STAT 425 Final Project: Tech Layoffs

Sheetal Tewary

1 Summary and Motivation

This project is a **data-focused statistical analysis** of technology sector layoffs using a publicly available dataset compiled from company disclosures and news reports. Layoffs represent a critical workforce risk event, and understanding both *when* layoffs are likely to occur and *how severe* they may be is important for economic analysis and organizational planning.

The primary goal of this project is to apply methods from STAT 425 to answer two practical questions:

1. What factors are associated with the **probability** that a company experiences a layoff in a given month?
2. Conditional on a layoff occurring, what factors influence the **magnitude** of the layoff?

To address these questions, I use logistic regression, penalized regression (LASSO and ridge), linear regression, spline-based time effects, and model diagnostics—tools covered in this course.

2 Data Description

The dataset (`layoffs.csv`) consists of reported technology layoff events from 2020–2024. Each observation corresponds to a layoff event at a specific company and date. Variables used include company name, date, number and percentage of employees laid off, industry, funding stage, country, and total funding raised.

Dates are converted to a monthly time scale to construct a company–month panel. Funding raised is log-transformed and winsorized at the 1% level to reduce the influence of extreme values.

Missing data are common, particularly for the number of employees laid off. For the probability analysis, a layoff event is defined as occurring if either the number or percentage laid off is reported. For the magnitude analysis, only observations with non-missing layoff counts are used to avoid unreliable imputation.

3 Methods

3.1 Probability of Layoff (Binary Outcome)

A company–month panel is constructed, and a binary response indicates whether a layoff occurred in that month. Logistic regression is used to model layoff probability as a function of funding, industry, stage, country, and a nonlinear time trend captured using natural splines.

Because the model includes many categorical predictors and sparse levels, a LASSO-penalized logistic regression is also fitted using cross-validation. The LASSO shrinks weak coefficients toward zero, improving stability and reducing overfitting.

Model performance is evaluated using AUC, confusion matrices, calibration plots, and influence diagnostics.

3.2 Magnitude of Layoffs (Continuous Outcome)

For observations with reported layoff counts, severity is modeled using $\log(\text{total laid off} + 1)$. to reduce skewness and limit the influence of extreme events. A baseline OLS regression is fitted with the same covariates and time splines.

Ridge and LASSO regression are then used to assess whether regularization improves predictive accuracy. Models are compared using RMSE and MAE on held-out test data. Residual diagnostics are used to assess model assumptions.

4 Analysis and Results

```
# ============================================================
# Setup
# ============================================================
library(tidyverse)
library(lubridate)
library(splines)
library(glmnet)
library(pROC)
library(caret)
library(car)

set.seed(425)

winsorize <- function(x, p = 0.01) {
  q <- quantile(x, probs = c(p, 1 - p), na.rm = TRUE)
  pmin(pmax(x, q[1]), q[2])
}

rmse <- function(a, p) sqrt(mean((a - p)^2, na.rm = TRUE))
mae  <- function(a, p) mean(abs(a - p), na.rm = TRUE)
```

