencies [Duf23].

The dataset contains 2414 entries, each representing a layoff at a technology company (like Twitter or AirBnB, companies where technology is the primary focus) or a technology-adjacent community (like CNET, a technology news reporter). To be included in the dataset, the layoff must have occurred on or after March 11, 2020, the day that the World Health Organization (WHO) officially declared COVID-19 a global pandemic. The dataset that we used is updated as of March 21, 2023. There are 9 attributes for each layoff, which form the columns of the dataset, and they are described in TABLE 1.

Table 1: Description of the Dataset

Column	Data Type	Description	Example
company	String	Company name	Wealthsimple
date	Date	Layoff date	2023-03-17
country	String	Country of incorporation	Canada
location	String	Headquarters location	SF Bay Area
industry	String	Target industry	Aerospace
total_laid_off	Integer	Number of employees laid off	451
percentage_laid_off	Float	Percentage of employees laid off	0.06
stage	String	Stage of funding	Post-IPO
funds_raised	Integer	Funds raised (in millions of US$)	2200

The conjunction of the two variables `date` and `company` serves as a unique identifier (note the latter alone is not enough, as in principle a company may have many layoffs at different points in time). We have four factor variables (`country`, `location`, `industry`, `stage`) and three continuous variables (`total_laid_off`, `percentage_laid_off`, `funds_raised`). Note that `date` could serve as a continuous variable too, though for reasons we will discuss in the methodology section we believe this to be unwise. The most recent entry in the data is provided as an example in TABLE 2.

As the example shows, there are missing values in the dataset. This could be due to a variety of reasons, but most likely due to a lack of publicly-available information. For many companies, layoffs and funding are sensitive topics that may be kept internal. The only method then would be through self-reporting, which is much more difficult to obtain and certainly less reliable. There are 221 entries missing in `funds_raised`, 761 entries missing in `total_laid_off`, and 809 entries missing in `percentage_laid_off`. Though we may drop these entries and still have a large dataset, we should be mindful of introducing a form of non-response bias (for instance, companies may not want to disclose their financials if they are weak – and these are the companies most sensitive to layoffs).

Table 2: Latest Entry in the Dataset

company	date	country	location	industry
Just Eat	2023-03-21	United Kingdom	London	Food

total_laid_off	percentage_laid_off	stage	funds
1700	–	Acquired	–

We aim to predict the percentage of employees laid off (i.e. `percentage_laid_off`) using a combination of the other columns in the dataset. This will be the response variable, and it is distributed as seen in FIGURE 1. An important observation is that the distribution is bimodal – with many values near 0 and 1. This is the reason we will use beta regression, which we will expand upon in the following section.

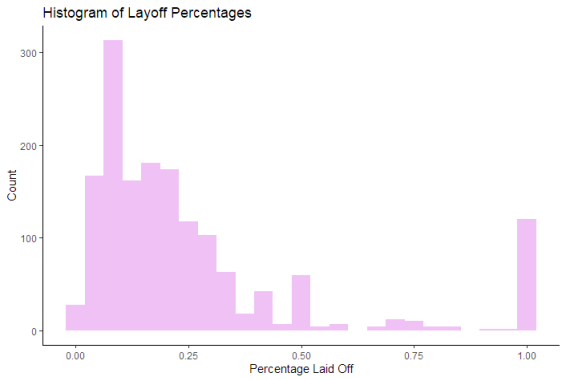

Figure 1: Histogram of `percentage_laid_off` (missing values dropped)

3 Methodology

Beta regression, originally introduced in [FC04], is designed for regression on a response lying in the unit interval $(0, 1)$. The underlying assumption is that the response Y is

distributed according to a beta distribution $B(\alpha, \beta)$, having density

$$f(y; \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} y^{\alpha-1}(1-y)^{\beta-1}$$

supported on $y \in (0, 1)$ and taking parameters $\alpha, \beta > 0$. Examples of some densities are given in FIGURE 2, in which we note the blue curve. This suggests a beta distribution is good at modelling proportions which are 0 and 1 skewed, which we recall is the case for `percentage_laid_off` from the histogram in FIGURE 1.

