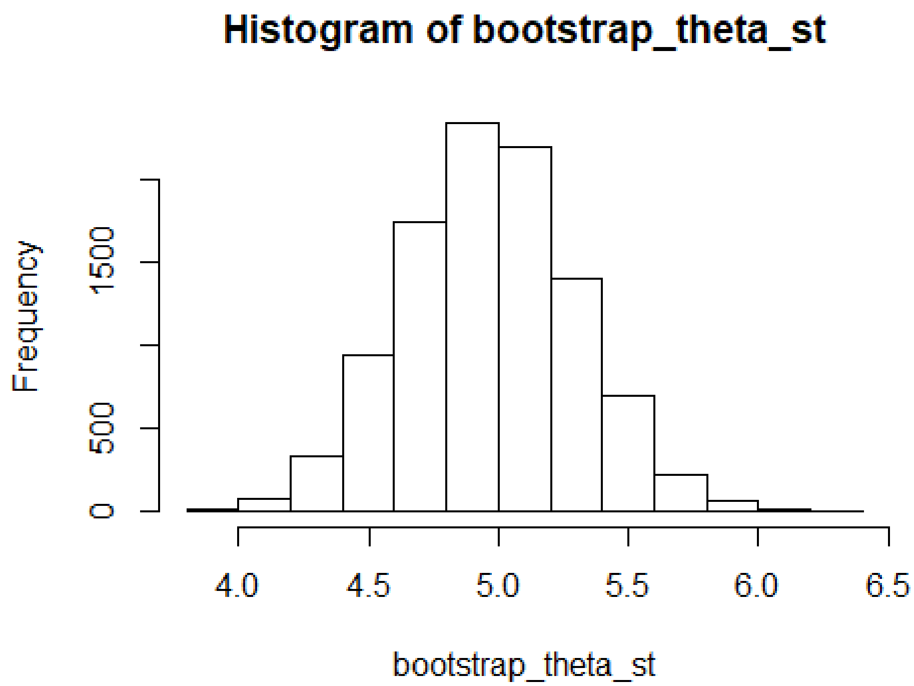
Yichen Dong HW 9

Yichen Dong

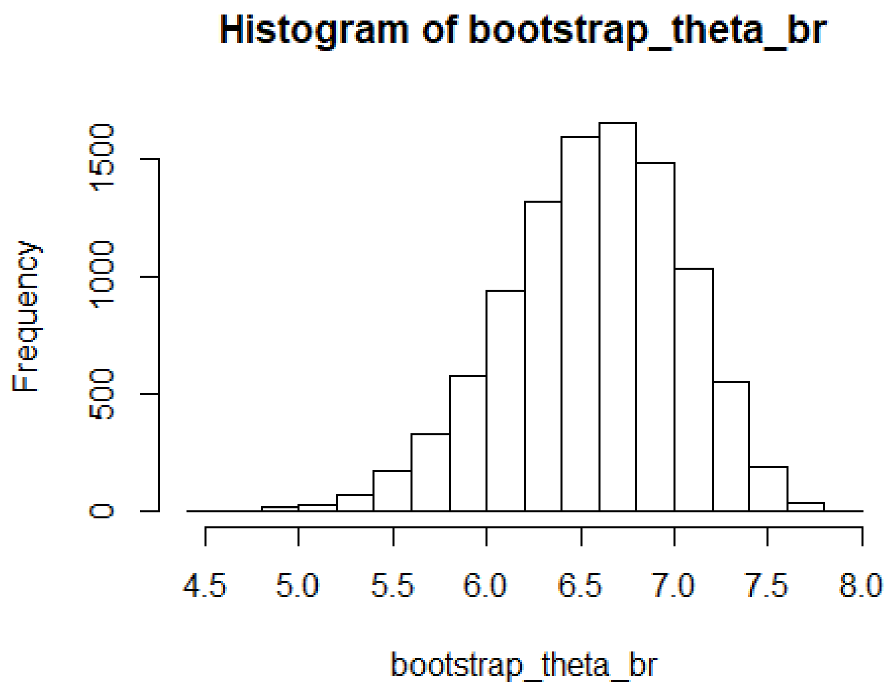
November 1, 2018

Problem 1

```
stomach = c(25,42,45,46,51,103,124,146,340,396,412,876,1112)
breast = c(24,40,719,727,791,1166,1235,1581,1804,3460,3808)
log_stomach = log(stomach)
log_breast = log(breast)
mean_log_stomach = mean(log_stomach)
mean_log_breast = mean(log_breast)

b = 10000
bootstrap_theta_st = NULL
for(i in 1:b){
  n = length(stomach)
  data_bootstrap = sample(log_stomach,size = n, replace = TRUE)
  bootstrap_mean = mean(data_bootstrap)
  bootstrap_theta_st[i] = bootstrap_mean
}
bootstrap_theta_bar_st = mean(bootstrap_theta_st)
bootstrap_variance_st = 1/(b-1)*sum((bootstrap_theta_st-bootstrap_theta_bar_s
t)^2)
#Finding the CI using an alpha of .05 and using the Percentile Method
sorted_bootstrap_theta_st = sort(bootstrap_theta_st)
bootstrap_theta_CI_st = c(bootstrap_theta_bar_st- sqrt(bootstrap_variance_st)
*qnorm(.975),bootstrap_theta_bar_st +sqrt(bootstrap_variance_st)*qnorm(.975))
bootstrap_theta_CI_st_pct = c(sorted_bootstrap_theta_st[round(b*.025)],sorted
