HOW MUCH DO MODEL VIOLATIONS AFFECT REGRESSION RESULTS: A DEMONSTRATION USING SHINY

John H. Walker and Harry Wu
Statistics Department
California Polytechnic State University, San Luis Obispo, CA 93407
jwalker@calpoly.edu

Checking the model assumptions is an important part of a regression analysis. However, other than knowing that model violations are bad and how to detect them, students often learn little about the degree to which the regression results are affected. This may leave the incorrect impression that even modest violations will ruin a regression. What negative effects should we expect if these assumptions are violated, and how are these negative effects modified by sample size and the strength of the regression association? Through the use of Shiny apps, students can see for themselves the effects of different levels of non-normality and unequal variance on the widths of confidence intervals and Type I and Type II error rates in regression.

INTRODUCTION

Hypothesis tests and confidence intervals associated with many statistical models have accompanying assumptions about the model that must be met in order for the analysis to perform optimally. Instructors teach these assumptions alongside the analyses, and students often learn how to verify whether the assumptions are appropriate for the data. However, in many classes the instruction stops there, and students are left with the misleading impression that the analysis is ruined if any of the model assumptions are even slightly violated. Instead, we can use simulation to demonstrate that modest violations of model assumptions may lead to only small effects on the analysis results.

We have developed two apps using the Shiny package in *R* to demonstrate how simple linear regression results are affected by non-normality and unequal variance of the model errors. These apps allow students to change the error distribution, error variance, and sample size for the regression model to see the impact of these model violations on simulated Type I and Type II error probabilities. These apps can be used as part of in-class demonstrations, lab activities, and homework assignments to give students a better understanding of how both mild and severe assumption violations can affect regression results.

THE SIMPLE LINEAR REGRESSION MODEL

The simple linear regression model was chosen to demonstrate the effect of assumption violations because it has several different model assumptions and is commonly taught in introductory and intermediate statistics classes. The model equation is

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

where y_i and x_i are the observed data, β_0 and β_1 are the true intercept and true slope parameters, and ε_i are the model errors. The model assumptions are that:

- $E(y_i) = \beta_0 + \beta_1 x_i$ (*linearity* of the model);
- the model errors (ε_i) are *independent*;
- the model errors (ε_i) have *equal variance*;
- the model errors (ε_i) are *normally distributed*.

For ease of simulation, we chose the integers 1 through 10 for the explanatory variable, x_i. The number of replicates, which we call *n*, is an input of the simulation. This means that the total sample size, $N = 10n$. The model parameters, β_0 and β_1, are also inputs of the simulation, as are the distribution and standard deviation, σ, of the model errors. For unequal variance, the model error variance can be set as a function of *x*.

The simulations track the Type I and Type II error rates for testing the null hypothesis of H_0: $\beta_1 = 0$ versus the alternative hypothesis of H_A: $\beta_1 \neq 0$ at significance level α.

In M. A. Sorto, A. White, & L. Guyot (Eds.), *Looking back, looking forward. Proceedings of the Tenth International Conference on Teaching Statistics (ICOTS10, July, 2018), Kyoto, Japan.* Voorburg, The Netherlands: International Statistical Institute. iase-web.org

THE NON-NORMALITY APP

The Non-Normality Shiny App allows students to experiment with different error distributions and see their effects on the regression results. The *R* server code for this app was based on earlier work by Hongyan Wang (2008). Through the app interface, the user can set the error distribution, sample size, model parameters (β_0, β_1, σ), and the number of iterations. The error distribution is modeled using the Tukey-Lambda distribution. Ali and Sharma (1996) define a simplified, two parameter version of the Tukey-Lambda distribution as

$$p^a - (1-p)^b$$

where *p* is a uniform(0,1) random variable. By varying the values of *a* and *b* different shapes of error distribution can be produced. For example, $a = b = 0.135$ gives an error distribution that is approximately normal, while values of $a = 6$ and $b = 0$ give an error distribution that is extremely right skewed. To satisfy the requirement of the regression model that $E(\varepsilon_i) = 0$, the regression errors are transformed to have mean 0 and standard deviation σ. Figure 1 shows a screenshot of the user interface for the Non-Normality App.

