Choosing a Statistical Model Amidst Ceiling and Floor Effects

Rao, V.N.V., Running, K., & Codding, R.S. (2021).

INSTRUCTIONS

This file contains R code and output to accompany *Choosing a statistical model amidst ceiling and floor effects* by Rao, Running, and Codding (2021). Specifically, it will diagnose a ceiling or floor effect (CFE), determine whether it is inappropriate to use an ANOVA model and if so whether it is appropriate to use a Tobit model, and subsequently fit and interpret a Tobit model.

This file assumes that readers are already familiar with the basics of using R, as well as the basics of the ANOVA model and general linear models.

Use both the Tobit R Guide pdf and Tobit R code to follow along and practice analyzing data with a CFE. Both files are available at `https://github.com/RaoVNV/NASP2021/`.

Install and load Libraries

If you have not previously installed these libraries, then run the following chunk:

```r
install.packages(c("ggplot2", "ggthemes", "dplyr", "VGAM", "sm", "MASS"))
```

Load the following libraries:

```r
library(ggplot2) # data visualization
library(ggthemes) # data visualization
library(dplyr) # data wrangling
library(VGAM) # tobit model
library(sm) # model diagnostics
library(MASS) # post hoc analyses

cbPalette <- c("#999999", "#E69F00", "#56B4E9") # 3-color color-blind palette

set.seed(17293759)
```

DATA

First, download the dataset `Fraction Knowledge Dataset.csv`. You can find the dataset and other supplementary material by visiting `https://github.com/RaoVNV/NASP2021/`.

Importing into R

Next, load the dataset into R. The following code creates a pop-up window. Use the pop-up window to navigate to the `Fraction Knowledge Dataset.csv` file on your computer.

```
fraction_knowledge <- read.csv(file=file.choose())
```

To ensure the file has been imported correctly, take a peak at the characteristics of the dataset.

```
#take a look at the first 5 observations
head(fraction_knowledge)
```

```
##   ï..student_ID site_ID teacher_ID      condition concepts_pre concepts_post
## 1           319   Site1   Teacher1        Control           17            17
## 2           312   Site1   Teacher1        Control           27            40
## 3           306   Site1   Teacher1 Concepts-First            6            40
## 4           315   Site1   Teacher1      Iterative           16            32
## 5           321   Site1   Teacher1      Iterative           26            37
## 6           311   Site1   Teacher1 Concepts-First           25            23
##   procedures_pre procedures_post
## 1             21              24
## 2             13              33
## 3             35              39
## 4             32              34
## 5             23              36
## 6             NA              38
```

```
#take a look at the last 5 observations
tail(fraction_knowledge)
```

```
##     ï..student_ID site_ID teacher_ID      condition concepts_pre concepts_post
## 109           215   Site2   Teacher5      Iterative           36            37
## 110           208   Site2   Teacher5 Concepts-First           27            40
## 111           218   Site2   Teacher5 Concepts-First           33            40
## 112           216   Site2   Teacher5        Control           30            40
## 113           222   Site2   Teacher5        Control           39            40
## 114           217   Site2   Teacher5      Iterative           40            40
##     procedures_pre procedures_post
## 109             23              29
## 110             40              39
## 111             32              39
## 112             35              40
## 113             35              39
## 114             40              40
```

```
#take a look at all the variable names
names(fraction_knowledge)
```

```
## [1] "ï..student_ID"   "site_ID"         "teacher_ID"      "condition"
## [5] "concepts_pre"    "concepts_post"   "procedures_pre"  "procedures_post"
```

```
#create quick summaries of all variables
summary(fraction_knowledge)
```

```
##  ï..student_ID     site_ID           teacher_ID         condition
##  Min.   :101.0   Length:114         Length:114         Length:114
##  1st Qu.:209.2   Class :character   Class :character   Class :character
##  Median :320.5   Mode  :character   Mode  :character   Mode  :character
##  Mean   :327.1
##  3rd Qu.:423.8
##  Max.   :528.0
##
##   concepts_pre   concepts_post   procedures_pre  procedures_post
##  Min.   : 2.00   Min.   :10.00   Min.   : 0.00   Min.   : 0.00
##  1st Qu.:22.00   1st Qu.:33.00   1st Qu.:21.00   1st Qu.:33.00
##  Median :28.00   Median :39.00   Median :29.00   Median :39.00
##  Mean   :28.28   Mean   :35.36   Mean   :26.55   Mean   :34.98
##  3rd Qu.:38.00   3rd Qu.:40.00   3rd Qu.:34.00   3rd Qu.:40.00
##  Max.   :40.00   Max.   :40.00   Max.   :40.00   Max.   :40.00
##  NA's   :1       NA's   :4       NA's   :3       NA's   :4
```

Initial Processing

We will analyze differences in scores between each of the experimental conditions using the *condition* variable. We want to use the `Control` condition as a reference group. We can set the order that R processes each condition by specifying levels of a factor.

```
#Make sure the experimental conditions are in the order we want
fraction_knowledge$condition <-
  factor(fraction_knowledge$condition,
    levels=c("Control","Concepts-First","Iterative"))
```

Visit `https://github.com/RaoVNV/NASP2021/` to review the `Fraction Knowledge Data Dictionary` for more information about each of the variables contained in this dataset, and the `Fraction Knowledge Data Introduction` for more information about the experimental study for which this data was collected.

DESCRIPTIVE STATISTICS

In this example, we will examine the students' scores on the procedures assessment at post-test. Our main analysis goal is to identify whether there were group differences between each of three experimental conditions. One important covariate that may affect students' post-test scores are their pre-test scores.

