Logistic Regression

Logistic regression is a type of **generalized linear model**, which is a family of models for which key linear assumptions are relaxed. Logistic regression is an excellent tool for modeling relationships with outcomes that are not measured on a continuous scale (a key requirement for linear regression). Logistic regression is often leveraged to model the probability of observations belonging to different classes of a categorical outcome, and this type of modeling is known as **classification**. The context for classification can be binomial for two classes (e.g., active/inactive, promoted/not promoted), multinomial for multiple unordered classes (e.g., job family, location), or ordinal for multiple ordered classes (e.g., survey items measured on a Likert scale, performance level, education level). Regardless of the outcome variable's classes, logistic regression is in fact a type of *regression* analysis, which by definition returns a numeric outcome—and probabilities are numeric. Logistic regression accomplishes this by using a link function to generalize the linear model for non-continuous outcomes.

You may be wondering why linear regression cannot be implemented when the categorical outcome is dummy coded as outlined in chapter "Data Preparation". In a binary case, in which the categorical response has been coded as 1/0, least squares regression would produce an estimate for $\hat{\beta}X$ that represents the estimated probability of the outcome coded as 1 given X. For example, if attrition is the binary outcome and $Y = 1$ for employees who left and $Y = 0$ for employees who stayed, $\hat{Y} > 0.5$ could lend to a termination prediction assuming this is an appropriate probability threshold. Linear regression may produce estimates lower than 0 and higher than 1, however, which complicates the interpretation of estimates as probabilities.

This issue is not limited to binary categorical outcomes. Response variables with more than 2 categories cannot naturally be converted into quantitative values that are appropriate for linear regression. Instead of modeling the response directly as in linear regression, logistic regression models the probability of an outcome's class given values for one or more predictors. For example, we can leverage our `employees`

C. Starbuck, *The Fundamentals of People Analytics*,
https://doi.org/10.1007/978-3-031-28674-2_12

data set to model the probability of `active` given a value for `interview_rating`, which would be written as $Pr($`active` = Yes | `interview_rating`$)$ or simply $p($`inteview_rating`$)$. A probability of `active` = Yes will be estimated for a given value of `interview_rating`, and the probability threshold for determining the predicted class needs to be defined based on the business context. If we want to minimize false positives (i.e., incorrectly flagging at-risk employees who do not actually leave), we may set the threshold to something north of 0.5 (e.g., 0.7) to gain more confidence that those classified into the termination class are highly likely to exit.

Binomial Logistic Regression

Since estimating a binary outcome using linear regression can result in $p(X) < 0$ for some values of X and $p(X) > 1$ for others, we need a function that constrains the output to a [0,1] interval. For logistic regression, the *logistic* function is used. This function converts the linear model, $p(X) = \beta_0 + \beta_1 X$, to the following form:

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

Irrespective of the value of X, the logistic function will always produce a sigmoidal (*S-shaped*) curve.

Taking the ratio of $\frac{p(X)}{1-p(X)}$ will give the odds of the outcome, which ranges between 0 (very low) and ∞ (very high). The logarithm of this ratio, $\log(\frac{p(X)}{1-p(X)})$, is known as the *log odds* or *logit* and is fundamental to logistic regression. Log odds is a *monotonic transformation*, meaning the greater the odds, the greater the log of odds (and vice versa).

Recall that in linear regression, the coefficient β on a predictor is interpreted as the average change in Y for a one-unit increase in the respective predictor's value. In logistic regression, the interpretation is similar but rather than β representing the average change in Y, and it represents the average unit change in the *log of the odds* for a one-unit increase in the predictor's value.

