

POLS6382 Quantitative Method III

Maximum Likelihood Estimation

Lab 8: Event Count Models

Ling Zhu and Emily Lee
Department of Political Science
University of Houston

2025/10/29

1. Learning Objectives

- Learn how to estimate a Poisson regression model.
- Learn how to test for over-dispersion.
- Learn how to estimate a Negative Binomial model.
- Learn how to interpret statistical results from a Poisson/Negbinomial model.

```
> rm(list=ls())
> setwd("/Users/lingzhu/Dropbox/UH Teaching/POLS6382_2025/2025 Labs/Lab 8")
> my_packages <- c("foreign", "ggplot2", "ggthemes", "dplyr", "ggpubr", "VGAM", "MASS", "AER", "psych" )
> invisible(lapply(my_packages, require, character.only = TRUE))
```

2. Data Example: Congressional Acts Overturned by the Supreme Court

The data file (`courtdata.dta`) uses data on the first through the 104th Congresses (1789-1996). The dependent variable (`nulls`) is the number of Congressional acts overturned by the Supreme Court during that Congress. The variable captures the Supreme Court's ability in exercising judicial review. We use this dataset to examine whether the Supreme Court can function as a counter-majoritarian institution. Also included in the data are four other variables:

- `congress`, a counter for the number of the Congress;
- `tenure`, a variable indicating the mean number of years served by justices sitting on the Court during that Congress;
- `unified`, a variable coded 1 if both houses of Congress are controlled by the same political party and 0 otherwise, and
- `PartyDisagreement`, an indicator of how politically "close" the Supreme Court is to Congress.

$$PartyDisagreement = |Democratic\ Percent\ in\ Congress_i - Democratic\ Percent\ on\ the\ Supreme\ Court_i| \quad (1)$$

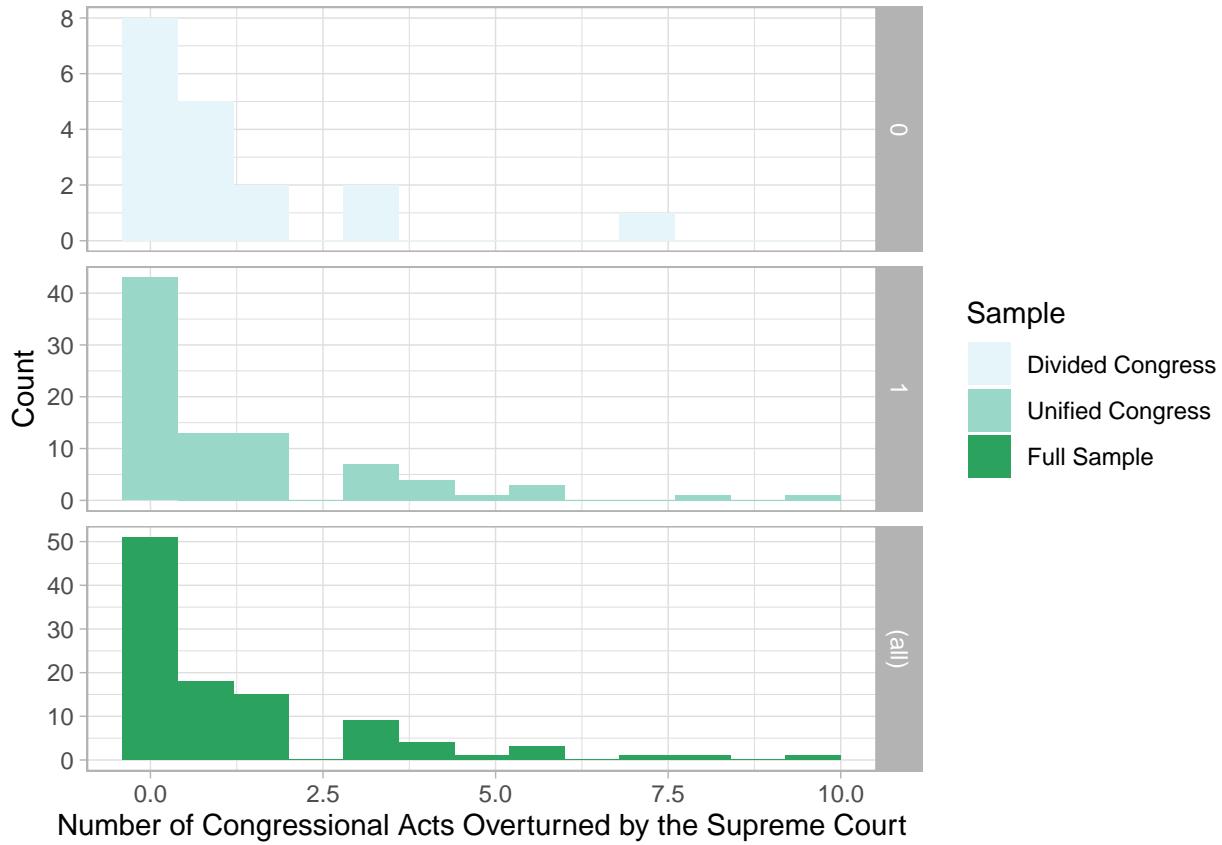
Court is a responsive institution (through the appointment and confirmation process), but that it is so at a lag, since turnover on the Court is usually slow. Because of this, the Court will be more likely to be out-of-step with the dominant political majority, and thus, more likely to find against it, as the majority that appointed and confirmed the sitting justices recedes into the past. Moreover, the Court will be most effective against a "weak" lawmaking majority (and less effective against a strong one), and that the Court will be

