Simulations and Inferential Analyses

Ellen Tworkoski

2/4/2022

Using Simulations from an Exponential Distribution to Provide Proof for Central Limit Theorem

Overview

This portion of the document investigates the mean and variance of simulated samples drawn from an exponential distribution. As described by the Central Limit Theorem, we expect the mean of these sample averages to approach the true population mean, and the variance of these sample averages to approach the true standard error of the underlying population distribution. The analysis below uses simulations and graphical displays to validate these expectations.

Simulations

We define an exponential distribution with lambda value 0.2. Therefore, the theoretical mean of this distribution is 5 (1/lambda) and the theoretical variance of this distribution is 25 ((1/lambda)^2).

We create 1,000 simulations, each one consisting of 40 draws from this exponential distribution. The mean within each simulation is calculated, and the mean and variance across all simulations is calculated.

```
#Load packages
library(crosstable)
library(tidyverse)
library(ggplot2)
library(stringr)
library(knitr)
```

```
#Define distribution properties
lambda <- 0.2
theor_mean <- 1/lambda
theor_mean
```

```
## [1] 5
```

```
theor_var <- (1/lambda)^2
theor_var
```

```
## [1] 25
```

```
#Define simulation variables
n <- 40        #Sample size
nsims <- 1000  #Number of simulations
set.seed(105)  #Set seed for reproducibility

#Draw 40,000 values (40*1000) from an exponential distribution with lambda value defined
↪  above
```

```r
samples <- matrix(rexp(n*nsims, rate = lambda), nrow = nsims, ncol = n)

#Calculate mean value of each sample (n=40) for each of the 1,000 simulations
means <- apply(samples, MARGIN = 1, FUN = mean)

#Determine average and variance of sample means
avg_of_means <- mean(means)
avg_of_means
```

```
## [1] 5.007316
```

```r
var_of_means <- var(means)
var_of_means
```

```
## [1] 0.6427414
```