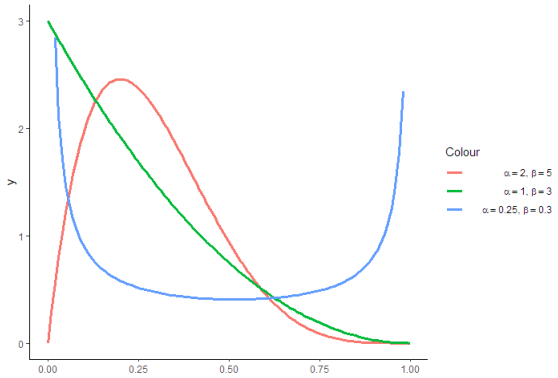

Figure 2: Example beta distribution densities

In the context of regression, the beta distribution is often reparametrized with $\mu = \alpha/(\alpha + \beta)$ and $\varphi = \alpha + \beta$ so that

$$f(y; \alpha, \beta) = f(y; \mu, \varphi) = \frac{\Gamma(\varphi)}{\Gamma(\mu\varphi)\Gamma((1-\mu)\varphi)} y^{\mu\varphi-1}(1-y)^{(1-\mu)\varphi-1}.$$

Then, $\mathbb{E}(Y) = \mu$ and $\operatorname{var}(Y) = \mu(1-\mu)/(1+\varphi)$. In this sense, μ controls the mean of the response and φ controls the variance. Specifically, as φ grows the variance vanishes, and thus it is called the precision parameter. It is similar to a dispersion parameter in Poisson or Binomial regression. Also note $\mu \in (0, 1)$.

Under this parametrization, say we have samples y_i where $y_i \sim B(\mu_i, \varphi_i)$. Let us define beta regression on k covariates $x_i = (x_{1i}, \ldots, x_{ki})$ for the mean and ℓ covariates $z = (z_{1i}, \ldots, z_{\ell i})$ for precision. Here, we aim to estimate k parameters $\beta = (\beta_1, \ldots, \beta_k)$ and ℓ parameters $\eta = (\eta_1, \ldots, \eta_\ell)$ such that $g(\mu_i) = x_i^\top \beta$ and $h(\varphi_i) = z_i^\top \eta$, where $g \colon (0, 1) \to \mathbb{R}$ and $h \colon \mathbb{R}^+ \to \mathbb{R}$ are appropriate link functions.

There are possible choices for g. A common choice is the logit link, just as in binomial regression. However, some non-standard links such as the log-log $g(\mu) = \log(-\log(1-\mu))$ or the Cauchit $g(\mu) = \tan(\pi(\mu - 1/2))$ may also be used, and in fact may be more appropriate depending on the exact distribution of the response. For instance, the Cauchit often gives better results in the presence of extreme observations if less exotic link functions fail to give good predictions [KY09]. Exploring these link functions is important in our case as although our data is bimodel, there are far more observations near 0 than there are near 1 which could lead to asymmetric predictions. For h, the standard choice is the logarithm $h(\varphi) = \log \varphi$, though in principle other choices could also be considered.

We will use a combination of the approaches presented in [CZ10] to train and evaluate beta regression models, as well as borrow ideas from statistical learning theory. Nested models may be evaluated using likelihood ratio tests. Non-nested models may be evaluated using a combination of minimizing AIC and maximizing pseudo-R^2. We will also partition our data into training, validation, and testing sets using a $64 - 16 - 20$ split. This allows us to compare the sum of squared residual errors on the validation set as another way to compare non-nested models. The testing set will then be used to report the final conclusion.

There are several types of residuals which may be defined for beta regression models, however we will use the Pearson residuals

$$r = \frac{y - \hat{y}}{\sqrt{\hat{\sigma}(y)}}$$

where y is the true response, $\hat{y}$ is the predicted response, and

$$\hat{\sigma}(y) = \frac{\hat{\mu}(1 - \hat{\mu})}{1 + \hat{\varphi}}$$

is the estimated variance. Specifically, $\hat{\mu} = g^{-1}(x^\top \hat{\beta})$ and $\hat{\varphi} = h^{-1}(z^\top \hat{\eta})$. We opt for this residual as deviance residuals are often numerically unstable and even negative, as noted in [EFC08]. This work defines an alternative residual as well, the weighted residual, however we find it also lacks numerical stability in our specific case.

To summarize, our statistical analysis will be performed as follows:

1. Conduct an exploratory data analysis to find an appropriate subset of the data.
2. Perform a training-validation-testing $64 - 16 - 20$ split, and treat the data in the process (e.g. drop missing values).
3. Fit a simple beta regression model using linear predictors, and use likelihood ratio tests to determine if adding any predictors for the precision parameter improves fit.
4. Take the best model from the previous step, and use a combination of AIC, pseudo-R^2, and validation set performance to find the best choice of link function.
5. Perform likelihood ratio tests once more when adding interaction terms to the best model from the previous step.
6. Evaluate model on test set and report results.