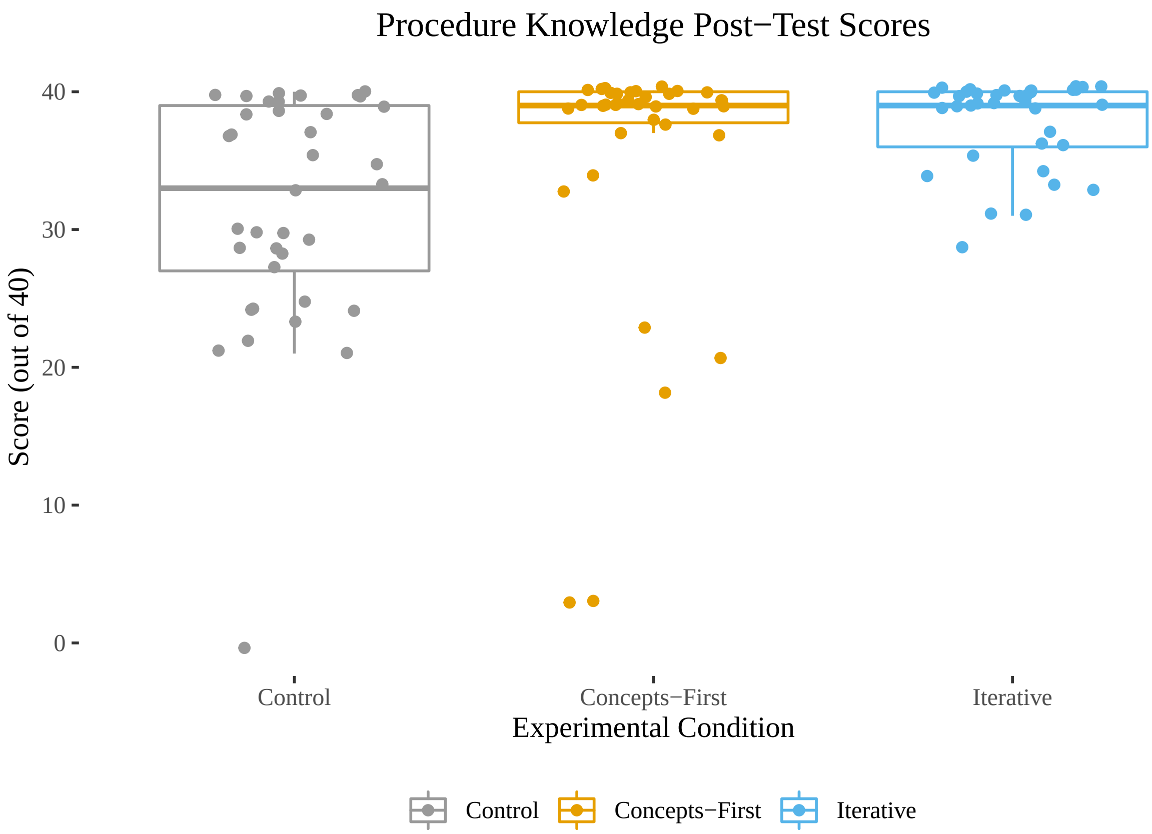
The study used block randomization to assign students within each classroom to an experimental condition based on their pre-test score. Therefore, the design matches that of the Analysis of Covariance (ANCOVA) model, where post scores are modelled as a function of experimental condition and pre-test scores: `procedures_post ~ condition + procedures_pre`.

However, before we fit the ANCOVA model, we must explore the data with descriptive summaries and visualizations, which will help us examine ANCOVA's suitability.

Diagnosing the CFE

First, let's look at the distribution for procedure knowledge post-test scores by experimental condition:

```r
ggplot(data=fraction_knowledge, # name of our data
  aes(y=procedures_post, # response variable
      x=condition, # explanatory variable
      color=condition # assign one color per condition
  )
) +
  scale_colour_manual(values=cbPalette) + # use the color blind palette
  geom_boxplot(outlier.shape=NA, # creates a boxplot
      alpha=0.5) + # makes points transparent
  geom_jitter(width=0.25)+ # ensures points do not overlap
  theme_tufte() + # applies theme consistent with Tufte's maxims
  ggtitle("Procedure Knowledge Post-Test Scores") + # main title
  theme(plot.title = element_text(hjust = 0.5)) + # centers the title
  ylab("Score (out of 40)") + # y-axis label
  xlab("Experimental Condition") + # x-axis label
  theme(legend.title = element_blank(), # suppresses legend label
        legend.position = "bottom") # fixes legend position
```