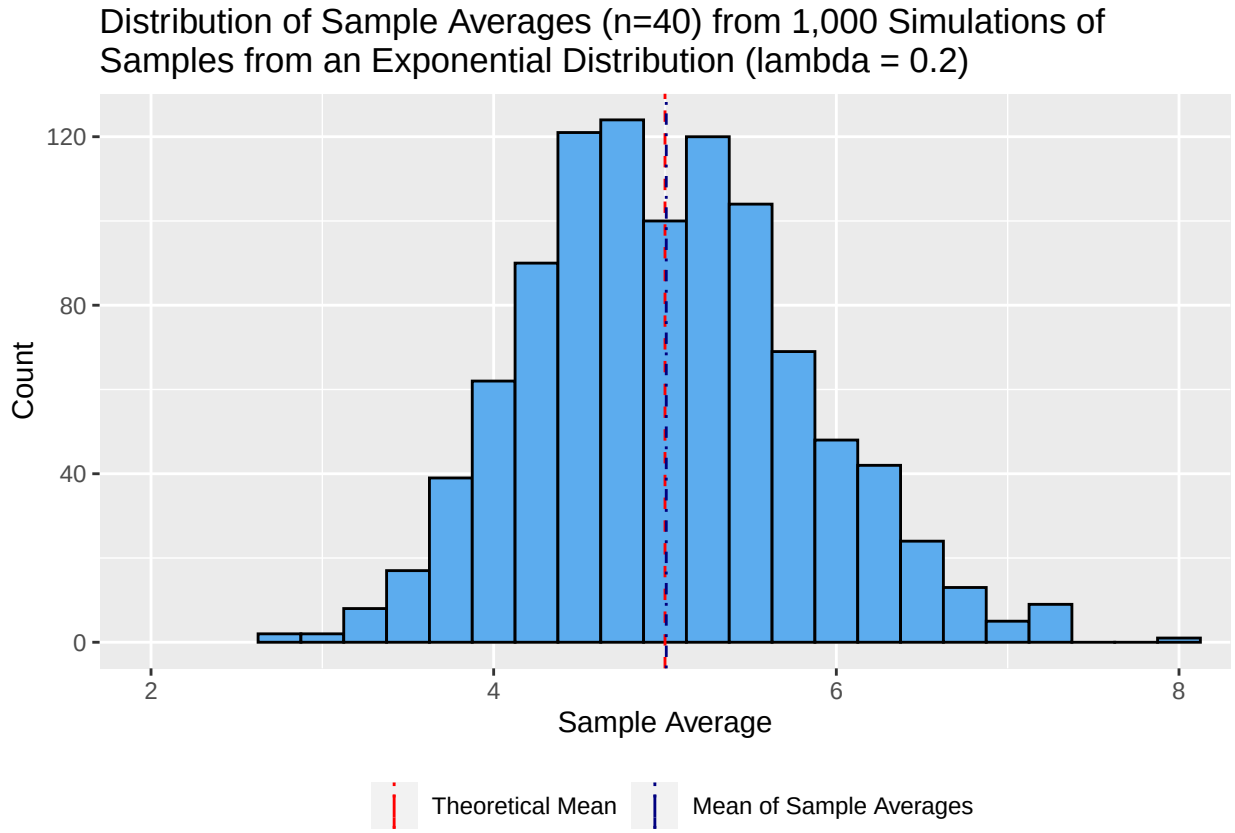
Sample Mean vs. Theoretical Mean

As described by the Central Limit Theorem, we expect the mean value of these sample averages to approach the true theoretical mean of this exponential distribution.

Given that lambda is 0.2, we know that the theoretical mean of this distribution is 5. As shown above, the mean value of the simulated sample averages is 5.0073159, which does in fact approach the theoretical mean of 5.

The R code below provides a graphical representation of this finding. It shows the distribution of the sample means, and includes vertical lines indicating the location of the theoretical population mean and the mean of the sample averages. Note that these lines are practically on top of one another, as expected.

```r
#Creates a plot showing the distribution of the sample means
ggplot(data = as.data.frame(means), mapping = aes(means)) +
    geom_histogram(binwidth = 0.25, fill = "steelblue2", color = "black") +
    geom_vline(aes(xintercept = theor_mean, color = "Theoretical Mean"), size = 0.5,
    ↪  linetype = 2) +
    geom_vline(aes(xintercept = avg_of_means, color = "Mean of Sample Averages"), size =
    ↪  0.5, linetype = 4) +
    xlab("Sample Average") +
    ylab("Count") +
    labs(title = str_wrap("Distribution of Sample Averages (n=40) from 1,000 Simulations
    ↪  of Samples from an Exponential Distribution (lambda = 0.2)",70))+
    scale_color_manual(name = "",
                       breaks = c('Theoretical Mean', 'Mean of Sample Averages'),
                       values = c('red', 'navyblue'))+
    theme(legend.position = "bottom")+
    coord_cartesian(xlim = c(2,8))
```



Sample Variance vs. Theoretical Variance

As described by the Central Limit Theorem, we expect the variance of the distribution of sample averages to equal the square of the standard error of the mean of the underlying population distribution.

We know that the square of the standard error of the mean is defined as the variance of the underlying population divided by the sample size. Therefore, we can directly relate the variance of the distribution of sample averages to the variance of the underlying exponential distribution.
The theoretical variance of the exponential distribution should be equal to the variance of the sample averages distribution multipled by the sample size (40).

Given that lambda is 0.2, we know that the theoretical variance of this distribution is 25. As shown above, the variance of the simulated sample averages is 0.6427414. Multiplying this by the sample size yields 25.7096556 which does in fact approach the theoretical variance.

The figure provided in the previous section provides a graphical display of the distribution of the sample averages from which the variance can be observed.

Normality of Distribution

Finally, the Central Limit Theorem states that the distribution of sample averages should be approximately normal if the sample size is sufficiently large enough. We can see from the above figure that the distribution does appear to be Gaussian (bell-shaped).

The relative normality of the distribution of sample averages is particularly obvious when observed in contrast to a distribution of random draws from the exponential distribution.

