Planejamento e Análise de Experimentos (EEE933)
Estudo de Caso 1

Pedro Vinícius, Samara Silva e Savio Vieira

10 de Agosto de 2020

Introdução

Uma versão atual de um software conhecido apresenta uma distribuição dos custos de execução com média populacional $\mu_c = 50$ e variância populacional $\sigma_c^2 = 100$. Posteriormente, uma nova versão do software é desenvolvida, no qual deseja-se investigar prováveis melhorias de desempenho. Com esse intuito, duas análises estatísticas são propostas: (i) uma sobre o custo médio e (ii) uma sobre a variância do custo.

Para ambos os casos, as hipóteses nulas foram definidas de maneira conservadora, partindo-se do pressuposto de que os parâmetros populacionais conhecidos foram mantidos na nova versão. A partir disso, diversas etapas foram conduzidas até a conclusão dos experimentos, entre elas a coleta de dados da distribuição dos custos do software novo, a análise exploratória dessa amostra, a inferência por meio dos testes estatísticos, a validação das premissas consideradas e as conclusões. As próximas seções contém o detalhamento técnico de cada uma dessas etapas.

Parte 1: Teste Sobre o Custo Médio

Planejamento dos Experimentos

No que se refere a primeira parte do estudo de caso, o teste terá que dispor de um nível de significância $\alpha = 0,01$, um tamanho de efeito de mínima relevância $\delta^* = 4$ e uma potência desejada $\pi = 1 - \beta = 0,8$. As hipóteses estatísticas foram definidas com o intuito de responder às questões propostas abaixo:

- Há alguma diferença entre o custo médio da versão nova do software e o custo médio da versão corrente?
- Caso haja, qual a melhor versão em termos de custo médio?

Em concordância com a proposta de comparação de custo médio entre as versões, as hipóteses de teste podem ser formuladas sobre o parâmetro média:

$$\begin{cases} H_0 : \mu_n = 50 \\ H_1 : \mu_n < 50 \end{cases}$$

onde a hipótese nula implica na igualdade entre os custos médios das versões e a hipótese alternativa unilateral na superioridade da nova versão em média.

A fase subsequente desse experimento consiste em gerar uma amostra representativa do desempenho da nova versão do software. Para tal fim, é necessário especificar o tamanho dessa amostra, considerando as propriedades preestabelecidas do teste. A priori, o Poder do Teste é bastante conveniente, porém implica em um grande dilema. O cálculo do tamanho amostral requer uma estimativa da variância, que só é obtida através das observações contidas na amostra. As possibilidades mais práticas de se conduzir o experimento nesse caso são [2]:

1. Utilização de conhecimento do problema para se obter uma estimativa (inicial) da variância;
2. Condução do estudo com um tamanho amostral predefinido, como $N = 30$, o que poderia violar a potência desejada;

3. Realização de um estudo piloto para estimar a variância dos dados a partir do tamanho de efeito de mínima relevância δ^*.

Considerando as vantagens e desvantagens de cada uma, optou-se por utilizar a primeira abordagem. Por mais que essa estimativa seja sobre-estimada e os prováveis ganhos não sejam observados ao término do estudo, uma vez que se espera ganhos de variância da nova versão do software em relação à versão atual, pode-se considerar igualdade de variâncias como uma estimativa inicial, ou seja, $\sigma_n^2 \approx \sigma_c^2 = 100$. Entretanto, essa premissa será avaliada posteriormente na análise exploratória dos dados.

Diante da estimativa inicial da variância amostral, o Poder do Teste pode ser finalmente realizado. Esse teste é originalmente usado para mensurar o controle do teste de hipóteses sobre o erro do tipo II (β), isto é, $P(\text{rejeitar } H_0 | H_0 \text{ é falso})$. No entanto, tal teste também pode ser utilizado para estimar outros parâmetros amostrais, como tamanho de efeito δ^*, nível de significância α, tamanho da amostra N, potência π e desvio padrão amostral σ_n [11]. No presente estudo, ele é utilizado para estimar o tamanho amostral N.

```r
# Poder do Teste para estimar o tamanho amostral
(params <- power.t.test(delta = 4,
             sd = 10,
             sig.level = 0.01,
             power = 0.8,
             type = "one.sample",
             alternative = "one.sided"))
```

```
##
##      One-sample t test power calculation
##
##               n = 65.45847
##           delta = 4
##              sd = 10
##       sig.level = 0.01
##           power = 0.8
##     alternative = one.sided
```

```r
N <- ceiling(params$n)
```

Assim, tem-se uma estimativa de $N = 66$ observações com o arredondamento superior.