In R, the `glm()` function is used in conjunction with the `family = binomial` argument to fit a logistic regression model. As we covered in chapter "Statistical Inference", discrete probability distributions can be leveraged to model different types of nominal variables, and the binomial distribution is appropriate for a sequence of independent observations with only two outcomes—such as our `active` variable featuring only *yes* and *no* values. Therefore, we need to pass the `family = binomial` argument into the `glm()` function. The formula passed into the function is structured consistent with the `lm()` function used for linear regression: `glm(y ~ x, data)`.

```
##
## Call:
## glm(formula = active ~ interview_rating, family = "binomial",
##     data = employees)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.03045   0.00000   0.00002   0.00531   2.06562
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -74.486     10.333  -7.208 5.67e-13 ***
## interview_rating   21.963      2.997   7.329 2.32e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1298.58  on 1469  degrees of freedom
## Residual deviance:  103.27  on 1468  degrees of freedom
## AIC: 107.27
##
## Number of Fisher Scoring iterations: 11
```

The logistic regression output has some differences relative to the output of a linear regression model.

- `Estimate`: Average change in the log of odds for each one-unit increase in the value of the predictor
- `z value`: Ratio of `Estimate` / `Standard Error`. Assuming $\alpha = 0.05$, a $|z$-value$|$ >= 2 (2s of the mean) is a good rule of thumb for achieving statistical significance per the properties of the normal distribution.
- `Null deviance`: Measure of how well the response can be predicted by a model with only an intercept term; the lower the number, the better the fit.
- `Residual deviance`: Measure of how well the response can be predicted by a model with p predictors; the lower the number, the better the fit. The larger the delta between residual and null deviance, the better the model with p predictors relative to the intercept-only model.

Given the positive coefficient on `interview_rating`, we can interpret this to mean that for each one-unit increase in the average interviewer rating during the onsite stage of the employee's recruiting process, the log of the odds of the employee staying with the organization (`active` status of `Yes` coded as 1) increases by 21.96.

To illustrate why the logistic function is necessary, let us demonstrate differences in applying linear and logistic regression models by regressing a binary outcome `active` onto `interview_rating`.

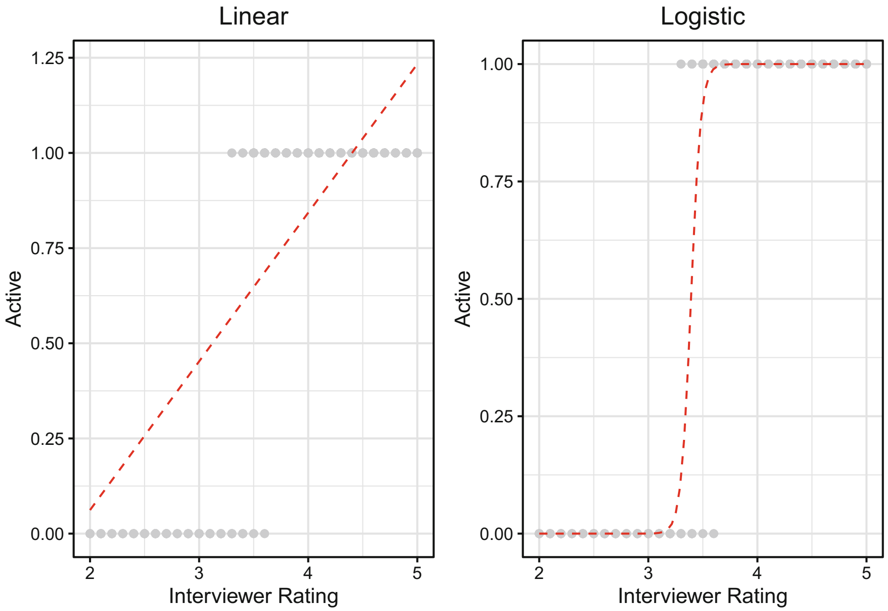

Fig. 1 Linear (left) and logistic (right) functions applied to models regressing active status (1/0) onto median interviewer rating

Figure 1 illustrates that for high values of `interview_rating`, a linear model would estimate probabilities for `active` that are greater than 1. Since probabilities range from 0 (impossible) to 1 (certain), anything outside the [0,1] interval does not make sense. On the other hand, the logistic function produces the *S-shaped* curve described previously. Using the logistic function, the probabilities are constrained to the [0,1] interval, and the visual reflects the fact that `active` can be perfectly predicted for low and high values of `interview_rating`, but it is a mixed bag for values in the middle of the range (3.3–3.6).

