Lesson 09: Statistical Inference, Hypothesis Testing for Single Populations

References

- Black, Chapter 9 Statistical Inference: Estimation for Single Populations (pp. 298-351)
- Kabakoff, Chapter 7 Basic Statistics (pp.158-160)
- Davies, Chapter 18 Hypothesis Testing (pp. 384-433)
- Stowell, Chapter 10 Hypothesis Testing (pp. 144-146, 158)

Exercises:

Data Set: hot_dogs.csv (Original source: Consumer Reports, June 1986, pp. 366-367)

Description: Results of a laboratory analysis of calories and sodium content of major hot dog brands. Researchers for Consumer Reports analyzed three types of hot dog: beef, poultry, and meat (mostly pork and beef, but up to 15% poultry meat). Fifty four observations are reported.

Variable Names:

1. Type = Type of hotdog (beef, meat, or poultry)
2. Calories = Calories per hot dog
3. Sodium = Milligrams of sodium per hot dog

1) Use hot_dogs.csv data and hypothesis tests to determine which type of hot dog has average calories less than 140 with 95% confidence. Present boxplots of calories by type of hot dog.

```
# Read the comma-delimited text file creating a data frame object in R,
# then examine its structure:
hotdogs <- read.csv("hot_dogs.csv")
str(hotdogs)
```

```
## 'data.frame':    54 obs. of  3 variables:
##  $ Type    : Factor w/ 3 levels "Beef","Meat",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Calories: int  186 181 176 149 184 190 158 139 175 148 ...
##  $ Sodium  : int  495 477 425 322 482 587 370 322 479 375 ...
```

```
# Again, we see that some of the questions concern subsets of the data by hotdog
# type. To respond to the questions, we will create three subset data frames:
beef <- subset(hotdogs, subset = (Type == "Beef"))
meat <- subset(hotdogs, subset = (Type == "Meat"))
poultry <- subset(hotdogs, subset = (Type == "Poultry"))

# First, we'll identify the 95% confidence intervals, per Type
with(beef, t.test(Calories, mu = 140, alternative = "less")$conf.int)
```

```
## [1]     -Inf 165.6044
## attr(,"conf.level")
## [1] 0.95
```

```r
with(meat, t.test(Calories, mu = 140, alternative = "less")$conf.int)
```

```
## [1]     -Inf 169.3917
## attr(,"conf.level")
## [1] 0.95
```

```r
with(poultry, t.test(Calories, mu = 140, alternative = "less")$conf.int)
```

```
## [1]     -Inf 128.3139
## attr(,"conf.level")
## [1] 0.95
```

```r
# Second, we'll extract just the upper bound of the 95% CI
with(beef, t.test(Calories, mu = 140, alternative = "less")$conf.int[2])
```

```
## [1] 165.6044
```

```r
with(meat, t.test(Calories, mu = 140, alternative = "less")$conf.int[2])
```

```
## [1] 169.3917
```

```r
with(poultry, t.test(Calories, mu = 140, alternative = "less")$conf.int[2])
```

```
## [1] 128.3139
```

```r
# Third, we'll add a logical comparison to our statement (" < 140")
with(beef, t.test(Calories, mu = 140, alternative = "less")$conf.int[2] < 140)
```

```
## [1] FALSE
```