Coleta dos Dados

O custo de execução do software novo é modelado por um algoritmo de Evolução Diferencial que faz uso de recombinação binomial, mutação aleatória e seleção elitista padrão para a minimização do *Shifted Sphere Problem*. Dessa forma, como cada observação é obtida por meio do módulo `ExpDE` em linguagem R [1], foram necessárias N execuções para construção da amostra. Os dados coletados foram salvos em um arquivo `.csv` com uma semente definida empiricamente `seed = 1007`. Uma vez que a amostra está fixa, as análises estatísticas da média e da variância poderão ser efetuadas sobre os mesmos dados.

```r
data_generation <- function(n){

  mre <- list(name = "recombination_bin", cr = 0.9)
  mmu <- list(name = "mutation_rand", f = 2)
  mpo <- 100
  mse <- list(name = "selection_standard")
  mst <- list(names = "stop_maxeval", maxevals = 10000)
  mpr <- list(name = "sphere", xmin = -seq(1, 20), xmax = 20 + 5 * seq(5, 24))
```

```
  sample <- c()
  # Geração de N observações
  for (i in 1:n){
    observation <- ExpDE(mpo, mmu, mre, mse, mst, mpr,
                         showpars = list(show.iters = "none"))$Fbest
    sample <- c(sample, observation)
  }
  return(sample)
}
```

Análise Exploratória de Dados

Algumas primeiras propriedades da amostra, como média, mediana, valores extremos e variância, podem ser obtidas de imediato.

```
data.frame('Variância' = var(sample), 'Média' = mean(sample),
                                  'Mediana' = median(sample), 'Mínimo' = min(sample),
                                  'Máximo' = max(sample), 'Desvio' = sd(sample))
```

```
##   Variância    Média  Mediana   Mínimo   Máximo   Desvio
## 1   36.6344 49.63844 48.92703 38.76112 67.21031 6.052636
```

É importante ressaltar que a variância é significativamente menor que a considerada no cálculo do tamanho amostral N, assim, caso as premissas consideradas sejam válidas, a potência do teste será superior a desejada ($\pi > 0,8$).

Entretanto, a fim de compreender melhor os dados em estudo, algumas representações gráficas da amostra coletada serão investigadas. Inicialmente, faz-se o uso do histograma, que representa a distribuição de frequências das observações. O histograma é uma boa ferramenta de análise visual do comportamento de variáveis contínuas em geral, auxiliando no processo de visualização de características, como a média da amostra e a variação da distribuição em torno da mesma. Para a amostra em questão, o histograma mostra uma distribuição levemente inclinada à direita, que corrobora que a média é maior do que a moda, bem como sugere a presença de outliers com custos maiores que 60 (Figura 1).

```
histogram <- ggplot(data = as.data.frame(sample), mapping = aes(x = sample))
histogram + geom_histogram(lwd = 0.3, bins = 20, color = 'black', fill = 'gray') +
            scale_x_continuous(name = 'Custo de Execução') +
            scale_y_continuous(name = 'Frequência')
```

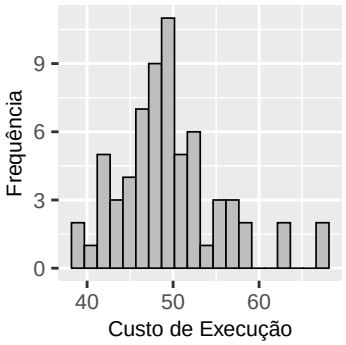

Figura 1: Histograma.

