04 Smoking 2

Contents

(04) Comparison of Samples with *Student*'s t test . . . **1**
- Preamble . . . 1
- Permutation Test . . . 2
- Wilcoxon-Mann-Whitney Test . . . 3
  - The outer Product again . . . 3
- Standardisation . . . 4
  - Standardise the Observations . . . 4
  - Calculate the standard error for the difference . . . 5
  - Calculate the t-statistic . . . 5
  - Permutation Again . . . 6
- Confidence Intervals . . . 6
  - Bootstrap . . . 6
- t-based . . . 10
- Challenge . . . 11
  - Sales Data . . . 11
  - Spider Data . . . 13

(04) Comparison of Samples with *Student*'s t test

Preamble

```
# Preamble

## Install Pacman
load.pac <- function() {

  if(require("pacman")){
    library(pacman)
  }else{
    install.packages("pacman")
    library(pacman)
  }

  pacman::p_load(xts, sp, gstat, ggplot2, rmarkdown, reshape2, ggmap,
                 parallel, dplyr, plotly, tidyverse, reticulate, UsingR, Rmpfr,
                 swirl, corrplot, gridExtra, mise, latex2exp, tree, rpart, lattice,
                 coin)


   mise()
}

load.pac()
```

```
## Loading required package: pacman
```

```
load(file = "~/Notes/DataSci/ThinkingAboutData/TAD.rdata")
load(file = "./TAD.rdata")
```

Permutation Test

So in wk 3 the code to test for a difference was to the effect of:

```
# Aggregate is a wrapper for apply
  ## As a Function
aggregate(bwt ~ smoke, birthwt, mean)
```

```
##   smoke      bwt
## 1    no 3515.639
## 2   yes 3260.285
```

```
  ## For Data Frames
aggregate(x = birthwt$bwt, by = list(smoking_status = birthwt$smoke), FUN = mean)
```

```
##   smoking_status        x
## 1             no 3515.639
## 2            yes 3260.285
```

```
obs_diff <- aggregate(x = birthwt$bwt, by = list(smoking_status = birthwt$smoke),
    FUN = mean)[,2] %>% diff()



sim_diff_samples <- replicate(1000, {
  smoke.sim <- birthwt
  smoke.sim$smoke <- sample(birthwt$smoke)
  sim_diff <-
```

```
    aggregate(bwt ~ smoke, smoke.sim, mean)[, 2] %>% diff()
})

# H_a, bwt less in smoking

(pval <- sim_diff_samples < obs_diff) %>% mean()
```

```
## [1] 0
```

Wilcoxon-Mann-Whitney Test

Suppose that we wanted to use the U-Statistic from the *Wilcoxon-Mann-Whitney* test:

> *U* is the number of data points where $s_i < n_i$;

The outer Product again

In order to perform this test we can use the outer product (as opposed to an awful nested for loop), for more information on the use of the outer product refer to:

- Using the Outer Product for Wilcoxon-Mann-Whitney Test
- The Outer Product Generally

```
# Return all the birthweights for nonSmokers
## method 1
bwt_nonsmoke <- birthwt[birthwt$smoke=="no",]$bwt
bwt_smoke <- birthwt[birthwt$smoke=="yes",]$bwt
## method 2
bwt_nonsmoke <- subset(x = birthwt, subset = birthwt$smoke=="no", select = bwt,
    drop = TRUE)
bwt_smoke <- subset(x = birthwt, subset = birthwt$smoke=="yes", select = bwt, drop
    = TRUE)

# Sum the values
outer(bwt_smoke, bwt_nonsmoke, "<") %>% sum()
```

```
## [1] 229164
```

This can be done by using the built-in functionj (any differences will be due to special treatment when observations are equal):

```
wilcox.test(bwt ~ smoke, birthwt, alternative = "greater")
```

```
##
```

```
##  Wilcoxon rank sum test with continuity correction
##
## data: bwt by smoke
## W = 231918, p-value < 2.2e-16
## alternative hypothesis: true location shift is greater than 0
```

Standardisation

#standardise

It is often useful to standardise data, an example of this is when performing PCA, in ***R*** it is also known as scaling the data.