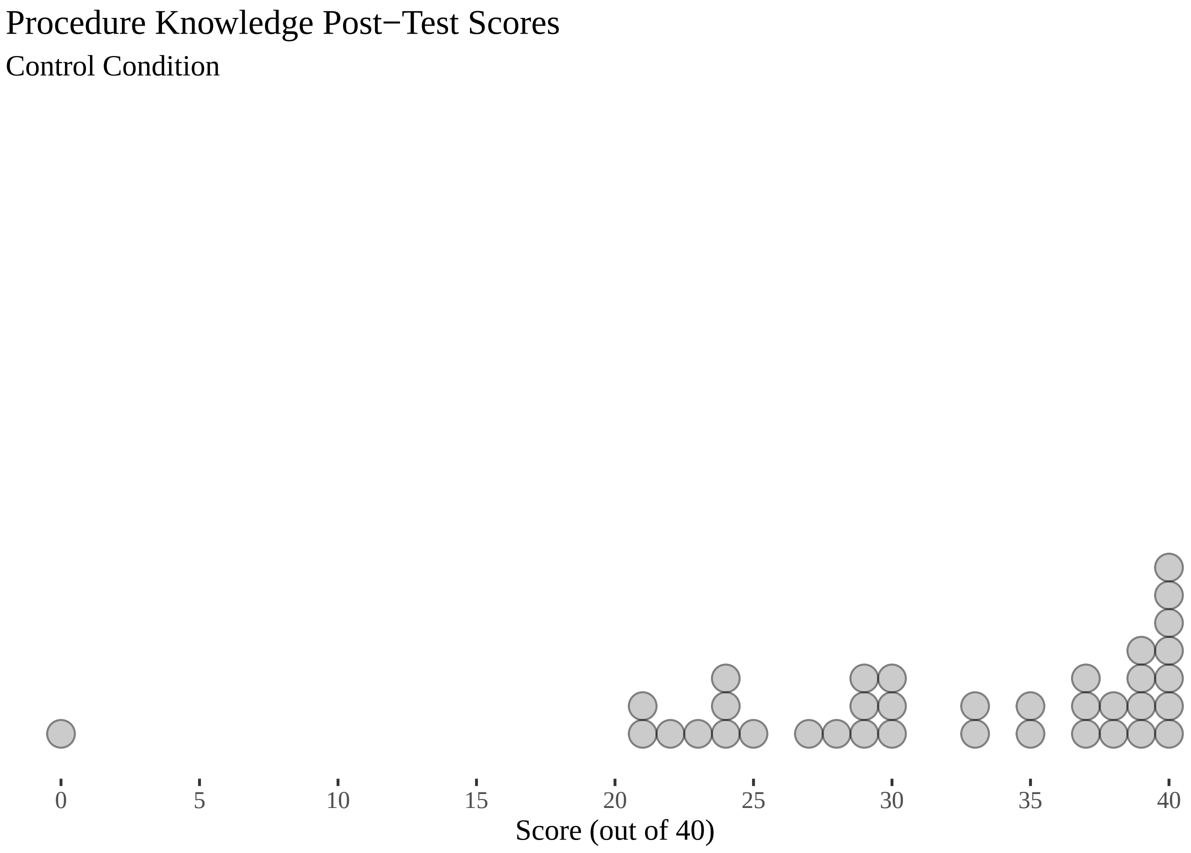


We can see that there are quite a few *dots* bunched up right around a value of 40. That signifies that several students have achieved the maximum score. This is evidence of a ceiling effect. ANOVA and ANCOVA models are based on the assumption that scores are normally distributed within groups. While *true* procedural knowledge scores may still be normally distributed, our observed scores are not, due to the ceiling effect.

We can further examine the extent of the ceiling effect by creating dotplots for each experimental condition.

Control Condition

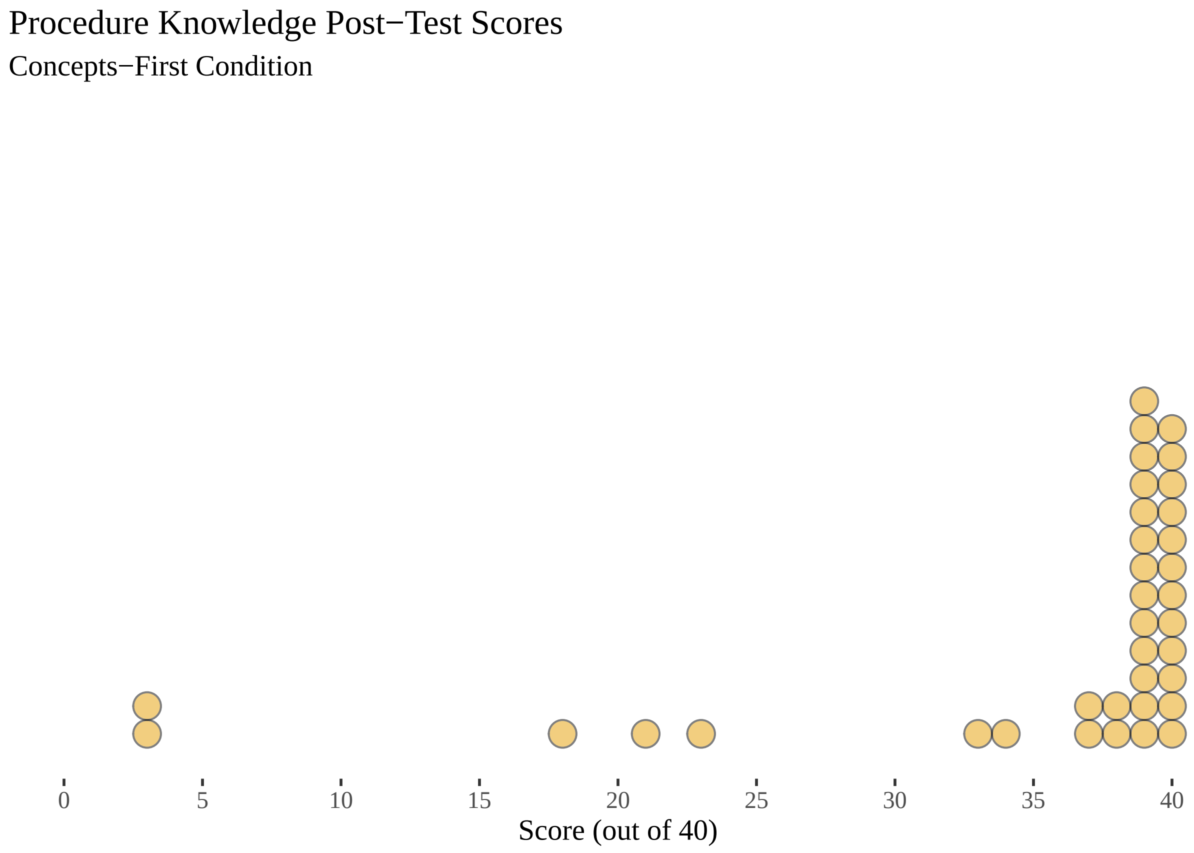
```
ggplot(
  data=(fraction_knowledge %>% filter(condition=="Control")), # subsets the data
  aes(x=procedures_post, fill=condition)
) +
  scale_fill_manual(values=cbPalette[1]) + # matches Color used in the previous graph
  scale_y_continuous(NULL, breaks = NULL) + # hides the Y-Axis
  scale_x_continuous(limits=c(0,40), breaks = seq(0,40,5)) + # specifies the X-axis
  geom_dotplot(alpha=0.5, binwidth=1) + # creates a dotplot
  theme_tufte() +
  theme(legend.position = "none") + # hides the Legend
  ggtitle("Procedure Knowledge Post-Test Scores", # creates a main title
          "Control Condition") + # creates a subtitle
  ylab("Relative Frequency") +
  xlab("Score (out of 40)")
```



Seven of the 38 students in the `Control` condition scored the maximum score, with a further four one point shy of the maximum.