Posteriormente, o boxplot, também conhecido como gráfico de caixa, é considerado. Além de descrever simultaneamente várias propriedades importantes de um conjunto de dados, como centro, dispersão e desvio de simetria, ele também possiblita a identificação de valores atípicos [9]. Para a amostra em questão, foi possível reforçar a presença das observações em torno da mediana levemente inferior a 50 (segundo quartil). Outro aspecto importante é a presença de uma assimetria positiva na distribuição, uma vez que a mediana está mais próxima do primeiro do que do terceiro quartil. A diferença nos comprimentos das linhas que saem da caixa pressupõe novamente a cauda acentuada à direita (Figura 2).

```
boxplot <- ggplot(data = as.data.frame(sample), mapping = aes(y = sample))
boxplot + geom_boxplot(lwd = 0.3) + scale_x_continuous(name = '') +
          scale_y_continuous(name = 'Custo de Execução') +
          theme(axis.text.x = element_blank())
```

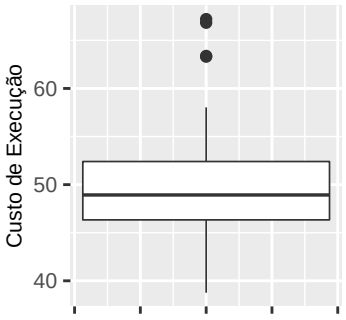

Figura 2: Boxplot.

Por fim, um gráfico bastante utilizado em estatística é o quantil-quantil (*QQ-Plot*), cujo objetivo é verificar a adequação da distribuição de frequência dos dados à uma distribuição de probabilidades de interesse. Caso os dados tenham distribuições idênticas, uma linha reta com inclinação unitária é traçada. A densidade com que os pontos são representados indicam a frequência de ocorrência dos mesmos [8]. No presente trabalho, o gráfico Q-Q foi utilizado para comparar a distribuição dos custos de execução da nova versão do software com uma distribuição normal. Ao gerá-lo, é evidente que os dados não seguem um padrão linear e, portanto, não provém de uma distribuição normal (Figura 3).

```
qqplot <- ggplot(data = as.data.frame(sample), mapping = aes(sample = sample))
qqplot + geom_qq_line() + geom_qq() + scale_y_continuous(name = 'Quantis da Amostra') +
         scale_x_continuous(name = 'Quantis Teóricos Normais')
```

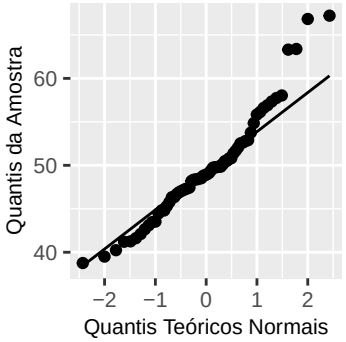

Figura 3: QQ-Plot.

Validação de Premissas

Devido à presumível falta de normalidade dos dados, o teste de Shapiro-Wilk foi feito para confirmação. A hipótese nula H_0 desse teste assume que a amostra veio de uma população com distribuição normal e para isso o parâmetro W é calculado e comparado com um valor de referência. Caso a resultante de W seja inferior ao valor tabelado, rejeita-se a hipótese de normalidade a um nível de significância α. A aplicação do teste gerou como resultado $W = 0,94772$ e o valor de $p = 0,00755$, que confirmam a não normalidade dos dados ($p < 0,05$).

```
(shapiro_test <- shapiro.test(x = sort(sample)))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sort(sample)
## W = 0.94772, p-value = 0.007555
```

Apesar disso, como o número de observações é suficientemente grande ($N = 66 \gg 30$), o Teorema do Limite Central (TLC) pôde ser evocado para estimação da distribuição amostral das médias. Já a premissa de independência foi assumida no processo de coleta dos dados. Caso uma amostra do software atual estivesse disponível, o teste de Durbin-Watson também caberia para avaliar uma possível autocorrelação nos resíduos.

Análise Estatística

A partir disso, pode-se realizar o Teste t de Student sem que suas premissas sejam violadas.

```
(t_test <- t.test(x = sample,
                  mu = 50,
                  alternative = "less",
                  conf.level = 0.99))
```

```
##
##  One Sample t-test
##
## data:  sample
## t = -0.4853, df = 65, p-value = 0.3145
## alternative hypothesis: true mean is less than 50
## 99 percent confidence interval:
##     -Inf 51.4154
## sample estimates:
## mean of x
##  49.63844
```

```
cat('Intervalo de confiança:', t_test$conf.int[1:2])
```

```
## Intervalo de confiança: -Inf 51.4154
```

Como o valor p encontrado é maior que o nível de significância estabelecido ($0,3145 > 0,05$), conclui-se que a hipótese nula H_0 não pode ser rejeitada a um nível de confiança de 95%. No contexto do problema estudado e para as configurações de teste definidas, uma falha ao rejeitar a hipótese nula significa que não há evidências de que os desempenhos médios das duas versões do software sejam distintos e, portanto, trata-se de uma conclusão fraca. Além disso, dado que a hipótese alternativa H_1 é unilateral, tem-se um intervalo de confiança aberto à esquerda $[-\infty,\ 51,4154]$.

