tutorial 8

Data taken from: https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/E9N6PH
We take this data and try to predict if an individual has voted for Biden or Trump in the 2020 election based on sex and gender.

To create a simplified model of Nate Cohn, I filter the variables voted_for, gender, and educ for the regression. The following is a glimpse of the data:

```
# A tibble: 43,554 x 3
   voted_for gender  educ
   <fct>     <chr>  <int>
 1 Trump     Male       4
 2 Biden     Female     5
 3 Biden     Female     5
 4 Trump     Male       3
 5 Trump     Female     3
 6 Trump     Female     2
 7 Biden     Female     5
 8 Biden     Female     5
 9 Biden     Female     5
10 Biden     Male       3
# i 43,544 more rows
```

We are interested in predicting the vote of an individual based on gender and education. The variable is modeled by the following logistic regression:

$$
\begin{aligned}
y_i|\pi_i &\sim \text{Bern}(\pi_i) \\
\text{logit}(\pi_i) &= \beta_0 + \beta_1 \times \text{sex}_i + \beta_2 \times \text{education}_i \\
\beta_0 &\sim \text{Normal}(0, 2.5) \\
\beta_1 &\sim \text{Normal}(0, 2.5) \\
\beta_2 &\sim \text{Normal}(0, 2.5)
\end{aligned}
$$

I compute the logistic regression with two methods:

```r
model <- glm(voted_for ~ gender + educ, data = ces2020, family = binomial)
summary(model)
```

```

Call:
glm(formula = voted_for ~ gender + educ, family = binomial, data = ces2020)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)    
(Intercept) -0.402851   0.028331  -14.22   <2e-16 ***
genderMale  -0.458765   0.020206  -22.70   <2e-16 ***
educ         0.261227   0.006896   37.88   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 58734  on 43553  degrees of freedom
Residual deviance: 56895  on 43551  degrees of freedom
AIC: 56901

Number of Fisher Scoring iterations: 4
```

```

SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
Chain 1: 
Chain 1: Gradient evaluation took 0.004957 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 49.57 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1: 
Chain 1: 
Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 0.22 seconds (Warm-up)
Chain 1:                0.249 seconds (Sampling)
Chain 1:                0.469 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 4.4e-05 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.44 seconds.
Chain 2: Adjust your expectations accordingly!
Chain 2:
Chain 2:
Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 2:
Chain 2:  Elapsed Time: 0.22 seconds (Warm-up)
Chain 2:                0.205 seconds (Sampling)
Chain 2:                0.425 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 4.4e-05 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.44 seconds.
Chain 3: Adjust your expectations accordingly!
Chain 3:
Chain 3:
Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 3: 
Chain 3:  Elapsed Time: 0.204 seconds (Warm-up)
Chain 3:                0.256 seconds (Sampling)
Chain 3:                0.46 seconds (Total)
Chain 3: 

SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
Chain 4: 
Chain 4: Gradient evaluation took 4e-05 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.4 seconds.
Chain 4: Adjust your expectations accordingly!
Chain 4: 
Chain 4: 
Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 4: 
Chain 4:  Elapsed Time: 0.196 seconds (Warm-up)
Chain 4:                0.21 seconds (Sampling)
Chain 4:                0.406 seconds (Total)
Chain 4: 
```

The reason we choose a logistic regression model to predict a variable such as vote for Biden vs Trump is becuase the variable of interest is a binary outcome. Logistic regression is design for

	Support Biden
(Intercept)	−0.450
	(0.187)
genderMale	−0.428
	(0.131)
educ	0.262
	(0.045)
Num.Obs.	1000
R2	0.040
Log.Lik.	−656.291
ELPD	−659.2
ELPD s.e.	8.4
LOOIC	1318.5
LOOIC s.e.	16.8
WAIC	1318.5
RMSE	0.48

modelling the odds of a particular event happening, where there is are only two outcomes– in this scenario, the voter either votes for Biden or Trump. Poisson or negative binomial regression is designed for count data, where the Poisson distribution specializes in the number of times an event occurs, where the outcome is a count that can range form zero to infinity. Negative binomial regression is used when there is over dispersion in the data set, by considering the variance as a variable based on mean, and introducing an extra term that allows variance to increase with the mean. Thus, a logisitc regression is the best fitted regression model for the scenario.