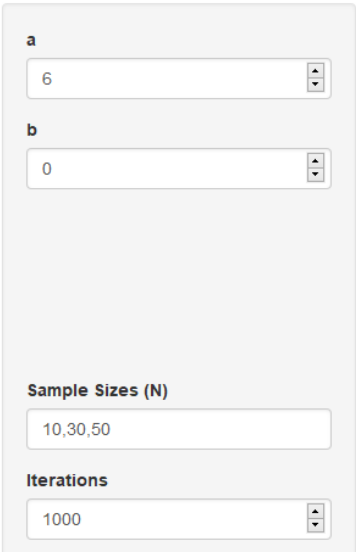

Figure 1: The User Interface of the Non-Normality App

As outputs, the Non-Normality App provides the simulated Type I or Type II error rate, the expected error rate under the normality assumption, and the p-value of a hypothesis test to detect whether the true Type I or Type II error rate differs from the expected error rate. Figure 2 shows the output from the Non-Normality App using the inputs shown in Figure 1. The sample sizes in the output refer to the total sample size, *N*, rather than the number of replicates, *n*, at each *x* value.

Type I Error Simulation

a	b	N	iters	typeI	se	alpha	zscore	pvalue
6.00	0.00	10.00	1000.00	0.03	0.01	0.05	-2.76	0.01
6.00	0.00	30.00	1000.00	0.05	0.01	0.05	-0.58	0.56
6.00	0.00	50.00	1000.00	0.05	0.01	0.05	0.29	0.77

Type II Error Simulation

a	b	N	iters	typeII	se	beta	zscore	pvalue
6.00	0.00	10.00	1000.00	0.66	0.01	0.77	-9.06	0.00
6.00	0.00	30.00	1000.00	0.33	0.02	0.38	-2.81	0.00
6.00	0.00	50.00	1000.00	0.17	0.01	0.15	1.79	0.07

Figure 2: Output of the Non-Normality App Using the Inputs from Figure 1

The output in Figure 2 shows that for the smallest sample size of $N = 10$, the Type I and Type II error rates are significantly smaller than expected with p-values of 0.01 or less. As the sample size increases, both the Type I and Type II error rates move closer to their expected values. At a total sample size of $N = 50$, the simulated Type II error rate of 0.17 is slightly higher than its expected rate of 0.15 (p-value = 0.07), but the difference is not statistically significant.

THE UNEQUAL VARIANCE APP

The Unequal Variance Shiny App simulates regressions with different degrees of unequal error variance. The user can set the number of replicates (n, where the total sample size $N = 10n$), the model parameters (β_0, β_1, σ), and the number of iterations. The sample size may be distributed evenly or unevenly among the values $x = 1$ through $x = 10$ by changing a simulation parameter called *n.state*. The amount of unequal variance in the model is controlled by a simulation parameter called *sigma.state*. By altering the values of *n.state* and *sigma.state* many different forms of unequal variance can be modeled. Figure 3 shows the user interface of the Unequal Variance App.

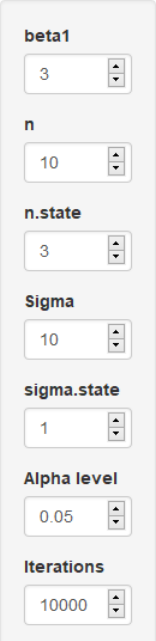

Figure 3: The User Interface of the Unequal Variance App

Balanced vs. Unbalanced Data

As described earlier, the explanatory variable data, x_i, is a list of integers from 1 to 10 that is replicated n times. This leads to a balanced data set with n replicates at each value of x. To make an unbalanced data set, the app takes a vector $\boldsymbol{v} = [-2, -2, -1, -1, 0, 0, 1, 1, 2, 2]$ and adjusts the number of replicates at each x value by $n.\text{state} \times \boldsymbol{v}$. Values of *n.state* > 0 will make the number of replicates increase as x increase. Values of *n.state* < 0 will make the number of replicates decrease as x increases. For example for a base value of $n = 5$ replicates at each value of x:

- if *n.state* = 1, the sample sizes for $x = 1, \ldots, 10$ will be [3, 3, 4, 4, 5, 5, 6, 6, 7, 7];
- if *n.state* = -2, the samples sizes for $x = 1, \ldots, 10$ will be [9, 9, 7, 7, 5, 5, 3, 3, 1, 1];
- if *n.state* = 0, the data is balanced with 5 replicates at each x value.

