9

NEGATIVE BINOMIAL REGRESSION MODELS AND ZERO-INFLATED MODELS

OBJECTIVES OF THIS CHAPTER

This chapter introduces negative binomial regression models and zero-inflated models. It first starts with an introduction to the negative binomial regression model followed by a discussion of how to interpret parameter estimates. Next, after a description of the research example, the data, and the sample, a multiple-predictor negative binomial regression model are illustrated with the `glm.nb()` function in the `MASS` package. Then, the `vglm()` function in the `VGAM` package is also used to fit the same multiple-predictor model. Finally, the `zeroinfl()` function in the `pscl` package is used to fit both the zero-inflated Poisson regression model and the zero-inflated negative binomial model. R commands and output are explained in detail. This chapter focuses on fitting the negative binomial regression models and zero-inflated models with R, as well as on interpreting and presenting the results. After reading this chapter, you should be able to:

- Identify when binomial regression models and zero-inflated models are used.
- Fit a negative binomial regression model and a zero-inflated model using R.
- Interpret the output.
- Interpret the incidence rate ratios and marginal effects.
- Compute, plot, and interpret the predicted counts.
- Compare nested models using the likelihood ratio test and compare non-nested models using the Vuong test.
- Present results in publication-quality tables using R.
- Write the results for publication.

9.1 NEGATIVE BINOMIAL REGRESSION MODELS: AN INTRODUCTION

The Poisson regression model which we discussed in the preceding chapter assumes that the mean of the count response variable is equal to the variance of the variable. However, this assumption is often violated. In real data analysis, it is common that the variance could be either greater or less than the mean of the count response variable. Overdispersion occurs when the variance of the count response variable is greater than the mean, whereas underdispersion is present when the variance is less than the mean. Both overdispersion and underdispersion impact the estimation of the standard errors of parameter estimates which result in biased results. This chapter focuses on the issue of overdispersion only since it is more common than underdispersion. To deal with overdispersion in Poisson regression models for count data, we can use several methods to estimate the standard errors. These methods include the use of quasi-Poisson models, robust standard errors, and bootstrapped standard errors. A better option is to use the negative binomial regression model which relaxes the equality of the mean and the variance assumption and allows overdispersion. In Poisson regression, the response variable follows a Poisson distribution with the mean equal to the variance, μ = Variance (Y). In negative binomial regression, the response variable follows a negative binomial distribution which accommodates overdispersion. To relax the equality assumption, the variance of the response variable, Variance (Y), is a function of the mean μ and a dispersion parameter α. Variance (Y) = $\mu(1 + \alpha\mu)$. When the dispersion parameter $\alpha = 0$, the variance is equal to the means. Thus, the negative binomial model is the same as the Poisson model under this circumstance. With both the mean μ and a dispersion parameter α, the negative binomial distribution can be also referred to as the Poisson-Gamma mixture distribution.

As with the Poisson regression model, the negative binomial regression model (Cameron & Trivedi, 2013; Hardin & Hilbe, 2018; Hilbe, 2011; Long & Freese, 2014) can be expressed as follows:

$$\ln(\mu) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon \tag{9.1}$$

where μ is the mean or the expected number of events; β_0 is the intercept; β_1, β_2, ..., β_p are the coefficients for the predictors; and ε is the error term. The left side of the equation, $\ln(\mu)$, is the log link function. The right side of the equation is the linear predictor. The error term ε on the right side of the equation reflects overdispersion. In the negative binomial regression model, the variance of the response variable is a function of the mean μ and a dispersion parameter α. It can be expressed as either a linear form, Variance (Y) = $\mu(1 + \mu)$, or a quadratic form, Variance (Y) = $\mu(1 + \alpha\mu) = \mu + \alpha\mu^2$. The model with a linear form for the variance is called the NB1 negative binomial model, while the model with a quadratic form is referred to as the NB2 negative binomial model in the literature (Hilbe, 2014). The NB2 model is more commonly used and is the focus in this chapter.

We obtain the predicted mean of the count response variable by exponentiating both sides of Equation 9.1. Since we assume that exp(ε) is 1 here, as with the Poisson regression model, the predicted mean of the count response variable in the negative binomial regression model is also expressed as:

$$\mu = \exp\left(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p\right) \tag{9.2}$$

When a count response variable is the number of events during a time period or in a location, a count of events can also be referred to as an incidence rate. If we define the incidence rate as the expected number of events per unit time or location, μ/t, the negative binomial model can also be expressed as follows:

$$\ln(\mu/t) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon \tag{9.3}$$

where μ is the mean count; t is a period of time; μ/t is the incidence rate; and $\ln(\mu/t)$ is the log of the incidence rate. Since $\ln(\mu/t) = \ln(\mu) - \ln(t)$, the equation can be rewritten as:

$$\ln(\mu) = \ln(t) + \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon \tag{9.4}$$

where $\ln(t)$ is the offset in the model equation.

9.1.1 The Negative Binomial Distribution

When there is overdispersion, we prefer the negative binomial probability distribution to the Poisson probability distribution. In the negative binomial probability distribution, we count the number of independent Bernoulli trials before a certain number of successes are achievement, so the number of the total trials equals the summation of the two numbers. There are different ways to express the negative binomial probability distribution. We express the distribution in terms of the probability of the number of trials before successes are achieved. With y trials before k successes in a total of $(y + k)$ trials and the success probability p for each individual trial, we express the negative binomial probability distribution as:

$$P(Y = y) = \binom{y+k-1}{y} p^k (1-p)^y \tag{9.5}$$

where $\binom{y+k-1}{y}$ is the negative binomial coefficient, y is the number of trials before a success is achieved, k is the number of successes in $y + k$ trials, and p is the probability of success for an individual trial.

9.1.2 Incidence Rate Ratios in Negative Binomial Regression Models

Like the Poisson regression model, the negative binomial regression model also estimates the log expected counts of an event or the log incidence rate of the response variable. The incidence rate is defined as the expected number of events during a period of time or in a location. In a simple negative binomial regression model with one predictor, $\ln(\mu) = \alpha + \beta X + \varepsilon$, where μ is the expected counts of an event or the incidence rate. The estimated coefficient is the negative binomial regression coefficient, which is the coefficient on the scale of the natural logarithm. It can be also referred to as the log coefficient. We estimate the relationship between the predictor variable and the log function of the expected counts of an event or the log incidence rate.

By exponentiating both sides of the equation, we get the expected counts of an event or the incidence rate:

$$\mu = \exp(\alpha + \beta X) \tag{9.6}$$

If the independent variable X is a categorical variable with the values of 0 and 1, what are the incidence rates of the response variable?

When $X = 0$, the incidence rate $= \exp(\alpha)$, which is the exponentiated intercept.

When $X = 1$, the incidence rate $= \exp(\alpha + \beta)$, which is the exponentiated sum of the intercept and the coefficient.

The incidence rate ratio of group 1 ($X = 1$) to group 2 ($X = 0$):

$$\text{IRR} = \frac{\exp(\alpha + \beta)}{\exp(\alpha)} = \frac{\exp(\alpha) \times \exp(\beta)}{\exp(\alpha)} = \exp(\beta) \tag{9.7}$$

For a one-unit increase in an independent variable (e.g., from 0 to 1 in the previous example) the change in the incidence rate is the incidence rate ratio, which is the exponentiated coefficient. When the independent variable is continuous, for a one-unit increase from any value of x to the value of $(x + 1)$, the change in the incidence rate is still the exponentiated coefficient.

9.1.3 Model Fit Statistics

As with those discussed in the previous chapters, model fit statistics, such as the log likelihood statistic, the residual deviance, the model chi-square statistic, the AIC and BIC statistics, and the pseudo R^2 statistics, can be computed for the negative binomial regression model. The likelihood ratio test, the AIC and BIC statistics, and the Vuong test can also be used for model comparisons.

9.1.4 Interpretation of Model Parameter Estimates

Interpreting the coefficients in a negative binomial regression model is the same as that in a Poisson regression model. When there are multiple predictors in the model, the incidence rate ratio for a predictor can be interpreted as the change in the expected number of events or the incidence rate of a response variable for a one-unit change in a predictor variable when holding other predictor variables constant.

We can also interpret an incidence rate ratio as a percentage change in an incidence rate. It can be calculated by using (incidence rate − 1) × 100%. A positive percentage change in an incidence rate indicates there is an increase in the incidence rate, whereas a negative percentage change corresponds to a decrease in the rate. A zero percentage change indicates no change in the rate at all.

9.2 RESEARCH EXAMPLE AND DESCRIPTION OF THE DATA AND SAMPLE

This chapter focuses on the same research problem as that in Chapter 8. We will still investigate the relationships between the count response variable, the number of zoo visits in a year, and four predictor variables. The GSS 2016 data are used for the following analyses. The following are the variables used for data analysis in this chapter:

- `vistzoo`: the recoded variable of the number of zoo visits in a year
- `maritals`: the recoded variable of marital (marital status) with 1 = currently married and 0 = not currently married
- `educ`: the highest education completed
- `female`: recoded variable of sex with 1 = female and 0 = male
- `wrkfull`: working full time or not.

9.3 FITTING A MULTIPLE-PREDICTOR NEGATIVE BINOMIAL REGRESSION MODEL WITH R

9.3.1 Packages and Functions for Negative Binomial Regression Models in R

The `glm.nb()` function in the `MASS` package (Venables & Ripley, 2002) and the `vglm()` function in the `VGAM` package (Yee, 2010) are used for fitting the negative binomial regression model. Since `MASS` is a part of R base distribution, you just need to load the package by typing `library(MASS)`. You also need to load the `VGAM` package with the `library(VGAM)` function if it is installed. The `glm.nb()` function is introduced first.

9.3.2 The `glm.nb()` Function

The `glm.nb()` function in the `MASS` package is used to fit the negative binomial regression model. As with the `glm()` function, the model formula in `glm.nb()` specifies the dependent variable and the predictor variable(s), which are separated by the tilde (~). The plus (+) symbol is used to connect multiple predictor variables. The data argument specifies the data frame. Please note that no `family` argument needs to be specified and the default link function, the log function, can be omitted. For more details on how to use this function, after loading the `MASS` package with the `library(VGAM)` function, type `help(glm.nb)` in the command prompt.

