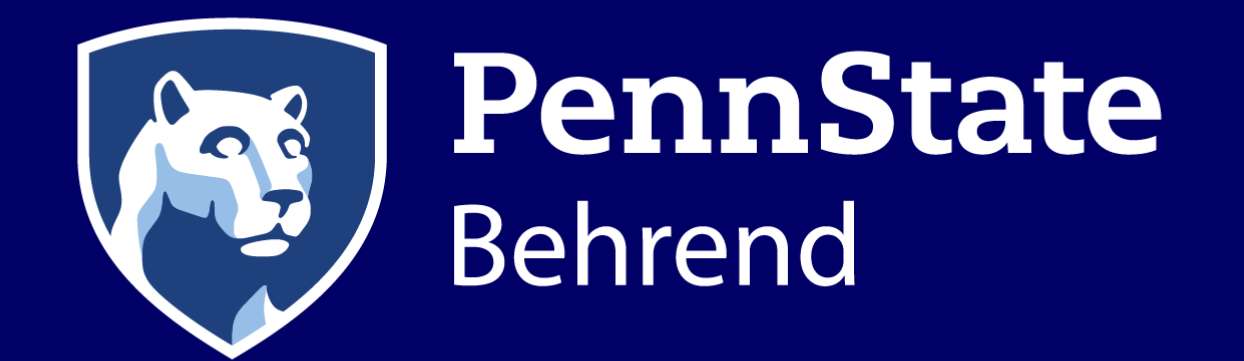


# Creating Custom Take Home Exams for Statistics Courses with R Markdown

Michael A. Rutter

Penn State Erie, The Behrend College



## Introduction

- Often in an applied statistics class there is a need for take home exams that test specific concepts.
- In an attempt to minimize non-authorized collaboration, custom exams are useful.
- Each custom exam also needs an answer key and data sets, when applicable.
- In this poster, I will present my procedure for making **custom take home exams**, data sets, and answer keys using **R Markdown**.

## Creating a Custom Exam

- Every exam needs a custom exam number (1000-9999) for the pdf and the data files.

```
exam.no <- round(runif(1,1000,10000),0)
write.table(exam.no,"exam_no.txt",
            row.names = FALSE, col.names = FALSE)
```
- This code is also used to write the custom data sets (as for question 2).

```
file.name <- paste(exam.no,"q2.csv",sep="")
write.csv(data,file.name,row.names=FALSE)
```

## Example Question One

- An unbalanced, one factor ANOVA with a unique sample size,  $p < 0.04$  and no need to transform.

```
p.value <- 1
bt.p <- 0
while(p.value>.04 || bt.p <.05){
  n.j <- round(runif(4,4,10))
  Y.bar.j <- runif(4,100,200)
  r.mu <- rep(Y.bar.j,n.j)
  Y <- round(rnorm(length(r.mu),r.mu,35))
  region <- rep(c("NE","MW","S","W"),n.j)
  res <- aov(Y~region)
  res.bt <- bartlett.test(Y~region)
  p.value <- anova(res)[1,5]
  bt.p <- res.bt$p.value}
```
- For this particular question, the data is presented on the exam so students are required to enter it into R.

```
```{r results='asis',echo=FALSE}
panderOptions('knitr.auto.asis', FALSE)
pander(unstack(data))
```
```

## Example Question Two

- In this example, the goal is to test the student's ability to examine the assumptions of ANOVA.
- The intent is to generate data that needs to be log transformed, although other transformations are accepted.

```
p.val <- 1
bt.p <- 1
log.p <- 0
while(p.val>.05 || bt.p >.03 || log.p<.05){
  n.j <- round(runif(1,4,8))
  Y.bar.j <- runif(5,2.2,3.1)
  r.mu <- rep(Y.bar.j,n.j)
  Y.t <- rnorm(length(r.mu),r.mu,.095)
  Y <- round(exp(Y.t),1)
  hormone <- rep(c("A","B","C","D","E"),n.j)
  res <- aov(log(Y)~hormone)
  res.bt <- bartlett.test(Y~hormone)
  res.bt.log <- bartlett.test(log(Y)~hormone)
  p.val <- anova(res)[1,5]
  bt.p <- res.bt$p.value
  log.p <- res.bt.log$p.value}
```
- Note that we are looking for a data set has  $p < 0.05$  when log-transformed, but fails a Bartlett's Test when using the non-transformed data.
- The p-value of non-transformed ANOVA can be any value, which causes some students not to check assumptions.

## Generating Answers

- For each question on the exam, answers are generated at the end.

```
\pagebreak
# Answers Exam `r exam.no`
## Question 2
```{r echo=FALSE}
...
lb <- exp(Y.bar.j - qt(1-.05/2,df.e)*sqrt(...)
ub <- exp(Y.bar.j + qt(1-.05/2,df.e)*sqrt(...)
...
bp <- barplot2(exp(Y.bar.j),ylim=c(0,25),
               plot.ci=TRUE,ci.l=lb,ci.u=ub,
               main="Mean with 95% CI")
groups <- multcompLetters4(res,TukeyHSD(res))
```
- At this point in the course, confidence intervals are generated via formulas (not R commands) and back-transforming is emphasized.

## Example Question Three: One of Two Questions

- In order to make the exams unique, you can choose between two or more questions randomly.

```
q3 <- sample(1:2,1)
if(q3==1){
  cat("A consumer product-testing ...")
} else {
  cat("A chemist is studying the...")
}
```
- Generating the data for the problem is a separate step.

```
if(q3==1){
  data <- read.csv("humidity.csv")
  p.value <- 0
  while(p.value < .05)
  {
    data$cons <- sample(data$cons)
    p.value <- with(data,tukey.add.test(...)
  }
  fx.3 <- data
  ...
```
- In this question, while the values of  $Y$  are fixed, they are assigned to different levels randomly.
- A RCBD question, data sets with an interaction are discarded.

## Creating Exams, Keys and Data

- In Linux, a bash script was used to create and separate the exams from answer key.

```
#!/bin/bash
Rscript -e "rmarkdown::render('midterm.Rmd')"
file="./exam_no.txt"
examno=$(cat "$file")
pdftk mid_term.pdf cat 1-3 output $examno.pdf
mv $examno.pdf ./To_print
mv mid_term.pdf mid_term_$examno.pdf
mv *.pdf ./Mid_term_Exam
mv *.csv ./Data
```

## Conclusion

- Students had no issues with separate, unique exams.
- Pre-generated answer keys made grading exams easier.
- Given statistical requirements of questions, some exams may take a little time to generate.
- No evidence of cheating, even among students who work closely together.