It is often helpful to explain the relationship between a predictor and binary outcome in terms of a percentage increase or decrease. When $\beta = 1$, this indicates that the likelihood of the outcome is identical between the two groups of the predictor. If we exponentiate the coefficients, we can convert the log odds into odds ratios to facilitate a more intuitive interpretation. Therefore, $(exp(\beta) - 1) * 100$ will provide the percentage increase or decrease in the odds of the *included* group relative to the *omitted* group.

Let us evaluate the log odds for `active` regressed onto two binary predictors: `overtime` and `job_lvl2plus`.

```r
# Create dummy-coded variable for job level 2+
employees$job_lvl2plus <- ifelse(employees$job_lvl > 1, 1, 0)

# Fit a logistic regression model
glm.fit <- glm(active ~ overtime + job_lvl2plus, data =
↪  employees, family = 'binomial')

# Produce model summary
summary(glm.fit)
```

```
##
## Call:
## glm(formula = active ~ overtime + job_lvl2plus, family = "binomial",
##     data = employees)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3688   0.3532   0.3532   0.6300   1.1270
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.5163     0.1192  12.718  < 2e-16 ***
## overtimeYes   -1.3965     0.1522  -9.176  < 2e-16 ***
## job_lvl2plus   1.2270     0.1523   8.055 7.93e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1298.6  on 1469  degrees of freedom
## Residual deviance: 1149.8  on 1467  degrees of freedom
## AIC: 1155.8
##
## Number of Fisher Scoring iterations: 5
```

We can convert these coefficients into odds ratios by exponentiating the coefficients:

```r
# Return exponentiated coefficients
exp(coef(glm.fit))
```

```
##  (Intercept)  overtimeYes job_lvl2plus
##    4.5551147    0.2474668    3.4108303
```

The exponentiated coefficient on `overtime` is $exp(\beta) = 0.25$, so there is a $(1-0.25)*100 = 75\%$ *decrease* in the odds of being active for employees who work overtime (since `overtime = Yes` is the included group) relative to those who do not work overtime. The exponentiated coefficient on `job_lvl2plus` is $exp(\beta) = 3.41$, so there is a $(3.41 - 1) * 100 = 241\%$ *increase* in the odds of being active for those with a job level of 2 or greater relative to those with a job level of 1 (i.e., attrition is a larger concern for level 1 employees).

Generalized linear mixed models can be fitted using the `glmer()` function from the `lme4` library. The syntax is identical to the illustration of `lmer()` for multilevel linear models in chapter “Linear Regression” with the exception of needing to define `family` as an additional argument.

A comparison of `glm()` and `glmer()` fits for logistic regression is shown in the following block of code. The mixed model example features an additional random (group-level) effect on `business_travel` via `1 | business_travel` and fixed (observation-level) effects on remaining predictors which are consistent with the standard logistic regression model:

```
# Load library
library(lme4)

# Logistic model
glm(active ~ overtime + job_lvl2plus, data = employees, family
↪  = 'binomial')
```

```
##
## Call:  glm(formula = active ~ overtime + job_lvl2plus, family = "binomial",
##     data = employees)
##
## Coefficients:
##  (Intercept)   overtimeYes  job_lvl2plus
##        1.516        -1.396         1.227
##
## Degrees of Freedom: 1469 Total (i.e. Null);  1467 Residual
## Null Deviance:       1299
## Residual Deviance: 1150  AIC: 1156
```

```
# Logistic mixed model
lme4::glmer(active ~ overtime + job_lvl2plus + (1 |
↪  business_travel), data = employees, family = 'binomial')
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: active ~ overtime + job_lvl2plus + (1 | business_travel)
##    Data: employees
##       AIC       BIC    logLik  deviance  df.resid
## 1148.0596 1169.2317 -570.0298 1140.0596      1466
## Random effects:
##  Groups          Name        Std.Dev.
##  business_travel (Intercept) 0.4351
## Number of obs: 1470, groups:  business_travel, 3
## Fixed Effects:
##  (Intercept)   overtimeYes  job_lvl2plus
##        1.546        -1.382         1.223
```

Multinomial Logistic Regression

Multinomial logistic regression is used to estimate the probability of an unordered categorical response with $K > 2$ classes. With an understanding of binomial logistic regression, extending the binomial model to a multinomial logistic regression model should be relatively intuitive.