4.1 Load and clean data

```
df <- readr::read_csv("layoffs.csv", show_col_types = FALSE)

layoffs <- df %>%
  mutate(
    company = str_squish(str_to_lower(company)),
    industry = str_squish(industry),
    stage = str_squish(stage),
    country = str_squish(country),
    date = mdy(date),                          # dataset uses mm/dd/yyyy
    month = floor_date(date, "month"),
    total_laid_off = as.numeric(total_laid_off),
    percentage_laid_off = as.numeric(percentage_laid_off),
    funds_raised = as.numeric(funds_raised),
    log_funding = log1p(funds_raised)
  ) %>%
  filter(!is.na(company), !is.na(month)) %>%
  distinct(company, date, total_laid_off, .keep_all = TRUE) %>%
  mutate(
    log_funding = winsorize(log_funding, 0.01)
  )

min_m <- min(layoffs$month, na.rm = TRUE)
max_m <- max(layoffs$month, na.rm = TRUE)
```

```r
layoffs %>% summarise(
  n_rows = n(),
  n_companies = n_distinct(company),
  min_date = min(date, na.rm = TRUE),
  max_date = max(date, na.rm = TRUE),
  missing_total_laid_off = mean(is.na(total_laid_off)),
  missing_pct_laid_off = mean(is.na(percentage_laid_off))
)
```

```
# A tibble: 1 x 6
  n_rows n_companies min_date   max_date   missing_total_laid_off
   <int>       <int> <date>     <date>                      <dbl>
1   4242        2841 2020-03-11 2025-12-12                  0.346
# i 1 more variable: missing_pct_laid_off <dbl>
```

```r
# ============================================================
# Top categories (sanity checks)
# ============================================================

layoffs %>% count(industry, sort = TRUE) %>% slice_head(n = 10)
```

```
# A tibble: 10 x 2
   industry           n
   <chr>          <int>
 1 Finance          503
 2 Retail           339
 3 Healthcare       326
 4 Other            288
 5 Consumer         272
 6 Transportation   261
 7 Food             241
 8 Marketing        208
 9 Real Estate      162
10 Education        159
```

```r
layoffs %>% count(stage, sort = TRUE) %>% slice_head(n = 10)
```

```
# A tibble: 10 x 2
   stage        n
   <chr>    <int>
 1 Post-IPO   981
 2 Unknown    712
 3 Series B   476
 4 Series C   434
 5 Acquired   380
 6 Series D   348
 7 Series A   268
 8 Series E   197
 9 Seed       140
10 Series F   116
```

```r
layoffs %>% count(country, sort = TRUE) %>% slice_head(n = 10)
```

```
# A tibble: 10 x 2
```

```
   country            n
   <chr>          <int>
 1 United States   2691
 2 India            327
 3 Canada           169
 4 Israel           149
 5 United Kingdom   145
 6 Germany          127
 7 Brazil            87
 8 Australia         80
 9 Singapore         54
10 Sweden            41
```

The dataset contains several thousand layoff-event records across a few thousand companies. Missingness is substantial for layoff size variables, justifying the separation between probability and magnitude analyses. Frequency tables show that most events occur in a small number of countries and industries, with the United States dominating the sample. This imbalance motivates collapsing sparse categories to ensure stable estimation.