9.3.3 The Negative Binomial Regression Model: Multiple-Predictor Model

In the following example, the `glm.nb(vistzoo ~ educ + maritals + female + wrkfull, data = count)` command tells R to predict the count response variable `vistzoo` from the four independent variables. In the `glm.nb()` function, the model equation is specified as `vistzoo ~ educ + maritals + female + wrkfull`. The data argument specifies `data = count`. The `family` argument is not needed. The fitted model is named `nbr`. The `summary(nbr)` function displays the output of the fitted model.

```
> # Negative binomial regression model with glm.nb() in MASS
> library(MASS)
> nbr <- glm.nb(vistzoo ~ educ + maritals + female + wrkfull, data=count)
> summary(nbr)

Call:
glm.nb(formula = vistzoo ~ educ + maritals + female + wrkfull,
    data = count, init.theta = 1.58331383, link = log)

Deviance Residuals:
    Min        1Q      Median       3Q       Max
-1.4702   -1.0592   -0.8788    0.3811    3.3187

Coefficients:
             Estimate    Std. Error   z value   Pr(>|z|)
(Intercept)  -1.30862    0.24565     -5.327    9.98e-08 ***
educ          0.04982    0.01665      2.992    0.00277 **
maritals      0.19126    0.09337      2.048    0.04052 *
female        0.02412    0.09386      0.257    0.79719
wrkfull       0.53499    0.09525      5.617    1.94e-08 ***
---
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
(Dispersion parameter for Negative Binomial(1.5833) family taken to be 1)

    Null deviance: 931.01  on 901  degrees of freedom
Residual deviance: 873.58  on 897  degrees of freedom
AIC: 2200

Number of Fisher Scoring iterations: 1

              Theta:  1.583
          Std. Err.:  0.251

 2 x log-likelihood:  -2188.013
```

9.3.4 Interpreting the Output

The output of the `glm.nb()` function is similar to that of the `glm()` function. The only difference is that the output of the `glm.nb()` function provides the additional estimate of the dispersion parameter. In the output, the first part is the call, which shows the R command for the negative binomial regression model. The second part shows the minimum, first quarter, median, third quarter, and maximum values of the deviance residuals. The third part shows the coefficients table including the parameter estimates for the four predictor variables and the intercept, their standard errors, the Wald *z* statistics, and the associated *p* values. The fourth part of the output shows the fit statistics including the null deviance, the residual deviance, and the AIC. Finally, the fifth part provides the dispersion parameter estimate labeled `Theta`, its standard error, and the 2 log likelihood value.

We start from the coefficients section. The coefficients in the negative binomial model can be interpreted in a similar way to those in the Poisson regression model. For the predictor variable `educ`, Wald $z = 2.992$. The associated *p* value, `Pr(>|z|) < .01`, so we rejected the null hypothesis. The rejection of the null hypothesis indicates that the predictor variable `educ` is a significant predictor of the count response variable `vistzoo`. For a one-unit increase in education the log expected number of visits to a zoo increases by a factor of .050 after controlling for other predictors in the model.

For the predictor variable `maritals`, Wald $z = 2.048$. The associated *p* value, `Pr(>|z|) < .05`, so we reject the null hypothesis. For the predictor variable `wrkfull`, Wald $z = 5.617$. The associated *p* value, `Pr(>|z|) < .001`, so we also reject the null hypothesis. Therefore, `maritals` and `wrkfull` are significant predictors of the response variable.

For the predictor variable `female`, the Wald $z = .257$. The associated *p* value `Pr(>|z|) = .797`, so we fail to reject the null hypothesis and conclude that there is no significant effect of female on the outcome variable. In other words, whether a person is a female or male does not significantly predict the log expected number of visits to a zoo.

Next, the fit statistics section displays the null deviance for the intercept-only model, the residual deviance for the model, and the AIC.

Finally, the dispersion parameter labeled `Theta`, its standard error, and the 2 log likelihood value are shown at the bottom. Recall that the variance of the response variable in the negative binomial regression model is expressed as a function of the mean μ and a dispersion parameter α. In the NB2 model, Variance $(Y) = \mu(1 + \alpha\mu) = \mu + \alpha\mu^2$. The `glm.nb()` function uses a slightly different form, Variance $(Y) = \mu + \mu^2/\theta$. As we see, $\theta = 1/\alpha$. `Theta` = 1.583 and its standard error is .251. We can easily compute $\alpha = 1/1.583 = .632$.

```
> alpha <- 1/1.583
> alpha
[1] 0.6317119
```

2 log likelihood = −2,188.01. We will use it to test the overall model fit later in the chapter.

We extract the coefficients with the `coef(nbr)` command and obtain the profiled confidence intervals with `confint(nbr)` as follows.

```
> coef(nbr)
(Intercept)          educ      maritals        female       wrkfull
-1.30862163    0.04982352    0.19125927    0.02411989    0.53499022
> confint(nbr)
Waiting for profiling to be done...
                  2.5 %         97.5 %
(Intercept)  -1.79149450   -0.83255567
educ          0.01694296    0.08288263
maritals      0.00766270    0.37477527
female       -0.15909658    0.20790336
wrkfull       0.34869644    0.72241114
```

We request the incidence rate ratios with the `exp(coef(nbr))` and `exp(confint(nbr))` commands, respectively. Both results are combined with `cbind(exp(nbr)), exp(confint(nbr)))`.

```
> exp(coef(nbr))
(Intercept)         educ     maritals        female      wrkfull
  0.2701922    1.0510856    1.2107733    1.0244131    1.7074315
> exp(confint(nbr))
Waiting for profiling to be done...
```

```
                   2.5 %      97.5 %
 (Intercept)   0.1667108   0.4349363
 educ          1.0170873   1.0864143
 maritals      1.0076921   1.4546645
 female        0.8529140   1.2310942
 wrkfull       1.4172189   2.0593927

> cbind(exp(coef(nbr)), exp(confint(nbr)))
Waiting for profiling to be done...
                                2.5 %      97.5 %
 (Intercept)   0.2701922   0.1667108   0.4349363
 educ          1.0510856   1.0170873   1.0864143
 maritals      1.2107733   1.0076921   1.4546645
 female        1.0244131   0.8529140   1.2310942
 wrkfull       1.7074315   1.4172189   2.0593927
```

The standard errors of the incidence rate ratios can be obtained with `exp(coef(nbr))*sqrt(diag(vcov(nbr)))`.

```
> exp(coef(nbr))*sqrt(diag(vcov(nbr)))
(Intercept)         educ     maritals       female      wrkfull
 0.06637319   0.01750264   0.11304683   0.09614906   0.16262904
```

9.3.5 Interpreting the Incidence Rate Ratios in the Negative Binomial Regression Model

As with the Poisson regression model, the negative binomial regression model also estimates the log expected counts of an event. Recall that the exponentiated (β_j) is the incidence rate ratio (IRR) for a one-unit change in a predictor variable when holding other predictors constant. A positive Poisson regression coefficient corresponds to an incidence rate ratio greater than 1, whereas a negative coefficient is associated with an incidence rate ratio less than 1.

In this model, for `educ`, the incidence rate ratio is 1.051. The result indicates that the incidence rate increases by a factor of 1.051 for a one-unit increase in education when holding all other predictors constant. It can also be interpreted as the change in the number of events as follows. The expected number of visits to a zoo increases by 1.051 for a one-unit increase in education when holding all other predictors constant. In other words, for a one-unit increase in education the expected number of visits to a zoo increases by 5.1%.

For `maritals`, the incidence rate ratio is 1.211, which indicates that the expected number of visits to a zoo for the married is 1.211 times as high as that for the unmarried when holding the other predictors constant. In other words, the expected number of visits to a zoo for the married is 21.1% higher than that for the unmarried.

The incidence rate ratio for `wrkfull` can be interpreted in a similar way. The incidence rate ratio is 1.707, which indicates that the expected number of visits to a zoo for those working full time is 1.707 times as high as that for those not working full time when holding the other predictors constant. In other words, the expected number of visits to a zoo for those working full time is 70.7% higher than that for those not working full time.

With regard to `female`, the incidence rate ratio is 1.024, which is not significant (see the associated $p = .797$ in the coefficients table). This indicates that being female does not impact the expected number of visits to a zoo.

9.3.6 Interpreting the Marginal Effects in the Negative Binomial Regression Model

We load the `margins` package (Leeper, 2021) with the `library(margins)` command, compute the marginal effects with the `margins(nbr)` command, and obtain the summary results with `summary(marg.nbr)` as follows.

```
> # marginal effects
> library(margins)
> marg.nbr <- margins(nbr)
> summary(marg.nbr)
  factor     AME      SE       z       p      lower    upper
    educ  0.0403  0.0137  2.9452  0.0032   0.0135  0.0672
  female  0.0195  0.0760  0.2570  0.7972  -0.1294  0.1684
maritals  0.1548  0.0762  2.0330  0.0421   0.0056  0.3041
 wrkfull  0.4330  0.0813  5.3269  0.0000   0.2737  0.5924
```

The average marginal effect for `educ` is .040. The result indicates that there are on average .040 more visits to a zoo for a one-unit increase in education when holding all other predictors constant.

The average marginal effect for `maritals` is .155. This indicates that the married have on average .155 more visits to the zoo than the unmarried when holding the other predictors constant. The average marginal effects for the other two predictor variables can be interpreted in a similar way.

9.3.7 Model Fit Statistics

Testing the Overall Model Using the Likelihood Ratio Test

To test if the overall model is significant, we fit a null model with the intercept only and compare the single-predictor model with the null model using the `anova()` function. The null model is fitted using the `glm.nb()` function with 1 as the

```
> # Testing the overall model using the likelihood ratio test
> nbr.0 <- glm.nb(vistzoo ~ 1, data = count)
> summary(nbr.0)

  Call:
  glm.nb(formula = vistzoo ~ 1, data = count, init.theta = 1.256798254,
     link = log)

  Deviance Residuals:
      Min        1Q     Median      3Q       Max
  -1.1184   -1.1184   -1.1184    0.1561    2.4080

  Coefficients:
                 Estimate    Std. Error    z value    Pr(>|z|)
  (Intercept)    -0.21020     0.04744      -4.431     9.37e-06 ***

  ---
  Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

  (Dispersion parameter for Negative Binomial(1.2568) family taken to be 1)

     Null deviance: 870.56 on 901 degrees of freedom
  Residual deviance: 870.56 on 901 degrees of freedom
  AIC: 2247

  Number of Fisher Scoring iterations: 1

                Theta: 1.257
            Std. Err.: 0.177

  2 x log-likelihood: -2243.012
```

intercept term in the model formula. The command and the output are displayed below.