_bootstrap_theta_st[round(b*.975)])
bootstrap_variance_st

## [1] 0.1088757

hist(bootstrap_theta_st)
```



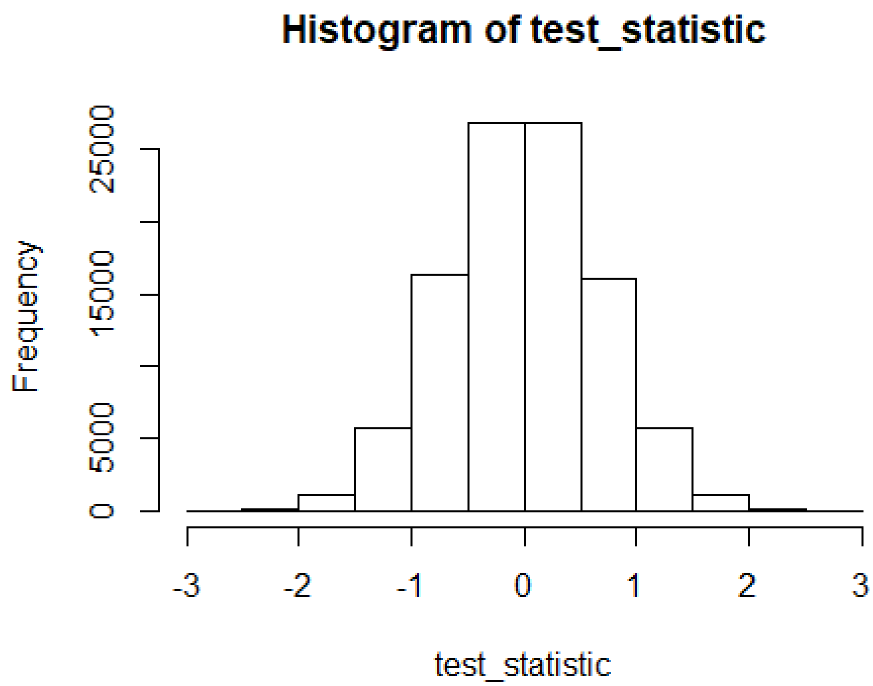

```
bootstrap_theta_CI_st

## [1] 4.319724 5.613156

bootstrap_theta_CI_st_pct

## [1] 4.330936 5.623129

#For breasts
b = 10000
bootstrap_theta_br = NULL
for(i in 1:b){
  n = length(breast)
  data_bootstrap = sample(log_breast,size = n, replace = TRUE)
  bootstrap_mean = mean(data_bootstrap)
  bootstrap_theta_br[i] = bootstrap_mean
}
bootstrap_theta_bar_br = mean(bootstrap_theta_br)
bootstrap_variance_br = 1/(b-1)*sum((bootstrap_theta_br-bootstrap_theta_bar_b
r)^2)
#Finding the CI using an alpha of .05 and using the Percentile Method
sorted_bootstrap_theta_br = sort(bootstrap_theta_br)
bootstrap_theta_CI_br = c(bootstrap_theta_bar_br- sqrt(bootstrap_variance_br)
*qnorm(.975),bootstrap_theta_bar_br +sqrt(bootstrap_variance_br)*qnorm(.975))
bootstrap_theta_CI_br_pct = c(sorted_bootstrap_theta_br[round(b*.025)],sorted
_bootstrap_theta_br[round(b*.975)])
hist(bootstrap_theta_br)
```