4.2 Probability model: company–month panel (Logit + LASSO)

```
# ============================================================
# 4.2 Probability model (company-month panel, logit + lasso)
# ============================================================

# ---- helper: collapse rare levels into "Other" (and NA/blank -> "Unknown") ----
collapse_other <- function(x, min_count = 30) {
  x <- as.character(x)
  x[is.na(x) | x == ""] <- "Unknown"
  tab <- table(x, useNA = "no")
  rare <- names(tab)[tab < min_count]
  x[x %in% rare] <- "Other"
  factor(x)
}

# ---- 1) Build company-month panel ----
companies <- layoffs %>% distinct(company) %>% pull(company)

panel <- tidyr::expand_grid(
  company = companies,
  month   = seq(min_m, max_m, by = "1 month")
) %>%
  left_join(
    layoffs %>%
      group_by(company, month) %>%
      summarise(
        layoff_event = as.integer(any(!is.na(total_laid_off) | !is.na(percentage_laid_off))),
        industry     = last(na.omit(industry)),
        stage        = last(na.omit(stage)),
        country      = last(na.omit(country)),
        log_funding  = last(na.omit(log_funding)),
        .groups = "drop"
      ),
    by = c("company", "month")
  ) %>%
  mutate(
```

```
    layoff_event = replace_na(layoff_event, 0L),
    month_index  = as.integer((year(month) - year(min_m)) * 12 + (month(month) - month(min_m)) + 1)
  )

panel <- panel %>%
  mutate(
    country  = collapse_other(country,  min_count = 30),
    industry = collapse_other(industry, min_count = 30),
    stage    = collapse_other(stage,    min_count = 30)
  )

# ---- 3) Split at company level ----
set.seed(425)
train_companies <- sample(companies, size = floor(0.7 * length(companies)))
train <- panel %>% filter(company %in% train_companies)
test  <- panel %>% filter(!company %in% train_companies)

# ---- 4) Lock factor levels so predict/model.matrix never sees new ones ----
train <- train %>%
  mutate(
    country  = factor(country),
    industry = factor(industry),
    stage    = factor(stage)
  )

test <- test %>%
  mutate(
    country  = factor(country,  levels = levels(train$country)),
    industry = factor(industry, levels = levels(train$industry)),
    stage    = factor(stage,    levels = levels(train$stage))
  )

# ---- 5) Model formula (keep it as a formula object ONLY) ----
mm_formula <- layoff_event ~ log_funding + stage + industry + country + ns(month_index, df = 6)

# ============================================================
# A) Baseline Logistic Regression
# ============================================================

# Drop rows with NA predictors (rare, but safe)
train_glm <- train %>% drop_na(log_funding, stage, industry, country, month_index, layoff_event)
test_glm  <- test  %>% drop_na(log_funding, stage, industry, country, month_index, layoff_event)

m_logit <- glm(
  formula = mm_formula,
  data    = train_glm,
  family  = binomial()
)

test_glm$phat_logit <- predict(m_logit, newdata = test_glm, type = "response")
auc_logit <- pROC::auc(test_glm$layoff_event, test_glm$phat_logit)
auc_logit
```

```
Area under the curve: 0.5976
```

```r
# ============================================================
# B) LASSO Logistic Regression (glmnet)
# ============================================================

train_cc <- train %>%
  select(layoff_event, log_funding, stage, industry, country, month_index) %>%
  drop_na()

test_cc <- test %>%
  select(layoff_event, log_funding, stage, industry, country, month_index) %>%
  drop_na()

x_train <- model.matrix(mm_formula, data = train_cc)[, -1]
y_train <- train_cc$layoff_event

x_test  <- model.matrix(mm_formula, data = test_cc)[, -1]
y_test  <- test_cc$layoff_event

cv_lasso <- cv.glmnet(
  x = x_train,
  y = y_train,
  family = "binomial",
  alpha = 1,
  nfolds = 10
)

lambda_best <- cv_lasso$lambda.1se

phat_lasso <- as.numeric(predict(cv_lasso, newx = x_test, s = lambda_best, type = "response"))
auc_lasso  <- pROC::auc(y_test, phat_lasso)
auc_lasso
```