To extend the binomial model to a multinomial context, we need to first identify a reference level. This decision may be arbitrary or guided by the research question or hypothesis, but the decision is nonetheless important as it impacts how the model coefficients are interpreted—always relative to the reference level. With the reference level defined, we can then express the multinomial logistic regression model as:

$$Pr(Y = k|X = x) = \frac{e^{\beta_{k0}+\beta_{k1}x_1+...+\beta_{kp}x_p}}{1 + \sum_{j=1}^{K-1} e^{\beta_{j0}+\beta_{j1}x_1+...+\beta_{jp}x_p}}$$

where K is the reference class, j is a $K-1$ non-reference level, and k is the specified class for which the probability is being estimated on the basis of values for one or more X predictors.

Consider `dept` from our `employees` data, which has values of `Research & Development`, `Sales`, and `Human Resources`. This is a nominal variable because differences in these levels are not ordered in the same way job levels ranging from 1 to 10 or Likert scales ranging from 1 to 5 are. Employees in the Sales department may be greater in number relative to employees in the Human Resources department, for example, but it would not be appropriate to assign to the Sales department a numeric value that indicates it is *higher* or *better* relative to the Human Resources department.

Multinomial models are essentially a collection of binomial models which compare the log odds of each non-reference category to the specified reference category. If the Human Resources department is identified as the reference category K, then β_{k0} for $k = Sales$ can be interpreted as the log odds of Sales department membership relative to Human Resources department membership in the following equation:

$$\log\left(\frac{Pr(Y = k|X = x)}{Pr(Y = K|X = x)}\right) = \beta_{k0} + \beta_{k1}x_1 + \ldots + \beta_{xp}x_p$$

Depending on the research objective, it may be appropriate to compute the odds of one category relative to all other categories. This can actually be accomplished using binomial regression if the category of interest is coded as 1 and all other categories are coded as 0. In this case, the reference for the binomial model is a *collection* of $K - 1$ categories. If understanding the odds of *each category* relative to a reference category is more appropriate based on the research objective, multinomial logistic regression is the proper model.

Let us illustrate how to implement multinomial logistic regression by determining how variables in the `employees` data set help in classifying employees into departments. To build this model in R, we will use the `multinom` function from the `nnet` package. It is important that the nominal response variable is defined as a factor before implementing multinomial logistic regression, so we will first convert the data type of `dept` from its native character type to a factor. We also need to identify the reference department against which the probability of each of the other departments will be evaluated; we will define this using the `ref` argument within the `relevel()` function:

```
# Load library
library(nnet)

# Convert dept to factor
employees$dept <- factor(employees$dept)

# Specify reference level
employees$dept <- relevel(employees$dept, ref = "Human
↪  Resources")

# Fit multinomial logistic regression model
# An omitted group for categorical variables is defined by
↪  default
multinom.fit <- nnet::multinom(dept ~ overtime + ed_field,
↪  data = employees)
```

```
## # weights:  24 (14 variable)
## initial  value 1614.960064
## iter  10 value 887.448722
## iter  20 value 833.883230
## iter  30 value 833.513790
## final  value 833.513639
## converged
```

```
# Summarize results from model object
summary(multinom.fit)
```

```
## Call:
## nnet::multinom(formula = dept ~ overtime + ed_field, data = employees)
##
## Coefficients:
##                         (Intercept) overtimeYes ed_fieldLife Sciences
## Research & Development    -18.92857   0.1674815              22.19842
## Sales                    -19.40948   0.1602262              21.60505
##                        ed_fieldMarketing ed_fieldMedical ed_fieldOther
## Research & Development          25.72879        22.21217      21.93577
## Sales                           37.42515        21.27842      20.96823
##                        ed_fieldTechnical Degree
## Research & Development                  22.0455
## Sales                                   21.5113
##
## Std. Errors:
##                        (Intercept) overtimeYes ed_fieldLife Sciences
## Research & Development    1.808496   0.3917603              1.817514
## Sales                     1.810390   0.4071887              1.819982
##                        ed_fieldMarketing ed_fieldMedical ed_fieldOther
## Research & Development          8.991934        1.820007      1.864206
## Sales                           8.996707        1.823207      1.874536
##                        ed_fieldTechnical Degree
## Research & Development                  1.84963
## Sales                                   1.85451
##
## Residual Deviance: 1667.027
## AIC: 1695.027
```

Notice the output from the `multinom()` function is quite limited relative to `glm()` and `lm()`. Coefficients and standard errors are provided, but p-values are not available in the output, so we will need to calculate them separately.

