

# Inferential Statistics Assignment Part 2

*Karl Ranson*

*30 August 2016*

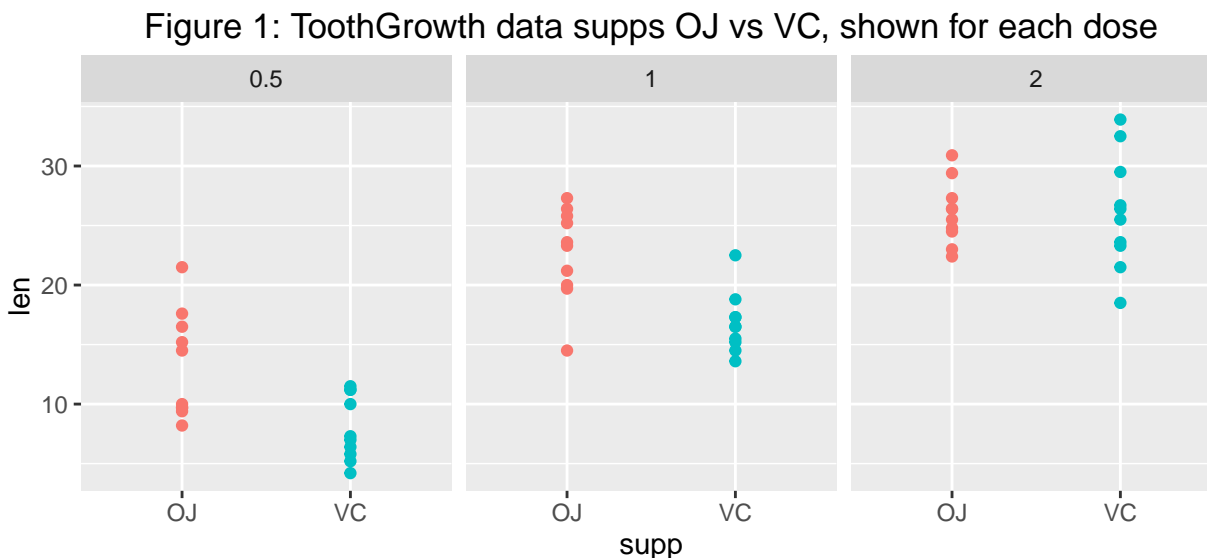
## Introduction

This PDF meets the requirements of Inferential Statistics assignment part 2, namely:

- (1) Perform exploratory analysis on exponential & ‘ToothGrowth’ data to demonstrate some basic characteristics;
- (2) Compare the sampled vs theoretical estimators for exponential data; and
- (3) Create confidence intervals for ToothGrowth, and show how multiple sample sets compare with the Central Limit Theorem.

To keep within the 3 page report limit, all R code is in the Appendix.