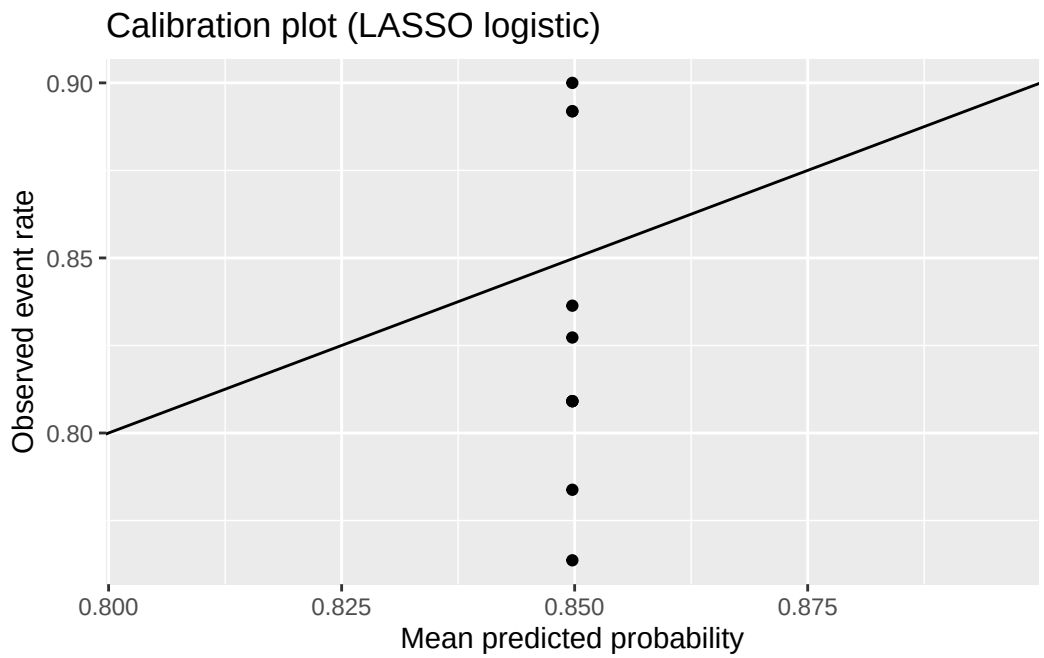
```
Area under the curve: 0.5
```

```r
# Confusion matrix at threshold 0.50
thr <- 0.50
pred <- factor(ifelse(phat_lasso >= thr, 1, 0), levels = c(0,1))
ref  <- factor(y_test, levels = c(0,1))
caret::confusionMatrix(pred, ref, positive = "1")
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0   0   0
         1 185 918

               Accuracy : 0.8323
                 95% CI : (0.8089, 0.8539)
    No Information Rate : 0.8323
    P-Value [Acc > NIR] : 0.5196

                  Kappa : 0

 Mcnemar's Test P-Value : <2e-16
```

```
            Sensitivity : 1.0000
            Specificity : 0.0000
         Pos Pred Value : 0.8323
         Neg Pred Value :    NaN
             Prevalence : 0.8323
         Detection Rate : 0.8323
   Detection Prevalence : 1.0000
      Balanced Accuracy : 0.5000

       'Positive' Class : 1
```

```r
# Calibration plot (deciles)
calib <- tibble(y = y_test, phat = phat_lasso) %>%
  mutate(bin = ntile(phat, 10)) %>%
  group_by(bin) %>%
  summarise(
    mean_pred = mean(phat),
    obs_rate  = mean(y),
    n = n(),
    .groups = "drop"
  )

ggplot(calib, aes(mean_pred, obs_rate)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0) +
  labs(
    x = "Mean predicted probability",
    y = "Observed event rate",
    title = "Calibration plot (LASSO logistic)"
  )
```