The `anova(nbr.0, nbr, test = "Chisq")` command compares the log-likelihood statistics of the fitted model `nbr` and the null model `nbr.0` using the likelihood ratio test.

```
> anova(nbr.0, nbr, test = "Chisq")
Likelihood ratio tests of Negative Binomial Models

Response: vistzoo
                                      Model       theta     Resid. df    2 x log-lik.
1                                         1    1.256798        901        -2243.012
2    educ + maritals + female + wrkfull      1.583314        897        -2188.013

      Test      df      LR stat.          Pr(Chi)
1
2    1 vs 2     4      54.99921     3.250222e-11
```

Here we first fit the null model and then compare the single-predictor model with the null model. This two-step process can also be simplified to the one-line command with the `update()` function within the `anova()` function. In the `anova(nbr, update (nbr, ~1), test= "Chisq")` command, we use `update(nbr, ~ 1)` to fit the null model with the intercept only.

```
> anova(nbr, update(nbr, ~1), test = "Chisq")
Likelihood ratio tests of Negative Binomial Models

Response: vistzoo
                                    Model     theta  Resid. df    2 x log-lik.
1                                       1  1.256798        901       -2243.012
2    educ + maritals + female + wrkfull  1.583314        897       -2188.013
     Test    df    LR stat.       Pr(Chi)
1
2    1 vs 2   4    54.99921   3.250222e-11
```

The null hypothesis of the test for the overall model is that the four predictor variables do not contribute to the model, and the alternative hypothesis is that the multiple-predictor model is better than the null model with no independent variables. The likelihood ratio test statistic $LR\ \chi^2_{(4)} = 54.999$, $p < .001$, which indicates that the overall model with the four predictors is significantly different from zero. Therefore, the multiple-predictor model provides a better fit than the null model with no independent variables in predicting the log number of visits to a zoo in a year.

Pseudo R^2

We use the `nagelkerke()` function in the `rcompanion` package (Mangiafico, 2021) to compute the pseudo R^2 statistics for the single-predictor model. We load the package first with `library(rcompanion)` and then use the `nagelkerke (nbr)` command.

```
> # Pseudo R2 with nagelkerke()
> library(rcompanion)
> nagelkerke(nbr)
$`Models`

Model: "glm.nb, vistzoo ~ educ + maritals + female + wrkfull, count, 1.58331383, log"
Null:  "glm.nb, vistzoo ~ 1, count, 1.256798254, log"

$Pseudo.R.squared.for.model.vs.null
                              Pseudo.R.squared
McFadden                             0.0245202
Cox and Snell (ML)                   0.0591530
Nagelkerke (Cragg and Uhler)         0.0645200
```

```
$Likelihood.ratio.test
Df.diff   LogLik.diff    Chisq      p.value
   -4          -27.5    54.999   3.2502e-11

$Number.of.observations
Model: 902
Null:  902

$Messages
[1] "Note: For models fit with REML, these statistics are based on refitting with ML"

$Warnings
[1] "None"
```

McFadden's R^2 is .025, Cox and Snell's R^2 is .059, and Nagelkerke's R^2 is .065. The same results can be computed using the equations for these three pseudo R^2 statistics. In the R command below, LLM1 is the log-likelihood value for the multiple-predictor model and LL0 is the log-likelihood value for the null model. The number of observations is 902. In addition, McFadden1 is the object name for McFadden's R^2, CS1 for Cox and Snell's R^2, and NG1 for Nagelkerke's R^2.

```
> # Pseudo R2 with equations
> LLM1 <- logLik(nbr)
> LL0 <- logLik(nbr.0)
> McFadden1 <- 1-(LLM1/LL0)
> McFadden1
'log Lik.' 0.02452024 (df=6)
> CS1 <- 1-exp(2*(LL0-LLM1)/902)
> CS1
'log Lik.' 0.05915298 (df=2)
> NG1<-CS1/(1-exp(2*LL0/902))
> NG1
'log Lik.' 0.06451996 (df=2)
```

AIC and BIC Statistics

The AIC and BIC statistics can also be computed by the AIC(nbr) and BIC(nbr) commands. The AIC(PR.2, nbr) and BIC(PR.2, nbr) commands compare the AIC and BIC statistics between the Poisson regression model and the negative binomial regression model, respectively. The output is as follows.

```
> # AIC and BIC Statistics
> AIC(nbr)
[1] 2200.013
> BIC(nbr)
[1] 2228.841
> PR.2 <- glm(vistzoo ~ educ + maritals + female + wrkfull, family = poisson, data
= count)
```

```
> AIC(PR.2, nbr)
       df       AIC
PR.2    5  2286.799
nbr     6  2200.013

> BIC(PR.2, nbr)
       df       BIC
PR.2    5  2310.822
nbr     6  2228.841
```

The AIC and BIC statistics for the multiple-predictor negative binomial regression model are 2,200.013 and 2,228.841, respectively. Compared with those in the Poisson regression model, both AIC and BIC indicate that the negative binomial regression model fits the data better.

9.3.8 Interpreting the Predicted Counts With the ggpredict() Function in the ggeffects Package

By using the ggpredict() function in the ggeffects package (Lüdecke, 2018b), we can compute the predicted number of events of the count response variable at specified values of the predictor variables. We first load the package with library(ggeffects) since it has been installed in previous chapters. The command nbr.ed <- ggpredict(nbr, terms = "educ[12, 14, 16]") tells R to compute the predicted counts of the response variable using the ggpredict() function. The argument inside the function includes the estimated model name, nbr, and the terms = "educ[12, 14, 16]" argument, which specifies the predictor variable educ at the values of 12, 14, and 16 when holding other predictor variables at their means. The terms option can specify up to four variables, including the second to fourth grouping variables. The output is assigned to the object named nbr.ed. The as.data.frame() function or the sqrt(diag(vcov())) function can be used to request the standard errors. The output is omitted here.

```
> # Predicted counts with ggpredict() in ggeffects
> library(ggeffects)
> nbr.ed <- ggpredict(nbr,terms = "educ[12, 14, 16]")
> nbr.ed
# Predicted counts of vistzoo

educ | Predicted |         95% CI
-------------------------------
  12 |      0.70 | [0.62, 0.78]
  14 |      0.77 | [0.70, 0.85]
  16 |      0.85 | [0.76, 0.95]

Adjusted for:
* maritals = 0.44
*   female = 0.56
*  wrkfull = 0.47
> plot(nbr.ed)
```

The results table displays the values of educ, the predicted counts, their standard errors, and the confidence intervals. The predicted counts in the negative binomial regression model are similar to those in the Poisson regression model which was introduced in the previous chapter.

When educ = 12, and the other predictor variables are held at their means (maritals = .44, female = .56, and wrkfull = .47), the predicted number of visits to a zoo is .70.

When educ = 14, and the other three predictor variables are held at their means, the predicted number of visits to a zoo is .77.

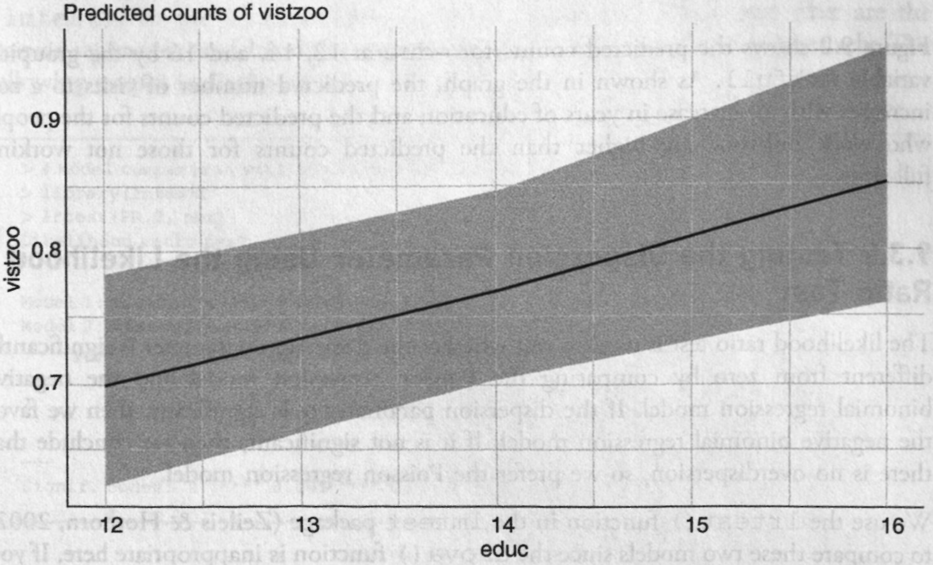
When educ = 16, and the other predictor variables are held at their means, the predicted number of visits to a zoo is .85.

The predicted counts are plotted using the plot(nbr.ed) function. Figure 9.1 shows the predicted number of visits to a zoo when educ is at the values of 12, 14, and 16.

The graph shows that with the increase of the years of education, the predicted number of visits to a zoo increases. In other words, people with higher levels of education are associated with having more visits to a zoo.