## Part 2.1 ToothGrowth data visualisation

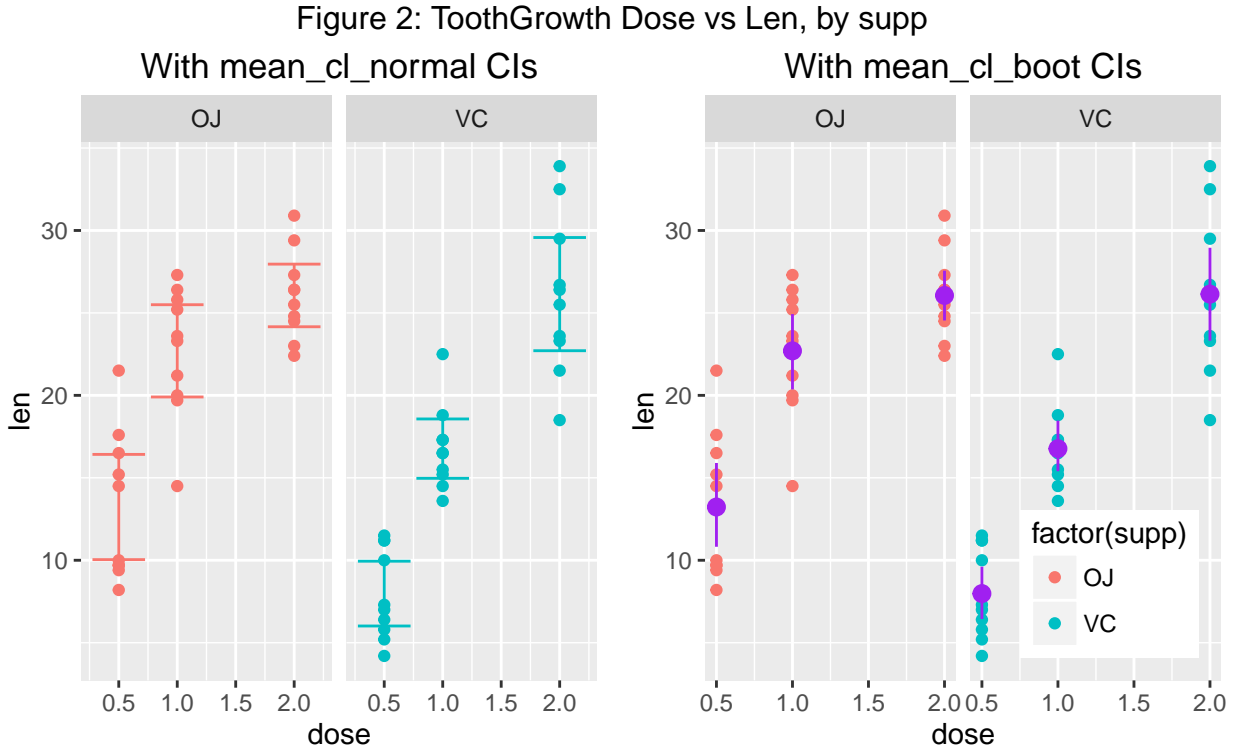


A ‘shapiro.test’, an often used test for normality of the underlying population, was conducted for each of the 6 data sets:

Table 1: Shapiro test p-values

supp	OJ	VC	OJ	VC	OJ	VC
dose	0.5	0.5	1.0	1.0	2.0	2.0
p_value	0.1820408	0.1695627	0.4152983	0.2697855	0.8147908	0.9194497

As all P values  $>0.05$ , we have insufficient evidence to reject the hypothesis that the underlying data sets are normal.



The below graphs shows the data sets separated by Supp and Dose, with mean and 2 x 95% confidence intervals as explained below.

“mean\_cl\_normal” = returns “sample mean and 95% confidence intervals assuming normality.”

“mean\_cl\_boot” is based on “smean.cl.boot” which is “a very fast implementation of the basic nonparametric bootstrap for obtaining confidence limits for the population mean without assuming normality”. The results here are based on 1,000 bootstraps. You can see that, based on differences between the two data sets, the bootstrapping intervals are tighter.

Thus even though the bootstrapping technique does not rely on assuming the underlying data is normal, it still provides more power.

## Part 2.2 - T tests

We do a two sided t-test using the ‘t.test’ function, comparing ‘OJ’ and ‘VC’ across the 3 dosages assuming unequal variances.

Table 2: T - test results - comparing between OJ and VC

dose	statistic	p.value
0.5	3.170	0.006
1.0	4.033	0.001
2.0	-0.046	0.964

## ToothGrowth Permutations

As an exercise, permutations were taken comparing OJ dosages. OJ was chosen because, from figure 2, of the 4 comparison sets across dosages & within each ‘supp’, there is least power to detect differences between (OJ

& 1.0) and (OJ & 2.0).