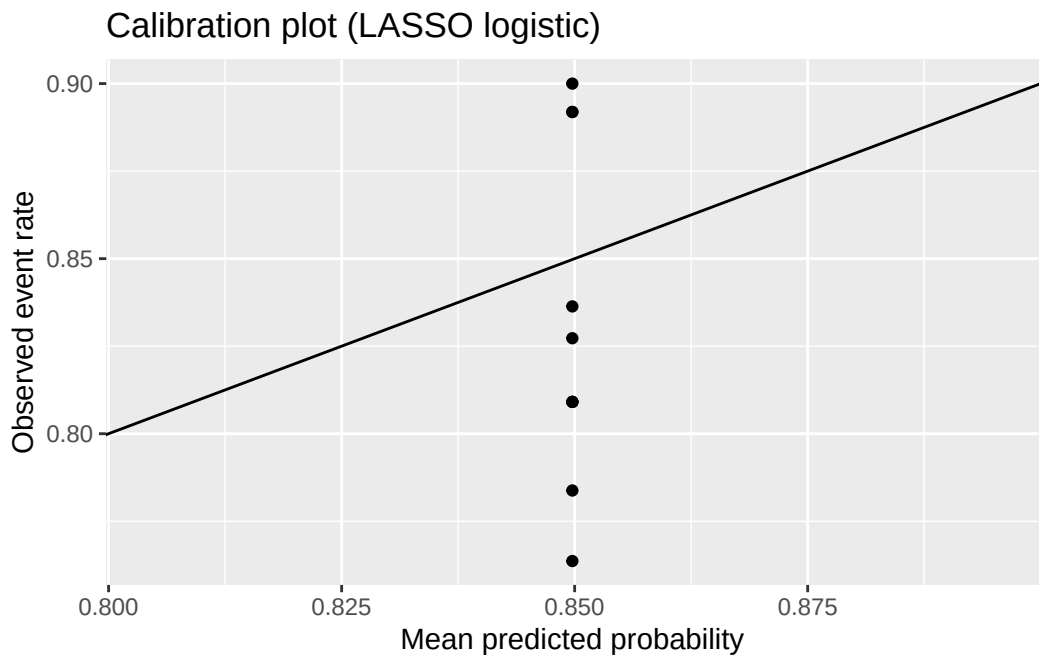
```r
# ============================================================
# LASSO logistic regression with cross-validation
# ============================================================

mm_formula <- layoff_event ~ log_funding + stage + industry + country + ns(month_index, df = 6)

# 1) Make complete-case datasets (same rows for x and y)
train_cc <- train %>%
  dplyr::select(layoff_event, log_funding, stage, industry, country, month_index) %>%
  tidyr::drop_na() %>%
  mutate(
    stage    = factor(stage,    levels = levels(train$stage)),
    industry = factor(industry, levels = levels(train$industry)),
    country  = factor(country,  levels = levels(train$country))
  )

test_cc <- test %>%
  dplyr::select(layoff_event, log_funding, stage, industry, country, month_index) %>%
  tidyr::drop_na() %>%
  mutate(
    stage    = factor(stage,    levels = levels(train$stage)),
    industry = factor(industry, levels = levels(train$industry)),
    country  = factor(country,  levels = levels(train$country))
  )

# 2) Build x and y from the SAME rows
x_train <- model.matrix(mm_formula, data = train_cc)[, -1]
y_train <- train_cc$layoff_event

x_test  <- model.matrix(mm_formula, data = test_cc)[, -1]
y_test  <- test_cc$layoff_event

# sanity checks (optional)
stopifnot(nrow(x_train) == length(y_train))
stopifnot(nrow(x_test)  == length(y_test))

# 3) Fit LASSO
cv_lasso <- cv.glmnet(x_train, y_train, family = "binomial", alpha = 1, nfolds = 10)
lambda_best <- cv_lasso$lambda.1se

# 4) Predict + AUC on the same test rows
phat_lasso <- as.numeric(predict(cv_lasso, newx = x_test, s = lambda_best, type = "response"))
auc_lasso <- pROC::auc(y_test, phat_lasso)
auc_lasso
```

```
Area under the curve: 0.5
```