Standardise the Observations

```
birthwt$std_bwt <- scale(birthwt$bwt, center = TRUE, scale = TRUE)

xbar <- mean(birthwt$std_bwt)
s    <- sd(birthwt$std_bwt)
x    <- birthwt$std_bwt
birthwt$std_bwt <- (x-xbar)/s
```

Calculate the Pooled Variance

Recall that the pooled variance is merely the variance of both populations:

$$s_p^2 = \frac{1}{n_1 - 1 + n_2 - 1} \cdot \sum_{i=1}^{n_1+n_2} \left[(x_i - \overline{x})^2 \right]$$

$$= \frac{(n_1 - 1) \times s_1^2 + (n_2 - 1) \times s_2^2}{n_1 - 1 + n_2 - 1}$$

```
# Simple
(v_p1 <- var(birthwt$bwt))
```

```
## [1] 270498.5
```

```
s_p1 <- sd(birthwt$bwt)

# Formula
```

```
## Summary Stats
### Smokers
n_s <- sum(birthwt$smoke=="yes")
df_s <- n_s -1
sd_s <- sd(birthwt$bwt[birthwt$smoke=="yes"])
xbar_s <- mean(birthwt$bwt[birthwt$smoke=="yes"])
var_s <- var(birthwt$bwt[birthwt$smoke=="yes"])
### Non Smokers
n_n <- sum(birthwt$smoke=="no")
df_n <- n_n -1
xbar_n <- mean(birthwt$bwt[birthwt$smoke=="no"])
sd_n <- sd(birthwt$bwt[birthwt$smoke=="no"])
var_n <- var(birthwt$bwt[birthwt$smoke=="no"])

(v_p2 <- (df_s * sd_s^2 + df_n * sd_n^2)/(df_s + df_n))
```

```
## [1] 255114.6
```

```
s_p2 <- sqrt(v_p2)
```

Calculate the standard error for the difference

For proofs refer to the org file.

$$\sigma_{\overline{x}_1 \pm \overline{x}_2} = \sigma_p \cdot \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

```
SED <- s_p1* sqrt(1/n_s + 1/n_n)
SED <- s_p2* sqrt(1/n_s + 1/n_n)
```

Calculate the t-statistic

```
#   bwt ~ smoke
(t_stat <- (xbar_n - xbar_s)/(SED))
```

```
## [1] 8.652745
```

Using the Built in t.test

```r
t.test(bwt ~ smoke, birthwt)
```

```
##
##  Welch Two Sample t-test
##
## data: bwt by smoke
## t = 8.5811, df = 1003.2, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  196.9596 313.7478
## sample estimates:
##  mean in group no mean in group yes
##          3515.639          3260.285
```

Permutation Again

In order to perform the permutation based test we can use an additional library:

```r
library(coin)
coin::oneway_test(formula = bwt ~ smoke,
                  data = birthwt,
                  distribution = coin::approximate(nresample = 1000),
                  alternative = "greater")
```

```
##
##  Approximative Two-Sample Fisher-Pitman Permutation Test
##
## data: bwt by smoke (no, yes)
## Z = 8.4031, p-value < 0.001
## alternative hypothesis: true mu is greater than 0
```

Confidence Intervals

The mean value of a sample will depend on the sample taken, this is however an estimator for the population mean and so can be used to predict where the population mean truly lies. The Central Limit Theorem Provides that the distribution of Sample Means follows a normal distribution:

$$\overline{X} \sim \mathcal{N}\left(\mu, \left(\frac{\sigma}{\sqrt{n}}\right)\right)$$

Bootstrap

Bootstrapping involves taking multiple samples from a sample (allowing reppetition also known as replacement) and then plotting the distribution of a test statistic, this represents an estimate of the population distribution and is known as resampling. Generally this is done for the variance or median of a data set because a t -distribution can be used to create a confidence interval of the data set.

- The basic idea being that the number of different ways take 10 observations with repetition from a sample of 30 is very large $^{(10+30-1)}C_{10} > 9E^9$ and these samples would be expected to ideally behave similarly to samples taken from the population, given such a large size, sampling statistics should represent a good prediction of the population.