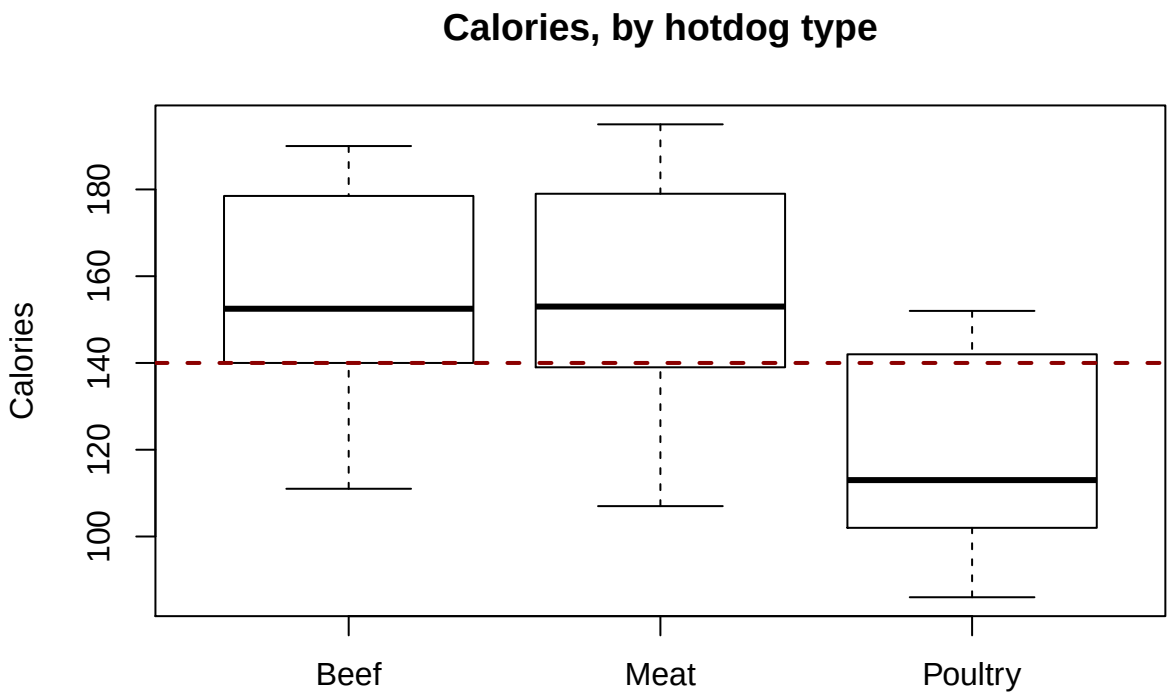
```r
with(meat, t.test(Calories, mu = 140, alternative = "less")$conf.int[2] < 140)
```

```
## [1] FALSE
```

```r
with(poultry, t.test(Calories, mu = 140, alternative = "less")$conf.int[2] < 140)
```

```
## [1] TRUE
```

```r
with(hotdogs, boxplot(Calories ~ Type, main = "Calories, by hotdog type",
    ylab = "Calories"))
abline(h = 140, lty = 2, lwd = 2, col = "darkred")
```




```
# Note that only poultry hot dogs meet this test.
```

2) Using hot_dogs.csv data and hypothesis tests at the 95% confidence level, determine which type of hot dog has an average Sodium level different from 425 milligrams.

```
# This looks like a set of two-tailed t-tests for means. Let the null hypothesis
# be H0: mu = 425, and we will perform a t-test to get the p-value for each
# type of hot dog.

# First we refer to the R documentation for the t.test function.
# t.test(x, y = NULL,
#       alternative = c("two.sided", "less", "greater"),
#       mu = 0, paired = FALSE, var.equal = FALSE,
#       conf.level = 0.95, ...)
with(beef, t.test(Sodium, alternative = "two.sided", mu = 425))
```

```
##
##  One Sample t-test
##
## data:  Sodium
## t = -1.0413, df = 19, p-value = 0.3108
## alternative hypothesis: true mean is not equal to 425
## 95 percent confidence interval:
##  353.2091 449.0909
## sample estimates:
```

```
## mean of x
##    401.15
```

```
with(meat, t.test(Sodium, alternative = "two.sided", mu = 425))
```

```
##
##  One Sample t-test
##
## data:  Sodium
## t = -0.2842, df = 16, p-value = 0.7799
## alternative hypothesis: true mean is not equal to 425
## 95 percent confidence interval:
##  370.2647 466.7941
## sample estimates:
## mean of x
##  418.5294
```

```
with(poultry, t.test(Sodium, alternative = "two.sided", mu = 425))
```

```
##
##  One Sample t-test
##
## data:  Sodium
## t = 1.6543, df = 16, p-value = 0.1175
## alternative hypothesis: true mean is not equal to 425
## 95 percent confidence interval:
##  415.4311 502.5689
## sample estimates:
## mean of x
##       459
```

```
# None of the three p-values is less than 0.05, so we do not reject the null hypothesis.

# Note. When running many classical tests, we often adjust the critical values
# for the tests so that we avoid making type-one errors. Here, no tests were
# statistically significant, so we do not need to adjust critical values.
```