A statistical measure named **Akaike Information Criterion (AIC)** is included in the output of this model, which is a score that is helpful for model selection. AIC is calculated by:

$$AIC = -2\frac{\ell}{n} + 2\frac{k}{n},$$

where n is the number of observations, k is the number of parameters (predictors + intercept), and ℓ is the log likelihood function where:

$$\ell = -\frac{n}{2}\left(1 + \ln(2\pi) + \ln\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\right)\right)$$

Just as we compared R^2 across linear regression models in chapter “Linear Regression”, AIC can be compared across models to determine which one is a better fit to the data; lower AIC values indicate better fit. Consistent with how the Adjusted R^2 statistic adjusts for variables that do not provide information, AIC penalizes models that use more parameters. Therefore, if two models explain the same amount of variance in the response, the model with fewer parameters will achieve a lower AIC score.

To determine whether the coefficients are statistically significant, we need to perform an additional step to compute p-values using z scores:

```
# Calculate z-scores
z_scores <- summary(multinom.fit)$coefficients /
↪  summary(multinom.fit)$standard.errors

# Produce p-values
p_values <- (1 - pnorm(abs(z_scores))) * 2

# Transpose and display rounded p-values
data.frame(t(round(p_values, 3)))
```

```
##                           Research...Development Sales
## (Intercept)                               0.000 0.000
## overtimeYes                               0.669 0.694
## ed_fieldLife Sciences                     0.000 0.000
## ed_fieldMarketing                         0.004 0.000
## ed_fieldMedical                           0.000 0.000
## ed_fieldOther                             0.000 0.000
## ed_fieldTechnical Degree                  0.000 0.000
```

These p-values indicate that those who work overtime are not significantly more likely to work in either the Research & Development or Sales departments relative to the Human Resources department. In other words, a considerable portion of employees in all three departments work overtime, so this variable is not helpful in classifying employees into their correct departments. This is evident in Fig. 2.

The p-values indicate that educational background is a strong predictor of department. This is evidenced in Fig. 3, which shows that each educational field is generally dominated by a single department. This means that we can achieve strong departmental *purity* on the basis of `ed_field` alone. All employees who studied Marketing work in Sales and all employees who studied HR work in the HR department. However, those who work in R&D have a variety of educational backgrounds, so the signal is not as clear for these cases and additional variables would be needed to more accurately assign these employees to the correct department.

As we did for binomial logistic regression, we can compute the exponential of model coefficients for the multinomial logistic regression model to convert the log odds to more intuitive odds ratios:

```
# Return exponentiated coefficients from model object
# Transpose rows to columns for improved readability
data.frame(t(exp(coef(multinom.fit))))
```