Concepts First Condition

```r
ggplot(
  data=(fraction_knowledge %>% filter(condition=="Concepts-First")),
  aes(x=procedures_post, fill=condition)
) +
  scale_fill_manual(values=cbPalette[2]) +
  scale_y_continuous(NULL, breaks = NULL) +
  scale_x_continuous(limits=c(0,40), breaks = seq(0,40,5)) +
  geom_dotplot(alpha=0.5, binwidth=1) +
  theme_tufte() +
  theme(legend.position = "none") +
  ggtitle("Procedure Knowledge Post-Test Scores",
          "Concepts-First Condition") +
  ylab("Relative Frequency") +
  xlab("Score (out of 40)")
```



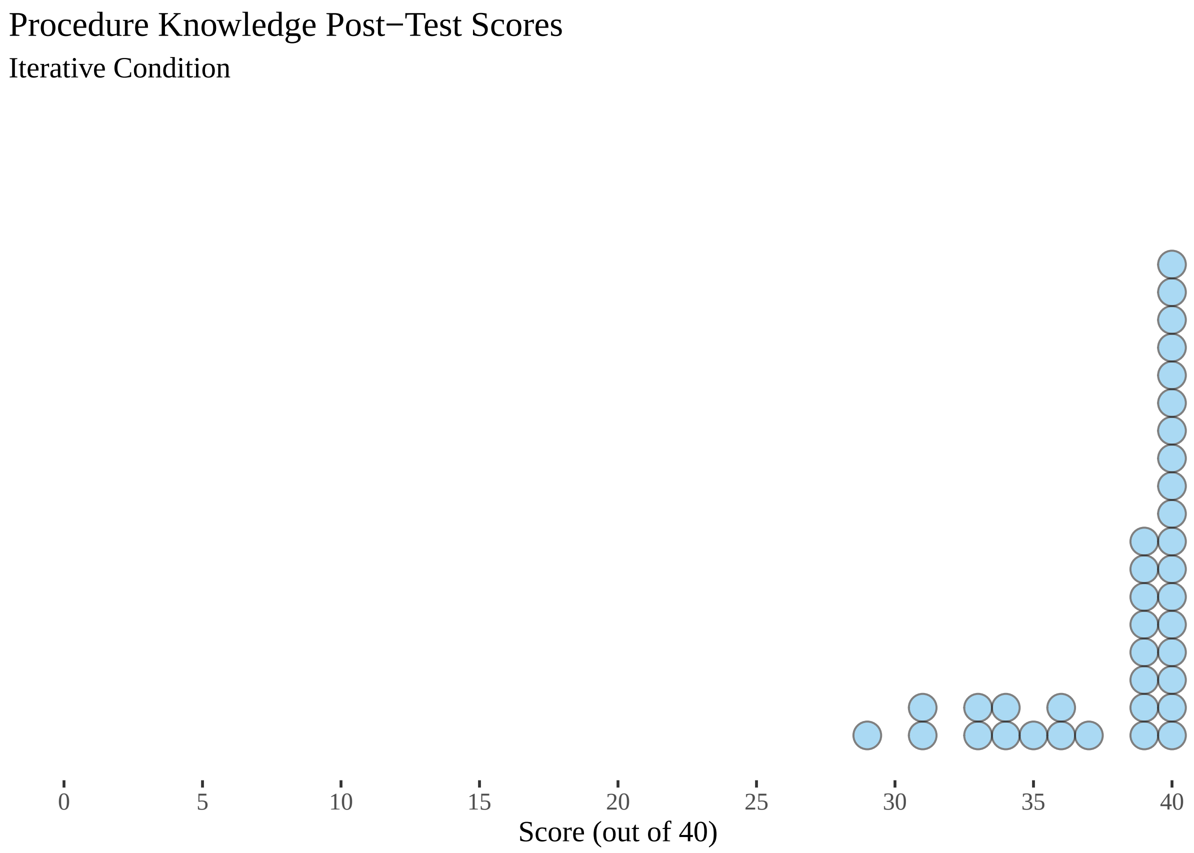
Procedure Knowledge Post−Test Scores

Concepts−First Condition

Twelve of the 38 students in the `Concepts-First` condition scored the maximum score, and another 13 students were only one point shy of the maximum.

Iterative Condition

```
ggplot(
  data=(fraction_knowledge %>% filter(condition=="Iterative")),
  aes(x=procedures_post, fill=condition)
) +
  scale_fill_manual(values=cbPalette[3]) +
  scale_y_continuous(NULL, breaks = NULL) +
  scale_x_continuous(limits=c(0,40), breaks = seq(0,40,5)) +
  geom_dotplot(alpha=0.5, binwidth=1) +
  theme_tufte() +
  theme(legend.position = "none") +
  ggtitle("Procedure Knowledge Post-Test Scores",
          "Iterative Condition") +
  ylab("Relative Frequency") +
  xlab("Score (out of 40)")
```



Eighteen of the 38 students in the `Iterative` condition scored the maximum score, and another 8 students were only one point shy of the maximum.