more likely to strike down laws when it is in political or partisan disagreement with the dominant political regime.

```
> courtdat<-read.dta("courtdata.dta")
> describe(courtdat, skew=FALSE)
```

	vars	n	mean	sd	median	min	max	range	se
nulls	1	104	1.34	1.93	1.00	0.00	10.00	10.00	0.19
congress	2	104	52.50	30.17	52.50	1.00	104.00	103.00	2.96
tenure	3	104	10.40	3.35	10.55	0.80	18.40	17.60	0.33
unified	4	104	0.83	0.38	1.00	0.00	1.00	1.00	0.04
PartyDisagreement	5	104	24.34	16.66	21.31	0.21	65.05	64.84	1.63

```
> ggplot(courtdat, aes(nulls, fill = unified))+
+   geom_histogram(binwidth = 0.8) +
+   theme_light() +
+   scale_fill_brewer(type="seq", palette = 2, name = "Sample",
+                     labels = c("Divided Congress", "Unified Congress", "Full Sample")) +
+   facet_grid(unified ~ ., margins = TRUE, scales = "free") +
+   labs(x="Number of Congressional Acts Overturned by the Supreme Court",
+        y="Count")
```



The histogram figure presents the counts of judicial overturn of Congressional acts. The first panel shows the distribution of counts under a divided Congress. The second panel shows the distribution of counts under a unified Congress. The bottom panel shows the distribution based on the full sample.

3. Estimating a Poisson Regression Model

Because the dependent variable `nulls` measures counts, we consider the PRM as a proper model specification. We use the `glm()` function to estimate a PRM. Table 1 reports findings based on the Poisson regression model. As Table 1 shows, `tenure` has a positive and significant effect on the number of Congressional acts overturned by the Supreme Court. `[PartyDisagreement]` has a significant and negative effect on the number of judicial overturns.

```
> model1<-glm(nulls~tenure+unified+PartyDisagreement, data=courtdatas, family=poisson)
> summary(model1)
```

Call:

```
glm(formula = nulls ~ tenure + unified + PartyDisagreement, family = poisson,
     data = courtdatas)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.2588	-1.3287	-0.6346	0.3974	5.0758

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.331596	0.440700	0.752	0.45179
tenure	0.073591	0.027764	2.651	0.00804 **
unified	-0.198131	0.241805	-0.819	0.41257
PartyDisagreement	-0.034160	0.006757	-5.055	4.3e-07 ***

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

(Dispersion parameter for poisson family taken to be 1)