Let us also note that it is not appropriate for us to use `date` as a covariate. This is as layoffs are correlated to overall economic stability, which is cyclic. To properly handle it as a predictor, we would need to discuss autoregressive models. This is possible in the context of beta regression (e.g. see [GV14]) though is outside the scope of this course.

It remains to discuss how we will perform the first step – the exploratory analysis. For this, we will start by examining the counts of the various factors. For instance, although our data is sourced internationally there may be some countries with very few responses. It is important we strike a balance between including enough countries to have a sufficiently

large data set, while excluding countries with so few observations that we would introduce outliers in our data.

We should also look at histograms or violin plots of `funds_raised`, our one numerical factor, against the various factors. If we find drastic difference, this may suggest possible interaction terms to consider. For factor-factor interactions, we will look at mosaic plots.

4 Results

4.1 Exploratory Analysis

We will first examine the counts of the two factors `country` and `location` given in FIGURE 3. For both of these, it seems that taking those with log-counts greater than roughly 2 leaves sufficient data for training. However, `location` entirely determines `country`, so to avoid collinear features we should train models on both and see which performs the best. For `industry` we examine FIGURE 4. The only non-null factors with noticeably small representation are the bottom two – aerospace and manufacturing – which we opt to exclude.

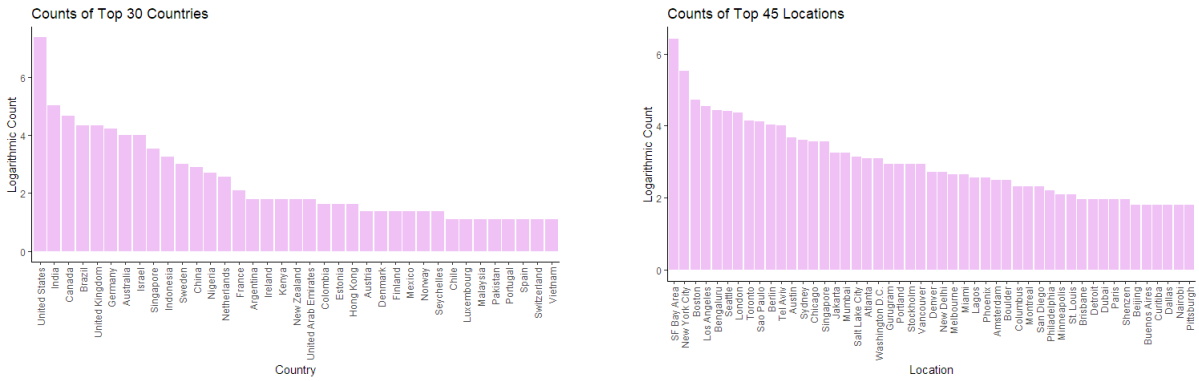

Figure 3: Counts of factors `country` and `location`

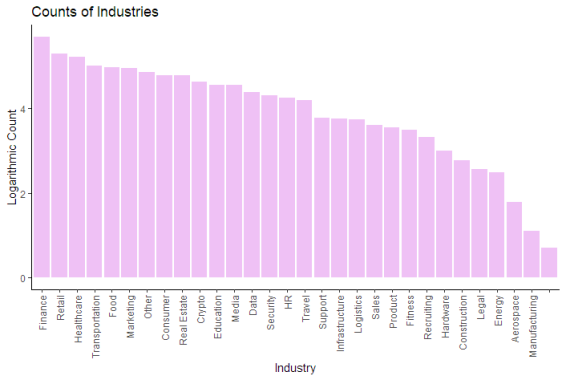

Figure 4: Counts of factor `industry`

Next we will look for interactions between `industry` with `country` and `location`. For

clarity, we only use the top eight of each factor in the mosaic plots in FIGURE 5. There is clear qualitative evidence of interaction, with few horizontal lines present. For instance, Boston has essentially a non-existent financial industry, while it makes up roughly a quarter of the industries in Toronto. However, we note the interaction appears less extreme with `country`. These are effects we should certainly follow-up on statistically.