Quantifying the CFE

Having diagnosed ceiling effects in all three experimental conditions, we need to quantify the magnitude of the ceiling effect in order to help determine which statistical model to utilize.

```
#Percentage of observations at or near ceiling by experimental condition
fraction_knowledge %>%
  group_by(condition) %>%
  filter(!is.na(procedures_post)) %>% # remove observations with missing values
  summarise(
      p.atCFE = sum(procedures_post==40)/n(), # percent (by condition) at ceiling
      p.nearCFE= sum(procedures_post>=39)/n() # percent (by condition) near ceiling
  )
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 3 x 3
##   condition      p.atCFE p.nearCFE
##   <fct>            <dbl>     <dbl>
## 1 Control          0.189     0.297
## 2 Concepts-First   0.333     0.694
## 3 Iterative        0.486     0.703
```

We see in this table that approximately 19% of scores in the `Control` condition are at the ceiling, and 30% are within 1 point of the ceiling. Similarly, approximately 33% of scores in the `Concepts-First` condition are at the ceiling and 49% of scores in the `Iterative` condition are at the ceiling.

The 30-20 rule says that to use ANOVA, no group should have more than 30% of it's observations at the ceiling. This is violated by both the `Concepts-First` and `Iterative` conditions Furthermore, the difference in the percentage of observations at the ceiling between two groups should not be more than 20%. This is violated by the difference between the `Iterative` condition and the `Control` condition, which is nearly 30 percentage points. Therefore, we should not use ANOVA or ANCOVA to analyze this data.

The tobit regression model, a type of censored regression models, explicitly aims to account for observed scores truncated at either a floor or ceiling. The tobit model assumes that an individual's true score is not limited to the observed range. It produces estimates for true scores based on the observed cumulative frequency distributions within the range of observed scores.

To decide whether to use the tobit model when faced with a CFE, we can follow the 70% rule. If no more than 70% of observations are at the ceiling in each group, then we can use Tobit regression. Since the proportions are all well below 70%, we can use Tobit regression to analyze this data. Note that even thought nearly 70% of observations in the `Concepts-First` and `Iterative` condition are *near* the ceiling, this still does not violate the 70% rule. First, the rule is for the number of observations *at* the ceiling, and is relatively robust. Second, even with 70% near the ceiling, the rule still suggests that the bias in the Tobit model will be small.

TOBIT REGRESSION

Tobit regression carries the same model assumptions as generalized linear models: normality of residuals and homoscedasticity. We will first fit the model before examining the model diagnostics.

Fitting the Model

Fitting a tobit model is very similar to fitting a generalized linear model with the `glm()` function, and only has a few small differences compared to fitting an ANOVA model.

We will be using the `vglm()` function from the `{VGAM}` package. We first specify the name of the dataset, with the `data=` option, just as we have in previous functions.

The second argument to the function is the model specification. The general form of a model is `response variable ~ explanatory variables + covariates`. This is the same format used by the `glm()` and `anova()` functions in R. In this case, we want to model procedure post-test scores as a function of experimental condition while controlling for procedure pre-test scores.

The last argument is where we specify that we are fitting a tobit model. We do this by using the `family=tobit()` option. We must also specific the minimum possible score and the maximum possible score with the `Lower=` and `Upper=` options.

```
tb.mdl <- vglm(
  data = fraction_knowledge,
  procedures_post ~ condition + procedures_pre,
  family=tobit(Lower=0, Upper=40)
)
```

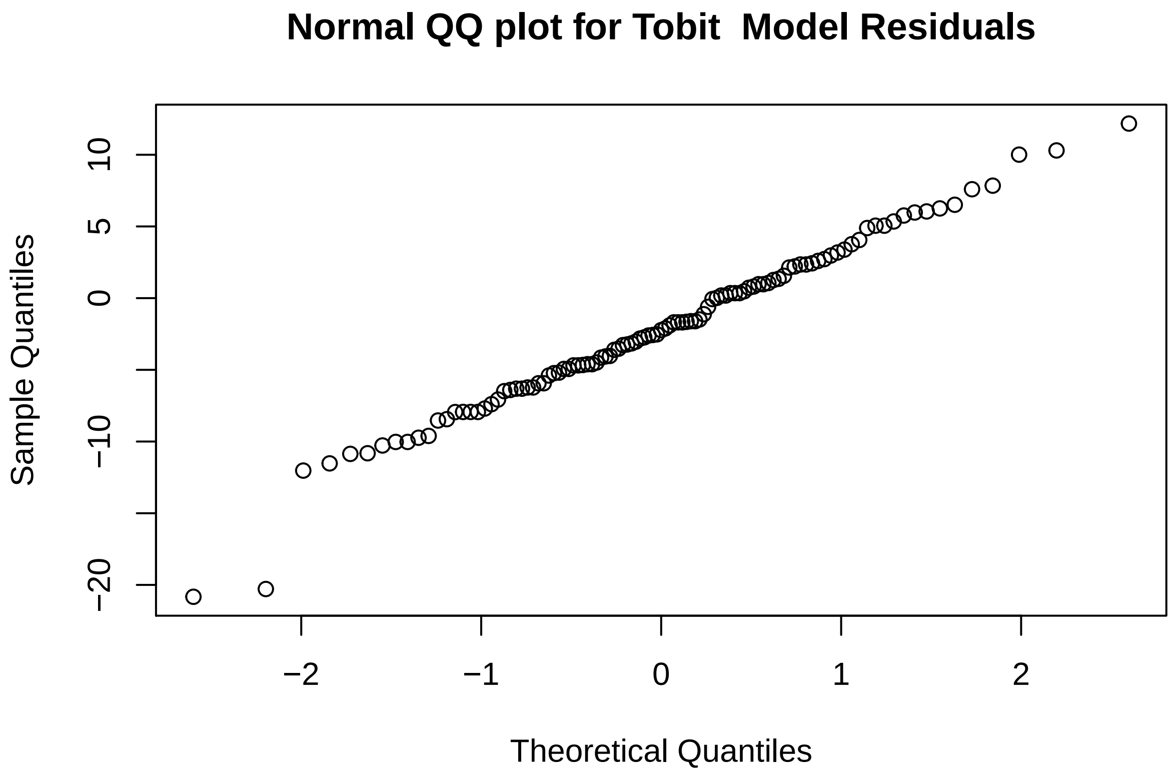
Residual Analysis

Before we inspect the model summary, we must ensure that the model assumptions are met. There are two assumptions to tobit regression, the same as to all general linear regression including ANOVA: residuals must be normally distributed, and the variables of the residuals should not change as a function of the explanatory variables.