Writing our own Function

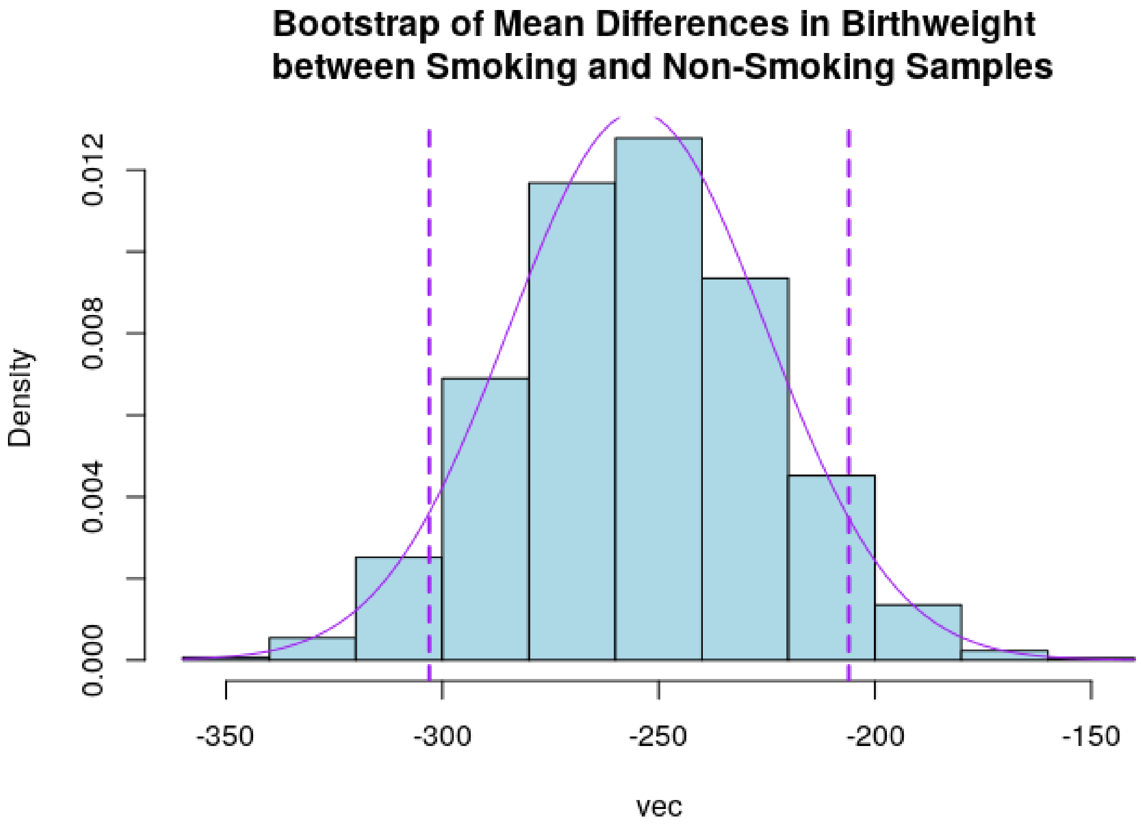
Implementing bootstrapping simply involves resampling the data and then recording the test statistic:

```r
bwt_smoke   <- birthwt[birthwt$smoke=="yes", 1]
bwt_nonsmoke <- birthwt[birthwt$smoke=="no" , 1]
d0 <- mean(bwt_smoke) - mean(bwt_nonsmoke)


d_vec <- replicate(3000, {
 bwt_smoke <- sample(bwt_smoke, replace = TRUE)
 bwt_nonsmoke <- sample(bwt_nonsmoke, replace = TRUE)
 d <- mean(bwt_smoke) - mean(bwt_nonsmoke)
 d
})
```

Plot Histogram (Base)

In order to plot a Histogram using base the `curve()` and `dnorm()` function can be used to visualise the density curve:

```r
# Plot Histogram

dens_hist <- function(vec, main = paste("Histogram With Density Curve"), col =
    "purple", fill = "lightblue") {

  # Make the Histogram
  hist(vec, freq = FALSE, col = fill, main = main)
  c_int <- quantile(x = vec, c(0.05, 0.95))

  # Plot the Curve
  x <- 1:2000
  curve(dnorm(x = x, mean = mean(vec), sd = sd(vec)), add = TRUE, col = col)
  abline(v = c_int[1], col = col, lwd = 2, lty = 2)
  abline(v = c_int[2], col = col, lwd = 2, lty = 2)

}

dens_hist(d_vec, main = "Bootstrap of Mean Differences in Birthweight \n between
    Smoking and Non-Smoking Samples")
```