3) Using hot_dogs.csv data and hypothesis tests, determine if the variance in Sodium values for beef hot dogs is different from 6000 with 95% confidence.

```
# Here we will use the appropriate chi-square test. Refer to R documentation:
# dchisq(x, df, ncp = 0, log = FALSE)
# pchisq(q, df, ncp = 0, lower.tail = TRUE, log.p = FALSE)
# qchisq(p, df, ncp = 0, lower.tail = TRUE, log.p = FALSE)
# rchisq(n, df, ncp = 0)
# x, q  vector of quantiles.
# p vector of probabilities.
# n  number of observations.
# df degrees of freedom (non-negative, but can be non-integer).
# ncp non-centrality parameter (non-negative).
# log, log.p logical; if TRUE, probabilities p are given as log(p).
# lower.tail    logical; if TRUE (default), probabilities are P[X = x],
```

```
#    otherwise, P[X > x].
# We could build on the confidence interval function we developed in lesson 7:
var.conf.int = function(x, conf.level = 0.95) {
    df <- length(x) - 1
    chilower <- qchisq((1 - conf.level)/2, df, lower.tail = TRUE)
    chiupper <- qchisq((1 - conf.level)/2, df, lower.tail = FALSE)
    v <- var(x)
    c(df * v/chiupper, df * v/chilower)
    }
with(beef, var.conf.int(Sodium))
```

```
## [1]  6068.506 22384.122
```

```
# If this logical is TRUE, then we reject the null hypothesis that mu = 6000:
with(beef, (6000 < var.conf.int(Sodium)[1]) ||
       (6000 >var.conf.int(Sodium)[2]))
```

```
## [1] TRUE
```

4) Assume a random sample of size 100 is drawn from a normal distribution for which the mean and variance are unknown. Assume the sample mean is 50 and the standard deviation of the sample is 2. Test the hypothesis that the true mean is 56, and also test the hypothesis the true mean is 40. Report p-values and comment on the results.

```
# Two-Tailed Test of Population Mean with Unknown Variance
# http://www.r-tutor.com/elementary-statistics/hypothesis-testing/two-tailed-test-population-mean-unkno

samp.mean <- 50     # sample mean
test.mean1 <- 56    # mean value to test
test.mean2 <- 40    # alternate mean value to test
samp.sd <- 2    # sample standard deviation
n <- 100        # sample size

t <- (samp.mean - test.mean1)/(samp.sd/sqrt(n)) # test statistic
t #[1] -30
```

```
## [1] -30
```

```
p.value <- 2 * pt(-abs(t), df=n-1)  # we use -abs(t) because pt()
                        # returns the probability that
                        # the actual value is less than the
                        # supplied t
p.value # [1] 1.70085e-51
```

```
## [1] 1.70085e-51
```

```
# And, for test.mean2 (40)
t <- (samp.mean - test.mean2)/(samp.sd/sqrt(n)) # test statistic
t #[1] 50
```

```
## [1] 50
```

```r
p.value <- 2 * pt(-abs(t), df=n-1)
p.value # [1] 4.595366e-72
```

```
## [1] 4.595366e-72
```