Normality

In order to check the normality of residuals, we use the Normal QQ Plot. When the points form a straight line, it is an indication that the residuals can indeed be appropriately modeled by a normal distribution.

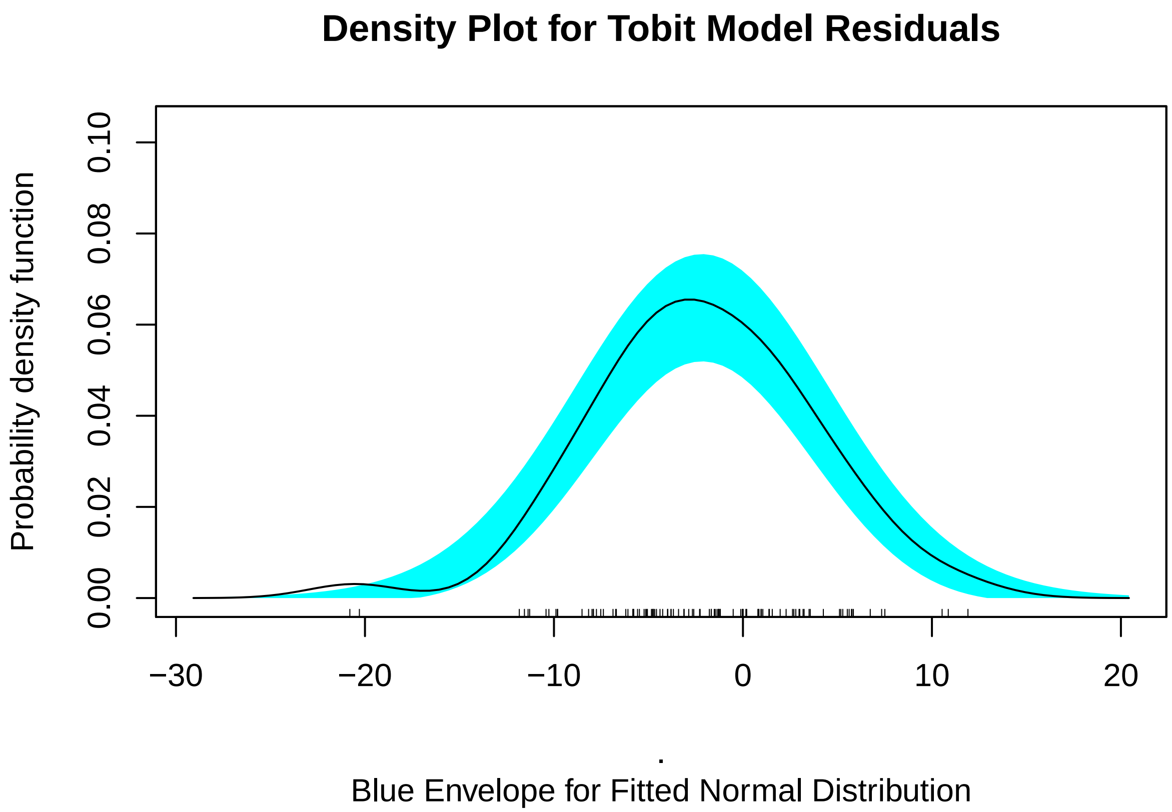
```
resid(tb.mdl, type = "response") %>% # extract residuals from the model
  qqnorm(main="Normal QQ plot for Tobit  Model Residuals")
```



In this case, it does not appear that there are any major violations of the assumption that the residuals should be normally distribution.

We can also examine the distribution via a density plot. When the line based on our data falls entirely within the blue region (representing what we'd expect under a normal distribution), then we have no reason to worry that the normality assumption is violated.

```r
resid(tb.mdl, type = "response") %>%
  sm.density(model = "normal") # add an envelope based on the normal distribution
title(main="Density Plot for Tobit Model Residuals",
      sub="Blue Envelope for Fitted Normal Distribution")
```