The figure below provides the distribution of 1,000 values taken from the same exponential distribution as before. The theoretical mean and the mean of the sample values are again noted on the graph. Although the

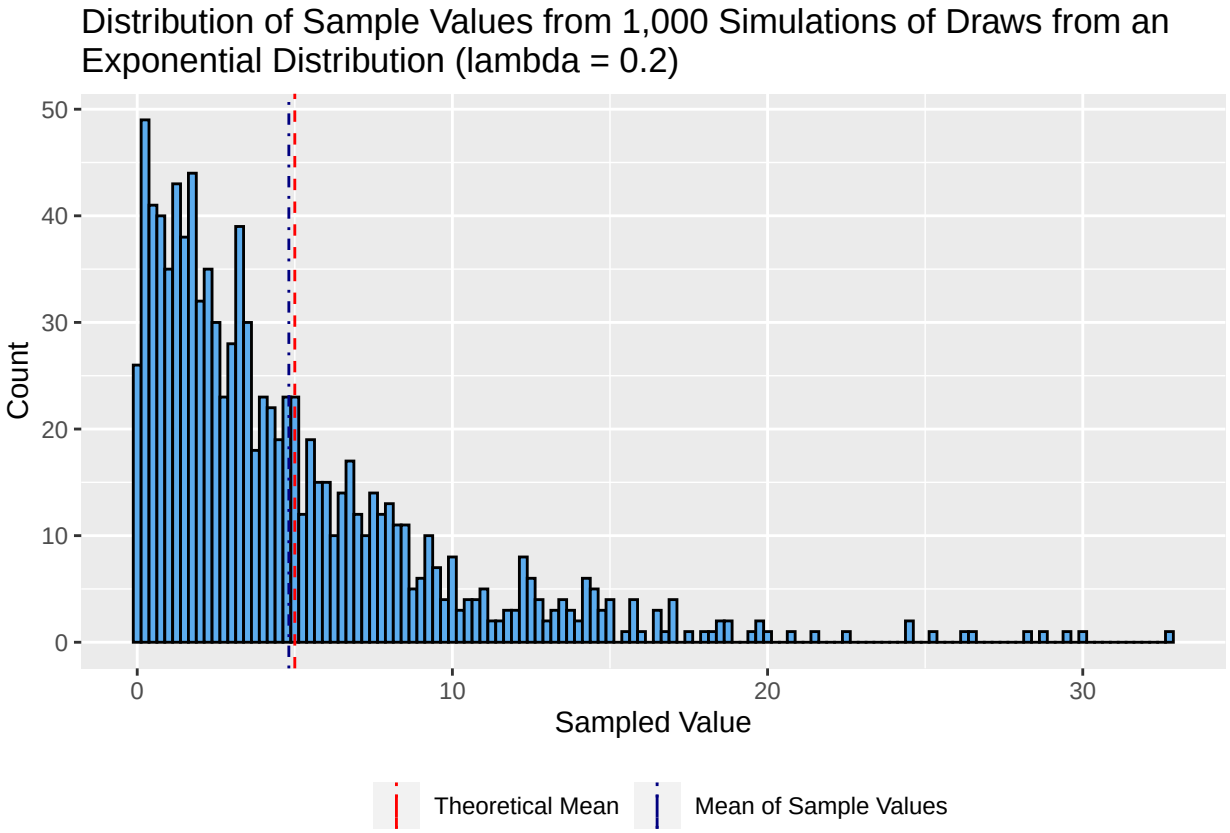
mean of the sample values does approach the theoretical mean, the distribution does not approach that of a normal curve, and does not provide information regarding the underlying population variance in the same way that the previous figure did.

```
# Generate a random sample of 1,000 values drawn from an exponential distribution with
↪  lambda defined above.
random_sample <-matrix(rexp(nsims, rate = lambda), nrow = nsims, ncol = 1)

#Calculate the mean of these 1,000 values
avg_of_sample <- mean(random_sample)
avg_of_sample
```

```
## [1] 4.811296
```

```
#Creates a plot showing the distribution of the randomly sampled 1,000 values
ggplot(data = as.data.frame(random_sample), mapping = aes(random_sample)) +
    geom_histogram(binwidth = 0.25, fill = "steelblue2", color = "black") +
    geom_vline(aes(xintercept = theor_mean, color = "Theoretical Mean"), size = 0.5,
    ↪  linetype = 2) +
    geom_vline(aes(xintercept = avg_of_sample, color = "Mean of Sample Values"), size =
    ↪  0.5, linetype = 4) +
    xlab("Sampled Value") +
    ylab("Count") +
    labs(title = str_wrap("Distribution of Sample Values from 1,000 Simulations of Draws
    ↪  from an Exponential Distribution (lambda = 0.2)",70))+
    scale_color_manual(name = "",
                       breaks = c('Theoretical Mean', 'Mean of Sample Values'),
                       values = c('red', 'navyblue'))+
    theme(legend.position = "bottom")
```