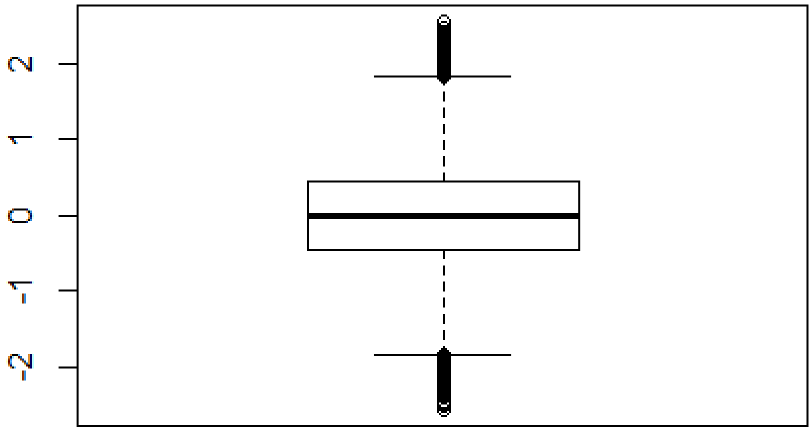

```
bootstrap_variance_br

## [1] 0.2191473

bootstrap_theta_CI_br

## [1] 5.647315 7.482358

bootstrap_theta_CI_br_pct

## [1] 5.562023 7.387759
```

We can see that the two confidence intervals are pretty close. ### Part b

```r
# I'm not sure how to permute, so I'm just going to mix up the variables many
many times and hope that's a good approximation
combined = c(log_stomach,log_breast)
itr = 100000
test_statistic = NULL
for(i in 1:itr){
  sample = sample(combined)
  permute_stomach = sample[1:13]
  permute_breast = sample[14:24]
  test_statistic[i] = mean(permute_stomach) - mean(permute_breast)
}
hist(test_statistic)
```

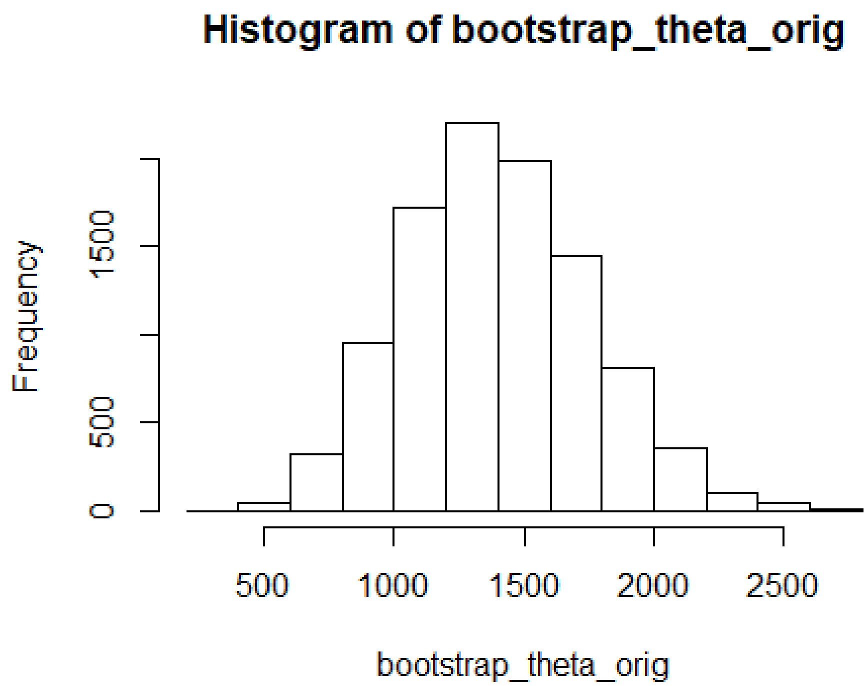



```r
mean(test_statistic)

## [1] -0.00214056

boxplot(test_statistic)
```



We can see that the different between the two groups is centered around 0. ### Part C

```
exp_breast_ci = exp(bootstrap_theta_CI_br_pct)
b = 10000
bootstrap_theta_orig = NULL
for(i in 1:b){
  n = length(breast)
  data_bootstrap = sample(breast,size = n, replace = TRUE)
  bootstrap_mean = mean(data_bootstrap)
  bootstrap_theta_orig[i] = bootstrap_mean
}
bootstrap_theta_bar_orig = mean(bootstrap_theta_orig)
bootstrap_variance_orig = 1/(b-1)*sum((bootstrap_theta_orig-bootstrap_theta_b
ar_orig)^2)
hist(bootstrap_theta_orig)
```

