A GRAPH THEORY APPROACH TO REDUCING TEST LENGTH

Aaron Carl Smith
University of Central Florida
Department of Mathematics
4393 Andromeda Loop N
Orlando, FL 32816
aaron.smith@ucf.edu

This paper presents a statistical technique based on a graph theory to identify test and quiz questions with less utility. Thus providing educators tools to improve assignments. These techniques are applied in a case study using a prequisite knowledge quiz for precalculus students. The quiz evaluates students' mastery of skills taught in prior math classes. During the first week of classes, the quiz helped faculty identify which students are at risk of not passing the course. Cycle intersection matrices identify which questions to consider removing.

1 The Problem

The bipartisan bill, No Child Left Behind, ushered an era of high stakes testing into United States education (Juana Summers, NPR, April 14, 2015). Groups of educators, parents, and politicians question the value, efficacy, and unintended consequences of such tests [6]. Using the same techniques to evaluate the value of medical tests, data scientists concluded that lengthy and frequent tests are wasteful, and unethical [8]. Educators and administrators can mitigate the negative effects of high impact tests by reducing tests to only the most informative questions. This paper presents a method that can identify test questions that are less informative. The method uses cycle intersection matrices from graph theory to identify questions that should be considered for removal. It is an unsupervised statistical learning technique that does not require historical data from past testing. The cycle intersection method requires only the test data. The cycle intersection approach is demonstrated in a case study using a prerequisite knowledge quiz for incoming precalculus students at the University of Central Florida (UCF). The

principle of reproducible research is adhered to and R source code for the computations are included [5, 11, 10, 12].

2 Introduction

During the first week of fall 2014 and spring 2015 semesters at the University of Central Florida's Mathematics Department, 1000 level mathematics students took quizzes on concepts taught in lower level math courses. The quizzes helped the faculty identify students who needed extra help, and students who should consider dropping down to a lower class. Four faculty constructed these quizzes that were given via the online homework system MyLabsPlus. The quizzes were proctored in a mathematics education computer lab. This paper will use parametric and non-parametric statistical results, and cycle intersection matrices to identify questions on the precalculus quiz that faculty should consider replacing or removing.

To identify questions to remove, cycle intersection matrices from graph theory identified correlated variables. Phifer's technique uses complete subgraphs to determine which questions are informatively redundant. Questions in a complete subgraph from a cycle intersection matrix can be reduced to a smaller set of questions [4].

First week prerequisite quiz results and test 1 scores will provided the data. The independent variables are the individual question scores, the dependent variable is the test 1 score. Initially, the intention was to use the final exam as the response variable. Preliminary data exploration showed that the relationship between quiz results and final exam scores was not strong enough for the task.

3 The Dataset

The quiz contained twenty questions. Questions 2, 3, 4, 6, 8, 9, 11, 13, 15, 16, 17, 18, 19, and 20 each had one part and were scored as binary zero-one observations. Questions 1, 5, 7, 10, 12, and 14 had multiple parts and were scored from zero to one without partial credit.

question	learning objective
1	Find intercepts from a graph and analyze symmetry.
2	Know how to graph key equations.
3	Determine which points are on the graph of an equation.
4	Solve applications and extensions.
5	Calculate and interpret the slope of a line.
6	Graph lines given a point and the slope.
7	Identify the slope and y-intercept of a line from its equation.
8	Given the equations of two lines, determine if they are parallel, perpendicular, or neither.
9	Write the equation of a circle with given characteristics.
10	Determine whether a relation represents a function.
11	Find the domain of a function defined by an equation.
12	Determine if a graph represents a function, and find the domain, range, intercepts, and symmetry.
13	Use a graph to determine where a function is increasing, decreasing, or constant.
14	Use a graph to locate maxima and minima.
15	Identify even and odd functions from an equation.
16	Graph functions using vertical and horizontal shifts.
17	Graph functions using compressions and stretches.
18	Graph functions using reflections about the x-axis and the y-axis.
19	Graph functions using a series of transformations.
20	Build and analyze functions using geometric formulas.

Following the principle of reproducible research, the source code for the computations will appear.