We can also compute the predicted counts for a continuous variable at given values by different groups. In the following example, we compute the predicted number of visits to a zoo for educ at the values of 12, 14, and 16 by the two groups in wrkfull

FIGURE 9.1 ● Predicted Counts for educ at 12, 14, and 16



when holding other variables at their means. The command is as follows: nbr.ew <- ggpredict(nbr, terms = c("educ[12, 14, 16]", "wrkfull")). In the ggpredict() function, the terms = c("educ[12, 14, 16]", "wrkfull") option specifies both educ and wrkfull, with the latter as the grouping variable. The output is assigned to an object named nbr.ew and is plotted with plot(nbr.ew).

```
> nbr.ew <- ggpredict(nbr, terms = c("educ[12, 14, 16]", "wrkfull"))
> nbr.ew
# Predicted counts of vistzoo

# wrkfull = 0

educ | Predicted |       95% CI
-------------------------------
  12 |      0.54 | [0.47, 0.63]
  14 |      0.60 | [0.52, 0.69]
  16 |      0.66 | [0.56, 0.77]

# wrkfull = 1

educ | Predicted |       95% CI
-------------------------------
  12 |      0.92 | [0.80, 1.07]
  14 |      1.02 | [0.90, 1.15]
  16 |      1.13 | [0.99, 1.29]

Adjusted for:
* maritals = 0.44
*   female = 0.56
> plot(nbr.ew)
```

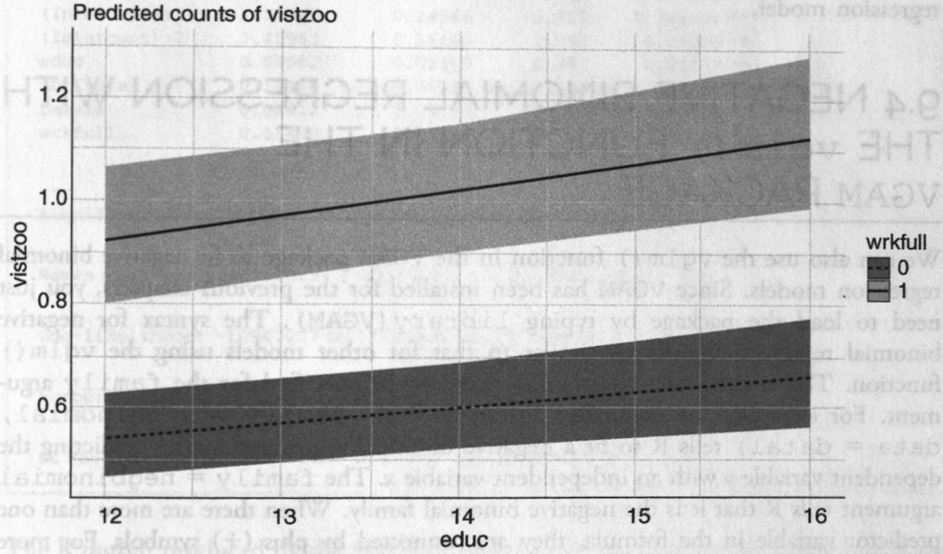
Figure 9.2 shows the predicted counts for educ at 12, 14, and 16 by the grouping variable wrkfull. As shown in the graph, the predicted number of visits to a zoo increases with an increase in years of education and the predicted counts for the people who work full-time are higher than the predicted counts for those not working full-time.

9.3.9 Testing the Dispersion Parameter Using the Likelihood Ratio Test

The likelihood ratio test is used to test whether the dispersion parameter is significantly different from zero by comparing the Poisson regression model and the negative binomial regression model. If the dispersion parameter α is significant, then we favor the negative binomial regression model. If it is not significant, then we conclude that there is no overdispersion, so we prefer the Poisson regression model.

We use the lrtest() function in the lmtest package (Zeileis & Hothorn, 2002) to compare these two models since the anova() function is inappropriate here. If you

FIGURE 9.2 ⬢ Predicted Counts for educ at 12, 14, and 16 by wrkfull



use the anova() function, you will get the same degrees of freedom of the deviances for these two models, which will make the model comparison impossible. To use the lrtest() function, you need to install the lmtest package first by typing install.packages("lmtest") and then load it by typing library (lmtest). In the lrtest(PR.2, nbr) command, PR.2 and nbr are the Poisson regression model and the negative binomial regression model, respectively. The following output is displayed.

```
> # Model comparison with the likelihood ratio test
> library(lmtest)
> lrtest(PR.2, nbr)
Likelihood ratio test

Model 1: vistzoo ~ educ + maritals + female + wrkfull
Model 2: vistzoo ~ educ + maritals + female + wrkfull
  #Df    LogLik   Df    Chisq    Pr(>Chisq)
1  5    -1138.4
2  6    -1094.0   1    88.786    < 2.2e-16 ***

---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The likelihood ratio test, $\chi^2_{(1)} = 88.79, p < .001$, which indicates that the dispersion parameter α is significantly different from zero, so we favor the negative binomial regression model.

9.4 NEGATIVE BINOMIAL REGRESSION WITH THE vglm() FUNCTION IN THE VGAM PACKAGE

We can also use the vglm() function in the VGAM package to fit negative binomial regression models. Since VGAM has been installed for the previous chapters, you just need to load the package by typing library(VGAM). The syntax for negative binomial regression models is similar to that for other models using the vglm() function. The negbinomial family needs to be specified for the family argument. For example, the command vglm(y ~ x, family = negbinomial, data = data1) tells R to fit a negative binomial regression model predicting the dependent variable *y* with an independent variable *x*. The family = negbinomial argument tells R that it is the negative binomial family. When there are more than one predictor variable in the formula, they are connected by plus (+) symbols. For more details on how to use this function, type help(negbinomial) in the command prompt after loading the VGAM package.

In the following example, the nb.v <- vglm(vistzoo ~ educ + maritals + female + wrkfull, family = negbinomial, data = count) command tells R to predict the count response variable vistzoo from the four independent variables. In the model formula for the vglm() function, the dependent variable vistzoo and the four predictor variables are separated by the tilde (~). The four predictor variables include educ, maritals, female, and wrkfull, which are connected by plus (+) symbols. We also specify the data arguments data = count. The fitted model is named nb.v. The following output is shown by the summary(nb.v) command.

```
> # Negative binomial regression model with vglm() in VGAM
> library(VGAM)
> nb.v <- vglm(vistzoo ~ educ + maritals + female + wrkfull, family = negbinomial,
data = count)
> summary(nb.v)

Call:
vglm(formula = vistzoo ~ educ + maritals + female + wrkfull,
    family = negbinomial, data = count)

Pearson residuals:
                    Min        1Q      Median      3Q      Max
loglink(mu)      -0.885   -0.6873     -0.5854  0.4253    7.239
loglink(size)   -13.773   -0.5511     -0.3851  0.9561    1.454
```

```
Coefficients:
               Estimate   Std. Error   z value    Pr(>|z|)
(Intercept):1  -1.30861    0.24566     -5.327   9.98e-08 ***
(Intercept):2   0.45953    0.16463      2.791    0.00525 **
educ            0.04982    0.01665      2.992    0.00277 **
maritals        0.19126    0.09337      2.048    0.04052 *
female          0.02412    0.09386      0.257    0.79722
wrkfull         0.53499    0.09525      5.617   1.95e-08 ***

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Names of linear predictors: loglink(mu), loglink(size)

Log-likelihood: -1094.007 on 1798 degrees of freedom

Number of Fisher scoring iterations: 5

No Hauck-Donner effect found in any of the estimates
```

The R output produced by the `vglm()` function for the negative binomial regression model is similar to that for the Poisson regression model. It includes the call of the model command, the Pearson residuals, the coefficients, the names of linear predictors, the log-likelihood, and the number of Fisher scoring iterations.

The first section shows the call, which is the R command for the model. The second section shows the minimum, first quarter, median, third quarter, and maximum values of the Pearson residuals. The third section shows the coefficients table including the parameter estimates for the predictor variable, the intercept, and the dispersion parameter, their standard errors, the Wald *z* statistics, and the associated *p* values. The fourth section shows the two names of linear predictors, `loglink(mu)` and `loglink(size)`. The former is the log link for the expected count (μ) and the latter is the log link for the index parameter (k) or the dispersion parameter ($1/k$). The fifth section shows the log-likelihood value and the degrees of freedom. Finally, the number of Fisher scoring iterations is displayed at the end.

The negative binomial coefficients in the coefficients section (labeled `Coefficients:`) are the same as those produced from the `glm.nb()` function. See the preceding section on the interpretation of the coefficients.

Also in the coefficients section, the log dispersion parameter is labeled `(Intercept):2`. To obtain the dispersion parameter, we need to exponentiate the estimate of the `(Intercept):2`. The dispersion parameter $\theta = \exp(.460) = 1.583$. As with the `glm.nb()` function, the `vglm()` function also uses a slightly different form for the NB2 model, Variance $(Y) = \mu + \mu^2/\theta$ rather than Variance $(Y) = \mu + \alpha\mu^2$. Since $\theta = 1/\alpha$, we can compute $\alpha = 1/1.583 = .632$.

```
> exp(.45953)
[1] 1.58333
> alpha.v <- 1/1.583
> alpha.v
[1] 0.6317119
```

The coefficients of the predictor variables can be extracted by using the `coef(nb.v, matrix = TRUE)` command. The confidence intervals are obtained with `confint (nb.v, matrix = TRUE)`.

```
> coef(nb.v, matrix = TRUE)
               loglink(mu)   loglink(size)
(Intercept)    -1.30861425       0.4595301
educ            0.04982324       0.0000000
maritals        0.19125640       0.0000000
female          0.02411682       0.0000000
wrkfull         0.53498927       0.0000000

> confint(nb.v, matrix = TRUE)
                        2.5 %        97.5 %
(Intercept):1   -1.790091654   -0.82713686
(Intercept):2    0.136868259    0.78219189
educ             0.017185431    0.08246105
maritals         0.008256362    0.37425644
female          -0.159844305    0.20807794
wrkfull          0.348303906    0.72167464
```

We use the `exp(coef(nb.v, matrix = TRUE))` and `exp(confint(nb.v, matrix = TRUE))` commands to compute the IRRs and the corresponding confidence intervals, respectively.

```
> exp(coef(nb.v, matrix = TRUE))
               loglink(mu)   loglink(size)
(Intercept)      0.2701942         1.58333
educ             1.0510853         1.00000
maritals         1.2107699         1.00000
female           1.0244100         1.00000
wrkfull          1.7074299         1.00000

> exp(confint(nb.v, matrix = TRUE))
                   2.5 %      97.5 %
(Intercept):1  0.1669449   0.4372995
(Intercept):2  1.1466771   2.1862591
educ           1.0173339   1.0859564
maritals       1.0082905   1.4539099
female         0.8522765   1.2313091
wrkfull        1.4166627   2.0578765
```

The standard errors of the IRRs are computed with the `exp(coef(nb.v))*sqrt(diag(vcov(nb.v)))` command.