Potência do Teste

Conforme relatado anteriormente, como a estimativa da variância amostral foi substancialmente superior à variância calculada sobre os dados, é bem provável que a potência do teste seja superior aos 80% utilizados para o cálculo do tamanho amostral N. Dessa forma, uma estimativa da potência para detecção de diferenças maiores ou iguais ao tamanho de efeito δ^* pode ser obtida pelo Poder do Teste, utilizando o desvio padrão amostral $\sigma = 6,0526$, o tamanho amostral $N = 66$ e conservando as demais propriedades.

```
(params <- power.t.test(delta = 4,
             sd = sqrt(var(sample)),
             sig.level = 0.01,
             n = 66,
             type = "one.sample",
             alternative = "one.sided"))
```

```
##
##      One-sample t test power calculation
##
##               n = 66
##           delta = 4
##              sd = 6.052636
##       sig.level = 0.01
##           power = 0.9982989
##     alternative = one.sided
```

```
cat('Potência obtida:', params$power)
```

```
## Potência obtida: 0.9982989
```

Em concordância com o que se esperava, a potência obtida $\pi = 0,9982$ é superior à desejada.

Outra perspectiva para esse caso é que, consequentemente, o número de observações na amostra foi sobre-estimado. Assim, uma estimativa para N capaz de garantir exatamente a potência desejada $\pi = 0,8$ pode ser obtida pelo Poder de Teste ao utilizar o desvio padrão amostral em conjunto com os demais parâmetros.

```
(params <- power.t.test(delta = 4,
             sd = sqrt(var(sample)),
             sig.level = 0.01,
             power = 0.8,
             type = "one.sample",
             alternative = "one.sided"))
```

```
##
##      One-sample t test power calculation
##
##               n = 25.752
##           delta = 4
##              sd = 6.052636
##       sig.level = 0.01
##           power = 0.8
##     alternative = one.sided
```

```r
cat('Tamanho amostral:', ceiling(params$n))
```

```
## Tamanho amostral: 26
```

Ou seja, o tamanho amostral necessário para o experimento é de apenas 26 observações, 40 observações a menos do que foi utilizado na estimativa inicial.

Parte 2: Teste Sobre a Variância do Custo

Planejamento dos Experimentos

Em relação à segunda parte deste experimento, isto é, quanto a avaliação de uma potencial melhoria na variância do custo de execução, foi considerado como hipótese nula H_0 a equivalência das variâncias entre as duas versões e como hipótese alternativa H_1 a redução da variância do custo do software nessa nova versão em análise.

$$\begin{cases} H_0 : \sigma_n^2 = 100 \\ H_1 : \sigma_n^2 < 100 \end{cases}$$

Tentativas Iniciais

Existem vários testes estatísticos para comparação de variância, como o teste F, o teste de Bonett, o teste Chi-quadrado, o teste de Levene, dentre outros. Todavia, além das particularidades de cada um deles, os três primeiros são indicados para distribuições normais. Além disso, tendo em vista que o TLC não é aplicável para o estimador amostral da variância, uma das opções recomendadas é o uso de um teste não-paramétrico, como o teste de Levene.

O Teste de Levene tem como hipótese nula a proposição de que os grupos que estão sendo comparados têm variâncias iguais, enquanto que a hipótese alternativa é que pelo menos um par desses grupos não possui igualdade de variância. Entretanto, essa categorização não comporta a situação em questão, visto que o intuito é a comparação da amostra da versão nova com o valor de referência da versão atual. Nesse âmbito, um teste não-paramétrico apropriado deve receber somente uma amostra para compará-la com uma variância populacional conhecida, o que, portanto, descarta o teste de Levene para o problema em questão.