5) A coin is flipped 100 times. If it is unbiased the probability of a heads should equal the probability of a tails. At the 95% confidence level, test the null hypothesis the coin is unbiased versus the alternative that it is biased if 43 heads are obtained. Test the same hypothesis if 63 heads are obtained. Use one-sided hypothesis tests.

```r
# Looks like another binomial problem. Refer to R documentation for prop.test().
# prop.test(x, n, p = NULL,
#           alternative = c("two.sided", "less", "greater"),
#           conf.level = 0.95, correct = TRUE)
# x: a vector of counts of successes, a one-dimensional table with two entries, or a two-dimensional
#     table (or matrix) with 2 columns, giving the counts of successes and failures, respectively.
# n:    a vector of counts of trials; ignored if x is a matrix or a table.
# p:    a vector of probabilities of success. The length of p must be the same as the number of groups
#     specified by x, and its elements must be greater than 0 and less than 1.
# alternative:  character string specifying the alternative hypothesis,
# must be one of "two.sided" (default), "greater" or "less". You can specify just the initial letter.
# conf.level: confidence level of the returned confidence interval.
# Must be a single number between 0 and 1.
# Only used when testing the null that a single proportion equals a given value, or that two proportion
# are equal; ignored otherwise.
# correct: a logical indicating whether Yates' continuity correction should be applied where possible.
prop.test(x = 43, n = 100, alternative = "less")  # see p-value 0.0968 > 0.05 (do not reject null hypot
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  43 out of 100, null probability 0.5
## X-squared = 1.69, df = 1, p-value = 0.0968
## alternative hypothesis: true p is less than 0.5
## 95 percent confidence interval:
##  0.000000 0.517194
## sample estimates:
##    p
## 0.43
```

```r
prop.test(x = 63, n = 100, alternative = "greater")  # see p-value 0.00621 < 0.05 (reject null hypothes
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  63 out of 100, null probability 0.5
## X-squared = 6.25, df = 1, p-value = 0.00621
## alternative hypothesis: true p is greater than 0.5
## 95 percent confidence interval:
##  0.5430629 1.0000000
## sample estimates:
##    p
## 0.63
```

6) salaries.csv contains data derived from a November 8, 1993 article in Forbes titled "America's Best Small Companies". The file gives the CEO age and salary for 60 small business firms. Use these data to test the hypothesis at 95% confidence that at least 50% of the CEOs are 45 years old or older. Also test the hypothesis at 95% confidence that at least 50% of the CEOs earn less than $500,000 per year. Use one-sided hypothesis tests.

```
# Read the comma-delimited text file creating a data frame object in R,
# then examine its structure:
salaries <- read.csv("salaries.csv")
str(salaries)
```

```
## 'data.frame':    60 obs. of  2 variables:
##  $ AGE: int  53 43 33 45 46 55 41 55 36 45 ...
##  $ SAL: int  145 621 262 208 362 424 339 736 291 58 ...
```

```
# To test the hypothesis about age, we must count the number >= 45 years old.
# Then prop.test will be used.

age <- salaries$AGE >= 45
count <- sum(age)
total <- length(age)
prop.test(x = count, n = total, alternative = "greater")
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  count out of total, null probability 0.5
## X-squared = 22.817, df = 1, p-value = 8.911e-07
## alternative hypothesis: true p is greater than 0.5
## 95 percent confidence interval:
##  0.7121941 1.0000000
## sample estimates:
##         p
## 0.8166667
```

```
# The p-value is less than 0.05 so reject the null hypothesis.

# Now we must count the number earning less than $500,000 per year.
# Then prop.test will be used.

salary <- salaries$SAL < 500
count <- sum(salary)
total <- length(salary)
prop.test(x= count, n = total, alternative = "greater")
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  count out of total, null probability 0.5
## X-squared = 10.417, df = 1, p-value = 0.0006244
## alternative hypothesis: true p is greater than 0.5
## 95 percent confidence interval:
```

```
##  0.6045034 1.0000000
## sample estimates:
##         p
## 0.7166667
```

```
# The null hypothesis must be rejected based on the small p-value.
```