First we clear old functions and datasets from the *R* session, display loaded packages, and load the dataset.

```
rm(list = ls())
search()
```

```
##  [1] ".GlobalEnv"        "package:psych"     "package:ellipse"
##  [4] "package:corrplot"  "package:randtests" "package:MASS"
##  [7] "package:knitr"     "package:stats"     "package:graphics"
## [10] "package:grDevices" "package:utils"     "package:datasets"
## [13] "package:methods"   "Autoloads"         "package:base"
```

```
colnames(data) <- c('test','quiz',1:20,'semester')
```

Not all students took both the quiz and test 1; some took one of the two, some took neither. During the data preparation phase of the project, students were sorted by their performance on the test and the quiz. We used these vectors in *R* to indentify which rows correspond to which type of results. These groups of rows are sorted by performance, better to worse.

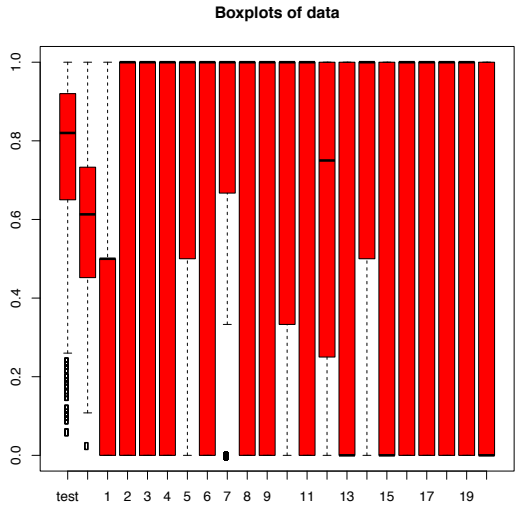

Figure 1: Boxplots of each questions scores

```
both <- 1:1044
quiz.only <- 1045:1128
test.only <- 1129:1146
neither <- 1147:1167


par(mfrow = c(1,1))
boxplot(data[,-23],main = 'Boxplots of data',col = 'red')
```

The box plots show that both semesters had approximately the same average performance, and same sample variance. Standard statistical tests verify that the semester averages are statistically equal and that semester variances are statistically equal. Thus, the data from both semesters combine for the analysis.

```
par(mfrow = 1:2)
boxplot(data[,1]~data[,23],main = 'Boxplots of Test',col = 'orange')
boxplot(data[,2]~data[,23],main = 'Boxplots of Quiz',col = 'orange')


test.fall <- data[data[,23] == 'fall2014',1]
test.fall <- test.fall[!is.na(test.fall)]
test.spring <- data[data[,23] == 'spring2015',1]
test.spring <- test.fall[!is.na(test.spring)]
var.test(test.fall,test.spring)

##
##  F test to compare two variances
##
```