```
> exp(coef(nb.v))*sqrt(diag(vcov(nb.v)))
(Intercept):1    (Intercept):2        educ     maritals       female
  0.06637490       0.26065788   0.01750293   0.11304847   0.09615055
     wrkfull
  0.16263165
```

We can use the `nagelkerke(nb.v)` command to compute the three types of pseudo R^2 statistics and the likelihood ratio test statistic for the multiple-predictor model. The output is omitted here.

9.5 ZERO-INFLATED POISSON REGRESSION WITH THE zeroinfl() FUNCTION IN THE pscl PACKAGE

When we fit Poisson regression and negative binomial models, sometimes we notice there are many zeros in the count response variable. We may wonder whether these zeros are so excessive that they impact both Poisson and negative binomial distributions of the response variable. Recall that in Poisson regression models, we assume the mean is equal to the variance, which is the equidispersion assumption. So having excessive zero counts does impact the Poisson distribution of the response variable. This is also true for the negative binomial regression model although it has a different over-dispersion assumption. If we have an excess of zero counts in the response variable, what are better alternatives to model this type of count data? In this section, we discuss the zero-inflated models which address the issue of excessive zero counts (Friendly & Meyer, 2016; Hilbe, 2014; Lambert, 1992; Long & Freese, 2014). The zero-inflated model can be applied to both Poisson regression and negative binomial regression. The former is referred to as the zero-inflated Poisson regression model and the latter is called the zero-inflated negative binomial regression model.

Having zero counts in the response variable does not mean the variable is zero-inflated. The zero-inflated model can determine whether these zero counts occur by chance or they are associated with other factors. In other words, can a variable or a set of variables predict these zero counts? This question can be addressed by using a logistic regression model. Therefore, the zero-inflated model includes two components. One is the binary logistic regression model component which predicts if the response variable is a zero count or not. The other is the count model component which predicts the count response variable. The count model component can be either a Poisson regression model or a negative binomial regression model. In the binary logistic regression model component, the outcome is the dichotomized count response variable with the zero counts coded as 1 and the positive counts coded as 0, so we predict the probability of having zero counts. If a predictor or a set of predictors can significantly predict the probability of having zero counts, then there are excessive zero counts associated with that predictor or that set of predictors.

The zero counts are included in both the binary logistic regression model component and the count model component, so there is an overlap of zero counts in both components. Therefore, the zero-inflated model is also called the zero-inflated mixture model.

We use the `zeroinfl()` function in the `pscl` package (Jackman, 2020; Zeileis et al., 2008) to fit both the zero-inflated Poisson regression model and the zero-inflated negative binomial model. We introduce the former model first in this section. Since `pscl` is a user-written package, you need to install it first by typing `install.packages("pscl")` and then load the package by typing `library(pscl)`. The syntax for the zero-inflated models with the `zeroinfl()` function is similar to that for other models using the `glm()` function except that we need to specify the additional zero-inflation part in the model formula. The model formula in `zeroinfl()` includes both the count part and the zero-inflation part. It first specifies the dependent variable and the predictor variable(s) in the count part, which are separated by the tilde (~). When there are multiple predictor variables, they are connected by plus (+) symbols. It then specifies the zero-inflation part which is separated from the count part by the vertical line (|). For example, the model formula, `y ~ x1 + x2 | x1 + x2`, includes the two predictor variables, *x*1 and *x*2 in both the count and the zero-inflation parts. When the same predictors are included in both parts, the model formula can be simplified as `y ~ x1 + x2`. You can also specify different predictors in both parts. The `dist = "poisson"` family is used for the Poisson `family` argument. You can omit this family argument since it is the default for the zero-inflated Poisson regression model. For example, the `zeroinfl(y ~ x1 + x2 | x1 + x2, dist = "poisson", data = data1)` command tells R to fit a zero-inflated Poisson regression model predicting the dependent variable *y* with the two independent variables *x*1 and *x*2. The `dist = "poisson"` argument tells R that it is the Poisson family. For more details on how to use this function, type `help (zeroinfl)` in the command prompt after loading the `pscl` package.

In the following example, the `zip <- zeroinfl(vistzoo ~ educ + maritals + female + wrkfull | educ + maritals + female + wrkfull, data = count)` command tells R to predict the count response variable `vistzoo` from the four independent variables. In the model formula for the `zeroinfl()` function, the dependent variable `vistzoo` and the four predictor variables are separated by the tilde (~). The four predictor variables include `educ`, `maritals`, `female`, and `wrkfull`, which are connected by plus (+) symbols in the count part. The same four predictors are specified in the zero-inflation part. We also specify the data arguments `data = count`. The fitted model is named `zip`. The following output is shown by the `summary(zip)` command.

```
> # ZIP model with zeroinfl() in pscl
> library(pscl)
> zip <- zeroinfl(vistzoo ~ educ + maritals + female + wrkfull | educ + maritals +
female + wrkfull, data = count)
> summary(zip)

Call:
zeroinfl(formula = vistzoo ~ educ + maritals + female + wrkfull |
   educ + maritals + female + wrkfull, data = count)

Pearson residuals:
    Min        1Q     Median        3Q        Max
-1.1051   -0.6683    -0.5906    0.4267    7.1068

Count model coefficients (poisson with log link):
                Estimate     Std. Error    z value     Pr(>|z|)
(Intercept)     -1.09095       0.30492     -3.578     0.000346 ***
educ             0.07632       0.02012      3.794     0.000148 ***
maritals         0.16696       0.10394      1.606     0.108187
female           0.04846       0.09851      0.492     0.622779
wrkfull          0.07383       0.11289      0.654     0.513107

Zero-inflation model coefficients (binomial with logit link):
                Estimate     Std. Error    z value     Pr(>|z|)
(Intercept)     -1.43959       0.91564     -1.572        0.116
educ             0.09034       0.06079      1.486        0.137
maritals        -0.13685       0.32803     -0.417        0.677
female           0.03580       0.31216      0.115        0.909
wrkfull         -1.75639       0.41260     -4.257     2.07e-05 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Number of iterations in BFGS optimization: 17
Log-likelihood: -1108 on 10 Df
```

The R output produced by the `zeroinfl()` function for the zero-inflated Poisson regression model is similar to that for the Poisson regression model by the `glm()` function. The major difference is that the former function displays the coefficients section for the additional zero-inflation component. It includes the call of the model command, the Pearson residuals, the count model coefficients, the zero-inflation model coefficients, the number of iterations, and the log-likelihood and corresponding degrees of freedom.

Specifically, the first section shows the call, which is the R command for the model. The second section shows the minimum, first quarter, median, third quarter, and maximum values of the Pearson residuals. The third section shows the coefficients table for the Poisson regression model including the parameter estimates for the predictor variables, the intercept, their standard errors, the Wald *z* statistics, and the associated *p* values. The fourth section shows the coefficients table for the zero-inflation model estimating the probability when the response variable equals 0, $P(Y = 0)$. The fourth section shows the number of iterations and the log-likelihood and corresponding degrees of freedom.

In the third section, the coefficients table for the Poisson regression model component is labeled with the title `Count model coefficients (poisson with log link)`. The interpretation of the Poisson coefficients is the same as those in the Poisson regression model and the negative binomial regression model. Each coefficient can be interpreted as the change in the log expected of the incidence rate or the log expected number of events for a one-unit increase in the predictor variable when holding other predictors constant.

For the predictor variable `educ`, Wald $z = 3.794$. The associated *p* value, `Pr(>|z|) < .001`, so we reject the null hypothesis. The rejection of the null hypothesis indicates that `educ` is a significant predictor of the count response variable `vistzoo`.

For the predictor variable `maritals`, Wald $z = 1.606$. The associated *p* value, `Pr(>|z|) = .108`, so we fail to reject the null hypothesis. For the predictor variable `female`, the Wald $z = .492$. The associated *p* value `Pr(>|z|) = .663`, so we also fail to reject the null hypothesis. For the predictor variable `wrkfull`, Wald $z = .654$. The associated *p* value, `Pr(>|z|) = .513`, so we fail to reject the null hypothesis, too. Therefore, `maritals`, `female`, and `wrkfull` are not significant predictors of the count response variable.

In the fourth section, the coefficients table for the zero-inflation model component is labeled with the title `Zero-inflation model coefficients (binomial with logit link)`. Recall that the zero-inflation model component is the logistic regression model for the binary outcome variable comparing zero and non-zero counts. The interpretation of the logit coefficients here is the same as those in the logistic regression model which were introduced in earlier chapters. Each coefficient can be interpreted as the change in the logit of having zero counts for a one-unit increase in the predictor variable when holding other predictors constant.

Among the four predictors in the zero-inflation model component, only `wrkfull` is significant. Wald $z = -4.257$. The associated *p* value, `Pr(>|z|) < .001`, so we reject the null hypothesis and conclude that `wrkfull` is significant in predicting excess zeros.

The coefficients of the predictor variables for both count and zero-inflation models can be extracted by using the `coef(zip)` command. The confidence intervals are obtained with `confint(zip)`. The output is omitted here.

We use the `exp(coef(zip))` and `exp(confint(zip))` commands to compute the IRRs and the corresponding confidence intervals, respectively. The results are combined with the `cbind()` function.

```
> exp(coef(zip))
count_(Intercept)       count_educ  count_maritals    count_female
        0.3358971        1.0793123       1.1817114       1.0496532
    count_wrkfull  zero_(Intercept)      zero_educ   zero_maritals
        1.0766275        0.2370247       1.0945462       0.8721047
      zero_female     zero_wrkfull
        1.0364488        0.1726664

> exp(confint(zip))
                          2.5 %     97.5 %
count_(Intercept)    0.18478087  0.6105982
count_educ           1.03758476  1.1227180
count_maritals       0.96391844  1.4487137
count_female         0.86535012  1.2732094
count_wrkfull        0.86291889  1.3432626
zero_(Intercept)     0.03939143  1.4262163
zero_educ            0.97159626  1.2330548
zero_maritals        0.45851228  1.6587704
zero_female          0.56213136  1.9109879
zero_wrkfull         0.07691409  0.3876233

> cbind(exp(coef(zip)), exp(confint(zip)))
                                     2.5 %     97.5 %
count_(Intercept) 0.3358971  0.18478087  0.6105982
count_educ        1.0793123  1.03758476  1.1227180
count_maritals    1.1817114  0.96391844  1.4487137
count_female      1.0496532  0.86535012  1.2732094
count_wrkfull     1.0766275  0.86291889  1.3432626
zero_(Intercept)  0.2370247  0.03939143  1.4262163
zero_educ         1.0945462  0.97159626  1.2330548
zero_maritals     0.8721047  0.45851228  1.6587704
zero_female       1.0364488  0.56213136  1.9109879
zero_wrkfull      0.1726664  0.07691409  0.3876233
```

The standard errors of the IRRs are computed with the exp(coef(zip)) *sqrt(diag(vcov(zip))) command.