Inferential Analyses on ToothGrowth data

Overview

The second portion of this document describes exploratory and inferential investigations performed on the ToothGrowth dataset included in the R datasets package. This dataset examines the effect of Vitamin C on tooth growth in guinea pigs. Each animal received one of three dose levels of vitamin C (0.5, 1, or 2 mg/day) by one of two delivery methods (orange juice or ascorbic acid).

Below, we explore this dataset and perform a series of hypothesis tests to determine if changes in dose level and/or delivery method impacted tooth growth.

Exploratory Analysis

Through the exploratory investigations indicated below, we observe that the ToothGrowth dataset contains three variables: len (tooth length), supp (delivery method), and dose (dose level). There are no missing values. We observe that there are 60 observations total, with 10 observations allocated to each of the 6 treatment levels (2 delivery methods * 3 dose levels).

The length variable has a mean value of 18, and is roughly normally distributed around the mean.

An initial look at the mean, median, and standard deviation of the length variable stratified by the 6 treatment levels indicates that tooth growth does vary across groups. Tooth growth appears to increase with increased dose amount, and it appears to be higher among guinea pigs receiving Vitamin C through orange juice, although this is not consistent across all dose levels. Hypothesis tests will be conducted to formally assess these associations.

```r
#Obtain basic information about the dataset variables and allocation of sample size
↪  across treatment groups
str(ToothGrowth)
```

```
## 'data.frame':    60 obs. of  3 variables:
##  $ len : num  4.2 11.5 7.3 5.8 6.4 10 11.2 11.2 5.2 7 ...
##  $ supp: Factor w/ 2 levels "OJ","VC": 2 2 2 2 2 2 2 2 2 2 ...
##  $ dose: num  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
```

```r
summary(ToothGrowth)
```