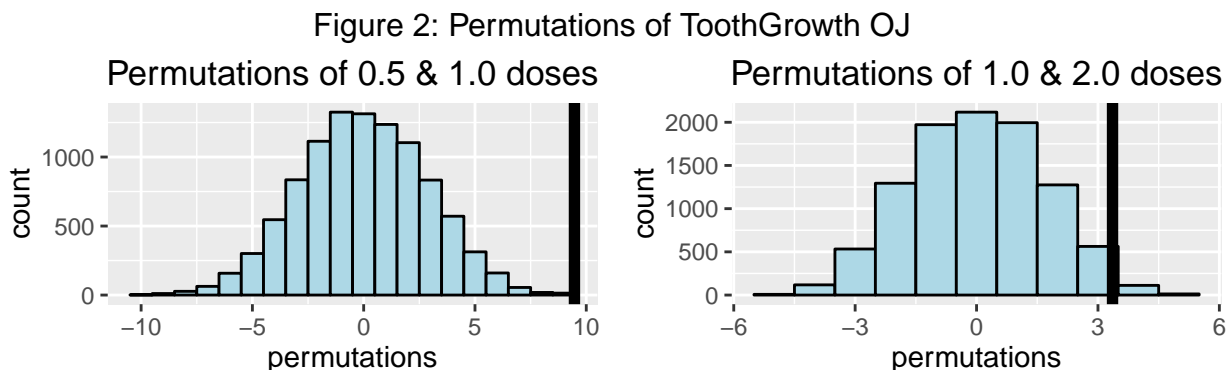


Table 3: OJ Permutations

	ObsrvdStt	MnOfPerm_Gtr_ObsrvdStt
OJ: 0.5 v 1.0	9.47	0.000
OJ: 1.0 v 2.0	3.36	0.018

The right hand column of Table 3 is the proportion of time the permuted observations are greater than the observed. It shows that there is less power for the (OJ: 1.0 v 2.0) compared with (OJ: 0.5 v 1.0). This is consistent with what we can visibly see on the Figure 1. However, both results are statistically significant.

A similar permutation summary table was created for VC.

Table 4: VC Permutations

	ObsrvdStt	MnOfPerm_Gtr_ObsrvdStt
VC: 0.5 v 1.0	8.79	0
VC: 1.0 v 2.0	9.37	0

## Conclusions

T-test analyses rely on the assumption that the underlying data set is normal. Based on the shapiro testing conducted per Table 1, this is a reasonable assumption.

The T-test summaries across supps in Table 2 lead us to two conclusions:

- (1) For dosage = 0.5 and 1.0, there is we have strong evidence at any reasonable alpha to reject the null hypothesis that the means of ‘OJ’ and ‘VC’ are the same.
- (2) For the 2.0 dosage, there is insufficient evidence to reject the null hypothesis.

These conclusions are intuitively supported by the visual differences in data at the top of the page. The t-tests are for unequal variance, and assume a normal underlying data distribution. Visually inspecting the graphs at the top of this page, the data does not obviously support this assumption. However, as mentioned above, the bootstrapping results are significant even when normality is not assumed.

Based on the permutation summary Tables 3 and 4, we can conclude that all dosage comparisons are statistically significant. That is, for all 4 dosage comparisons, there is strong evidence that the comparisons sets have significantly different population means.

## Appendix 1: All R code

### Libraries

```
knitr::opts_chunk$set(cache=TRUE)

library(grid)
library(ggplot2)
library(gridExtra)
library(moments)
library(stats)
library(plyr)
library(dplyr)
```

### Visualisation

```
p1 <- ggplot(ToothGrowth, aes(supp, len)) + geom_point(aes(fill = factor(supp))) + aes(colour=factor(supp)) +
  ggtitle("Figure 1: ToothGrowth data supps OJ vs VC, shown for each dose") + facet_wrap(~ dose) +
p1
```

### NormalTest

```
TG <- aggregate(formula = len ~ supp + dose,
  data = ToothGrowth,
  FUN = function(x) {y <- shapiro.test(x); y$p.value})

TG <- t(dplyr::rename(TG, p_value = len))

knitr::kable(TG, digits=3, caption = "Shapiro test p-values")
```

### Part 2.1: ToothGrowth data visualisation

```
p1 <- ggplot(ToothGrowth, aes(dose, len)) + geom_point() +
  ggtitle("With mean_cl_normal CIs") + aes(colour=factor(supp)) + facet_wrap(~ supp) + stat_summary(fun.data =
  mean_cl_normal, geom="pointrange", color="purple")

p2 <- ggplot(ToothGrowth, aes(dose, len)) + geom_point() +
  ggtitle("With mean_cl_boot CIs") + aes(colour=factor(supp)) +
  facet_wrap(~ supp) + theme(legend.justification=c(1,0), legend.position=c(1,0)) + stat_summary(fun.data =
  mean_cl_boot, geom="pointrange", color="purple")

grid.arrange(p1, p2, ncol = 2, top = "Figure 2: ToothGrowth Dose vs Len, by supp")
```

### Part 2.2: ToothGrowth t-tests

```
TG <- ddply(ToothGrowth, "dose",
  function(x) {
    w <- t.test(len~supp, data=x)
    with(w, data.frame(statistic, p.value))
  })

knitr::kable(TG, digits=3, caption = "T - test results - comparing between OJ and VC")
```