```
Null deviance: 251.8 on 103 degrees of freedom
Residual deviance: 208.0 on 100 degrees of freedom
AIC: 357.55
```

Number of Fisher Scoring iterations: 6

4. Testing for Overdispersion

A key assumption of the PRM is `equidispersion`, that is, the equality between mean and variance. Overdispersion often occurs in empirical applications. One informal way to check for over-dispersion is to compare the mean and variance of the dependent variable. The variable `nulls` has a mean of 1.337 and a variance of 3.720. It suggests that there is likely overdispersion.

The formal way of testing for overdispersion is to consider the alternative hypothesis:

$$Var(y_i|x_i) = \mu_i + \alpha\mu_i = (1 + \alpha)\mu_i \quad (2)$$

The function `dispersiontest()` in package `AER` tests equidispersion against the alternative expressed as equation (2). The default option is to evaluate $(1 + \alpha)$. If the argument `trafo` is specified, the test is formulated using α . The two tests are equivalent. The first test shows that the true dispersion parameter $(1 + \alpha)$ is greater than 1. The estimated dispersion parameter is 2.327. The second test shows that α is greater than 0 (estimated as 1.327). We have evidence that there is over-dispersion.

```
> # Informal test: Comparing mean and variance
> mean(courtdatas)nulls)
```

```
[1] 1.336538
> var(courtdata)nulls)
[1] 3.720594
> # Formal test:
> dispersiontest(model1)

Overdispersion test

data: model1
z = 2.0127, p-value = 0.02207
alternative hypothesis: true dispersion is greater than 1
sample estimates:
dispersion
2.326578
> dispersiontest(model1, trafo=1)
```

```
Overdispersion test

data: model1
z = 2.0127, p-value = 0.02207
alternative hypothesis: true alpha is greater than 0
sample estimates:
alpha
1.326578
```

#5. Negative Binomial Regression

When over-dispersion occurs, the negative binomial regression model (NBRM) is preferred to the PRM. In R, tools for estimating an NBRM are provided by the MASS package. With known θ , function `negative.binomial()` can be used. For unknown θ , function `glm.nb()` can be used. In addition, the Zelig provides tools to estimate both PRM and NBRM. Table 2 shows results based on the NBRM. We observe that the coefficient of Tenure is greater in the NBRM than in the PRM. As for the coefficient of Party Disagreement, the NBRM produces a coefficient with greater size than that produced by the PRM.

```
> model2<-glm.nb(nulls~tenure+unified+ PartyDisagreement, data=courtdata)
> summary(model2)
```

```
Call:
glm.nb(formula = nulls ~ tenure + unified + PartyDisagreement,
       data = courtdata, init.theta = 1.004379528, link = log)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.7195	-1.1431	-0.4364	0.2382	2.5556

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-0.109089	0.693792	-0.157	0.875059
tenure	0.108681	0.043958	2.472	0.013421 *
unified	-0.107078	0.385093	-0.278	0.780968
PartyDisagreement	-0.035210	0.009835	-3.580	0.000343 ***

```

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

(Dispersion parameter for Negative Binomial(1.0044) family taken to be 1)

Null deviance: 122.7 on 103 degrees of freedom
Residual deviance: 100.4 on 100 degrees of freedom
AIC: 319.56

Number of Fisher Scoring iterations: 1

Theta: 1.004
Std. Err.: 0.303

2 x log-likelihood: -309.557

```

We can use various methods to compare/evaluate model fit. First, deviance, log-likelihood, and AIC statistics can all be used to assess goodness of fit. For each model, we can compare the Null deviance and model deviance, which tells us how much a model is better than the null model. The following example shows that when specifying a PRM, adding variable `tenure` and `PartyDisagreement` significantly improves model fit (compared with the null model), while including `unified` does not significantly improve model fit.

Second, we can use the likelihood ratio test to compare the two model specifications: PRM v. NBRM. The LR test asks “Did the likelihood change much in one model v. the other model. (Or from the restricted model to the unrestricted model.)” We observe that the likelihood changes significantly from the PRM and NBRM. The NBRM is preferred. Third, we can also directly compare the AIC statistics produced by the two models. AIC is an index of the amount of information regarding your data lost when your model is used to describe it. A smaller AIC is better.

```
> anova(model1, test="Chisq")
```

Analysis of Deviance Table

Model: poisson, link: log

Response: nulls

Terms added sequentially (first to last)