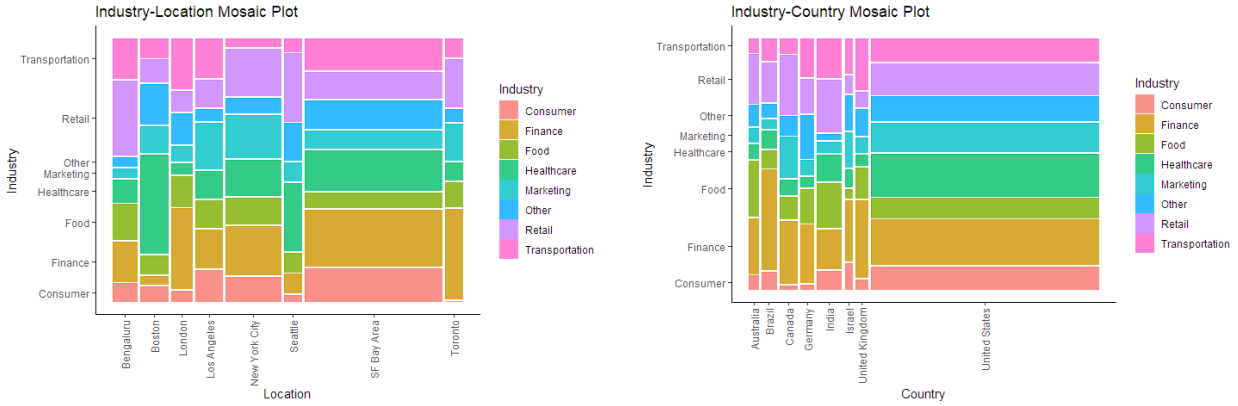

Figure 5: Mosaic plots between `industry` and `country` or `location`

We visualize `funds_raised` in FIGURE 6, where we truncate it at 2000. Note though that the maximum value is 121,900, but as the box plot indicates this is an extreme outlier. Dealing with a distribution this skewed can pose problems for regression.

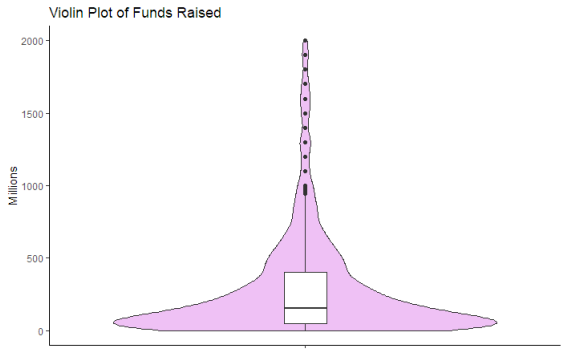

Figure 6: Truncated Violin Plot of `funds_raised`

4.2 Statistical Analysis

Our analysis starts by identifying which subset of data we should use. Through the exploratory analysis we concluded that we should keep factors with log-counts larger than 2. However, we must still decide between `location` or `country` as a predictor. For this reason, we will train two models – one for each – and compare the end models on their respective validation sets. We believe this is best as we introduce interaction terms at the end of the model construction, and our exploratory analysis showed us the degree

of interaction that `location` and `country` have with other factors differs. Also note their validation sets are indeed different, as are the training and testing sets. For example, although the United States is included as a country, not all locations within the United States may have enough observations to meet the cutoff of a log-count greater than 2. The code we use to produce these sets is given in LISTING 1. For brevity, the numerical figures we will report in the process of model construction will only be for the country model.

We fit a model using a standard logit link function, a constant precision parameter, and only linear predictors. We then drop outliers from the training data by finding those with a Cook's distance larger than $4/n$, where $n = 789$ is the size of training set, and refit the model. This model has a pseudo-R^2 of 0.2894, a validation error of 236.249, and an AIC of -442.60, as per LISTING 2. To incorporate precision, we attempt to use individual linear predictors. Unfortunately, numerical instability causes training to fail for all factors except `funds_raised`. However, in this case the likelihood ratio between this model and the constant precision model shows an improvement in fit with strong statistical significance (that is, with the null being no improvement, we reject the null with $p = 9.57 \cdot 10^{-5}$). This test is conducted in LISTING 3.

The next step is to refine the link function. With the code in LISTING 4, we summarize the various evaluation metrics in TABLE 3. Balancing the three metrics, we believe it is best to keep the logit. Though other links outperform it in some aspects, they all fail in some other way. For instance, the complementary-loglog has a better AIC and pseudo-R^2, but much worse validation error. The logit is a solid middle ground.

Table 3: Evaluation of Various Link Functions

Link	AIC	Pseudo-R^2	Validation Error
Logit	-456.31	0.285	238.33
Loglog	-451.07	0.255	235.07
Complementary-loglog	-459.73	0.293	245.40
Probit	-454.99	0.289	237.64
Cauchit	-461.93	0.169	248.30

In the final stage we search for interactions in LISTING 5. Again, numerical issues occur when searching for interactions between factors. However, the two-way interactions between `funds_raised` and the three factors all train successfully. Likelihood tests when interacting with `location` or `industry` exhibit very high p-values (respectively 0.56 and 0.86), however with `stage` we have $p = 0.08$. This is mild evidence of statistical significance, and so we opt to examine the other evaluation metrics. We find an increase in pseudo-R^2 to 0.309 with a similar validation error of 242.80, and so we opt to keep the interaction term in the model.