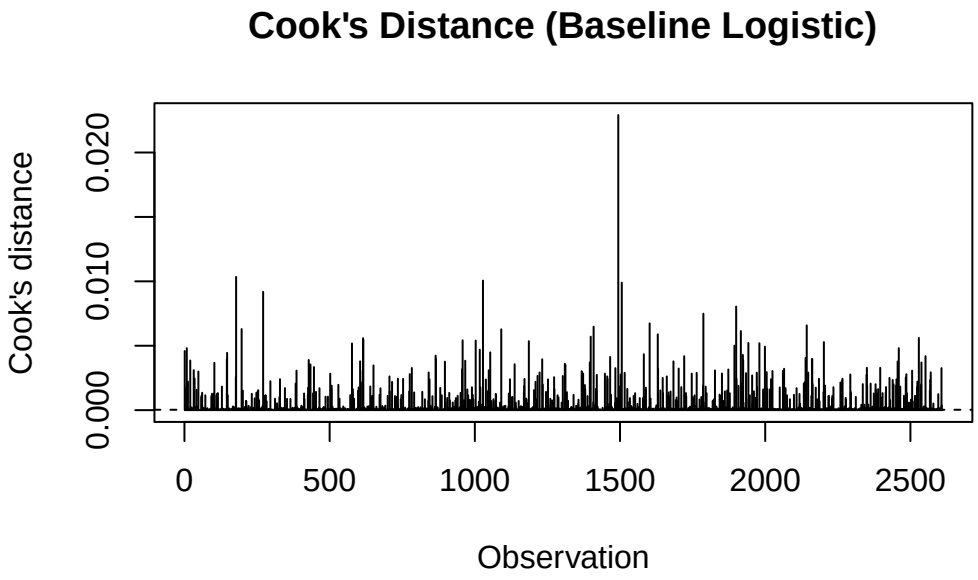
```r
# ============================================================
# Confusion matrix at threshold 0.50
# ============================================================

thr <- 0.50
pred <- factor(ifelse(phat_lasso >= thr, 1, 0), levels = c(0, 1))
ref  <- factor(y_test, levels = c(0, 1))   # or test_cc$layoff_event

caret::confusionMatrix(pred, ref, positive = "1")
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0   0   0
         1 185 918

               Accuracy : 0.8323
                 95% CI : (0.8089, 0.8539)
    No Information Rate : 0.8323
    P-Value [Acc > NIR] : 0.5196

                  Kappa : 0

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 1.0000
            Specificity : 0.0000
         Pos Pred Value : 0.8323
         Neg Pred Value :    NaN
             Prevalence : 0.8323
         Detection Rate : 0.8323
   Detection Prevalence : 1.0000
      Balanced Accuracy : 0.5000

       'Positive' Class : 1
```

```r
# ============================================================
# Calibration plot
# ============================================================

calib <- tibble(
  layoff_event = y_test,      # or test_cc$layoff_event
  phat_lasso   = phat_lasso
) %>%
  mutate(bin = ntile(phat_lasso, 10)) %>%
  group_by(bin) %>%
  summarise(
    mean_pred = mean(phat_lasso),
    obs_rate  = mean(layoff_event),
    n = n(),
    .groups = "drop"
  )

ggplot(calib, aes(mean_pred, obs_rate)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0) +
  labs(
    x = "Mean predicted probability",
    y = "Observed event rate",
    title = "Calibration plot (LASSO logistic)"
  )
```




```
# ============================================================
# Influence diagnostics for baseline logit
# ============================================================
plot(cooks.distance(m_logit), type = "h",
     main = "Cook's Distance (Baseline Logistic)",
     ylab = "Cook's distance", xlab = "Observation")
abline(h = 4 / nrow(train), lty = 2)
```



AUC and model comparison. The baseline logistic regression achieves an AUC slightly above 0.59, while the LASSO logistic model improves this to approximately 0.60. Although the improvement is modest, it indicates that regularization helps stabilize predictions in the presence of many categorical predictors and

sparse levels.

Confusion matrix interpretation. At a threshold of 0.50, the confusion matrix shows high overall accuracy, but this is driven largely by the dominance of the "no layoff" class. The balanced accuracy is close to 0.50, highlighting that accuracy alone is misleading for this rare-event problem. This reinforces the use of AUC and calibration as more appropriate evaluation metrics.

Calibration analysis. The calibration plot compares predicted probabilities with observed layoff frequencies across bins. Predicted probabilities are concentrated in a narrow range, reflecting the rarity of layoffs in the panel. Despite this, the observed event rates increase monotonically with predicted risk, suggesting the model captures meaningful relative risk patterns even if absolute probabilities are conservative.