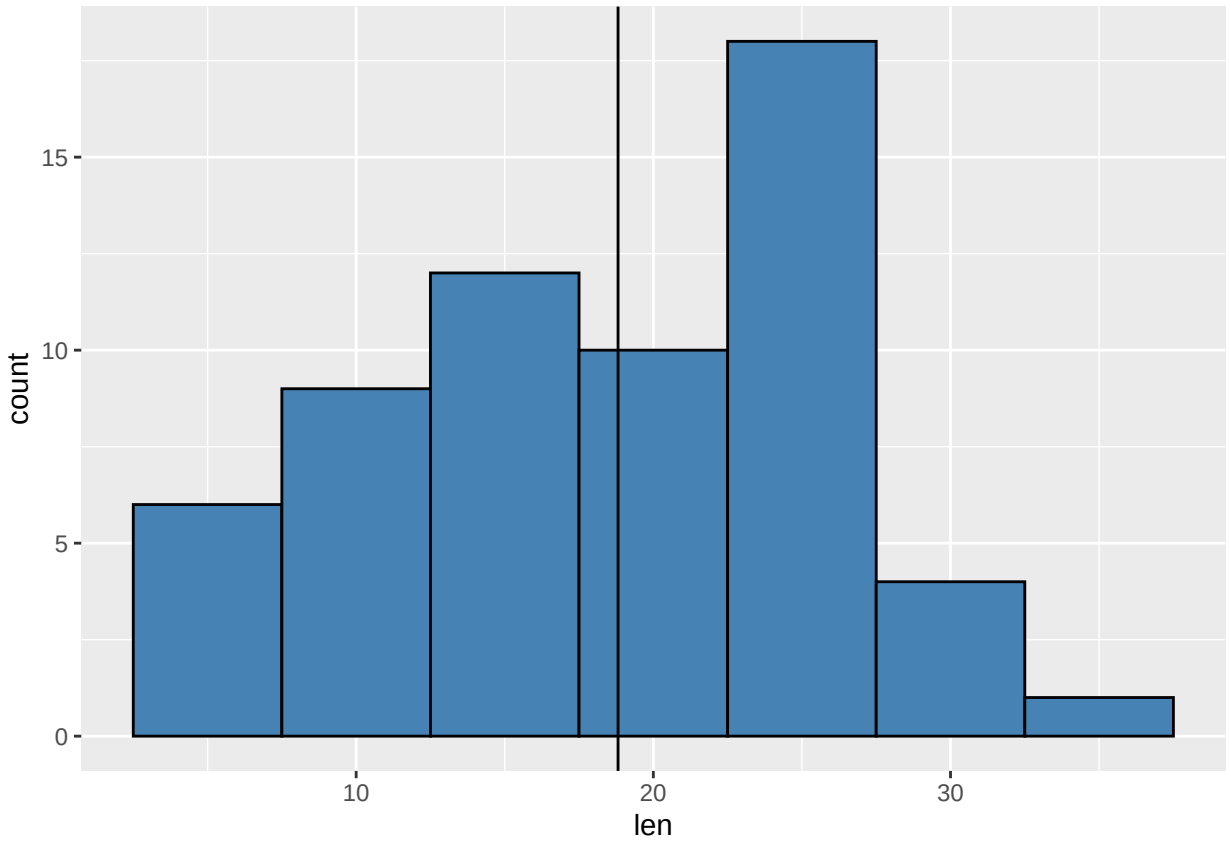
```
##       len        supp         dose
##  Min.   : 4.20   OJ:30   Min.   :0.500
##  1st Qu.:13.07   VC:30   1st Qu.:0.500
##  Median :19.25           Median :1.000
##  Mean   :18.81           Mean   :1.167
##  3rd Qu.:25.27           3rd Qu.:2.000
##  Max.   :33.90           Max.   :2.000
```

```r
crosstable(ToothGrowth, supp, by = dose) %>% as_flextable(keep_id=FALSE)
```

```
## Warning: Warning: fonts used in `flextable` are ignored because the `pdflatex`
## engine is used and not `xelatex` or `lualatex`. You can avoid this warning
## by using the `set_flextable_defaults(fonts_ignore=TRUE)` command or use a
## compatible engine by defining `latex_engine: xelatex` in the YAML header of the
## R Markdown document.
```

label	variable	dose		
		0.5	**1**	**2**
supp	OJ	10 (33.33%)	10 (33.33%)	10 (33.33%)
	VC	10 (33.33%)	10 (33.33%)	10 (33.33%)

```r
#Check normality of length var
avg_len <- mean(ToothGrowth$len)
ggplot(data = ToothGrowth, mapping = aes(len)) + geom_histogram(binwidth = 5, fill =
↪  "steelblue", color = "black") + geom_vline(xintercept = avg_len)
```




```
#Check dist of length by dose/supp
ToothGrowth %>% group_by(dose, supp) %>% summarize(median(len), mean(len), sd(len))
```

```
## `summarise()` has grouped output by 'dose'. You can override using the `.groups` argument.
```

```
## # A tibble: 6 x 5
## # Groups:   dose [3]
##    dose supp  `median(len)` `mean(len)` `sd(len)`
##   <dbl> <fct>         <dbl>       <dbl>     <dbl>
## 1   0.5 OJ            12.2        13.2       4.46
## 2   0.5 VC             7.15        7.98      2.75
## 3   1   OJ            23.5        22.7       3.91
## 4   1   VC            16.5        16.8       2.52
## 5   2   OJ            26.0        26.1       2.66
## 6   2   VC            26.0        26.1       4.80
```