	Df	Deviance	Resid. Df	Resid. Dev	Pr(>Chi)
NULL			103	251.79	
tenure	1	13.8785	102	237.92	0.000195 ***
unified	1	0.3927	101	237.52	0.530867
PartyDisagreement	1	29.5287	100	208.00	5.51e-08 ***

```
---
```

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

```
> anova(model2,test="Chisq")
```

Analysis of Deviance Table

Model: Negative Binomial(1.0044), link: log

Response: nulls

Terms added sequentially (first to last)

```

Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL
tenure      1    8.2148     102    114.49 0.0041551 **
unified      1    0.3007     101    114.19 0.5834483
PartyDisagreement 1   13.7830     100    100.40 0.0002052 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> lrtest(model1,model2)

```

Likelihood ratio test

```

Model 1: nulls ~ tenure + unified + PartyDisagreement
Model 2: nulls ~ tenure + unified + PartyDisagreement
#Df LogLik Df Chisq Pr(>Chisq)
1   4 -174.77
2   5 -154.78  1 39.99  2.553e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

#6. Substantive Interpretation: The Negative Binomial Model

There are various ways by which we can substantively interpret the coefficients produced by a PRM or a NBRM. Because the NBRM is preferred in this data example, we will use the NBRM to illustrate how to substantively interpret model results.

6.1 Incidence-Rate Ratio (Factor Change).

We can convert the coefficients in the NBRM using function `exp()`. The mean IIR estimates show for a one-unit change in x_k , the expected count changes by a factor of $\exp(\beta_k)$, holding all other variables constant. For example, the mean IIR associated with `tenure` is 1.115. This means that a one-year increase in `tenure`, will increase the expected number of judicial nullifications by a factor of 1.115, holding other variables constant. Or equivalently, increase the expected number of judicial nullifications by 11.5%. What about the IIR associated with `PartyDisagreement`? It is smaller than 1, meaning that the relationship is negative. A one-unit increase in the party disagreement scale will decrease the expected number of judicial nullifications by a factor of 0.9. In other words, it will decrease the expected number by 10%.

```

> # Incidence-Rate Ratio (IRR)
> (est <- cbind(Estimate = coef(model2),confint(model2)))

```

	Estimate	2.5 %	97.5 %
(Intercept)	-0.1090892	-1.55332685	1.31886887
tenure	0.1086807	0.01260981	0.20883861
unified	-0.1070779	-0.86262734	0.63224809
PartyDisagreement	-0.0352104	-0.05509979	-0.01634363

```
> exp(est)
```

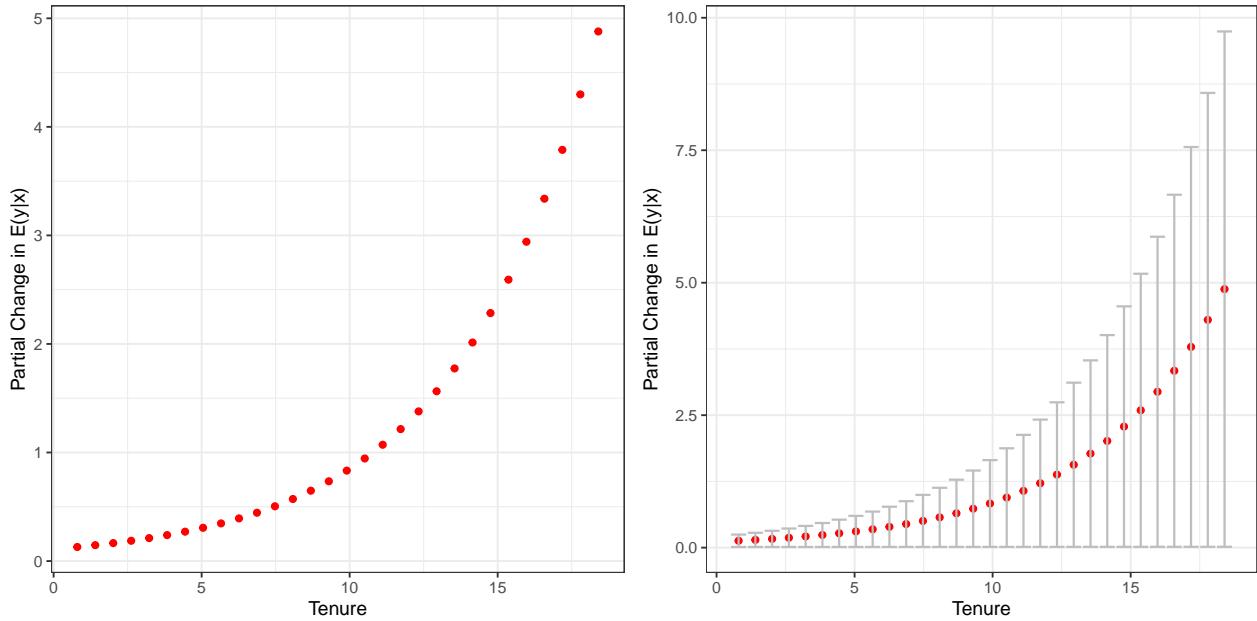
	Estimate	2.5 %	97.5 %
(Intercept)	0.8966504	0.2115430	3.7391895
tenure	1.1148063	1.0126896	1.2322461
unified	0.8984556	0.4220518	1.8818364
PartyDisagreement	0.9654023	0.9463907	0.9837892

6.2 Marginal Effects (Partial Change).

We can also use the estimated coefficients to calculate the marginal effects of x_k on $E(y|x)$. We can do so just for the mean marginal effects. We can also obtain the confidence intervals of the mean marginal effects based on the lower and upper bounds of the CIs estimated for the β coefficients.

$$\frac{\partial E(y|x)}{\partial x} = \exp(x\beta)\beta \quad (3)$$