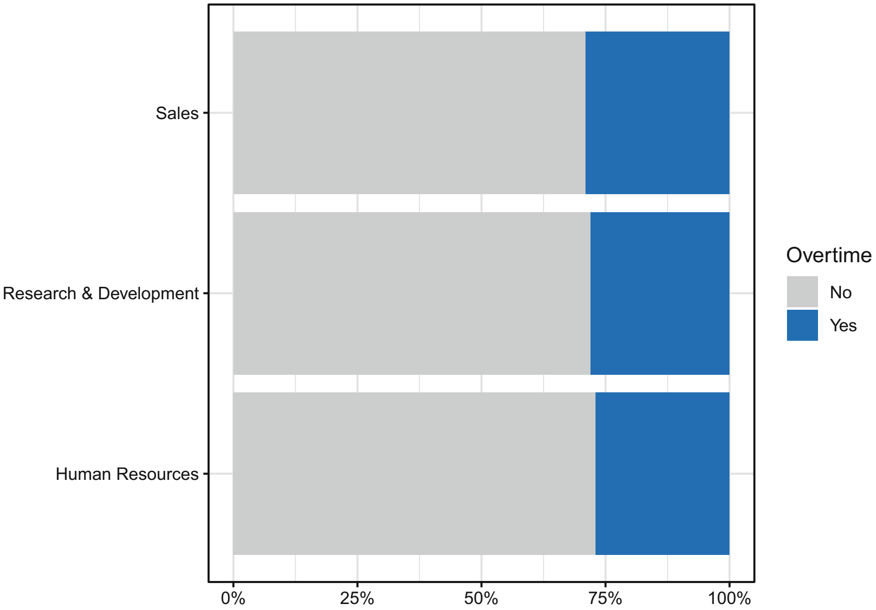

Fig. 2 Overtime by department

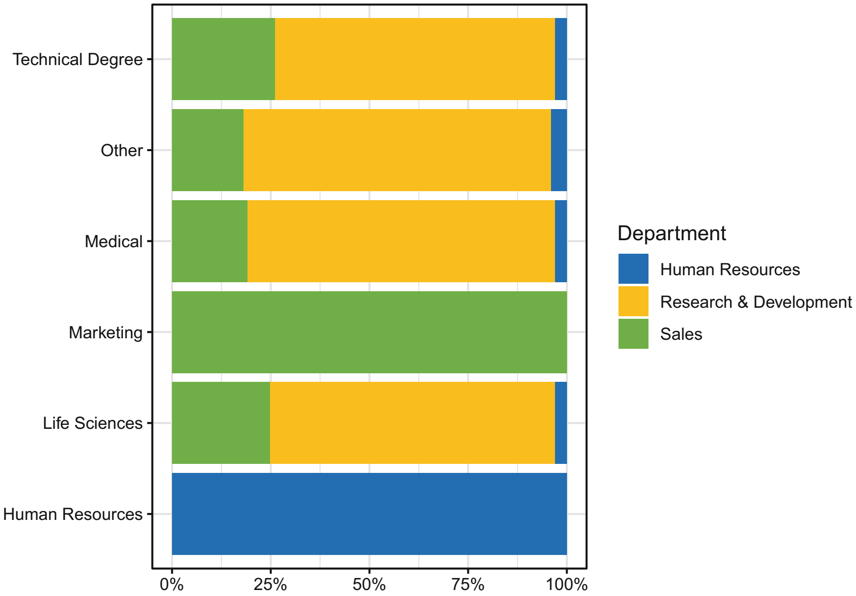

Fig. 3 Department distribution by educational field

```
##                           Research...Development        Sales
## (Intercept)                          6.017654e-09 3.720234e-09
## overtimeYes                          1.182323e+00 1.173776e+00
## ed_fieldLife Sciences                4.371726e+09 2.415207e+09
## ed_fieldMarketing                    1.492349e+11 1.792819e+16
## ed_fieldMedical                      4.432226e+09 1.742213e+09
## ed_fieldOther                        3.361886e+09 1.277572e+09
## ed_fieldTechnical Degree             3.751786e+09 2.199075e+09
```

Consistent with our approach for binomial logistic regression, we can interpret these exponentiated coefficients in terms of having greater or lesser odds. Importantly, in the multinomial context, the odds are *relative to the reference category*—the Human Resources department in this case.

For example, the odds ratio associated with a Medical educational field is $exp(\beta) = 4.43$ for Research & Development and $exp(\beta) = 1.74$ for Sales. Therefore, those with a Medical education have $(4.43 - 1) * 100 = 343$ greater odds of being in the Research & Development department and $(1.74 - 1) * 100 = 74$ greater odds of being in the Sales department relative to the Human Resources department. We can ignore the odds ratios associated with `overtime` since this variable does not provide significant information.