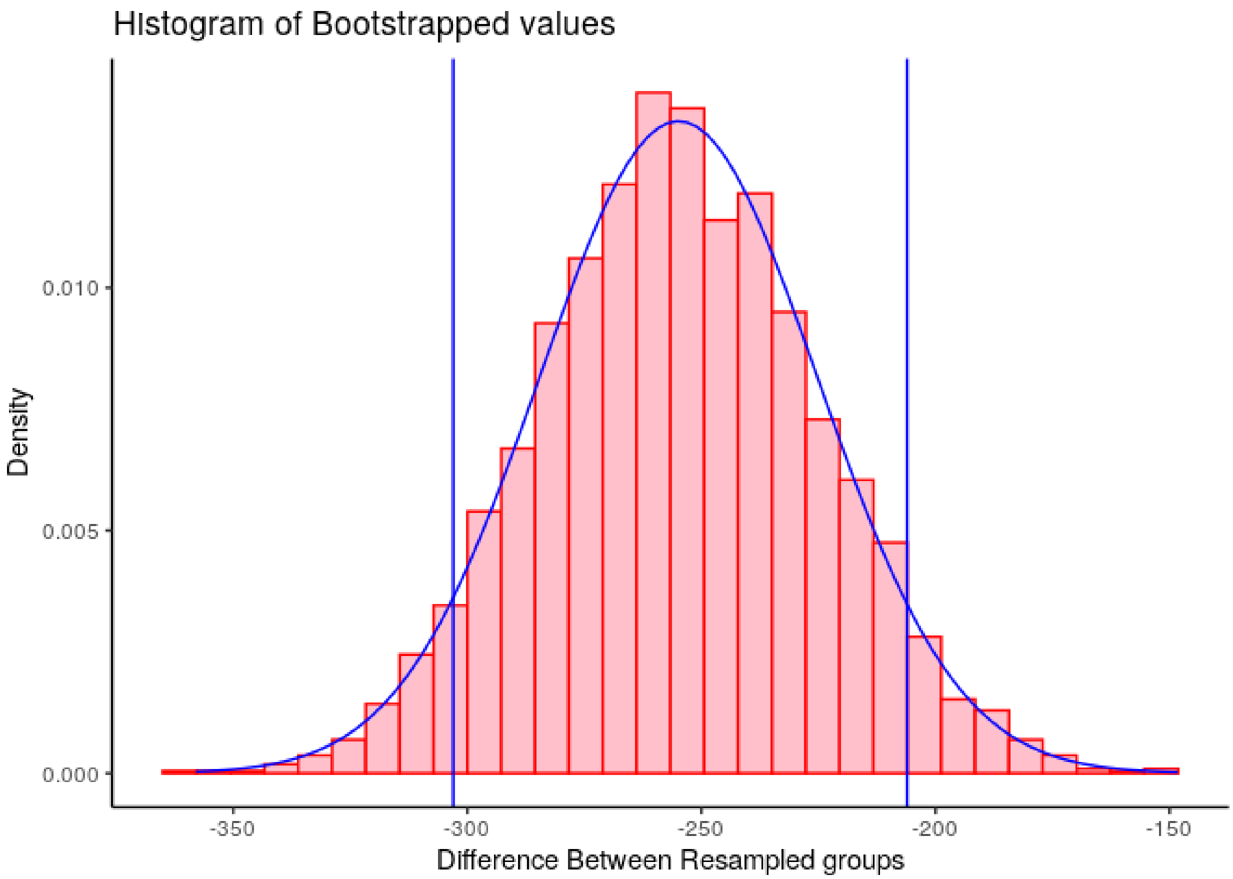


Plot Histogram (ggplto2)