```
> # Marginal effects (partial change) of "tenure"
> x<-seq(0.8,18.4, length=30)
> ll<-exp(x*0.01260981)*0.01260981
> ul<-exp(x*0.20883861)*0.20883861
> meaneffects=(ll+ul)/2
> meffects<-data.frame(cbind(x,meaneffects,ll,ul))
> # Mean Effects
> p1<-ggplot(meffects,aes(x=x,y=meaneffects))+
+   geom_point(color="red")+
+   theme_bw()+
+   labs(x="Tenure",y="Partial Change in E(y|x)")
>
> p2<-ggplot(meffects,aes(x=x,y=meaneffects))+
+   geom_point(color="red")+
+   geom_errorbar(aes(ymin=ll,ymax=ul),color="gray",linewidth=0.6)+
+   theme_bw()+
+   labs(x="Tenure",y="Partial Change in E(y|x)")
>
> ggarrange(p1, p2, ncol=2)
```



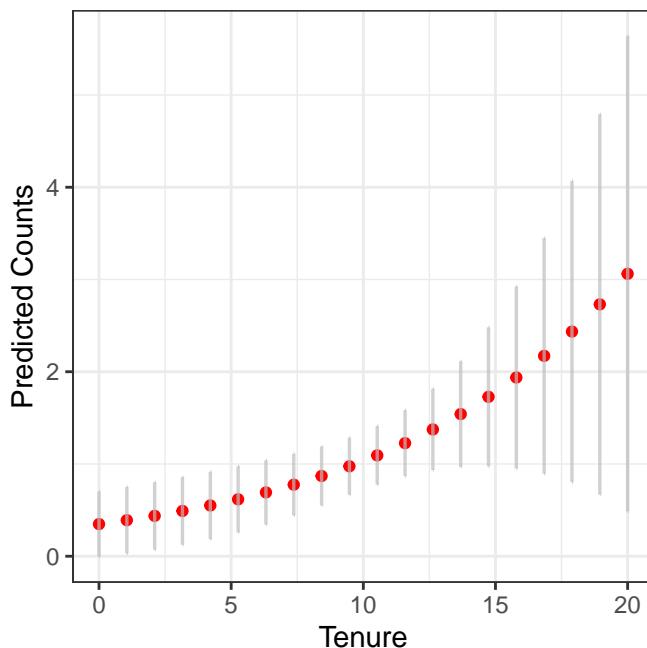
6.3 Predicted Counts.

After estimating the NBRM, we can obtain a predicted number of judicial nullifications. In-sample mean predictions can be obtained by calling out `$fitted.values`. Out-sample prediction can be produced using function `predict()`. The flowing figure shows as `tenure` increases, the predicted number of judicial nullification also increases.