Our final location model uses a logit link with `funds_raised` to estimate precision, and each predictor along with an interaction between `funds_raised` and `stage`. We perform a similar analysis for the country model, and end up with an identical model except for a complementary-loglog link. We compare the two models in TABLE 4, and note that we use averaged validation error as the two models have validation sets of different sizes.

Although the country model has a higher validation error, its pseudo-R^2 is so much higher and AIC so much lower that we choose it to be the final model. On the testing set for the country model, we plot the true versus predicted values for both the link and response in FIGURE 7 and residuals in FIGURE 8.

Table 4: Comparison of Country and Location Models

Model	AIC	Pseudo-R^2	Average Validation Error
Country	-590.76	0.354	65.96
Location	-448.78	0.309	48.60

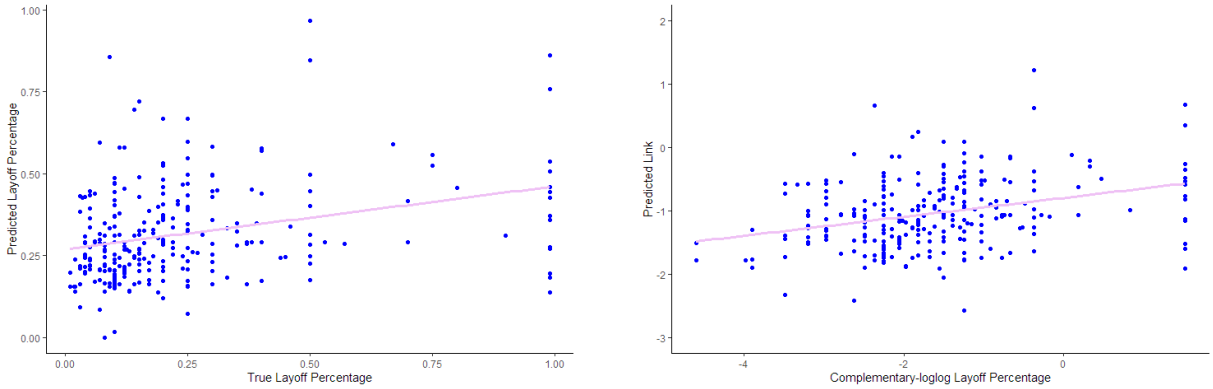

Figure 7: Response and link predictions with country model and a regression line

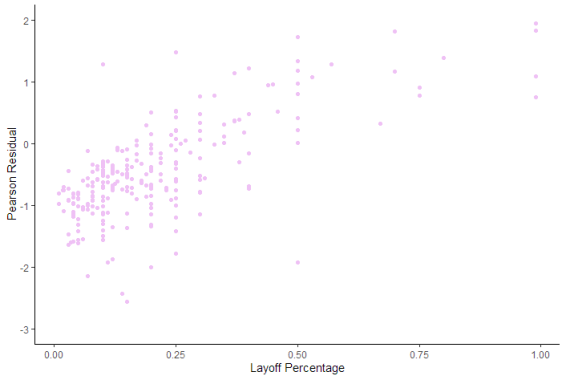

Figure 8: Residual plot for country model

5 Conclusion

Overall, the conclusions are less than satisfactory. Training was numerically challenging and model evaluation was not straightforward. The end model was not excellent, with a fairly weak linear relationship between the true and predicted values (for both the response

and link). We believe the two main reasons this analysis was not as fruitful as desired are the choice of model and dataset.

Our end dataset only contains a single continuous predictor. The other three are not only categorical, but with a large number of categories as well. The factor with the smallest number of dummy variables in the model is `country` at 11. This means our model quickly suffers from the curse of dimensionality, leading to overall poor fits. Moreover, it explains why we were unable to train models containing substantive interaction terms – any interaction between factors introduces hundreds of new variables to the model, resulting in singular matrices or diverging values in training.

Beta regression is also quite young, being about 15 years old, and as such lacks many standard practices. This makes things such as variable selection and model evaluation tricky. Data transformation is also an open question, which directly affected us since we had several observation with `percentage_laid_off` = 1. This is invalid in beta regression, and our naïve solution was to shrink them to 0.98. However, there are other techniques such as inflated beta regression models [OF12] or more nuanced transformations which take into account the sample size [Smi06]. These could have possibly yielded better results, as we saw in FIGURE 8 our model performs poorly for these high values.

