PSM II Tutorial 9: Regression models for count data

1 Introduction

This week we will use data collected by Fisman and Miguel on how many times diplomats working at the United Nations headquarters in New York City were issued with parking fines. Diplomats are immune from prosecution for some crimes, including parking violations, so diplomats in many countries park in places where they are not allowed to. We will use this dataset to see if there is any relation between the number of violations a country's diplomats were issued and the level of corruption in that country.

This tutorial will use the easystats collection of R packages for working with regression output and the tidyverse collection of packages for wrangling data.

To get started, download the `un_parking_violations.csv` data file from Moodle. Put this file in the folder on your computer that you are using for this module, then create a new R script in RStudio (`File > New File > R Script`) and save it as `tutorial_09.R` in the same folder as the data file.

Start your script file by loading the packages we will need:

```
library(easystats)
library(tidyverse)
```

You can now load the data using the `read_csv()` function from the readr package:

```
parking_violations <- read_csv("un_parking_violations.csv")
```

2 Get to know the data

Start by looking at the different columns in the data:

country_code
a three-letter code corresponding to a country (character)

country_name
the name of the country (character)

violations
the number of parking fines issued to diplomats from the country (numeric)

diplomats
the number of diplomats from the country working at the UN (numeric)

gdp
the gross domestic product of the country (numeric, in US dollars)

corruption
a measure of corruption in the country, ranging from 0 (lowest corruption) to 10 (highest corruption) (numeric)

Create a histogram of the `violations` column in the data to see how this variable is distributed.

```
ggplot(parking_violations, aes(x = violations)) + geom_histogram()
```

Question 1: what do you notice about this distribution? Is it closer to a normal distribution or the sort of distribution we would expect for a count variable?

We are primarily interested in understanding if there is a relationship between corruption in a country and the number of parking tickets issued to that country's diplomats at the UN. In order to understand that relationship, we need to account for other explanatory variables that might influence the number of parking tickets issued to each country's diplomats.

Question 2: why is it important to account for the number of diplomats in each country's UN delegation?

3 Fit a Poisson model

Since our outcome variable is a *count* of parking violations, we can start by using Poisson regression to model the relationship with corruption.

```
parking_model_poisson <- glm(
  violations ~ diplomats + corruption,
  data = parking_violations,
  family = "poisson"
)
```

We can use `model_parameters(parking_model_poisson, exponentiate = TRUE)` to look at the incidence rate ratios for the relationships in this

model, and use `report(parking_model_poisson, exponentiate = TRUE)` to create an automatic summary of the model.

Question 3: how would you interpret these coefficients and the corresponding confidence intervals?

We know that in order to rely on the results produced by a Poisson model, it is necessary for the variance of the outcome variable to be similar to the mean (called the *equidispersion assumption*). We can test whether this assumption is true in our model using an over-dispersion test:

```
check_overdispersion(parking_model_poisson)
```

Question 4: based on this test, is the equidispersion assumption true for our model? Is the outcome variable suitable for Poisson regression?

4 Fit a negative-binomial model

In cases where a count outcome variable is over-dispersed, it is usually better to fit a negative-binomial model instead of a Poisson model. We can do this using the `glm.nb()` function from the MASS package.

```
parking_model_nb <- MASS::glm.nb(
  violations ~ diplomats + corruption,
  data = parking_violations
)
```

Before going any further, let's check the assumptions of this type of regression model:

```
check_collinearity(parking_model_nb)
check_outliers(parking_model_nb)
check_overdispersion(parking_model_nb)
```

These checks show that there might be a problem with one row in the data (an outlier) having too much influence on the model. We can look into this further by producing a plot of influential observations:

```
plot(check_outliers(parking_model_nb))
```

This shows that we probably don't need to worry in this case: all the observations are within the dashed lines, and the solid line of best fit is largely straight and horizontal.

As in previous models, we can use the `model_parameters()` and `report()` functions to inspect the coefficients produced by this model.

Question 5: what is your interpretation of each coefficient and the associated p-value?

We can use a likelihood-ratio test to see if the Poisson and negative-binomial models we have created are different in how well they fit the data. We can also compare AIC values to see which model fits the data best.

```
compare_performance(parking_model_poisson, parking_model_nb, metrics = "common")
test_likelihoodratio(parking_model_poisson, parking_model_nb)
```

Question 6: which model fits the data better? Is that difference significant?

We can compare the coefficients across the two models using the `compare_parameters()` function from the parameters package:

```
compare_parameters(parking_model_poisson, parking_model_nb, exponentiate = TRUE)
```

Question 7: Do the confidence intervals produced by each model for an explanatory variable overlap? Are the coefficients for `diplomats` significantly different from one another? Treat values as significantly different if p<0.05. Remember what we learned in previous weeks about the relationship between p-values and confidence intervals.

5 Fit a linear model

To demonstrate why it is important to model count variables with models that understand the properties of counts, we can fit a linear model to the same data and see if the assumptions of a linear model are met.

```
parking_model_lm <- lm(
  violations ~ diplomats + corruption,
  data = parking_violations
)
```

We can check each of the assumptions of linear regression in turn:

Assumption	Code
Homoskedasticity	`check_heteroskedasticity()`
Normality of residuals	`check_normality()`
Multicollinearity	`check_collinearity()`
Influential outliers	`check_outliers()`
Independent residuals	`check_autocorrelation()`

Question 8: are the assumptions of linear regression true in the case of this model? Can we trust the results produced by this model?