Hypothesis Testing

Nine two-sided student t-tests were conducted. Three tests compared the two delivery methods (orange juice and ascorbic acid) to each other within each of the three dose levels. Three tests compared the three dose levels to each other within the orange juice delivery method. Three tests compared the three dose levels to each other within the ascorbic acid delivery method. A 95% CI was constructed for each t-test.

To control for multiple comparisons, adjusted p-values were calculated using the Benjamini–Hochberg method. Any adjusted p-value less than 0.05 is considered sufficient evidence to reject the null hypothesis that no difference exists in tooth growth between the two treatment groups being compared.

```r
#Create variables counting unique number of dose levels and delivery methods
doses <- unique(ToothGrowth$dose)
supps <- unique(ToothGrowth$supp)

#Initialize vectors for hypothesis test parameters of interest. Note that 9 tests will be
↪  conducted so vector length is set to 9.
reference_group <- vector(length = 9)
comparator_group <- vector(length = 9)
lower_ci <- vector(length = 9)
upper_ci <- vector(length = 9)
raw_pvalue <- vector(length = 9)

#Initialize loop counter variable (n_tests) to 1
n_tests <- 1

#Loop over dose levels. Within each dose level, conduct a t tests comparing the length
↪  variable between the orange juice (OJ) and ascorbic acid (VC) delivery methods.
for(d in doses){
    vc_dose <- ToothGrowth %>% subset(supp == 'VC' & dose == d, select = len)
    oj_dose <- ToothGrowth %>% subset(supp == 'OJ' & dose == d, select = len)

    reference_group[[n_tests]] <- paste0("VC_", d)
    comparator_group[[n_tests]] <- paste0("OJ_", d)
    raw_pvalue[[n_tests]] <- t.test(oj_dose, vc_dose, alternative = "two.sided", paired =
↪  FALSE, var.equal = FALSE, conf.level = 0.95)$p.value
    lower_ci[[n_tests]] <- t.test(oj_dose, vc_dose, alternative = "two.sided", paired =
↪  FALSE, var.equal = FALSE, conf.level = 0.95)$conf.int[1]
    upper_ci[[n_tests]] <- t.test(oj_dose, vc_dose, alternative = "two.sided", paired =
↪  FALSE, var.equal = FALSE, conf.level = 0.95)$conf.int[2]
    n_tests <- n_tests+1
}

#Loop over delivery methods. Within each delivery method, conduct 3 pairwise t-tests
↪  comparing the three dose levels to each other.
for(s in supps){
    supp_0.5 <- ToothGrowth %>% subset(supp == s & dose == 0.5, select = len)
    supp_1 <- ToothGrowth %>% subset(supp == s & dose == 1, select = len)
    supp_2 <- ToothGrowth %>% subset(supp == s & dose == 2, select = len)

    reference_group[[n_tests]] <- paste0(s, "_0.5")
    comparator_group[[n_tests]] <- paste0(s, "_1")
    raw_pvalue[[n_tests]] <- t.test(supp_1, supp_0.5, alternative = "two.sided", paired =
↪  FALSE, var.equal = FALSE, conf.level = 0.95)$p.value
    lower_ci[[n_tests]] <- t.test(supp_1, supp_0.5, alternative = "two.sided", paired =
↪  FALSE, var.equal = FALSE, conf.level = 0.95)$conf.int[1]
    upper_ci[[n_tests]] <- t.test(supp_1, supp_0.5, alternative = "two.sided", paired =
↪  FALSE, var.equal = FALSE, conf.level = 0.95)$conf.int[2]
    n_tests <- n_tests+1

    reference_group[[n_tests]] <- paste0(s, "_0.5")
    comparator_group[[n_tests]] <- paste0(s, "_2")
    raw_pvalue[[n_tests]] <- t.test(supp_2, supp_0.5, alternative = "two.sided", paired =
↪  FALSE, var.equal = FALSE, conf.level = 0.95)$p.value
```

```
    lower_ci[[n_tests]] <- t.test(supp_2, supp_0.5, alternative = "two.sided", paired =
↪  FALSE, var.equal = FALSE, conf.level = 0.95)$conf.int[1]
    upper_ci[[n_tests]] <- t.test(supp_2, supp_0.5, alternative = "two.sided", paired =
↪  FALSE, var.equal = FALSE, conf.level = 0.95)$conf.int[2]
    n_tests <- n_tests+1

    reference_group[[n_tests]] <- paste0(s, "_1")
    comparator_group[[n_tests]] <- paste0(s, "_2")
    raw_pvalue[[n_tests]] <- t.test(supp_2, supp_1, alternative = "two.sided", paired =
↪  FALSE, var.equal = FALSE, conf.level = 0.95)$p.value
    lower_ci[[n_tests]] <- t.test(supp_2, supp_1, alternative = "two.sided", paired =
↪  FALSE, var.equal = FALSE, conf.level = 0.95)$conf.int[1]
    upper_ci[[n_tests]] <- t.test(supp_2, supp_1, alternative = "two.sided", paired =
↪  FALSE, var.equal = FALSE, conf.level = 0.95)$conf.int[2]
    n_tests <- n_tests+1
}

#Combine all hypothesis test results into a single data frame
test_results <- data.frame(reference_group, comparator_group, lower_ci, upper_ci,
↪  raw_pvalue)

#Apply the Benjamini-Hochberg procedure to calculate adjusted p values, correcting for
↪  multiple comparisons
test_results$BH_pvalue <- p.adjust(test_results$raw_pvalue, method = "BH")
kable(test_results)
```