Em virtude disso, bem como da escassez de métodos não-paramétricos compatíveis aos dados, outras abordagens foram investigadas. A primeira delas trata-se de uma transformação nos dados com o intuito de torná-los normais. As transformações monotônicas não-lineares $\log_{10}$ e quadrática mudam a distância entre os valores na distribuição e, portanto, a forma da distribuição [12]. A primeira calcula o logaritmo na base 10 de um grupo de números contidos na calda à direita da distribuição para reduzir a inclinação positiva, enquanto a segunda toma a raiz quadrada de grandes valores, levando os valores altos para mais próximo do centro. Contudo, as tentativas de transformações não asseguraram normalidade aos dados.

Solução Proposta

Por fim, a possibilidade encontrada para testar as hipóteses de teste foi a reamostragem dos dados. Essa técnica consiste em descartar a distribuição amostral assumida de uma estatística e calcular uma distribuição empírica à sua real distribuição. Assim, uma vez que essa distribuição é estimada, testes estatísticos podem ser realizados, intervalos de confiança podem ser construídos e hipóteses podem ser testadas sem que seja necessário validar premissas inerentes à distribuição [10].

Dentre os métodos da literatura mais comuns estão *Jackknife* e *Bootstrap* [6]. Ambas as técnicas são bastante utilizadas para avaliar a variância de um estimador a partir da sua distribuição empírica. No entanto, o *Jackknife* é mais conservador e, consequentemente, produz erros padrões maiores [5]. Em vista disso, optou-se pelo uso da técnica de *Bootstrap* do tipo não-paramétrico para estimar a distribuição da variância amostral do software novo, cuja região coberta pelo intervalo de confiança permitirá testar a hipótese de interesse ao nível de significância previamente definido em $\alpha = 0,05$.

A partir do módulo `boot` em linguagem R [3], que contempla funções e conjuntos de dados para reamostragem do livro *Bootstrap Methods and Their Application* [4], foi possível realizar o experimento

proposto. Uma estimativa ideal da distribuição empírica por reamostragem *Bootstrap* leva o número de replicações ao infinito e, em regras gerais, utiliza-se um número relativamente grande ($R > 1.000$) [7]. Como o processo do presente estudo não demanda muito esforço computacional, o número de replicações para estimar a distribuição amostral da função estatística variância foi definido em $R = 10.000$, de forma a minimizar o erro de amortecimento intrínseco ao *Bootstrap* devido à reamostragem finita. Por fim, é possível encontrar o intervalo de confiança correspondente à distribuição estimada. A função `boot.ci` permite gerar cinco tipos diferentes de intervalo de confiança não-paramétricos e bilaterais: (i) *basic*, (ii) *studentized*, (iii) *percentile*, (iv) *adjusted percentile* (BCa) e (v) *first order normal approximation* [3].

A priori, o tipo de intervalo definido foi o BCa, uma vez que ele é um intervalo mais preciso (segunda ordem) e permite ajustar distorções na distribuição, como viés e assimetria. Se a distribuição empírica foi distorcida positivamente, o intervalo de confiança é ajustado para a direita. Caso a distribuição seja distorcida negativamente, o intervalo é ajustado para a esquerda [4].

Conforme especificado anteriormente, deseja-se ainda um nível de confiança de 95%, entretanto, a estimativa do intervalo de confiança obtida pelo `boot.ci` é bilateral. Uma potencial alternativa para esse caso é ajustar o nível de confiança para 90%, de tal forma que cada lado do intervalo tenha a probabilidade de erro do tipo I desejada $\alpha = 0,05$ [9].

Supondo novamente que os dados são independentes e identicamente distribuídos em razão do seu processo de coleta e que a reamostragem já foi calculada com R suficientemente grande, é possível estimar o intervalo de confiança para o estimador variância amostral. A reprodutibilidade do experimento é possível a partir da semente aleatória fixa em `seed = 1007`.

```
# Função estatística
statistic_function <- function(x, i){
  variance <- var(x[i])
  return(variance)
}

# Semente aleatória
set.seed(1007)

# Bootstrap com 10000 replicações
resampling <- boot(data = sample, statistic = statistic_function, R = 10000)

# Intervalo de confiança não-paramétrico
boot.ci(boot.out = resampling, conf = 0.90, type = 'bca')
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = resampling, conf = 0.9, type = "bca")
##
## Intervals :
## Level       BCa
## 90%   (26.21, 53.07 )
## Calculations and Intervals on Original Scale
```

É possível afirmar, ao nível de confiança de 95%, que a variância da nova versão do software é substancialmente inferior à variância da versão atual ($53,07 \ll 100$). Com o intuito de corroborar a ausência de influência do tipo de intervalo de confiança definido previamente no resultado, o teste foi feito para os demais tipos e pode-se concluir que em todos os casos há evidências estatísticas de que a versão nova é superior à versão atual em termos da variância do custo de execução.