```
> exp(coef(zip))*sqrt(diag(vcov(zip)))
count_(Intercept)        count_educ   count_maritals     count_female
       0.10242210        0.02171242       0.12282326       0.10340395
    count_wrkfull  zero_(Intercept)        zero_educ    zero_maritals
       0.12154469        0.21702862       0.06654227       0.28607446
      zero_female      zero_wrkfull
       0.32353663        0.07124141
```

We can use the nagelkerke(zip) command to compute the three types of pseudo R^2 statistics and the likelihood ratio test statistic for the model. The output is omitted here.

We use the Vuong test to compare the zero-inflated Poisson regression model and the Poisson regression model. This test can be used to compare two non-nested models. A significant z statistic suggests that one model is better than the other. The vuong()

function in the pscl package function is used for model comparison. In the vuong(zip,PR.2) command, zip and PR.2 are the zero-inflated Poisson regression model and the Poisson regression model, respectively. The following output is displayed.

```
> # Vuong test
> vuong(zip, PR.2)
Vuong Non-Nested Hypothesis Test-Statistic:
(test-statistic is asymptotically distributed N(0,1) under the
null that the models are indistinguishible)
------------------------------------------------------------
              Vuong z-statistic             H_A      p-value
Raw                    3.068161   model1 > model2  0.0010769
AIC-corrected          2.571295   model1 > model2  0.0050659
BIC-corrected          1.377672   model1 > model2  0.0841523
```

The Vuong test $z = 3.068$, $p < .01$, which indicates that the Vuong z statistic is significantly different from zero, so we prefer the zero-inflated Poisson regression model to the Poisson regression model.

Since among the four predictors in the zero-inflation model component only wrkfull is significant in predicting zeros, we can remove the other three predictors in the binary zero-inflation model. The simplified model can be fitted using the zip2 <- zeroinfl(vistzoo ~ educ + maritals + female + wrkfull | wrkfull, data = count) command. The results of the model are displayed by the summary(zip2) command.

```
> # ZIP reduced model
> zip2 <- zeroinfl(vistzoo ~ educ + maritals + female + wrkfull | wrkfull, data =
count)
> summary(zip2)

Call:
zeroinfl(formula = vistzoo ~ educ + maritals + female + wrkfull |
    wrkfull, data = count)

Pearson residuals:
    Min       1Q   Median      3Q     Max
-1.1557  -0.6592  -0.5575  0.4306  7.0883

Count model coefficients (poisson with log link):
              Estimate  Std. Error  z value   Pr(>|z|)
(Intercept)   -0.80700     0.23268   -3.468   0.000524 ***
educ           0.05660     0.01524    3.715   0.000203 ***
maritals       0.20464     0.08287    2.469   0.013536 *
female         0.04562     0.08251    0.553   0.580321
wrkfull        0.05033     0.11187    0.450   0.652782

Zero-inflation model coefficients (binomial with logit link):
              Estimate  Std. Error  z value   Pr(>|z|)
(Intercept)    -0.1693      0.1716   -0.987      0.324
wrkfull        -1.8017      0.4457   -4.043   5.29e-05 ***
```

```
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Number of iterations in BFGS optimization: 15
Log-likelihood: -1109 on 7 Df
```

We use the likelihood ratio test to compare the zero-inflated Poisson regression model and the simplified model. In the `lrtest(zip2, zip)` command, `zip2` and `zip` are the two models being compared. The following output is displayed.

```
> lrtest(zip2, zip)
Likelihood ratio test

Model 1: vistzoo ~ educ + maritals + female + wrkfull | wrkfull
Model 2: vistzoo ~ educ + maritals + female + wrkfull | educ + maritals +
      female + wrkfull
       #Df      LogLik    Df      Chisq      Pr(>Chisq)
1        7     -1108.8
2       10     -1107.5     3     2.4508          0.4842
```

The likelihood ratio test, $\chi^2_{(3)} = 2.451$, $p = .484$, which is not significant. This indicates that there is no significant difference between these two models. Since the simplified model is more parsimonious than the original model, we prefer the former model with fewer parameters.

9.6 ZERO-INFLATED NEGATIVE BINOMIAL REGRESSION WITH THE zeroinfl() FUNCTION IN THE pscl PACKAGE

We again use the `zeroinfl()` function in the `pscl` package to fit and the zero-inflated negative binomial model. Since `pscl` has been loaded in the last section with `library(pscl)`, we do not need to load it again. To fit the zero-inflated negative binomial regression model, with all else the same as those in the zero-inflated Poisson regression model, we only need to specify a different family argument, `dist = "negbin"`. The model formula in `zeroinfl()` still includes the negative binomial model and the zero-inflation model components. The `zinb <- zeroinfl(vistzoo ~ educ + maritals + female + wrkfull | educ + maritals + female + wrkfull, dist = "negbin", data = count)` command tells R to predict the count response variable `vistzoo` from the four independent variables with the zero-inflated negative binomial regression model. The fitted model is named `zinb`. The following output is shown by the `summary(zinb)` command.

```
> # Zero-inflated NB model with zeroinfl() in pscl
> zinb <- zeroinfl(vistzoo ~ educ + maritals + female + wrkfull | educ + maritals +
female + wrkfull, dist="negbin", data = count)
> summary(zinb)

Call:
zeroinfl(formula = vistzoo ~ educ + maritals + female + wrkfull |
   educ + maritals + female + wrkfull, data = count, dist = "negbin")

Pearson residuals:
     Min        1Q     Median       3Q       Max
-0.9752   -0.6224   -0.5706    0.4036    6.7580

Count model coefficients (negbin with log link):
               Estimate    Std. Error    z value    Pr(>|z|)
(Intercept)    -1.11891     0.29408      -3.805     0.000142 ***
educ            0.05972     0.01930       3.094     0.001978 **
maritals        0.19616     0.10398       1.887     0.059223 .
female          0.04780     0.10249       0.466     0.640922
wrkfull         0.18744     0.13569       1.381     0.167149
Log(theta)      0.80108     0.21345       3.753     0.000175 ***

Zero-inflation model coefficients (binomial with logit link):
               Estimate    Std. Error    z value    Pr(>|z|)
(Intercept)    -2.02306     1.34445      -1.505      0.132
educ            0.07225     0.08484       0.852      0.394
maritals       -0.01096     0.50484      -0.022      0.983
female          0.19995     0.50826       0.393      0.694
wrkfull        -8.28490    17.01979      -0.487      0.626

---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Theta = 2.2279
Number of iterations in BFGS optimization: 25
Log-likelihood: -1088 on 11 Df
```

The R output created by the zeroinfl() function for the zero-inflated negative binomial regression model is similar to that for the zero-inflated Poisson regression model. The only difference is that the output for the zero-inflated negative binomial regression model displays the additional dispersion parameter Log(theta). It includes the call of the model command, the Pearson residuals, the count model coefficients, the zero-inflation model coefficients, the dispersion parameter theta, the number of iterations, and the log-likelihood and corresponding degrees of freedom.

The fourth section shows the coefficients table for the zero-inflation model estimating the probability when the response variable equals 0, $P(Y = 0)$. The fourth section shows the number of iterations and the log-likelihood and the corresponding degrees of freedom.

The coefficients table for the negative binomial regression model component is labeled with the title `Count model coefficients (negbin with log link)`. It shows the estimates for the predictor variables, the intercept, and the dispersion parameter, their standard errors, the Wald z statistics, and the associated p values. The interpretation of the coefficients here is the same as those in the negative binomial regression model. Each coefficient can be interpreted as the change in the log expected of the incidence rate or the log expected number of events for a one-unit increase in the predictor variable when holding other predictors constant.

For the predictor variable `educ`, Wald $z = 3.094$. The associated p value, `Pr(>|z|)` $<$ `.01`, so we reject the null hypothesis. The rejection of the null hypothesis indicates that `educ` is a significant predictor of the count response variable `vistzoo`.

For the predictor variable `maritals`, Wald $z = 1.887$. The associated p value, `Pr(>|z|)` $=$ `.059`, so we fail to reject the null hypothesis. For the predictor variable `female`, the Wald $z = .466$. The associated p value `Pr(>|z|)` $=$ `.640`, so we also fail to reject the null hypothesis. For the predictor variable `wrkfull`, Wald $z = 1.381$. The associated p value, `Pr(>|z|)` $=$ `.167`, so we also fail to reject the null hypothesis.

Also, in the coefficients section, the log dispersion parameter is labeled `Log(theta)`. Wald $z = 3.753$. The associated p value, `Pr(>|z|)` $<$ `.01`, which is significant. To obtain the dispersion parameter, we need to exponentiate the estimate of the `Log(theta)`. The dispersion parameter $\theta = \exp(.801) = 2.228$, which is also displayed at the bottom of the output. As with the `glm.nb()` function, the `zeroinfl()` function also uses a slightly different form for the NB2 model, Variance $(Y) = \mu + \mu^2/\theta$ rather than Variance $(Y) = \mu + \alpha\mu^2$. Since $\theta = 1/\alpha$, we can compute $\alpha = 1/2.228 = .449$. The R output for computing θ and α is as follows.

```
> exp(0.80108)
[1] 2.227946
> alpha.zinb <- 1/2.2279
> alpha.zinb
[1] 0.4488532
```

The coefficients table for the zero-inflation model component is labeled with the title `Zero-inflation model coefficients (binomial with logit link)`. The interpretation of the logit coefficients here is the same as those in the logistic regression model. Each coefficient can be interpreted as the change in the logit of having zero counts for a one-unit increase in the predictor variable when holding other predictors constant.

Among the four predictors in the zero-inflation model component, none of them are significant in predicting zeros. This suggests that the more parsimonious negative binomial model may be a better choice. We use the Vuong test to compare these two models later in the chapter.

The coefficients of the predictor variables can be extracted with `coef(zinb)`. The confidence intervals are obtained with the `confint(zinb)` command. The output is omitted here.

We use the `exp(coef(zinb))` and `exp(confint(zinb))` commands to compute the IRRs and the corresponding confidence intervals, respectively. The results are combined with the `cbind()` function.