Ordinal Logistic Regression

Many projects in people analytics involve understanding how variables influence ordinal outcomes, such as performance ratings or survey items measured on a Likert scale. Stepwise changes in the levels of ordinal outcomes may or may not be consistent. For example, it may be easy for one to be promoted from job level 1 to 2 but relatively difficult to progress from 5 to 6. Linear regression should not be used in these settings, as linear assumptions are designed for data measured on a continuous scale and will not hold for ordinal data. This section will cover **ordinal logistic regression**, which is a modeling technique designed for understanding how variables influence stepwise changes in a multi-class ordinal outcome.

Beyond the universal data screening procedures we have covered, such as ensuring problematic collinearity is not present, ordinal logistic regression features a unique **proportional odds assumption** that must be satisfied. This assumption, also known as the **parallel regression assumption**, requires that each independent variable has an equal effect at each level of the ordinal outcome. If the effect varies across levels of the outcome, separate models are needed to accurately reflect the associations with each pair of levels.

Though other approaches exist for modeling ordinal outcomes, the proportional-odds model based on cumulative distribution probabilities is the most common. Its intercepts are dependent on the j levels, but slopes are equal as defined by:

$$\log\left(\frac{Pr(Y \leq j)}{Pr(Y > j)}\right) = \beta_{j0} - \sum_{i=1}^{n} \beta_{1i}x_{1i} + \ldots + \beta_{pi}x_{pi}$$

The implementation of ordinal logistic regression will be demonstrated by evaluating the statistical drivers of engagement. First, we need to define `engagement` as an ordered factor:

```
# Define ordered factor
employees$engagement <- ordered(employees$engagement, levels =
↪  c(1, 2, 3, 4, 5))

# Verify structure of engagement variable
str(employees$engagement)
```

```
##  Ord.factor w/ 5 levels "1"<"2"<"3"<"4"<..: 3 2 2 3 3 3 4 3 2 3 ...
```

Next, the proportional odds assumption will be checked using the **Brant test**. The Brant test is a set of comparisons of the separate binary logistic models underlying the overall model (Brant, 1990). This test evaluates whether β_{j1} through β_{jp} are consistent across each of the j levels. This is done via a χ^2 test to compare coefficients and determine whether observed differences are larger departures from what we would expect by chance. The null hypothesis states that coefficients are not statistically different across the j levels; therefore, $p < 0.05$ indicates that significant differences in effects are present across the j levels and, thus, the proportional odds assumption is violated.

We can leverage the `brant` package in R for this, which is compatible with the `polr()` function from the `MASS` package that will be used to perform ordinal logistic regression. Since we will be evaluating model statistics, rather than merely using the model for prediction (the subject of chapter "Predictive Modeling"), we need to specify the `Hess = TRUE` argument in the `polr()` function to include the *Hessian matrix* (observed information matrix) in the output.

```
# Load libraries
library(MASS)
library(brant)

# Fit a ordinal logistic regression model
ord.fit <- MASS::polr(engagement ~ org_tenure, data =
↪  employees, Hess = TRUE)

# Test proportional odds assumption using the Brant test
brant::brant(ord.fit)
```

```
## --------------------------------------------
## Test for X2  df  probability
## --------------------------------------------
## Omnibus      0.39   2   0.82
## org_tenure   0.39   2   0.82
## --------------------------------------------
##
## H0: Parallel Regression Assumption holds
```

Notice the line in the output for the omnibus test, which shows an identical χ^2, df, and p-value to the line associated with `org_tenure`. **Omnibus tests** are statistical tests which test for the significance of several parameters in a model at once. For example, a one-way ANOVA evaluating differences in mean commute time across three locations is an omnibus test since it has more than two parameters. As we covered in chapter "Analysis of Differences", the null hypothesis is rejected in the context of ANOVA if there is at least *one* difference in complex contrasts—even if the mean commute time is not significantly different between *all* groups. Test statistics are identical for this ordinal logistic regression model because there is a single predictor, but Brant's omnibus test investigates equality of coefficients for all predictors jointly in the case of more than two parameters (Martin, 2022). The null and alternative hypotheses for Brant's omnibus test are:

- H_0: The odds are proportional for all predictors in the model.
- H_A: The odds are non-proportional for at least one predictor.