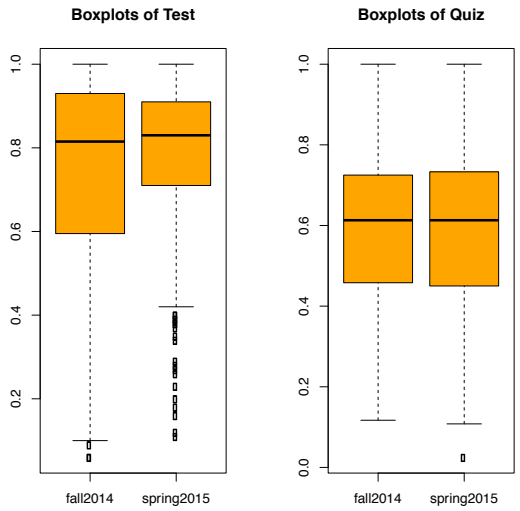

Figure 2: Boxplots of each semester's quiz and test scores

```
## data:  test.fall and test.spring
## F = 1.0291, num df = 587, denom df = 549, p-value = 0.7337
## alternative hypothesis:
##      true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.8725745 1.2130784
## sample estimates:
## ratio of variances
##           1.029072

ansari.test(test.fall,test.spring,
            alternative = "two.sided",
            exact = NULL,
            conf.int = FALSE,
            conf.level = 0.95)

##
##  Ansari-Bradley test
##
## data:  test.fall and test.spring
## AB = 165360, p-value = 0.4215
## alternative hypothesis: true ratio of scales is not equal to 1

t.test(test.fall,test.spring,var.equal = TRUE)

##
##  Two Sample t-test
```

```
##
## data:  test.fall and test.spring
## t = -1.3314, df = 1136, p-value = 0.1833
## alternative hypothesis:
##   true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.045563595  0.008725376
## sample estimates:
## mean of x mean of y
## 0.7429082 0.7613273

wilcox.test(test.fall,test.spring)

##
##  Wilcoxon rank sum test with continuity correction
##
## data:  test.fall and test.spring
## W = 154062, p-value = 0.1677
## alternative hypothesis: true location shift is not equal to 0

quiz.fall <- data[data[,23] == 'fall2014',2]
quiz.fall <- quiz.fall[!is.na(quiz.fall)]
quiz.spring <- data[data[,23] == 'spring2015',2]
quiz.spring <- quiz.fall[!is.na(quiz.spring)]
var.test(quiz.fall,quiz.spring)

##
##  F test to compare two variances
##
## data:  quiz.fall and quiz.spring
## F = 0.9984, num df = 630, denom df = 615, p-value = 0.9842
## alternative hypothesis:
##   true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.8530777 1.1683933
## sample estimates:
## ratio of variances
##          0.9984392

ansari.test(quiz.fall,quiz.spring,
            alternative = "two.sided",
            exact = NULL,
            conf.int = FALSE,
            conf.level = 0.95)

##
##  Ansari-Bradley test
```

```
##
## data:  quiz.fall and quiz.spring
## AB = 197222, p-value = 0.9518
## alternative hypothesis: true ratio of scales is not equal to 1

t.test(quiz.fall,quiz.spring,var.equal = TRUE)

##
##  Two Sample t-test
##
## data:  quiz.fall and quiz.spring
## t = -0.0962, df = 1245, p-value = 0.9234
## alternative hypothesis:
##   true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.02141533  0.01941313
## sample estimates:
## mean of x mean of y
## 0.5895119 0.5905130

wilcox.test(quiz.fall,quiz.spring)

##
##  Wilcoxon rank sum test with continuity correction
##
## data:  quiz.fall and quiz.spring
## W = 193646, p-value = 0.9121
## alternative hypothesis: true location shift is not equal to 0
```

There appears to be a difference between the performance of students that took both assignments and students that only took one. We conjecture that students who miss an assignment perform worse on the other than students who took both.

4 An Initial Look at Correlation

Standard statistical techniques relating total quiz scores to test scores show that it is reasonable to look into the relationship between quiz questions and test scores. The scatter plot of the quiz and test scores and their simple linear model shows weak correlation that is statistically significant. This simple linear regression model shows that it is reasonable to continue looking into the relationships between questions [7].

```
plot(data[,2:1],
  main = 'Scatter plot of test scores versus quiz scores')
lm(data[,1:2])
```

```
##
## Call:
## lm(formula = data[, 1:2])
##
## Coefficients:
## (Intercept)         quiz
##      0.4875       0.4641

summary(lm(data[,1:2]))

##
## Call:
## lm(formula = data[, 1:2])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.63487 -0.09374  0.04349  0.12933  0.38203
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.48753    0.02032    24.0   <2e-16 ***
## quiz         0.46406    0.03244    14.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1896 on 1042 degrees of freedom
##   (123 observations deleted due to missingness)
## Multiple R-squared:  0.1641,Adjusted R-squared:  0.1633
## F-statistic: 204.6 on 1 and 1042 DF,  p-value: < 2.2e-16

anova(lm(data[,1:2]))

## Analysis of Variance Table
##
## Response: test
##             Df Sum Sq Mean Sq F value    Pr(>F)
## quiz         1  7.356  7.3556  204.62 < 2.2e-16 ***
## Residuals 1042 37.458  0.0359
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

abline(lm(data[,1:2]),col = 'blue')


dev.off()

## null device
##           1
```

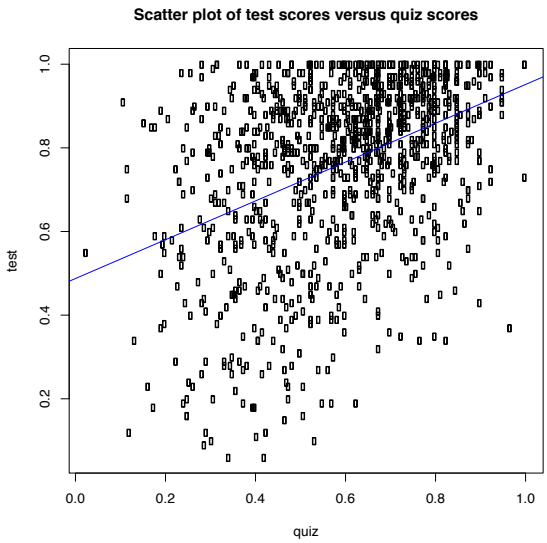

Figure 3: Plot of the observed quiz, test pairs

```
require(MASS)
```

Furthermore, the Cox-Stuart test shows that the relationship between quizzes and test is statistically significant. Hence, it is reasonable to construct a generalized linear model [2].

```
require(randtests)
cox.stuart.test(
  data[order(data[both,2]),1],
  alternative = 'right.sided')

##
##  Cox Stuart test
##
## data:  data[order(data[both, 2]), 1]
## statistic = 360, n = 513, p-value < 2.2e-16
## alternative hypothesis: increasing trend
```