Conclusões

Com o objetivo de inferir sobre a melhora de desempenho da nova versão de um software em comparação com a versão atual, foi utilizado de conceitos e ferramentas estatísticas para investigar e concluir sobre esta comparação.

Ficou a cargo de investigar sobre dois pontos principais, custo médio de execução e sobre a variância no custo de execução. Sobre o custo médio de execução, conclui-se que por não ter sido rejeitada a hipótese nula de que o custo médio de execução é igual a 50, neste caso, não pode-se inferir sobre a melhoria da nova versão sobre a versão atual. Vale lembrar que não rejeitar a hipótese nula não significa que ela é verdadeira, e sim que pelos testes realizados, com 95% de confiança, não conseguiu-se provar sua falsidade. Porém, sobre a variância do custo da execução, a hipótese nula de que a variância é igual a 100 foi rejeitada, neste caso, com os testes realizados, com 95% de confiança, a hipótese alternativa foi aceita. Assim, a partir da análise realizada, conclui-se que, por esse quesito, houve melhoria da nova versão do software com relação a versão atual.

Diante das inferências expostas, recomenda-se a utilização da nova versão do software em substituição da versão atual.

Aperfeiçoando os testes

Alguns testes não puderam ser feitos pois a amostra não tem uma representação normal associada. Já outros testes pelo fato de se ter somente uma amostra. Sugere-se possibilidade de extrair amostra da versão atual do software. De posse desta amostra seria possível utilizar de outros testes para calcular os estimadores de interesse, como por exemplo o teste de Levene, e na tentativa até de se evitar o uso da ferramenta de reamostragem, como foi usada no experimento acima.

Referências

[1] Felipe Campelo. Modular Differential Evolution for Experimenting with Operators. https://www.rdocumentation.org/packages/ExpDE/versions/0.1.2, 2016. Version 0.1.2.

[2] Felipe Campelo. Lecture Notes on Design and Analysis of Experiments. http://git.io/v3Kh8, 2018. Version 2.12; Creative Commons BY-NC-SA 4.0.

[3] Angelo Canty. Bootstrap Functions. https://www.rdocumentation.org/packages/boot/versions/1.3-25, 2020. Version 1.3-25.

[4] Anthony Christopher Davison and David Victor Hinkley. *Bootstrap Methods and Their Application*. Number 1. Cambridge University Press, 1997.

[5] Bradley Efron. *The Jackknife, the Bootstrap and Other Resampling Plans*. SIAM, 1982.

[6] Bradley Efron. Bootstrap Methods: Another Look at the Jackknife. In *Breakthroughs in statistics*, pages 569–593. Springer, 1992.

[7] Bradley Efron and Robert J Tibshirani. *An Introduction To The Bootstrap*. CRC press, 1994.

[8] Ramanathan Gnanadesikan and Martin B Wilk. Probability Plotting Methods for the Analysis of Data. *Biometrika*, 55(1):1–17, 1968.

[9] Douglas C Montgomery and George C Runger. *Applied Statistics and Probability for Engineers*. John Wiley & Sons, 2010.

[10] Camilo Daleles Rennó. Jackknife, Bootstrap e outros Métodos de Reamostragem. http://www.dpi.inpe.br/referata/arq/_2011/12_Camilo/Renno_2011_resampl.pdf, 2011. Acesso em 05 de Agosto de 2020.

[11] R Development Core Team. Power Calculations For One And Two Sample T Tests. https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/power.t.test, 2020. Documentation reproduced from package stats, version 3.6.2, License: Part of R 3.6.2.

[12] Sharon Lawner Weinberg and Sarah Knapp Abramowitz. *Statistics Using SPSS: An Integrative Approach.* Cambridge University Press, 2008.