The results of Brant's test indicate that we fail to reject H_0 since $p = 0.82$ for the omnibus test, so the proportional odds assumption holds for these data. If the model featured more than one predictor, we could also evaluate the statistics on individual predictors—but only to determine for which predictor(s) the proportional odds assumption is violated *if* $p < 0.05$ for the omnibus test. As the number of variables and tests increases, so too does our risk of incorrectly rejecting the proportional odds assumption; therefore, decisions regarding the proportional odds assumption should not be based on statistics for individual predictors alone. Even if the odds are truly proportional for each predictor independently, with 20 predictors we would expect to find one by chance for which $p < 0.05$ since our tolerance for a Type I error is 1 in 20 with $\alpha = 0.05$.

Since the proportional odds assumption holds, let us review the model output:

```
# Summarize ordinal logistic regression model
summary(ord.fit)
```

```
## Call:
## MASS::polr(formula = engagement ~ org_tenure, data = employees,
##     Hess = TRUE)
##
## Coefficients:
##                 Value Std. Error t value
## org_tenure -0.006965   0.008032 -0.8671
##
## Intercepts:
##     Value      Std. Error t value
## 1|2    -2.8661     0.1271   -22.5423
## 2|3    -0.8419     0.0804   -10.4703
## 3|4      2.1711     0.1036    20.9479
## 4|5  4122.5567     0.1036 39776.9095
##
## Residual Deviance: 3084.592
## AIC: 3094.592
```

The output provides the average effect of a one-unit increase in `org_tenure` (the Coefficients section) as well as intercepts on each pair of levels for `engagement` (the Intercepts section). The effect of a one-unit increase in `org_tenure` in moving `engagement` from one ordinal level to the next is quite small ($\beta = -0.007$). The intercepts are often referred to as cutpoints and can be roughly translated as thresholds. While the intercepts for each cutpoint vary, note that the single coefficient on `org_tenure` is only possible because our proportional odds assumption holds and the effect is consistent (proportional) across levels of our ordered factor, `engagement`.

Consistent with the default output from the `multinom()` function used for multinomial logistic regression, p-values are not provided in the standard output from the `polr()` function. However, we can calculate them for reasonably large samples by comparing the t-values against the standard normal distribution:

```r
# Store coefficients to df
coef_df <- coef(summary(ord.fit))

# Produce p-values
p <- pnorm(abs(coef_df[, "t value"]), lower.tail = FALSE) * 2

# Combine p values with coefficients df
coef_df <- cbind(coef_df, "p value" = p)

# Display df contents
coef_df
```

```
##                    Value  Std. Error       t value       p value
## org_tenure   -0.00696485 0.008032275     -0.867108  3.858828e-01
## 1|2          -2.86612408 0.127144280    -22.542297 1.598105e-112
## 2|3          -0.84185679 0.080404351    -10.470289  1.182814e-25
## 3|4           2.17107818 0.103641956     20.947870  1.962145e-97
## 4|5        4122.55669611 0.103641956  39776.909511  0.000000e+00
```

We can now estimate the likelihood of a particular observation having a specified level of Y, such as $Y \leq 3$, as follows:

$$\log\left(\frac{Pr(Y \leq 3)}{Pr(Y > 3)}\right) = 2.17 - 0.007x_{orgtenure}$$

Review Questions

1. Can linear regression be used when outcome variables are measured on a non-continuous scale? Why or why not?
2. What are some examples of hypotheses for which logistic regression would be an appropriate model?
3. Why is it helpful to calculate the exponential of log odds in a logistic regression model?
4. What does an odds ratio of 1.25 indicate in a binomial context?
5. What does an odds ratio of 0.75 indicate in a multinomial context?
6. How does Akaike Information Criterion (AIC) compare to R^2 with respect to its purpose and function?
7. In what type of R object do ordinal data need to be stored in order to implement ordinal logistic regression?
8. Should linear regression be used to understand associations of predictors with an ordinal outcome? Why or why not?
9. What does the proportional odds (or parallel regression) assumption assume about model coefficients?
10. Why is it important to evaluate Brant's omnibus test—over the test statistics on independent predictors alone—when determining whether the proportional odds assumption for ordinal logistic regression holds?