Since the goal is to identify which questions are the least useful, let's look at how the question predictor vectors are correlated. We will use Pearson's, Kendall's and Spearman's measures of correlation. Then we compare the correlation matrices using a matrix norm [9, 3].

```
correlation.matrix.pearson <- cor(
  data[,-23],
use = "pairwise.complete.obs",
method = 'pearson')
```

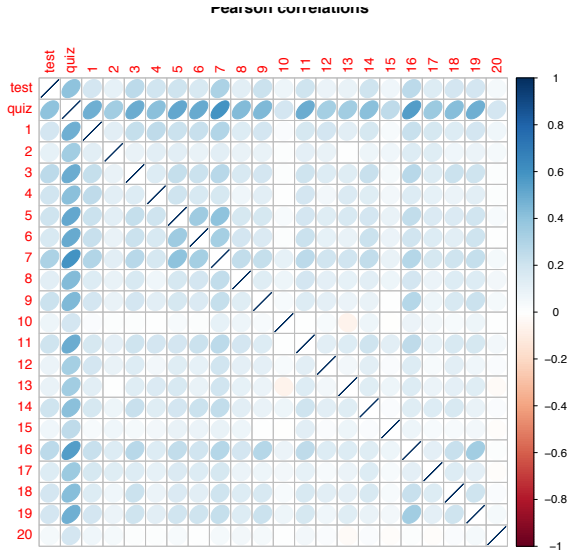

Figure 4: Ellipse plot of Pearson correlation between question responses

```
colnames(correlation.matrix.pearson) <- c('test','quiz',1:20)
rownames(correlation.matrix.pearson) <- c('test','quiz',1:20)
round(correlation.matrix.pearson,1)

require(corrplot)
require(ellipse)
corrplot(correlation.matrix.pearson,title = '',method = 'ellipse')
title('Pearson correlations',outer = TRUE)


dev.off()

## null device
##           1


correlation.matrix.kendall <- cor(
  data[,-23],
use = "pairwise.complete.obs",
method = 'kendall')
colnames(correlation.matrix.kendall) <- c('test','quiz',1:20)
rownames(correlation.matrix.kendall) <- c('test','quiz',1:20)
round(correlation.matrix.kendall,1)

require(corrplot)
require(ellipse)
corrplot(correlation.matrix.kendall,title = '',method = 'ellipse')
title('Kendall correlations',outer = TRUE)
```

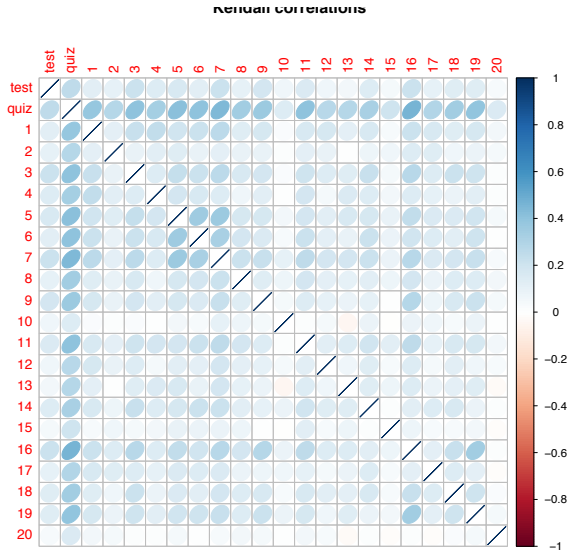

Figure 5: Ellipse plot of Kendall correlation between question responses

```
dev.off()

## null device
##           1


correlation.matrix.spearman <- cor
  (data[,-23],
use = "pairwise.complete.obs"
,method = 'spearman')
colnames(correlation.matrix.spearman) <- c('test','quiz',1:20)
rownames(correlation.matrix.spearman) <- c('test','quiz',1:20)
round(correlation.matrix.spearman,1)

require(corrplot)
require(ellipse)
corrplot(correlation.matrix.spearman,title = '',method = 'ellipse')
title('Spearman correlations',outer = TRUE)


dev.off()

## null device
##           1
```

Let's use the matrix two-norm to see how similar these three correlation matrices are to each other.

