BACS-hw05-107070004

Question 2)

Let's return to the strictly fictional scenario (but with real data) from last week's Verizon dataset. Imagine this time that Verizon claims that they take no more than 7.6 minutes on average (single-tail test) to repair phone services for its customers. The file verizon.csv has a recent sample of repair times collected by the New York Public Utilities Commission, who seeks to verify this claim at 99% confidence.

(a) Recreate the traditional hypothesis test of last week using high-level built-in functions of R: (you may have to see the R help documentation, google how to use them, or ask for help on Teams)

```
repairtimes <- read.csv("verizon.csv", header=TRUE)$Time
```

(i) Use the t.test() function to conduct a one-sample, one-tailed t-test: report 99% confidence interval of the mean, t-value, and p-value

```
t.test(repairtimes, mu=7.6, conf.level = 0.99, alternative = "less")
```

```
##
##  One Sample t-test
##
## data:  repairtimes
## t = 2.5608, df = 1686, p-value = 0.9947
## alternative hypothesis: true mean is less than 7.6
## 99 percent confidence interval:
##      -Inf 9.360414
## sample estimates:
## mean of x
##  8.522009
```

(ii) Use the power.t.test() function to tell us the power of the test

```
hyp_mean <- 7.6
repairtimes_mean <- mean(repairtimes)
repairtimes_sd <- sd(repairtimes)
power.t.test(n = length(repairtimes), delta = repairtimes_mean-hyp_mean,
sd = repairtimes_sd, alternative = 'one.sided')
```

```
##
##      Two-sample t test power calculation
##
##               n = 1687
##           delta = 0.9220095
##              sd = 14.78848
##       sig.level = 0.05
##           power = 0.5657309
##     alternative = one.sided
##
## NOTE: n is number in *each* group
```

(b) Let's use bootstrapped hypothesis testing to re-examine this problem:

(i) Retrieve the original t-value from traditional methods (above)

```r
repairtimes_se <- sd(repairtimes)/sqrt(length(repairtimes))
repairtimes_se
```

```
## [1] 0.3600527
```

```r
t_value <- (repairtimes_mean-hyp_mean)/repairtimes_se
t_value
```

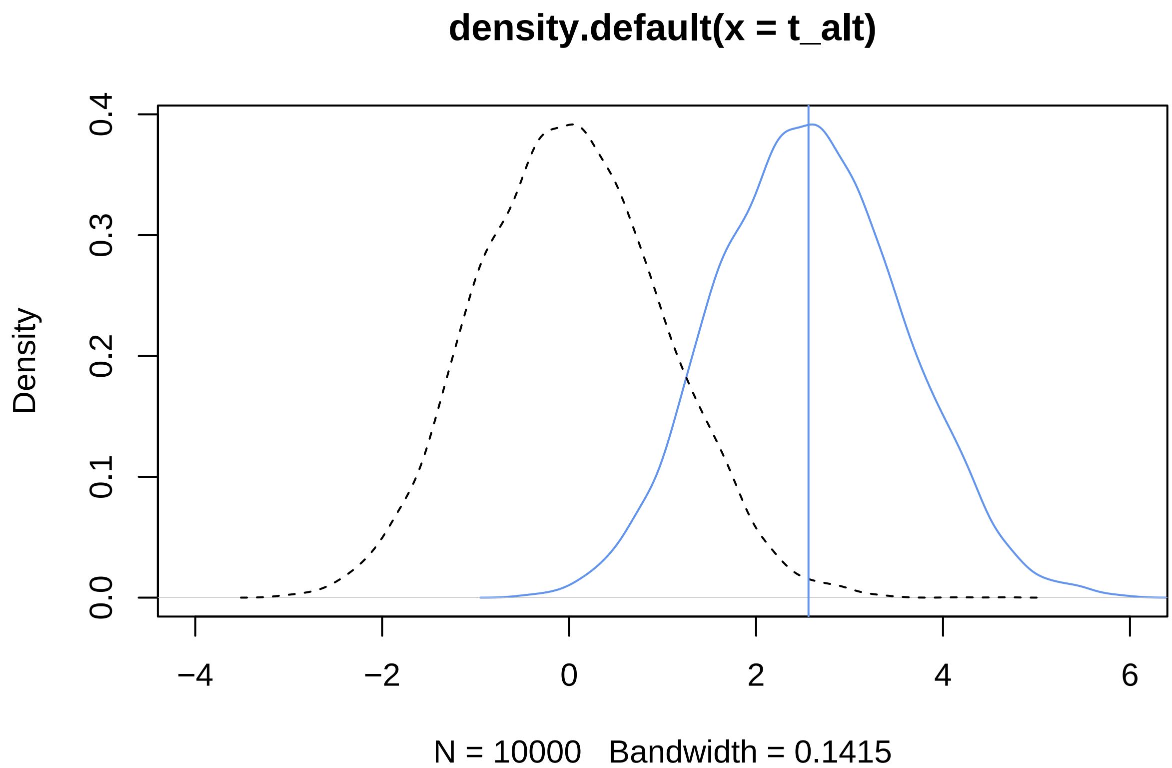
```
## [1] 2.560762
```

(ii) Bootstrap the null and alternative t-distributions

```r
bootstrapped_null_alt <- function(sample0, hyp_mean) {
    resample <- sample(sample0, length(sample0), replace=TRUE)
    resample_se <- sd(sample0)/sqrt(length(resample))

    t_stat_alt <- (mean(resample)-hyp_mean)/resample_se
    t_stat_null <- (mean(resample)-mean(sample0))/resample_se

    c(t_stat_alt, t_stat_null)
}
boot_t_stats <- replicate(10000, bootstrapped_null_alt(repairtimes, hyp_mean))
t_alt <- boot_t_stats[1,]
t_null <- boot_t_stats[2,]
plot(density(t_alt), col="cornflowerblue", xlim=c(-4,6))
lines(density(t_null), lty="dashed")
abline(v=t_value, col="cornflowerblue")
```



(iii) Find the 99% cutoff value for critical null values of t (from the bootstrapped null); What should our test conclude when comparing the original t-value to the 99% cutoff value?

```
cutoff_99 <- quantile(t_null, probs=c(0.005, 0.995))
cutoff_99
```

```
##      0.5%     99.5%
## -2.355968  2.669487
```

(iv) Compute the p-value and power of our bootstrapped test

```
null_probs <- ecdf(t_null)
one_tailed_pvalue <- 1 - null_probs(t_value)
one_tailed_pvalue
```

```
## [1] 0.0071
```

```
alt_probs <- ecdf(t_alt)
alt_probs(cutoff_99[1]) + (1-alt_probs(cutoff_99[2]))
```

```
## [1] 0.456
```