```

> newdata<-data.frame(tenure=seq(0,20,length=20),
+   PartyDisagreement=mean(courtdata$PartyDisagreement),unified=mean(courtdata$unified))
> newdata1<- cbind(newdata,predict(model2, newdata, type = "response",se.fit=TRUE))
> newdata1 <- within(newdata1, {
+   LL <- fit - 2 * se.fit
+   UL <- fit + 2 * se.fit
+   meancounts <- fit
+ })
>
> ggplot(newdata1,aes(x=tenure,y=meancounts))+
+   geom_point(color="red")+
+   geom_errorbar(aes(ymin=LL,ymax=UL),color="gray",linewidth=0.6, alpha=0.7, width=0)+
+   theme_bw()+
+   labs(x="Tenure",y="Predicted Counts")

```

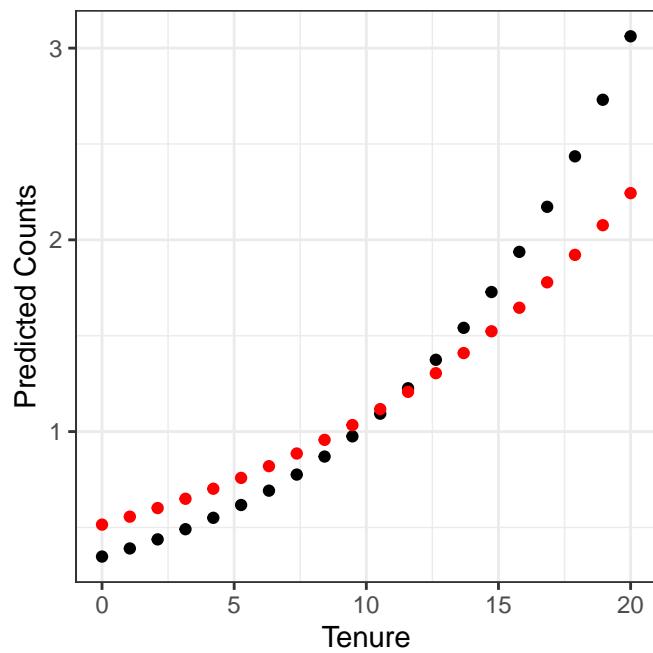


We can further compare the predicted number of judicial nullifications based on the two different model specifications: PRM (in red dots) and NBRM (in black dots). The comparison shows that, the PRM (red dots) produces greater predicted counts than those produced by the NBRM, when `tenure` is short (approximately, `tenure` < 10). When `tenure` is greater than 10, the PRM produces smaller predicted counts than the negative binomial model.

```

> newdata1$residual.scale<-NULL
> newdata1$fit<-NULL
> newdata1$se.fit<-NULL
> newdata1<- cbind(newdata1,predict(model1, newdata, type = "response",se.fit=TRUE))
> newdata1$posmean<-newdata1$fit
> newdata1$posse<-newdata1$se.fit
> ggplot(newdata1,aes(x=tenure,y=meancounts))+
+   geom_point(color=" black")+
+   geom_point(aes(x=tenure,y=posmean),color="red")+
+   theme_bw()+
+   labs(x="Tenure",y="Predicted Counts")