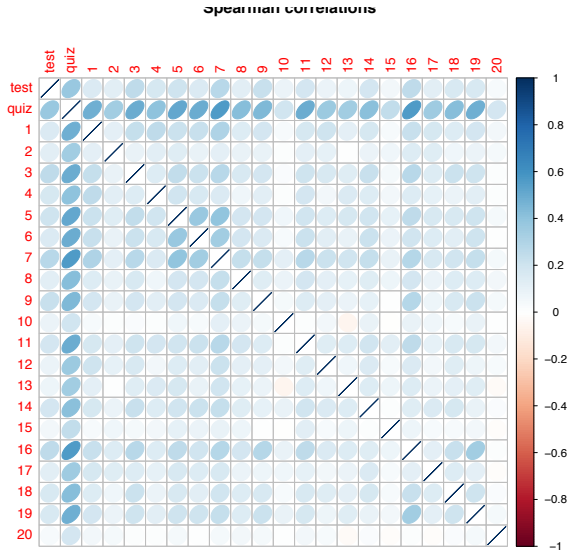

Figure 6: Ellipse plot of Spearman correlation between questions responses

```
correlation.differences <- matrix(0,nrow = 3,ncol = 3)
correlation.differences[1,2] <- norm
  (correlation.matrix.pearson
  - correlation.matrix.kendall,
  type = '2')
correlation.differences[1,3] <- norm
  (correlation.matrix.pearson
  - correlation.matrix.spearman,
  type = '2')
correlation.differences[2,3] <- norm
  (correlation.matrix.kendall
  - correlation.matrix.spearman,
  type = '2')
correlation.differences[2,1] <- correlation.differences[1,2]
correlation.differences[3,1] <- correlation.differences[1,3]
correlation.differences[3,2] <- correlation.differences[2,3]
rownames(correlation.differences) <- c('Pearson','Kendall','Spearman')
colnames(correlation.differences) <- c('Pearson','Kendall','Spearman')
correlation.differences
```

```
##            Pearson   Kendall  Spearman
## Pearson  0.0000000 0.5333733 0.1248185
## Kendall  0.5333733 0.0000000 0.4911891
## Spearman 0.1248185 0.4911891 0.0000000
```

We see that Pearson and Spearman give similar results. Each question score has discrete values with constant differences between possible values, there would be

little gained from using nonparametric regression. Since straight line correlation is close to monotone correlation, little would be gained from using robust methods.

5 A Graph Theory Approach to Correlation

Cycle intersection matrices from graph theory is an established method to identify concepts that can be combined. Phifer used minimum scores on questions with binary correct/incorrect responses to identify which precalculus in a calculus program questions could be combined [4]. Here we modify the proceedure and take each student's minimum score for each pair of questions. Then sum the minimum scores for each pair of questions. If questions form a complete subgraph (clique), then those concepts should be considered for combining. Phifer used weighted graphs with ten nodes to identify cliques. Twenty nodes is difficult to interprete, thus we used a simple binary adjacency matrix. The matrix is defined as

$$M_{jk} = \sum_{s=1}^{N} \min(x_{sj}, x_{sk}) \tag{1}$$

where N is the number of students, s is the index of students, x_{sj} is the score student s earned on question j.

When questions j and k are scored as correct or incorrect, such as multiple choice questions, X_{sj}, X_{sk} and their minimum are binary variables. If these are assumed to be independent Bernoulli random variables with nonconstant probabilities of success, then M_{jk} is a J-binomial random variable [1]. Further, since probabilities of success are on a continuum of $[0, 1]$, one may assume that each student and question combination has a distinct probability of resulting in a correct response. Say that p_{sj} is the probability that student s gets question j correct. Under these assumptions,