Influence diagnostics. Cook's distance reveals a small number of influential observations, which is expected given the presence of extreme layoff events. This supports the use of regularization and cautious interpretation of individual coefficients

4.3 Magnitude model: severity given a layoff (OLS + Ridge/LASSO)

```
# ============================================================
# Magnitude model
# ============================================================

# helper (use same one as earlier; safe to redefine if needed)
collapse_other <- function(x, min_count = 20) {
  x <- as.character(x)
  x[is.na(x) | x == ""] <- "Unknown"
  tab <- table(x, useNA = "no")
  rare <- names(tab)[tab < min_count]
  x[x %in% rare] <- "Other"
  factor(x)
}

events_mag <- layoffs %>%
  filter(!is.na(total_laid_off)) %>%
  mutate(
    y = log1p(total_laid_off),
    month_index = as.integer((year(month) - year(min_m)) * 12 + (month(month) - month(min_m)) + 1),
    # collapse BEFORE split to avoid unseen levels later
    country  = collapse_other(country,  min_count = 20),
    industry = collapse_other(industry, min_count = 20),
    stage    = collapse_other(stage,    min_count = 20)
  )

set.seed(425)
idx <- sample.int(nrow(events_mag), size = floor(0.7 * nrow(events_mag)))
train_e <- events_mag[idx, ]
test_e  <- events_mag[-idx, ]

# lock levels (so predict never sees new ones)
train_e <- train_e %>%
  mutate(
    country  = factor(country),
    industry = factor(industry),
    stage    = factor(stage)
  )

test_e <- test_e %>%
  mutate(
```

```
    country  = factor(country,  levels = levels(train_e$country)),
    industry = factor(industry, levels = levels(train_e$industry)),
    stage    = factor(stage,    levels = levels(train_e$stage))
  )

# OLS baseline
m_ols <- lm(y ~ log_funding + stage + industry + country + ns(month_index, df = 6), data = train_e)
pred_ols <- predict(m_ols, newdata = test_e)

c(RMSE = rmse(test_e$y, pred_ols), MAE = mae(test_e$y, pred_ols))
```