Changing the Pattern of Unequal Variance

The parameter *sigma.state* controls the pattern of unequal variance in the simulation. The *sigma.state* is a non-negative value that serves as an exponent for the standard deviation of the model errors. This standard deviation is computed using the formula

$$\sigma(\varepsilon_i) = \sigma x_i^{sigma.state}$$

where σ is the model standard deviation under homoscedasticity, x_i are the integers from 1 to 10, and *sigma.state* is the exponent of x_i. When *sigma.state* is zero, the model errors have equal variance, σ^2. As *sigma.state* increases, the heteroscedasticity of the model errors increases. For example for σ = 10:

- if *sigma.state* = 0.25, $\sigma(\varepsilon_i)$ = [10, 11.89, 13.16, 14.14, 14.95, 15.65, 16.27, 16.82, 17.32, 17.78] at *x* = 1 to 10;
- if *sigma.state* = 1, $\sigma(\varepsilon_i)$ = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100] at *x* = 1 to 10.
- if *sigma.state* = 0, $\sigma(\varepsilon_i)$ = [10, 10, 10, 10, 10, 10, 10, 10, 10, 10] at *x* = 1 to 10.

Output

As output, the Unequal Variance App provides the simulated Type I or Type II error rate, the expected Type I or Type II error rate under equal variance, a $1-\alpha$ confidence interval for the true error rate, and a $1-\alpha$ confidence interval for the difference between the true and expected error rates. The confidence intervals are computed using the exact binomial test procedure (*binom.test*) in the *R* package *stats*. Figure 4 shows the output of the Unequal Variance App using the inputs shown in Figure 3.

Power	ExpectedPower	L.Power	U.Power	Power.diff	L.Power.diff	U.Power.diff
0.16	0.19	0.15	0.17	-0.03	-0.03	-0.02

Replicates	Sigmas	PooledSigma
4, 4, 7, 7, 10, 10, 13, 13, 16, 16	10, 20, 30, 40, 50, 60, 70, 80, 90, 100	72.92

Figure 4: Output of the Unequal Variance App Using the Inputs from Figure 3

The output in Figure 4 indicates that the simulated power is 0.16, and the expected power under equal variance is 0.19. The 95% confidence interval of the difference between the simulated and expected power goes from -0.03 to -0.02 indicating that for these parameter inputs, the actual power of the test is lower than expected. While it is statistically significant, a 2 to 3 percentage point difference may not be of practical significance. The output also displays the replicate and unequal variance patterns that result from the input values of *n.state* and *sigma.state*.

CONCLUSION

The purpose of these apps is to give students in regression classes easy tools to investigate the effects of non-normality and unequal variance on the results of regression inference. Instructors can develop assignments that ask students to explore how the effects of these model departures change based on sample size, data balance, and the degree of non-normality or heteroscedasticity. For example, through simulation students could discover that the effects of unequal error variance are less noticeable if the data are balanced, but grow more severe for unbalanced data.

The original versions of these apps were developed as a student senior project (Wu, 2017). Future improvements are planned to the user interface, graphical output, and batch processing. The apps are available at the Cal Poly Shiny website (*https://statistics.calpoly.edu/shiny*).

REFERENCES

Ali, M. M. and Sharma, S. C. (1996). Robustness to Non-Normality of Regression F tests. *Journal of Econometrics*, *71*, 175-205.

Wang, Hongyan (2008). *Robustness to Non-Normality of the Regression t-Test*. (Unpublished senior project). California Polytechnic State University, San Luis Obispo.

Wu, Harry (2017). *Non-Normality and Heteroscedasticity in Regression and ANOVA*. (Senior Project, California Polytechnic State University San Luis Obispo). Retrieved from http://digitalcommons.calpoly.edu/statsp/59.