```
> exp(coef(zinb))
count_(Intercept)         count_educ     count_maritals      count_female
     0.3266351837       1.0615383658       1.2167216992      1.0489635783
    count_wrkfull  zero_(Intercept)         zero_educ     zero_maritals
     1.2061628254       0.1322496364       1.0749230663      0.9891001495
      zero_female       zero_wrkfull
     1.2213384461       0.0002522989

> exp(confint(zinb))
                          2.5 %         97.5 %
count_(Intercept)  1.835458e-01   5.812747e-01
count_educ         1.022124e+00   1.102472e+00
count_maritals     9.923939e-01   1.491758e+00
count_female       8.580655e-01   1.282332e+00
count_wrkfull      9.245002e-01   1.573638e+00
zero_(Intercept)   9.484141e-03   1.844128e+00
zero_educ          9.102592e-01   1.269374e+00
zero_maritals      3.677225e-01   2.660482e+00
zero_female        4.510252e-01   3.307282e+00
zero_wrkfull       8.215654e-19   7.747980e+10

> cbind(exp(coef(zinb)), exp(confint(zinb)))
                                         2.5 %         97.5 %
count_(Intercept)   0.3266351837   1.835458e-01   5.812747e-01
count_educ          1.0615383658   1.022124e+00   1.102472e+00
count_maritals      1.2167216992   9.923939e-01   1.491758e+00
count_female        1.0489635783   8.580655e-01   1.282332e+00
count_wrkfull       1.2061628254   9.245002e-01   1.573638e+00
zero_(Intercept)    0.1322496364   9.484141e-03   1.844128e+00
zero_educ           1.0749230663   9.102592e-01   1.269374e+00
zero_maritals       0.9891001495   3.677225e-01   2.660482e+00
zero_female         1.2213384461   4.510252e-01   3.307282e+00
zero_wrkfull        0.0002522989   8.215654e-19   7.747980e+10
```

The standard errors of the IRRs are computed with the `exp(coef(zinb))*sqrt(diag(vcov(zinb)))` command.

```
> exp(coef(zinb))*sqrt(diag(vcov(zinb)))
count_(Intercept)        count_educ     count_maritals      count_female
      0.096055754       0.020492340        0.126513598       0.107508696
    count_wrkfull  zero_(Intercept)          zero_educ     zero_maritals
      0.163663375       0.177802801        0.091191896       0.499336677
      zero_female      zero_wrkfull
      0.620762554       0.004294075
```

We can use the nagelkerke(zinb) command to compute the three types of pseudo R^2 statistics and the likelihood ratio test statistic for the multiple-predictor model. The output is omitted here.

We use the Vuong test to compare the zero-inflated negative binomial regression model and the negative binomial regression model. In the vuong(zinb,nbr) command, zinb and nbr are the zero-inflated negative binomial regression model and the negative binomial regression model, respectively. The following output is displayed.

```
> vuong(zinb, nbr)
Vuong Non-Nested Hypothesis Test-Statistic:
(test-statistic is asymptotically distributed N(0,1) under the
null that the models are indistinguishible)
-------------------------------------------------------------

              Vuong z-statistic             H_A        p-value
Raw                   1.6470053  model1 > model2    0.04977848
AIC-corrected         0.2044316  model1 > model2    0.41900814
BIC-corrected        -3.2610737  model2 > model1    0.00055496
```

The Vuong test $z = 1.647$, $p = .05$; the AIC-corrected test $z = .204$, $p = .419$. The results of both tests are not significantly different from zero, so we conclude that there is no significant difference between the zero-inflated negative binomial regression model and the negative binomial regression model. We prefer the negative binomial regression model since it is more parsimonious than the zero-inflated model. In addition, the BIC-corrected test $z = - 3.261$, $p < .001$, which suggests that the negative binomial regression model be preferred.

We also use the Vuong test to compare the negative binomial regression model and the simplified zero-inflated Poisson regression model. In the vuong(nbr,zip2) command, nbr and zip2 are the negative binomial regression model and the simplified zero-inflated Poisson regression model, respectively. The following output is displayed.

```
> vuong(nbr, zip2)
Vuong Non-Nested Hypothesis Test-Statistic:
(test-statistic is asymptotically distributed N(0,1) under the
null that the models are indistinguishable)
-------------------------------------------------------------

              Vuong z-statistic             H_A      p-value
Raw                    1.996957  model1 > model2   0.0229149
AIC-corrected          2.267855  model1 > model2   0.0116690
BIC-corrected          2.918638  model1 > model2   0.0017578
```

The Vuong test $z = 1.997$, $p < .05$, which indicates that the Vuong z statistic is significantly different from zero, so we prefer the negative binomial regression model to the simplified zero-inflated Poisson regression model.

9.7 MAKING PUBLICATION-QUALITY TABLES

9.7.1 Presenting the Results Using the stargazer Package

We can use the stargazer package (Hlavac, 2018) to make a table containing the results of the fitted negative binomial model with the glm.nb() function. Since the package has been installed in earlier chapters, we only need to load the package by typing library(stargazer). After fitting the negative binomial model nbr, we use the command as follows: stargazer(nbr, type = "text", align = TRUE, out = "nbrmod.txt"). In the stargazer() function, we first specify the model object to be presented and then the type of table. The option type = "text" specifies the table type and the align = TRUE option aligns the results of the model. The out = "nbrmod.txt" argument saves the output named nbrmod.txt.

```
> # Presenting the results with stargazer()
> library(stargazer)
> stargazer(nbr, type = "text", align = TRUE, out = "nbrmod.txt")


==========================================
                    Dependent variable:
                    ----------------------
                          vistzoo
------------------------------------------
educ                      0.050***
                          (0.017)

maritals                  0.191**
                          (0.093)

female                    0.024
                          (0.094)

wrkfull                   0.535***
                          (0.095)

Constant                 -1.309***
                          (0.246)

------------------------------------------
Observations                902
Log Likelihood           -1,095.007
theta                  1.583*** (0.251)
Akaike Inf. Crit.        2,200.013
==========================================
Note:           *p<0.1; **p<0.05; ***p<0.01
```

We can also create the table in HTML format and copy it into Microsoft Word. The command is as follows: stargazer(nbr, type = "html", align = TRUE, out = "nbrmod.htm"). It produces Table 9.1, as shown here in its original format, presenting the results of the multiple-predictor negative binomial regression model.

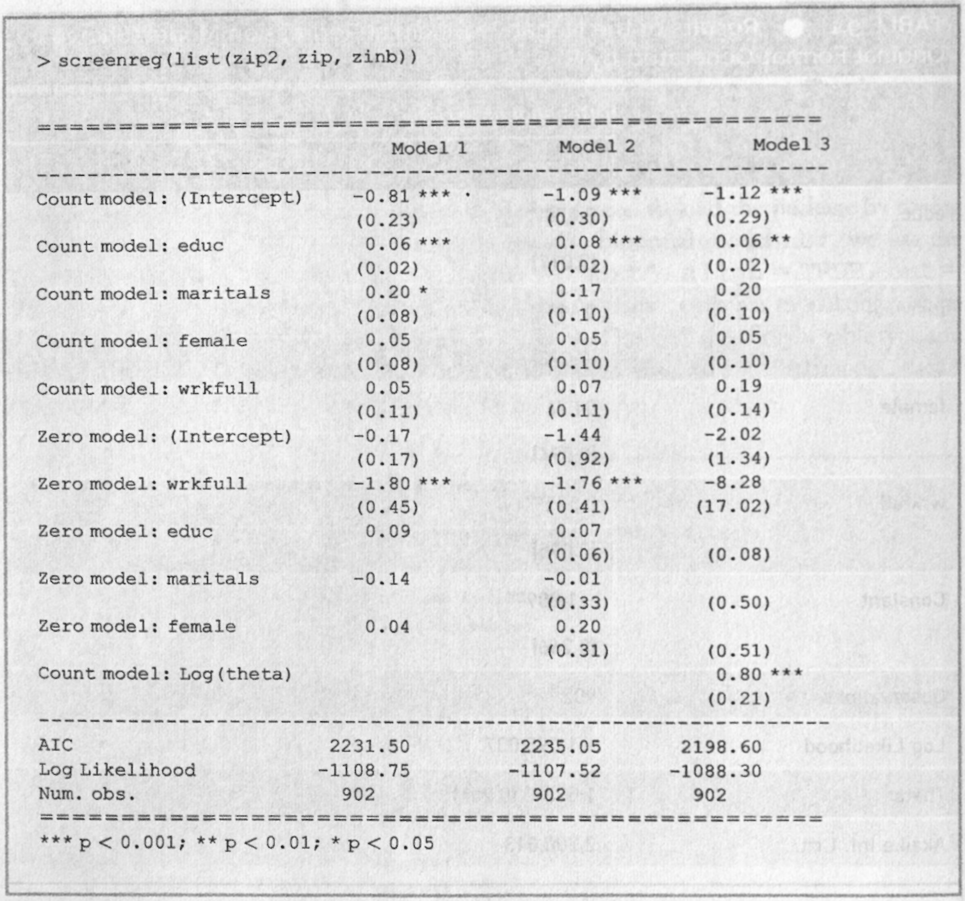
TABLE 9.1 ● Results of the Negative Binomial Regression Model (Shown in Original Format Generated by R)

	Dependent variable:
	vistzoo
educ	0.050^{***}
	(0.017)
maritals	0.191^{**}
	(0.093)
female	0.024
	(0.094)
wrkfull	0.535^{***}
	(0.095)
Constant	-1.309^{***}
	(0.246)
Observations	902
Log Likelihood	−1,095.007
Theta	1.583^{***} (0.251)
Akaike Inf. Crit.	2,200.013

*$p < 0.1$
**$p < 0.05$
***$p < 0.01$.

9.7.2 Presenting the Results Using the texreg Package

The results of the reduced zero-inflated Poisson model, the full zero-inflated Poisson model, and the zero-inflated negative binomial regression model can also be displayed in a table by using the `screenreg()` and `htmlreg()` functions from the `texreg` package (Leifeld, 2013). Since the package has been installed in earlier chapters, we only need to load the package by typing `library(texreg)`. We use the following command to display the results: `screenreg(list(zip2, zip, zinb))`. In the `screenreg()` function, we specify the three model objects to be presented with the `list()` function.