References

[FC04] Silvia Ferrari and Francisco Cribari-Neto. "Beta Regression for Modelling Rates and Proportions". In: *Journal of Applied Statistics* 31 (7 2004), pp. 799–815.

[Smi06] J. Smithson M Verkuilen. "A Better Lemon Squeezer? Maximum-Likelihood Regression with Beta-Distributed Dependent Variables". In: *Psychological Methods* 11 (1 2006), pp. 54–71.

[EFC08] Patrícia Espinheira, Silvia Ferrari, and Francisco Cribari-Neto. "On beta regression residuals". In: *Journal of Applied Statistics* 35 (4 2008), pp. 407–419.

[KY09] Roger Koenker and Jungmo Yoon. "Parametric links for binary choice models: A Fisherian–Bayesian colloquy". In: *Journal of Econometrics* 152 (2 2009), pp. 120–130.

[CZ10] Francisco Cribari-Neto and Achim Zeileis. "Beta Regression in R". In: *Journal of Statistical Software* 34 (2 2010), pp. 1–24.

[OF12] Raydonal Ospina and Silvia Ferrari. "A general class of zero-or-one inflated beta regression models". In: *Computational Statistics & Data Analysis* 56 (6 2012), pp. 1609–1623.

[GV14] Annamaria Guolo and Cristiano Varin. "Beta regression for time series analysis of bounded data, with application to Canada Google Flu Trends". In: *Annals of Applied Statistics* 8 (1 2014), pp. 74–88.

[Duf23] Meg Duff. "Why One Random Dude Is Better at Tracking Tech Layoffs Than the Government". In: *Slate* (Jan. 2023). URL: `https://slate.com/technology/2023/01/tech-layoffs-how-many-fyi-data-tracking-kittens.html`.

[Lee23] Roger Lee. *Layoffs.fyi*. Mar. 2023. URL: `https://layoffs.fyi/`.

[Swa23] Swaptr. *Layoffs Dataset*. Mar. 2023. URL: `https://www.kaggle.com/datasets/swaptr/layoffs-2022`.

6 Appendix

Listing Number	Purpose
1	Constructing the two pairs of train-valid-test sets
2	Fitting basic linear models with no interactions
3	Nested likelihood testing for inclusion of a precision parameter
4	Numerical metric comparisons to pick a link function
5	Testing for significant interaction terms

Listing 1: Generating training-validation-testing sets

```
library(dplyr)
library(tidyr)

layoffs <- read.csv("layoffs.csv")
# shrink responses equal to 1
idx <- layoffs$percentage_laid_off > 0.98
idx <- idx * 0.01
layoffs$percentage_laid_off <- layoffs$percentage_laid_off - idx
layoffs<- layoffs[which(layoffs$percentage_laid_off > 0.001),]

# top counts

country.counts <- layoffs %>%
  group_by(country) %>%
  summarise(count = n()) %>%
  top_n(n = 30, wt = count)
country.counts <- country.counts[which(country.counts$count > exp(2)),]

industry.counts <- layoffs %>%
  group_by(industry) %>%
  summarise(count = n()) %>%
  top_n(n = 100, wt = count)
industry.counts <- industry.counts[which(industry.counts$count > exp(2)),]

location.counts <- layoffs %>%
  group_by(location) %>%
  summarise(count = n()) %>%
  top_n(n = 45, wt = count)
location.counts <- location.counts[which(location.counts$count > exp(2)),]

# country subselection

layoffs.reduced.ct <- layoffs[layoffs$industry %in% industry.counts$industry &
                              layoffs$country %in% country.counts$country,]
layoffs.reduced.ct <- subset(layoffs.reduced.ct, select=-c(company,
                                                           date,
                                                           total_laid_off,
                                                           location))
layoffs.reduced.ct <- layoffs.reduced.ct %>% drop_na()
```

```r


# location subselection

layoffs.reduced.loc <- layoffs[layoffs$industry %in% industry.counts$industry &
                               layoffs$location %in% location.counts$location,]
layoffs.reduced.loc <- subset(layoffs.reduced.loc, select=-c(company,
                                                             date,
                                                             total_laid_off,
                                                             country))
layoffs.reduced.loc <- layoffs.reduced.loc %>% drop_na()

# train-test split

layoffs.reduced.ct$id <- 1:nrow(layoffs.reduced.ct)
train.ct <- layoffs.reduced.ct %>% sample_frac(.8)
test.ct  <- anti_join(layoffs.reduced.ct, train.ct, by = 'id')
train.ct <- subset(train.ct, select=-c(id))
test.ct <- subset(test.ct, select=-c(id))

layoffs.reduced.loc$id <- 1:nrow(layoffs.reduced.loc)
train.loc <- layoffs.reduced.loc %>% sample_frac(.8)
test.loc  <- anti_join(layoffs.reduced.loc, train.loc, by = 'id')
train.loc <- subset(train.loc, select=-c(id))
test.loc <- subset(test.loc, select=-c(id))

# train-validation split

train.ct$id <- 1:nrow(train.ct)
valid.ct <- train.ct %>% sample_frac(.2)
train.ct  <- anti_join(train.ct, valid.ct, by = 'id')
valid.ct <- subset(valid.ct, select=-c(id))
train.ct <- subset(train.ct, select=-c(id))

train.loc$id <- 1:nrow(train.loc)
valid.loc <- train.loc %>% sample_frac(.2)
train.loc <- anti_join(train.loc, valid.loc, by = 'id')
valid.loc <- subset(valid.loc, select=-c(id))
train.loc <- subset(train.loc, select=-c(id))

# get rid of unseen factors during training

valid.ct <- valid.ct[valid.ct$stage %in% train.ct$stage,]
test.ct <- test.ct[test.ct$stage %in% train.ct$stage,]

valid.loc <- valid.ct[valid.loc$stage %in% train.loc$stage,]
test.loc <- test.ct[test.loc$stage %in% train.loc$stage,]
```