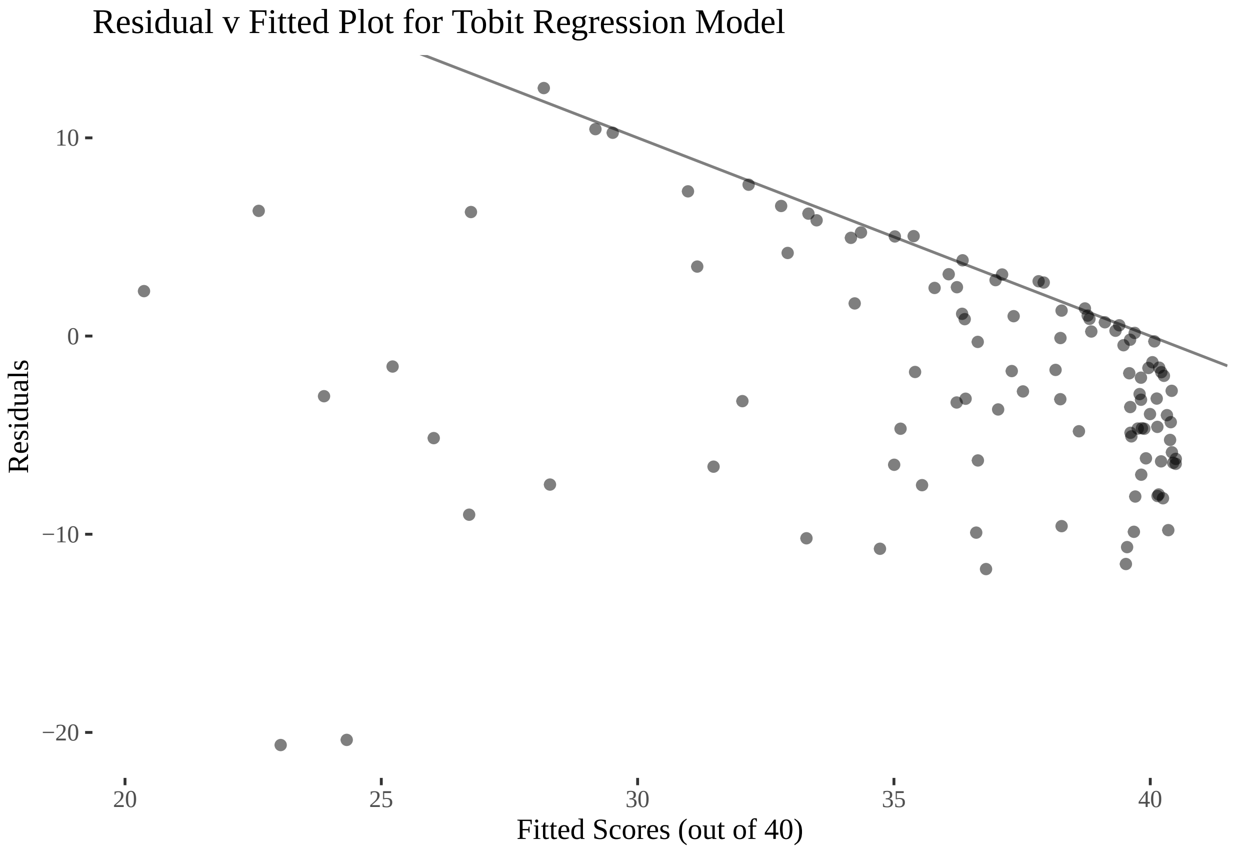


Indeed, it does not appear that the normality assumption is violated, as the black line representing the distribution of residuals for our data falls entirely within the blue region.

Homoscedasticity

Interpreting homoscedasticity from residual vs fitted plots for the tobit model is a little bit tricky because of the ceiling effect. For example, it is not possible to have a positive residual for a predicted score at the ceiling, since the observed score can never be more than 40. Similarly, there is an upper limit to the positive residuals for any given predicted score, as scores are limited by the ceiling. Therefore, the main goal of the residual vs fitted plot is to ensure there are no obvious patterns in the plot.

```
data.frame(
 r=resid(tb.mdl, type = "response"),
 f=fitted(tb.mdl, type = "censored") # extract predicted scores
) %>%
ggplot(aes(x=f, y=r)) +
  geom_jitter(alpha=0.5, width=0.5, height=0.5) +
  ggtitle("Residual v Fitted Plot for Tobit Regression Model") +
  theme_tufte() +  ylab("Residuals") +  xlab("Fitted Scores (out of 40)") +
  geom_abline(slope=-1, intercept=40, alpha=0.5) # add a line at the ceiling
```



We see that residuals are mainly limited by the presence of the ceiling (denoted by the sloping line). However, residuals generally fall between +10 and -10, and we can only assume that the they fail to reach as high as +10 for fitted scores above 30 due to the ceiling. No other obvious patterns in the residuals exist, such as an obvious curvilinear pattern, and therefore it appears that the homoscedasticity assumption is not violated.

There are two potential extreme values with residuals near -20, but in-person follow-up determined that these are not data errors, and thus the observations were retained in the dataset.

Interpreting the model

Now that we have verified that no model assumptions are violated, we can therefore conclude that the tobit model is an appropriate and useful model to help us analyze the dataset, and can begin interpreting the fitted model.

Experimental Condition

The first step in interpreting the model is to determine whether there are differences in post-test scores by experimental condition. To do this, we utilize the ANOVA Type 3 Sums of Squares. This test examines the ratio of the variance between groups to the variance within groups. Note, this test, though it bears the name `ANOVA`, is *not* the same thing as fitting an ANOVA model to compare a difference in group means, although the underlying mathematics are similar.

```r
anova.vglm(tb.mdl, type="III")
```

```
## Analysis of Deviance Table (Type III tests: each term added last)
##
## Model: 'tobit', 'VGAMcategorical'
##
## Links: 'identitylink', 'loglink'
##
## Response: procedures_post
##
##                Df 2 * LogLik Diff. Resid. Df  LogLik  Pr(>Chi)
## condition       2           12.541       211 -268.44  0.001892 **
## procedures_pre  1           39.862       210 -282.11 2.725e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We see from the Type III tests that there appears to be strong evidence indicating that there are differences in post-test scores by experimental condition (p=0.002). Also note that pre-test scores appear to be related to post-test scores.

Estimated group differences

What are the estimated differences in the mean scores for each group? For generalized linear models fit with the `glm()` function, we might use the `{emmeans}` package. However, this package is not compatible with the tobit regression model. Therefore, we can examine group differences by simply viewing the model summary.