$$E(M_{jk}) = \sum_{s=1}^{N} p_{sj}p_{sk}, \tag{2}$$

$$V(M_{jk}) = \sum_{s=1}^{N} p_{sj}p_{sk}(1 - p_{sj}p_{sk}), \tag{3}$$

$$G_{jk}(t) = \prod_{s=1}^{N} (1 - p_{sj}p_{sk} + p_{sj}p_{sk}t). \tag{4}$$

The probability of specific events can be computed with routine convolutions. The central limit theorem and the above expected values allows for computationally easier approximations of M_{jk}'s probabilities. If one wishes to approximate a J-binomial random variable with a single binomial random variable using the average probability of success, the variance of the binomial random variable will always be greater [1].

```
X <- (X+1)/2
```

```
graph.matrix.min <- matrix(0,nrow = 20,ncol = 20)
rownames(graph.matrix.min) <- 1:20
colnames(graph.matrix.min) <- 1:20
for(j in 1:1044) for(k in 1:20) for(l in k:20){
  graph.matrix.min[k,l] <- graph.matrix.min[k,l] + min(X[j,c(k,l)])
}
graph.matrix.min <- graph.matrix.min + t(graph.matrix.min)
diag(graph.matrix.min) <- diag(graph.matrix.min)/2
round(graph.matrix.min)
```

From this matrix, a binary adjacency matrix will be built. Larger values will result in a one, smaller values will correspond to zero. A cut off value for zero and one needs to be established. A common cut for boxplot outliers is 1.4 times a measure of data spread; here we choose

$$A_{jk} = 1 \; if \; M_{jk} > \min(M) + 1.5\text{midrange}(M)/2, \tag{5}$$

$$A_{jk} = 0 \; otherwise. \tag{6}$$

```
d.min <- diff(range(graph.matrix.min))/2
min.min <- min(graph.matrix.min)
adjacency.min <- matrix(0,nrow = 20,ncol = 20)
for(j in 1:20) for(k in 1:20){
  if(graph.matrix.min[j,k]>
  min.min + 1.4*d.min) adjacency.min[j,k] <- 1
}
rownames(adjacency.min) <- 1:20
colnames(adjacency.min) <- 1:20
adjacency.min[
  order(graph.matrix.min %*% rep(1,20)),
order(graph.matrix.min %*% rep(1,20))]
```

From the adjacency matrix corresponding to the cycle intersection matrix, we can see that questions 3, 5, 7, and 14 should be considered for replacing with one question.

question	learning objective
3	Determine which points are on the graph of an equation.
5	Calculate and interpret the slope of a line.
7	Identify the slope and y-intercept of a line from its equation.
14	Use a graph to locate maxima and minima.

Looking at which questions are similar, mastery of question 3 could be represented by mastery of question 14. Mastery of 7 could represent mastery of 5.

6 Conclusion

Creating a test or quiz that sufficiently evaluate subject mastery while using a minimum number of questions is a challege. Extra problems waste resources,

students' time, and increases the burden of teaching a class. Treating a quiz or test as a designed experiment opens the opportunity to use statistical techniques to build better assignments. Techniques from regression are well developed and well understood. Reducing a regression model is a natural way to identify questions that do not contribute to evaluating subject mastery. A disadvantage of regression techniques is that is depends on a response variable.

The graph theory approach of cliques gives faculty a tool to identify which questions are redundant. An advantage of cliques is that this approach uses on the predictor variables and does not require the faculty to come up with a response variable.

References

[1] James C Benneyan and DA Borgman. A useful j-binomial type distribution for non-homogeneous dichotomous events. In *Industrial Engineering Research Conference Proceedings*, pages 1–6, 2004.

[2] Frederico Caeiro and Ayana Mateus. *randtests: Testing randomness in R*, 2014. R package version 0.3.

[3] Duncan Murdoch and E. D. Chow. *ellipse: Functions for drawing ellipses and ellipse-like confidence regions*, 2013. R package version 0.3-8.

[4] Caitlin R Phifer. The cycle intersection matrix and applications to planar graphs and data analysis for postsecondary mathematics education. 2014.

[5] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014.

[6] Press The Associated. After backlash florida puts limits on standardized testing. *AP Regional State Report - Florida*, 2015.

[7] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. ISBN 0-387-95457-0.

[8] Howard Wainer and Richard Feinberg. For want of a nail: Why unnecessarily long tests may be impeding the progress of western civilisation. *Significance*, 12(1):16 – 21, 2015.

[9] Taiyun Wei. *corrplot: Visualization of a correlation matrix*, 2013. R package version 0.73.

[10] Yihui Xie. *Dynamic Documents with R and knitr*. CRC Press, 2013.

[11] Yihui Xie. knitr: A general-purpose package for dynamic report generation in r. *R package version*, 1(7), 2013.

[12] Yihui Xie. knitr: a comprehensive tool for reproducible research in r. *Implementing Reproducible Research*, page 1, 2014.