```
> screenreg(list(zip2, zip, zinb))

=================================================================
                             Model 1        Model 2       Model 3
-----------------------------------------------------------------
Count model: (Intercept)     -0.81 ***      -1.09 ***     -1.12 ***
                             (0.23)         (0.30)        (0.29)
Count model: educ             0.06 ***       0.08 ***      0.06 **
                             (0.02)         (0.02)        (0.02)
Count model: maritals         0.20 *         0.17          0.20
                             (0.08)         (0.10)        (0.10)
Count model: female           0.05           0.05          0.05
                             (0.08)         (0.10)        (0.10)
Count model: wrkfull          0.05           0.07          0.19
                             (0.11)         (0.11)        (0.14)
Zero model: (Intercept)      -0.17          -1.44         -2.02
                             (0.17)         (0.92)        (1.34)
Zero model: wrkfull          -1.80 ***      -1.76 ***     -8.28
                             (0.45)         (0.41)       (17.02)
Zero model: educ              0.09           0.07
                                            (0.06)        (0.08)
Zero model: maritals         -0.14          -0.01
                                            (0.33)        (0.50)
Zero model: female            0.04           0.20
                                            (0.31)        (0.51)
Count model: Log(theta)                                    0.80 ***
                                                          (0.21)
-----------------------------------------------------------------
AIC                        2231.50        2235.05       2198.60
Log Likelihood            -1108.75       -1107.52      -1088.30
Num. obs.                   902            902           902
=================================================================
*** p < 0.001; ** p < 0.01; * p < 0.05
```

To create the table in the HTML format and copy it into Microsoft Word, we use the command as follows: `htmlreg(list(zip2, zip, zinb), file = "zipzinb.doc", doctype = TRUE, html.tag = TRUE, head.tag = TRUE)`. The table is omitted here.

9.8 REPORTING THE RESULTS

Reporting the results for the negative binomial regression is similar to that used for Poisson regression. The only difference is that you need to report the additional dispersion parameter. The following are the generic guidelines for reporting the results. You may need to adjust your writing since your discipline or journals may have different requirements.

First, describe the negative binomial regression model, the count response variable and independent variables, and your research hypothesis or the purpose of your study. Include a couple of sentences justifying your use of this model for the analysis.

Second, report the likelihood ratio test statistic for the model and the associated p value, followed by the interpretation on whether the fitted model is better than the null model. If more than one model is developed, then compare models using likelihood ratio test statistics and/or the AIC and BIC statistics.

Third, report the parameter estimates for the predictor variables, their standard errors, the associated p values, and the dispersion parameter in a table. In addition, report the incidence rate ratio for each predictor in the table or text and interpret the results. The following is an example of summarizing the results for the negative binomial regression model illustrated previously.

The negative binomial regression analysis was conducted to predict the count outcome variable, the number of zoo visits in a year, from a set of predictor variables, such as marital status, years of education, gender, and working status. The negative binomial regression model was fitted since the response variable was a count of the number of visits to a zoo in a year and there was overdispersion in the Poisson regression model.

The likelihood ratio test for the fitted model $\chi^2_{(4)} = 54.999$, $p < .001$, indicates that the full model with the four predictors provides a better fit than the null model with no independent variables in predicting the count response variable.

The likelihood ratio test is used to test whether the dispersion parameter is significantly different from zero by comparing the Poisson regression model and the negative binomial regression model. The likelihood ratio test, $\chi^2_{(1)} = 88.79$, $p < .001$, which indicates that the dispersion parameter α is significantly different from zero, so the use of the negative binomial regression model is justified.

Table 9.1 displays the parameter estimates for the multiple-predictor negative binomial regression model. The results can be interpreted in terms of the incidence rate ratios which are the exponentiated coefficients.

In this model, for `educ`, the incidence rate ratio is 1.051. The result indicates that for a one-unit increase in education the expected number of visits to a zoo increases by 5.1%.

For `maritals`, the incidence rate ratio is 1.211, which indicates that expected number of visits to a zoo for the married is 1.211 times as high as that for the unmarried when holding other predictors constant.

The incidence rate ratio for `wrkfull` can be interpreted in the similar way. The incidence rate ratio is 1.707, which indicates that the expected number of visits to a zoo for those working full time is 1.707 times as high as that for those not working full time when holding the other predictors constant.

With regard to `female`, the incidence rate ratio is 1.024, which is not significant (see the associated $p = .797$ in the coefficients table). This indicates that being female does not impact the expected number of visits to a zoo.

9.9 SUMMARY OF R COMMANDS IN THIS CHAPTER

```
# Chap 9 R Script
# Remove all objects
rm(list = ls(all = TRUE))

# The following user-written packages need to be installed first by using
install.packages(" ") and then by loading it with library()
# library(MASS)
# library(VGAM)                    # It is already installed for Chapter 4
# library(rcompanion)              # It is already installed for Chapter 3
# library(margins)                 # It is already installed for Chapter 3
# library(ggeffects)               # It is already installed for Chapter 2
# library(texreg)                  # It is already installed for Chapter 4
# library(pscl)

# Import the count dataset
library(foreign)
count <- read.dta("C:/CDA/count.dta")

# Convert variables from integer to numeric so they will work well with ggpredict()
count$educ <- as.numeric(count$educ)
count$wrkfull <- as.numeric(count$wrkfull)
count$maritals <- as.numeric(count$maritals)

attach(count)
str(count)

# Negative binomial regression model with glm.nb() in MASS
library(MASS)
nbr <- glm.nb(vistzoo ~ educ + maritals + female + wrkfull, data = count)
summary(nbr)
alpha<-1/1.583
alpha
```

```
coef(nbr)
confint(nbr)
exp(coef(nbr))
exp(confint(nbr))
cbind(exp(coef(nbr)), exp(confint(nbr)))
exp(coef(nbr))*sqrt(diag(vcov(nbr)))

# marginal effects
library(margins)
marg.nbr <- margins(nbr)
summary(marg.nbr)

# Testing the overall model using the likelihood ratio test
nbr.0 <- glm.nb(vistzoo ~ 1, data = count)
summary(nbr.0)
anova(nbr.0, nbr, test = "Chisq")
anova(nbr, update(nbr, ~1), test = "Chisq")

# Pseudo R2 with nagelkerke()
library(rcompanion)
nagelkerke(nbr)

# Pseudo R2 with equations
LLM1 <- logLik(nbr)
LL0 <- logLik(nbr.0)
McFadden1 <- 1-(LLM1/LL0)
McFadden1
CS1 <- 1-exp(2*(LL0-LLM1)/902)
CS1
NG1 <- CS1/(1-exp(2*LL0/902))
NG1

# AIC and BIC Statistics
AIC(nbr)
BIC(nbr)

PR.2 <- glm(vistzoo ~ educ + maritals + female + wrkfull, family = poisson, data =
count)
AIC(PR.2, nbr)
BIC(PR.2, nbr)

# Presenting the results with stargazer()
library(stargazer)
stargazer(nbr, type="text", align=TRUE, out="nbrmod.txt")
stargazer(nbr, type="html", align=TRUE, out="nbrmod.htm")

# Predicted counts with ggpredict() in ggeffects
library(ggeffects)
nbr.ed <- ggpredict(nbr, terms = "educ[12, 14, 16]")
nbr.ed
plot(nbr.ed)

nbr.ew <- ggpredict(nbr, terms = c("educ[12, 14, 16]", "wrkfull"))
nbr.ew
plot(nbr.ew)
```

```
# Negative binomial regression model with vglm() in VGAM
library(VGAM)
nb.v <- vglm(vistzoo ~ educ + maritals + female + wrkfull, family = negbinomial,
data=count)
summary(nb.v)
coef(nb.v, matrix = TRUE)
confint(nb.v, matrix = TRUE)
exp(coef(nb.v, matrix = TRUE))
exp(confint(nb.v, matrix = TRUE))
exp(coef(nb.v))*sqrt(diag(vcov(nb.v)))
exp(.45953)
alpha.v <- 1/1.583
alpha.v
nagelkerke(nb.v)
AIC(nb.v)
BIC(nb.v)

# Model comparison with the likelihood ratio test
library(lmtest)
lrtest(PR.2, nbr)

# ZIP model with zeroinfl() in pscl
# install.packages("pscl")
library(pscl)
zip <- zeroinfl(vistzoo ~ educ + maritals + female + wrkfull | educ+ maritals +
female + wrkfull, data=count)
summary(zip)
coef(zip)
confint(zip)
exp(coef(zip))
exp(confint(zip))
cbind(exp(coef(zip)), exp(confint(zip)))
exp(coef(zip))*sqrt(diag(vcov(zip)))

# Pseudo R2 with nagelkerke()
library(rcompanion)
nagelkerke(zip)

# Vuong test
vuong(zip, PR.2)

# ZIP reduced model
zip2 <- zeroinfl(vistzoo ~ educ + maritals + female + wrkfull | wrkfull,
data=count)
summary(zip2)
lrtest(zip2, zip)
# Zero-inflated NB model with zeroinfl() in pscl
Zinb <- zeroinfl(vistzoo ~ educ + maritals + female + wrkfull | educ + maritals +
female + wrkfull, dist="negbin", data=count)
summary(zinb)
coef(zinb)
confint(zinb)
exp(coef(zinb))
exp(confint(zinb))
cbind(exp(coef(zinb)), exp(confint(zinb)))
exp(coef(zinb))*sqrt(diag(vcov(zinb)))
nagelkerke(zinb)
```

```
vuong(zinb, nbr)
vuong(nbr, zip2)

# Presenting the results with textreg
library(texreg)
screenreg(list(zip2, zip, zinb))
htmlreg(list(zip2, zip, zinb), file="zipzinb.doc", doctype=TRUE, html.tag=TRUE,
head.tag=TRUE)

detach(count)
```

Glossary

Overdispersion occurs when the variance of the count response variable is greater than the mean.

The NB2 negative binomial model is defined when the variance has a quadratic form, $\mu + \alpha\mu^2$.

The negative binomial regression model relaxes the equality of the mean and the variance assumption and allows overdispersion.

The variance of the response variable in the negative binomial regression model is a function of the mean μ and a dispersion parameter α.

The zero-inflated models address the issue of excessive zero counts. The zero-inflated model can be applied to both Poisson regression and negative binomial regression. The former is referred to as the **zero-inflated Poisson regression model** and the latter is called the **zero-inflated negative binomial regression model**.

Exercises

Use the `rwm1984` data (Hilbe, 2014) available at **https://edge.sagepub.com/liu1e** for the following problems.

1. Conduct an analysis for a negative binomial regression model and estimate the count response variable `docvis` (the number of visits to a doctor in a year) from the two predictor variables, `outwork` (1 = not working and 0 = working) and `female` (1 = female and 0 = male).
2. Interpret the likelihood ratio test for the overall model.
3. List three measures of pseudo R^2 and the AIC and BIC statistics.
4. In the regression table, identify the coefficients for the predictor variables `outwork` and `female`. Are they both statistically significant?
5. Identify the dispersion parameter labeled `Theta` and compute the dispersion parameter α.
6. Compute the IRRs and interpret the IRR for the predictor variable `female`.
7. Make a publication-quality table containing the estimated coefficients.
8. Write a report to summarize the results from the output.