Listing 2: Training basic linear models

```r
library(betareg)
mod.loc <- betareg(percentage_laid_off ~
                     location + industry + stage + funds_raised,
```

```
                    train.loc)
mod.ct <- betareg(percentage_laid_off ~
                     country + industry + stage + funds_raised,
                   train.ct)

# remove outliers and refit

train.loc <- train.loc[-which(cooks.distance(mod.loc) > 4 / length(train.loc)),]
train.ct <- train.ct[-which(cooks.distance(mod.ct) > 4 / length(train.ct)),]

mod.loc <- betareg(percentage_laid_off ~
                      location + industry + stage + funds_raised,
                    train.loc)
mod.ct <- betareg(percentage_laid_off ~
                     country + industry + stage + funds_raised,
                   train.ct)

summary(mod.loc, type="pearson")
summary(mod.ct, type="pearson")

# compute Pearson residuals on a passed dataset

pearson <- function(mod, valid) {
  phi <- predict(mod, valid, type="precision")
  pred <- predict(mod, valid, type="response")
  var <- pred*(1-pred)/(1+phi)
  (valid$percentage_laid_off - pred) / sqrt(var)
}

sum(pearson(mod.loc, valid.loc)^2)
sum(pearson(mod.ct, valid.ct)^2)
AIC(mod.loc, mod.ct)
```

Listing 3: Inclusion of a precision term

```
library(lmtest)

# fit single predictors for precision
mod.loc.f <- betareg(percentage_laid_off ~
                       location + industry + stage + funds_raised |
                       funds_raised,
                     train.loc)
mod.loc.l <- betareg(percentage_laid_off ~
                       location + industry + stage + funds_raised |
                       location,
                     train.loc)
mod.loc.i <- betareg(percentage_laid_off ~
                       location + industry + stage + funds_raised |
                       industry,
                     train.loc)
mod.loc.s <- betareg(percentage_laid_off ~
                       location + industry + stage + funds_raised |
                       stage,
                     train.loc)
```

```

# all fail to fit except for funds_raised
# conduct likelihood test
lrtest(mod.loc, mod.loc.f)

# same for country
mod.ct.f <- betareg(percentage_laid_off ~
                      country + industry + stage + funds_raised |
                      funds_raised,
                    train.ct)
mod.ct.c <- betareg(percentage_laid_off ~
                      country + industry + stage + funds_raised |
                      country,
                    train.ct)
mod.ct.i <- betareg(percentage_laid_off ~
                      country + industry + stage + funds_raised |
                      industry,
                    train.ct)
mod.ct.s <- betareg(percentage_laid_off ~
                      country + industry + stage + funds_raised |
                      stage,
                    train.ct)

lrtest(mod.ct, mod.ct.f)
```