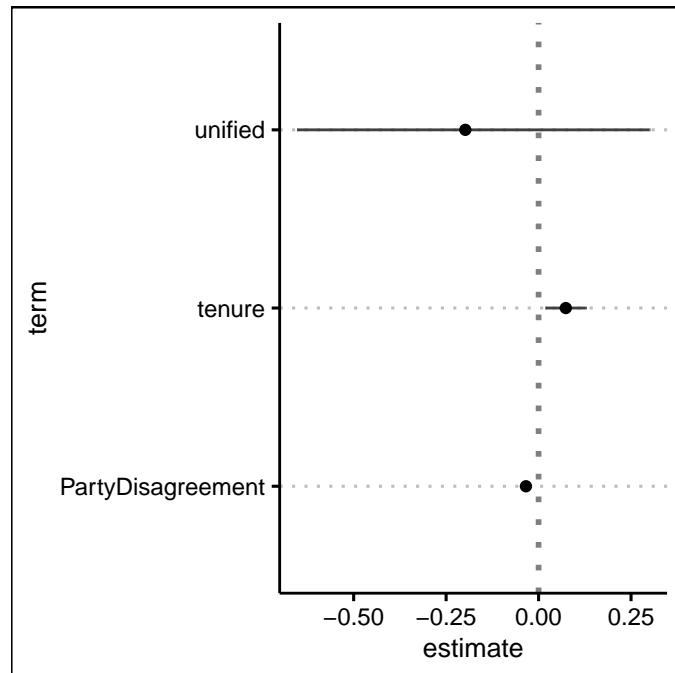
```



7. Appendix: Some Additional Visualization Examples

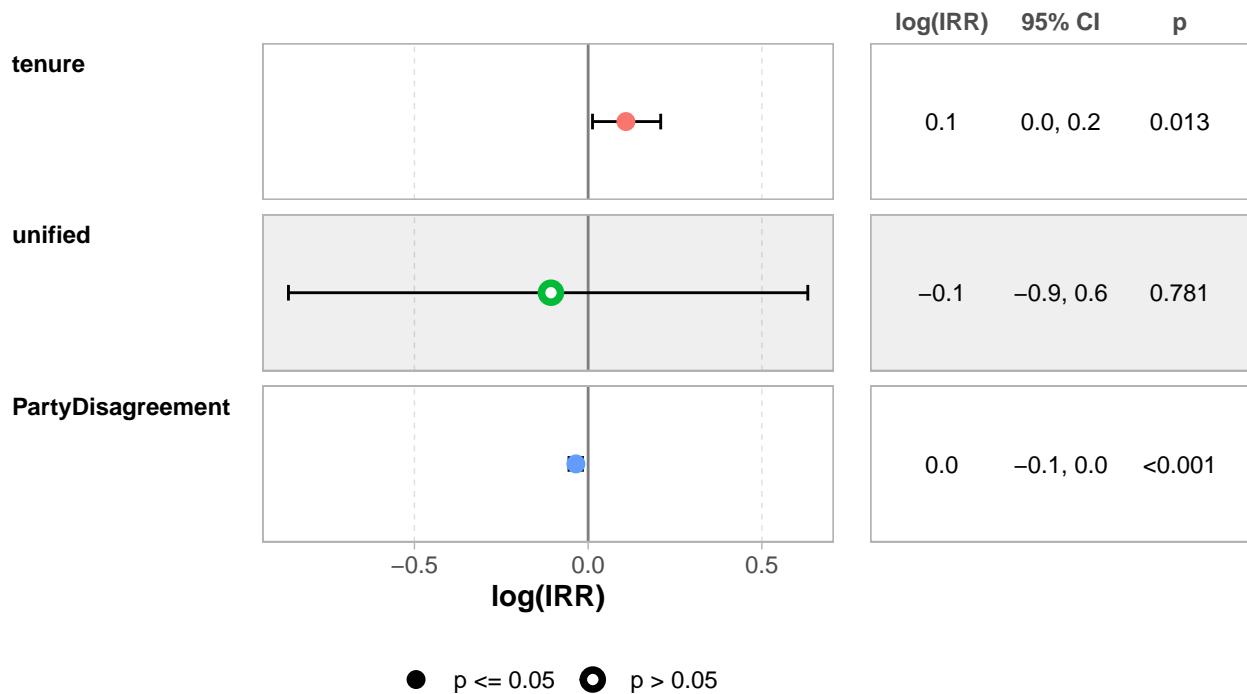
7.1 Plotting Model Coefficients

```
> library(GGally)
> ggcoef(model1, exclude_intercept = TRUE)+theme_clean()
```



7.2 Plotting Model Coefficients with a Table

```
> library(ggstats)
> ggcoef_table(model2)
```



7.3 Visualizing Predicted Counts Using Package ggeffects

```
> library(ggeffects)
> ggeffect(model2, terms = "tenure[0:20 by = 1]") %>%
+   plot() + theme_light() + labs(title = "", x = "Tenure", y = "Predicted Number of Judicial Overturns")
```