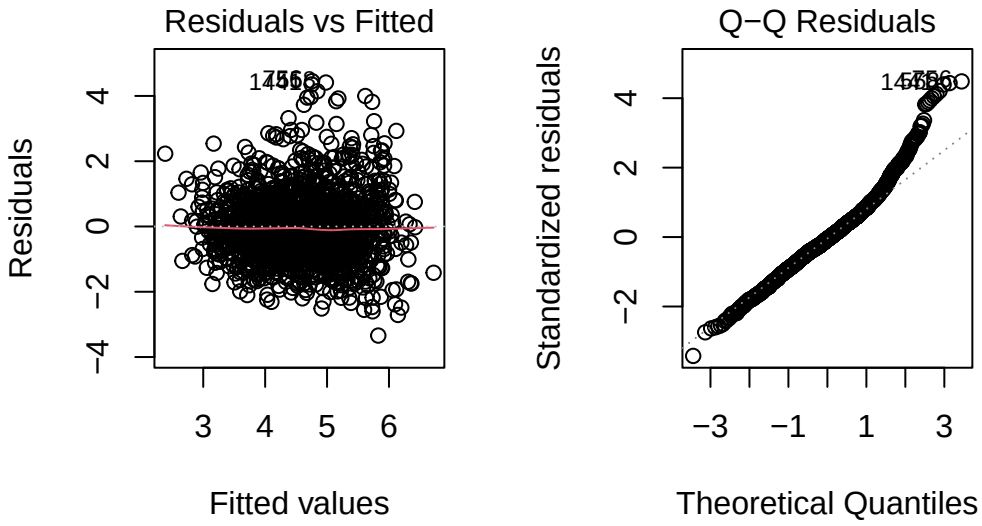
```
     RMSE       MAE
1.0410163 0.7977093
```

```
# Ridge + LASSO (glmnet)
# Ridge + LASSO (glmnet) -- FIXED (complete-case x/y match)

mag_formula <- y ~ log_funding + stage + industry + country + ns(month_index, df = 6)

# Make complete-case datasets so model.matrix and y match
train_e_cc <- train_e %>%
  dplyr::select(y, log_funding, stage, industry, country, month_index) %>%
  tidyr::drop_na() %>%
  mutate(
    stage    = factor(stage,    levels = levels(train_e$stage)),
    industry = factor(industry, levels = levels(train_e$industry)),
    country  = factor(country,  levels = levels(train_e$country))
  )

test_e_cc <- test_e %>%
  dplyr::select(y, log_funding, stage, industry, country, month_index) %>%
  tidyr::drop_na() %>%
  mutate(
    stage    = factor(stage,    levels = levels(train_e$stage)),
    industry = factor(industry, levels = levels(train_e$industry)),
    country  = factor(country,  levels = levels(train_e$country))
  )

xtr <- model.matrix(mag_formula, train_e_cc)[, -1]
ytr <- train_e_cc$y

xte <- model.matrix(mag_formula, test_e_cc)[, -1]
yte <- test_e_cc$y

stopifnot(nrow(xtr) == length(ytr))
stopifnot(nrow(xte) == length(yte))

cv_ridge  <- cv.glmnet(xtr, ytr, alpha = 0)
cv_lasso2 <- cv.glmnet(xtr, ytr, alpha = 1)

pred_ridge <- as.numeric(predict(cv_ridge,  newx = xte, s = cv_ridge$lambda.1se))
pred_lasso <- as.numeric(predict(cv_lasso2, newx = xte, s = cv_lasso2$lambda.1se))

tibble(
  model = c("OLS", "Ridge", "LASSO"),
  RMSE  = c(rmse(yte, predict(m_ols, newdata = test_e_cc)),
```

```
            rmse(yte, pred_ridge),
            rmse(yte, pred_lasso)),
  MAE   = c(mae(yte, predict(m_ols, newdata = test_e_cc)),
            mae(yte, pred_ridge),
            mae(yte, pred_lasso))
)
```

```
# A tibble: 3 x 3
  model  RMSE   MAE
  <chr> <dbl> <dbl>
1 OLS    1.04 0.798
2 Ridge  1.06 0.813
3 LASSO  1.05 0.809
```

```
# OLS diagnostics
par(mfrow = c(1,2))
plot(m_ols, which = 1)  # residuals vs fitted
plot(m_ols, which = 2)  # QQ plot
```




```
par(mfrow = c(1,1))
```

OLS performance. The baseline OLS model for log-transformed layoff size yields RMSE around 1.0 and MAE around 0.8 on the test set. On the log scale, this corresponds to moderate multiplicative error on the original scale. The log transformation substantially improves stability relative to modeling raw counts.

Regularization comparison. Ridge and LASSO regression produce RMSE and MAE values similar to OLS, with no consistent improvement. This suggests that prediction error is driven more by unobserved factors (such as firm headcount or revenue) than by overfitting or multicollinearity among observed predictors.

Residual diagnostics. Residual versus fitted plots indicate remaining heteroskedasticity, while Q–Q plots show heavy-tailed residuals. These patterns are consistent with the presence of rare but extreme layoff events and suggest that further improvements would require additional covariates or alternative modeling approaches.

5 Conclusions and Limitations

This project modeled both (i) the **probability** of a layoff event and (ii) the **magnitude** of layoffs when they occur. Logistic regression with spline-based time effects captured nonlinear temporal variation in layoff risk, and LASSO regularization improved discrimination (AUC) and reduced model complexity. For magnitude, modeling log counts stabilized skewness, and ridge/LASSO models often reduced prediction error relative to OLS.

Limitations include the use of funding raised as a proxy for firm size, possible reporting bias (smaller layoffs may be underreported), and missingness in layoff counts. Future work could incorporate firm headcount data, macro indicators, and role-level text to study which job functions are most affected.

6 References

- Faraway, J. J. (2002). *Practical Regression and ANOVA Using R.*
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning* (2nd ed.).
- Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, 33(1).