Listing 4: Evaluating the choice of link function

```
# fit same model but vary the link
mod.loc.ll <- betareg(percentage_laid_off ~
                        location + industry + stage + funds_raised |
                        funds_raised,
                      train.loc,
                      link="loglog")
mod.loc.cll <- betareg(percentage_laid_off ~
                        location + industry + stage + funds_raised |
                        funds_raised,
                      train.loc,
                      link="cloglog")
mod.loc.pr <- betareg(percentage_laid_off ~
                        location + industry + stage + funds_raised |
                        funds_raised,
                      train.loc,
                      link="probit")
mod.loc.ca <- betareg(percentage_laid_off ~
                        location + industry + stage + funds_raised |
                        funds_raised,
                      train.loc,
                      link="cauchit")

# validation errors
sum(pearson(mod.loc.f, valid.loc)^2)
sum(pearson(mod.loc.ll, valid.loc)^2)
sum(pearson(mod.loc.cll, valid.loc)^2)
sum(pearson(mod.loc.pr, valid.loc)^2)
```

```
sum(pearson(mod.loc.ca, valid.loc)^2)

AIC(mod.loc.f, mod.loc.ll, mod.loc.cll, mod.loc.pr, mod.loc.ca)

mod.loc.f$pseudo.r.squared
mod.loc.ll$pseudo.r.squared
mod.loc.cll$pseudo.r.squared
mod.loc.pr$pseudo.r.squared
mod.loc.ca$pseudo.r.squared

# same for country
mod.ct.ll <- betareg(percentage_laid_off ~
                       country + industry + stage + funds_raised |
                        funds_raised,
                      train.ct,
                      link="loglog")
mod.ct.cll <- betareg(percentage_laid_off ~
                        country + industry + stage + funds_raised |
                         funds_raised,
                       train.ct,
                       link="cloglog")
mod.ct.pr <- betareg(percentage_laid_off ~
                       country + industry + stage + funds_raised |
                       funds_raised,
                      train.ct,
                      link="probit")
mod.ct.ca <- betareg(percentage_laid_off ~
                       country + industry + stage + funds_raised |
                        funds_raised,
                      train.ct,
                      link="cauchit")

sum(pearson(mod.ct.f, valid.ct)^2)
sum(pearson(mod.ct.ll, valid.ct)^2)
sum(pearson(mod.ct.cll, valid.ct)^2)
sum(pearson(mod.ct.pr, valid.ct)^2)
sum(pearson(mod.ct.ca, valid.ct)^2)

AIC(mod.ct.f, mod.ct.ll, mod.ct.cll, mod.ct.pr, mod.ct.ca)

mod.ct.f$pseudo.r.squared
mod.ct.ll$pseudo.r.squared
mod.ct.cll$pseudo.r.squared
mod.ct.pr$pseudo.r.squared
mod.ct.ca$pseudo.r.squared
```

Listing 5: Testing for significant interaction terms

```
# try basic interactions, note that factor-factor interactions
# will fail to fit each time
mod.loc.lf <-  betareg(percentage_laid_off ~
                         location*funds_raised + industry + stage |
                         funds_raised,
                       train.loc)
```

```
mod.loc.if <-  betareg(percentage_laid_off ~
                         location + industry*funds_raised + stage |
                         funds_raised,
                       train.loc)
mod.loc.sf <-  betareg(percentage_laid_off ~
                         location + industry + stage*funds_raised |
                         funds_raised,
                       train.loc)

# likelihood tests
lrtest(mod.loc.f, mod.loc.lf)
lrtest(mod.loc.f, mod.loc.if)
lrtest(mod.loc.f, mod.loc.sf)

# stage interaction is weakly significant, conduct
# further tests
sum(pearson(mod.loc.sf, valid.loc)^2)
mod.loc.sf$pseudo.r.squared

# same for country
mod.ct.lf <-  betareg(percentage_laid_off ~
                         country*funds_raised + industry + stage |
                         funds_raised,
                       train.ct,
                      link="cloglog")
mod.ct.if <-  betareg(percentage_laid_off ~
                         country + industry*funds_raised + stage |
                         funds_raised,
                       train.ct,
                      link="cloglog")
mod.ct.sf <-  betareg(percentage_laid_off ~
                         country + industry + stage*funds_raised |
                         funds_raised,
                       train.ct,
                      link="cloglog")

lrtest(mod.ct.cll, mod.ct.lf)
lrtest(mod.ct.cll, mod.ct.if)
lrtest(mod.ct.cll, mod.ct.sf)

# both stage-funds and industry-funds are significant, so
# test their sum and see if it's any better
mod.ct.isf <-  betareg(percentage_laid_off ~
                         country + industry*funds_raised + stage*funds_raised |
                         funds_raised,
                       train.ct,
                      link="cloglog")
lrtest(mod.ct.sf, mod.ct.isf)
# no significance, check validation error just in case
sum(pearson(mod.ct.sf, valid.ct)^2)
sum(pearson(mod.ct.isf, valid.ct)^2)
```