```r
summary(tb.mdl)
```

```
##
## Call:
## vglm(formula = procedures_post ~ condition + procedures_pre,
##     family = tobit(Lower = 0, Upper = 40), data = fraction_knowledge)
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept):1           19.70660    2.38589   8.260  < 2e-16 ***
## (Intercept):2            1.97167    0.09183  21.470  < 2e-16 ***
## conditionConcepts-First  3.57895    1.83072   1.955 0.050591 .
## conditionIterative       6.57917    1.86897   3.520 0.000431 ***
## procedures_pre           0.54128    0.07927   6.828 8.59e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: mu, loglink(sd)
##
## Log-likelihood: -262.1748 on 209 degrees of freedom
##
## Number of Fisher scoring iterations: 7
##
## No Hauck-Donner effect found in any of the estimates
```

The first two values reported as Intercept1 and Intercept2 are the predicted mean when covariates=0 and the log of the residual standard error. In this case, for students in the `Control` condition with a pre-test score of 0, the model predicts that they will have an average post-test score of 19.71. Similarly, by exponentiating 1.972, we can estimate that the residual standard error for students scores is approximately 7.18.

The next two values reported are indicator variables for the `Concepts-First` and the `Iterative` conditions In this case, the model estimates that the average score for students in the `Concepts-First` condition is 3.58 points higher than the average score for students in the `Control` condition ($p = 0.051$; approximately 9% points), while the average score in the `Iterative` condition is 6.58 points higher than the `Control` condition's average ($p = 0.0004$; approximately 16% points).

The final value reported is the estimate of the regression coefficient for the covariate in the model, pre-test scores.

We can also create confidence intervals for the regression coefficients using the `confintvglm()` function, used here to create 95% confidence intervals for the difference in mean scores for the two experimental conditions compared to the `Control` condition.

```
confintvglm(tb.mdl)[3:4,]
```

```
##                                 2.5 %    97.5 %
## conditionConcepts-First -0.009203548  7.167104
## conditionIterative       2.916058345 10.242279
```

Recall that the ANOVA-based model would have likely underestimated the true differences in this case. Indeed, the ANCOVA model estimates the differences between the `Concepts-First` condition and the `Control` condition as 2.33 and between the `Iterative` condition and the `Control` condition as 4.17 after adjusting for pre-test scores, which are both near a 35% underestimation compared to the Tobit model estimated differences.

Effect sizes

We can convert these estimates into effect sizes akin to Cohen's d by utilizing the model estimated standard deviation of post-test scores or the standard deviation of pre-test scores. We cannot use the standard deviation of observed post-test scores as the ceiling effect will lead to an underestimate of the true standard deviation. We can only use the standard deviation of pre-test scores so long as there is no ceiling effect in the pre-test scores. There are also other more advanced measures of effect sizes that may be more appropriate with advanced regression models such as the Tobit model.

```
coef(tb.mdl)[3:4] / exp(coef(tb.mdl)[2])
```

```
## conditionConcepts-First      conditionIterative
##               0.4982769               0.9159802
```

There appears to be a medium effect on post-test scores comparing the `Concepts-First` condition to the `Control` condition (0.498), and a large effect on post-test scores comparing the `Iterative` condition to the `Control` condition (0.916).

It should be noted that Tobit regression *can* result in slight over-estimations of the effect size. However, with the `<70%` rule clearly met, the overestimation should not be any more than 0.025 in terms of the effect size, and therefore would not alter our interpretations.

Multiple Comparisons

While the standard model output allowed us to compare both experimental conditions to the `Control` condition, we may be interested in comparing the experimental conditions to each other. We can achieve this by specifying contrasts. The `contr.sdif` function from the `{MASS}` package will allow us to interpret regression coefficients in terms of successive differences. That is, the first estimate will provide the difference between the `Concepts-First` and the `Control` conditions, while the second estimate will provide the difference between the `Iterative` and the `Concepts-First` conditions, based on the order of the levels in our variable.

```
tb.mdl.2 <- vglm(
  data = fraction_knowledge,
  procedures_post ~ condition + procedures_pre,
  family=tobit(Lower=0, Upper=40),
  contrasts = list(condition="contr.sdif")
)
summary(tb.mdl.2)
```

```
##
## Call:
## vglm(formula = procedures_post ~ condition + procedures_pre,
##     family = tobit(Lower = 0, Upper = 40), data = fraction_knowledge,
##     contrasts = list(condition = "contr.sdif"))
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept):1                      23.09264    2.15296  10.726  < 2e-16 ***
## (Intercept):2                       1.97167    0.09183  21.470  < 2e-16 ***
## conditionConcepts-First-Control     3.57895    1.83072   1.955   0.0506 .
## conditionIterative-Concepts-First   3.00022    1.91714   1.565   0.1176
## procedures_pre                      0.54128    0.07927   6.828 8.59e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: mu, loglink(sd)
##
## Log-likelihood: -262.1748 on 209 degrees of freedom
##
## Number of Fisher scoring iterations: 7
```

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

```
confintvglm(tb.mdl.2)[4,]
```

```
##      2.5 %     97.5 %
## -0.7573098  6.7577458
```

```
coef(tb.mdl.2)[4] / exp(coef(tb.mdl.2)[2])
```

```
## conditionIterative-Concepts-First
##                         0.4177033
```

We now see that the estimated difference in the mean scores between the `Iterative` condition and the `Concepts-First` condition is approximately 3.00 (approximately 7.5% points; *p*=0.118; 95% CI: -.757 to 6.758; moderate effect size of 0.418).

SUMMARY

When faced with a CFE, four easy steps can be taken to confidently and more precisely estimate differences between group means:

- Diagnose the CFE with a dotplot
- Quantify the magnitude of the CFE
- Determine the appropriateness of an ANOVA and/or Tobit model
- Fit and Interpret a Tobit Regression Model if appropriate

All four of these steps can easily be performed in R, as shown in this document. While Tobit models have more advanced extensions (i.e. generalized tobit regression), the basic linear Tobit regression model can help school psychology researchers mitigate measurement problems and provide more precise estimates of intervention effects.