This can also be done in ggplot2 by using the `stat_function()` with `dnorm()`:

```
d_vec.tb <- tibble::enframe(d_vec, value = "difference")
d_vec.tb <- d_vec.tb %>% dplyr::select("difference")


ggplot(d_vec.tb, aes(x = difference)) +
  geom_histogram(aes(y = ..density..), fill = "pink", col = "red") +
  stat_function(col = "blue",
                fun = dnorm,
                args = list(mean = mean(d_vec.tb$difference),
                            sd = sd(d_vec.tb$difference))) +
  geom_vline(xintercept = quantile(d_vec.tb$difference, c(0.05)), col = "blue") +
  geom_vline(xintercept = quantile(d_vec.tb$difference, c(0.95)), col = "blue") +
  theme_classic() +
  labs(title = "Histogram of Bootstrapped values", x = "Difference Between
       Resampled groups", y = "Density")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
quantile(d_vec.tb$difference, c(0.05))
```

```
##      5%
## -303.03
```

#bootstrap

Using the boot() library

The boot library will take a data set and a function as an argument and perform resampling, however, the function passed to it must take two arguments, a dataframe and an index, boot will pass a matrix of values to that function where in each row represents a subsequent re-sample.

```
library(boot)
```

```
##
## Attaching package: 'boot'
## The following object is masked from 'package:survival':
##
##     aml
```

```
## The following object is masked from 'package:lattice':
##
##     melanoma
```

```
# Create a Function for Mean Differences
mean_diff <- function(dataframe,i) {
    resampled_data = dataframe[i,]
    -diff(aggregate(bwt ~ smoke, resampled_data, mean)$bwt)
}

# Use the boot Function
b <- boot(birthwt, mean_diff, R = 1000, strata = birthwt$smoke)
boot.ci(b, type="basic", conf = 0.87)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = b, conf = 0.87, type = "basic")
##
## Intervals :
## Level      Basic
## 87%   (206.7, 301.0 )
## Calculations and Intervals on Original Scale
```

t-based

Generally the t -distribution can be used in order to give confidence intervals for the true population value of the mean:

```
t.test(bwt ~ smoke, data = birthwt, var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data: bwt by smoke
## t = 8.6527, df = 1224, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  197.4554 313.2520
## sample estimates:
##  mean in group no mean in group yes
##         3515.639         3260.285
```

```
t.test(bwt ~ smoke, data = birthwt, var.equal = TRUE, conf = 0.99)
```

```
##
##  Two Sample t-test
##
## data: bwt by smoke
## t = 8.6527, df = 1224, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 99 percent confidence interval:
##  179.2189 331.4885
## sample estimates:
##  mean in group no mean in group yes
##         3515.639         3260.285
```

```
t.test(bwt ~ smoke, data = birthwt, var.equal = FALSE, conf = 0.99)
```

```
##
##  Welch Two Sample t-test
##
## data: bwt by smoke
## t = 8.5811, df = 1003.2, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 99 percent confidence interval:
##  178.5573 332.1501
## sample estimates:
##  mean in group no mean in group yes
##         3515.639         3260.285
```

Notice how being able to assume that the populations have the same variance can improve the prediction, this is because it reduces the standard error which is $\frac{\sigma}{\sqrt{n}}$

Challenge

Sales Data

Using the data `salesEW.csv`:

- determine if there is a difference in mean sales between the East and West offices.
- Compute the 90% confidence interval for the difference in means.

Difference in Mean Values

Presuming that the variances are equal:

```
# Perform the t-Test
# salesEW
sales_test <- t.test(sales ~ office, data = salesEW, var.equal = TRUE, conf = 0.9)

# Is there a difference in mean sales?
if (sales_test$p.value < 0.1) {
  paste("There is a difference in average sales between offices (p=",
      signif(sales_test$p.value, 1), ")") %>% print()
} else {
  print("There is insufficient evidence to suggest a difference in the average
      sales between offices")
}
```

```
## [1] "There is a difference in average sales between offices (p= 0.06 )"
```

```
# What is a 90% confidence interval for the average difference
upr <- sales_test$conf.int[2]
lwr <- sales_test$conf.int[1]
  ## What's the direction?
  (office_means <- aggregate(sales ~ office, salesEW, mean))
```

```
##   office   sales
## 1   east 162.6991
## 2   west 154.0425
```

```
    # East Sells More

  paste0("The Eastern Office Sells more by an interval of (", signif(lwr, 2), ", ",
      signif(upr, 2), ")") %>% print()
```

```
## [1] "The Eastern Office Sells more by an interval of (1.2, 16)"
```