reference_group	comparator_group	lower_ci	upper_ci	raw_pvalue	BH_pvalue
VC_0.5	OJ_0.5	1.7190573	8.780943	0.0063586	0.0081754
VC_1	OJ_1	2.8021482	9.057852	0.0010384	0.0015576
VC_2	OJ_2	-3.7980705	3.638070	0.9638516	0.9638516
VC_0.5	VC_1	6.3142880	11.265712	0.0000007	0.0000031
VC_0.5	VC_2	14.4184880	21.901512	0.0000000	0.0000004
VC_1	VC_2	5.6857333	13.054267	0.0000916	0.0001648
OJ_0.5	OJ_1	5.5243656	13.415634	0.0000878	0.0001648
OJ_0.5	OJ_2	9.3247594	16.335241	0.0000013	0.0000040
OJ_1	OJ_2	0.1885575	6.531442	0.0391951	0.0440945

From the above table, we see that eight of the nine hypothesis tests have an adjusted p-value (BH_pvalue) less than 0.05. The null hypothesis (no difference in tooth growth between the reference and comparator group) is rejected for these eight tests. Note that 'VC' indicates ascorbic acid and 'OJ' indicates orange juice in the above table.

Conclusions

From the above hypothesis tests, we conclude that orange juice resulted in more tooth growth than ascorbic acid for dose levels of 0.5 and 1 mg/day. However, no difference in tooth growth was observed between the two delivery methods for a dose level of 2 mg/day.

Higher dose levels seemed to be the larger driver of tooth growth. Increases from 0.5 to 1, 0.5 to 2, and 1 to 2 mg/day resulted in statistically significant increases in tooth growth within both delivery methods when using an alpha of 0.05. However, for an alpha of 0.01, the tooth growth increase between doses of 1 and 2 mg/day when using an orange juice delivery method was no longer statistically significant. All other dose level comparisons remained significant at the 0.01 alpha level.

Note that the hypothesis tests were conducted under the following assumptions, and thus all conclusions rest on these assumptions: the underlying data are iid normally distributed, all treatment samples are independent of each other, and sample variances are unequal.