## ToothGrowth Permutations

```
OJ_.5_1 <- ToothGrowth[(ToothGrowth$dose) %in% c(0.5,1.0) & ToothGrowth$supp == "OJ",c(1,3)]
y <- OJ_.5_1$len
group <- as.character(OJ_.5_1$dose)
testStat <- function(w, g) mean(w[g == 1.0]) - mean(w[g == 0.5])
observedStat <- testStat(y, group)
permutations <- sapply(1 : 10000, function(i) testStat(y, sample(group)))
R11 <- observedStat
R12<- mean(permutations > observedStat)
g3 = ggplot(data.frame(permutations = permutations), aes(permutations))
g3 = g3 +labs(title = "Permutations of 0.5 & 1.0 doses")
g3 = g3 + geom_histogram(fill = "lightblue", color = "black", binwidth = 1)
g3 = g3 + geom_vline(xintercept = observedStat, size = 2)

OJ_1_2 <- ToothGrowth[(ToothGrowth$dose) %in% c(1,2) & ToothGrowth$supp == "OJ",c(1,3)]
y <- OJ_1_2$len
group <- as.character(OJ_1_2$dose)
testStat <- function(w, g) mean(w[g == 2]) - mean(w[g == 1])

observedStat <- testStat(y, group)
permutations <- sapply(1 : 10000, function(i) testStat(y, sample(group)))
R21<-observedStat
R22 <- mean(permutations > observedStat)
g4 = ggplot(data.frame(permutations = permutations),
  aes(permutations))
g4 = g4 + labs(title = "Permutations of 1.0 & 2.0 doses")
g4 = g4 + geom_histogram(fill = "lightblue", color = "black", binwidth = 1)
g4 = g4 + geom_vline(xintercept = observedStat, size = 2)

grid.arrange(g3, g4, ncol = 2, top ="Figure 2: Permutations of ToothGrowth OJ")
sumry <- data.frame("ObsrvdStt"=c(R11,R21), "MnOfPerm_Gtr_ObsrvdStt"=c(R12,R22), row.names=c("OJ: 0.5 v
knitr::kable(sumry, digits=3, caption = "OJ Permutations")

VC_.5_1 <- ToothGrowth[(ToothGrowth$dose) %in% c(0.5,1.0) & ToothGrowth$supp == "VC",c(1,3)]
y <- VC_.5_1$len
group <- as.character(VC_.5_1$dose)
testStat <- function(w, g) mean(w[g == 1.0]) - mean(w[g == 0.5])
observedStat <- testStat(y, group)
permutations <- sapply(1 : 10000, function(i) testStat(y, sample(group)))
R11 <- observedStat
R12<- mean(permutations > observedStat)

g3 = ggplot(data.frame(permutations = permutations), aes(permutations))
g3 = g3 +labs(title = "Permutations of 0.5 & 1.0 doses")
g3 = g3 + geom_histogram(fill = "lightblue", color = "black", binwidth = 1)
g3 = g3 + geom_vline(xintercept = observedStat, size = 2)

VC_1_2 <- ToothGrowth[(ToothGrowth$dose) %in% c(1,2) & ToothGrowth$supp == "VC",c(1,3)]
y <- VC_1_2$len
group <- as.character(VC_1_2$dose)
testStat <- function(w, g) mean(w[g == 2]) - mean(w[g == 1])
```

```

observedStat <- testStat(y, group)
permutations <- sapply(1 : 10000, function(i) testStat(y, sample(group)))
R21<-observedStat
R22 <- mean(permutations > observedStat)

g4 = ggplot(data.frame(permutations = permutations),
             aes(permutations))
g4 = g4 + labs(title = "Permutations of 1.0 & 2.0 doses")
g4 = g4 + geom_histogram(fill = "lightblue", color = "black", binwidth = 1)
g4 = g4 + geom_vline(xintercept = observedStat, size = 2)

# Hide graph as reached 3 page limit. But would have been useful
# grid.arrange(g3, g4, ncol = 2, top = "Figure 2: Permutations of ToothGrowth VC")

sumry <- data.frame("ObsrvdStt"=c(R11,R21), "MnOfPerm_Gtr_ObsrvdStt"=c(R12,R22), row.names=c("VC: 0.5 v
knitr::kable(sumry, digits=20, caption = "VC Permutations")

```