Hence it can be concluded that:

- The probability of concluding that the offices sell differing amounts under the assumption that there is no difference between the two offices is only 6% and hence this is rejected and it is accepted that the offices sell differing amounts.
  - The p-value for the Eastern office selling more is 2.8%.
- The probability of taking a sample from the offices and finding the average difference from West to East outside the range of an increase (1.2, 16) is only 10%

    - If the null hypothesis is that the difference is 0, the probability of incorrectly rejecting that null hypothesis is hence 10%
        - (This means assuming that the null hypothesis is true, be careful of the distinction between the *False Discovery Rate* and the *False Positive Rate*, mixing these two up is known as the *Base Rate Fallacy*)

An equivalent interpretation of the Confidence interval would be:

- If a 90% confidence interval was drawn from a sample of the difference in sales, the probability of the actual population mean value being contained in that interval would be 90%
    - i.e. if 30 samples were taken, the true population mean would be expected to be found in 27 of those intervals.
        - Or rather if 30×10^9 samples were taken, the true population would be found in 27×10^9 rounded to 2 sig. figures.

Although the difference is statistically significant, the difference represents less than 10% of the recorded sales and so the difference would be a non-meaningful significant difference.

Spider Data

Using the data `spider.csv`:

- determine if there is an increase in mean anxiety levels when seeing a spider.
- Compute the confidence interval for the differenc in mean anxiety levels

Greater Mean value

In this case we won't assume that the variance between the groups is equal, perhaps certain individuals react differently to photographs for particular reasons and so this might cause a greater spread of the data for the photo group (i.e. this possibility substantiates an evidentiary burden, there is a persuasive burden required to take the variances as equal)

```
## create the t.test
str(spider)
```

```
## 'data.frame':   24 obs. of 2 variables:
##  $ Group  : Factor w/ 2 levels "Picture","Real Spider": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Anxiety: int 30 35 45 40 50 35 55 25 30 45 ...
```

```
(spider_test <- t.test(formula = Anxiety ~ Group, data = spider, var.equal = FALSE,
    alternative = "greater"))
```

```
##
##  Welch Two Sample t-test
##
## data: Anxiety by Group
## t = -1.6813, df = 21.385, p-value = 0.9464
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
## -14.15808      Inf
## sample estimates:
##    mean in group Picture mean in group Real Spider
##                     40                      47
```

The probability of concluding that real spiders cause more anxiety, assuming that they do not in fact cause more anxiety, is 11%. Although this is still a low p-value it would not substantiate a sufficient amount of evidence in order to make that conclusion.

Confidence Interval for Difference in Levels

The confidence interval depends on the alternative hypothesis, It should generally however be drawn from the null hypothesis, 0.

```
str(spider)
```

```
## 'data.frame':  24 obs. of 2 variables:
##  $ Group : Factor w/ 2 levels "Picture","Real Spider": 1 1 1 1 1 1 1 1 1 ...
##  $ Anxiety: int 30 35 45 40 50 35 55 25 30 45 ...
```

```
(spider_test <- t.test(Anxiety ~ Group, spider, var.equal = FALSE, alternative =
    "two.sided", conf=0.9))
```

```
##
##  Welch Two Sample t-test
##
## data: Anxiety by Group
## t = -1.6813, df = 21.385, p-value = 0.1072
## alternative hypothesis: true difference in means is not equal to 0
## 90 percent confidence interval:
## -14.1580825  0.1580825
## sample estimates:
##    mean in group Picture mean in group Real Spider
##                     40                      47
```

```
paste("There is 90% probability that the average anxiety level induced by a mere
    Picture of a spider will be different from that of a spider by between",
    signif(spider_test$conf.int[1], 2), "and", signif(spider_test$conf.int[2], 2))
```

```
## [1] "There is 90% probability that the average anxiety level induced by a mere
```

```
Picture of a spider will be different from that of a spider by between -14 and
0.16"
```

If multiple samples from the population were taken, 90% of those drawn intervals would contain the population mean value, by such reasoning the probability of this confidence interval, (-14, 0.16), containing the true population mean is 90%.