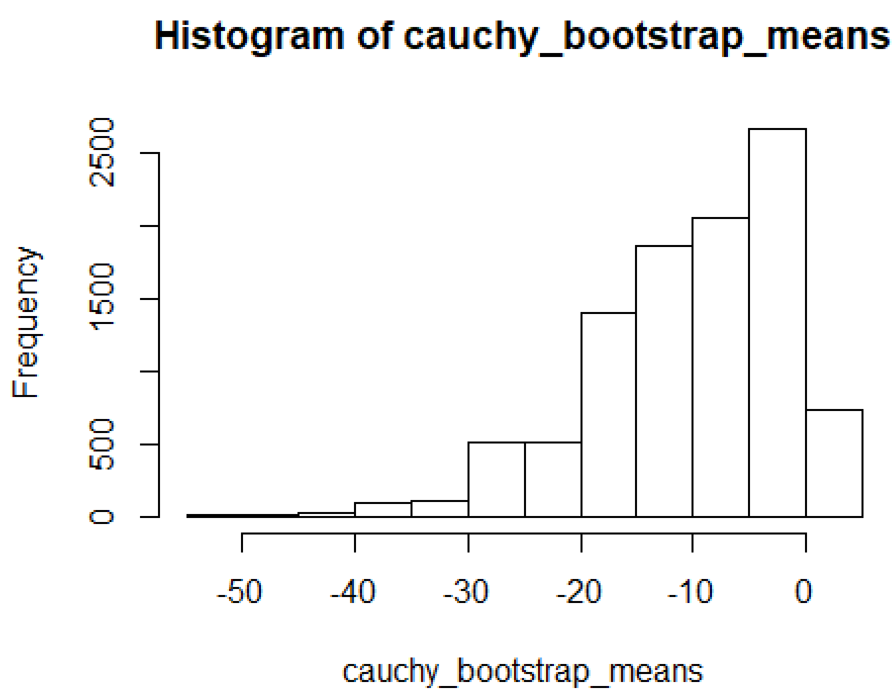



```
#Finding the CI using an alpha of .05 and using the Percentile Method
sorted_bootstrap_theta_orig = sort(bootstrap_theta_orig)
bootstrap_theta_CI_orig = c(bootstrap_theta_bar_orig- sqrt(bootstrap_variance
_orig)*qnorm(.975),bootstrap_theta_bar_orig +sqrt(bootstrap_variance_orig)*qn
orm(.975))
bootstrap_theta_CI_orig_pct = c(sorted_bootstrap_theta_orig[round(b*.025)],so
rted_bootstrap_theta_orig[round(b*.975)])
exp_breast_ci

## [1]  260.349 1616.080

bootstrap_variance_orig

## [1] 126378.9

bootstrap_theta_CI_orig

## [1]  699.9172 2093.4441

bootstrap_theta_CI_orig_pct

## [1]  746.5455 2122.9091
```

We can see that the confidence intervals for the exponential confidence interval is a lot different from the one for just the original data.

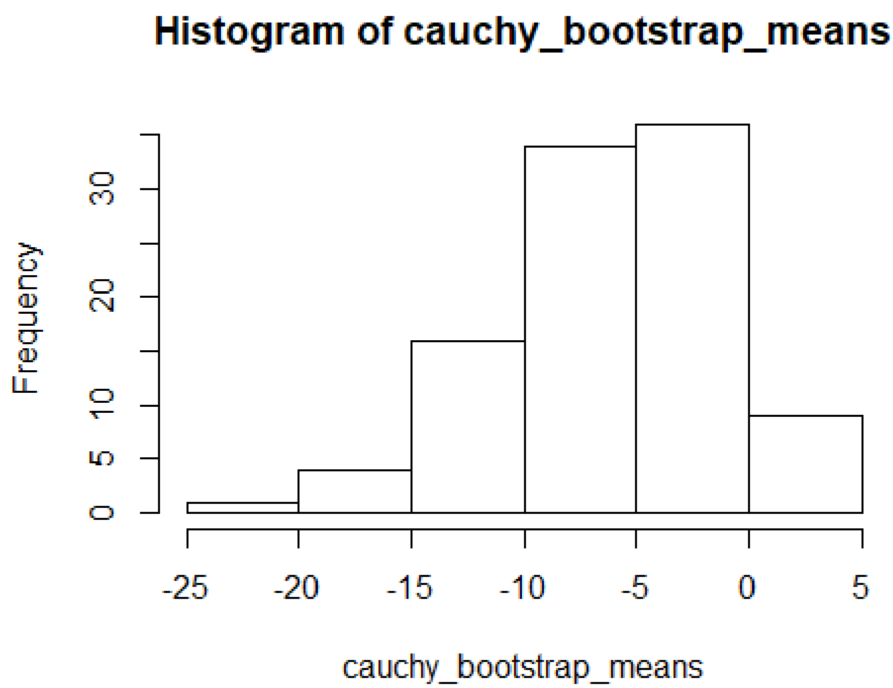
Problem 2

```r
cauchy = rcauchy(1000)
mean(cauchy)

## [1] -10.24537

b = 10000
cauchy_bootstrap_means = NULL
for(i in 1:b){
  n = length(cauchy)
  bootstrap_cauchy = sample(cauchy,size = n,replace = TRUE)
  cauchy_bootstrap_means[i] = mean(bootstrap_cauchy)
}
hist(cauchy_bootstrap_means)
```

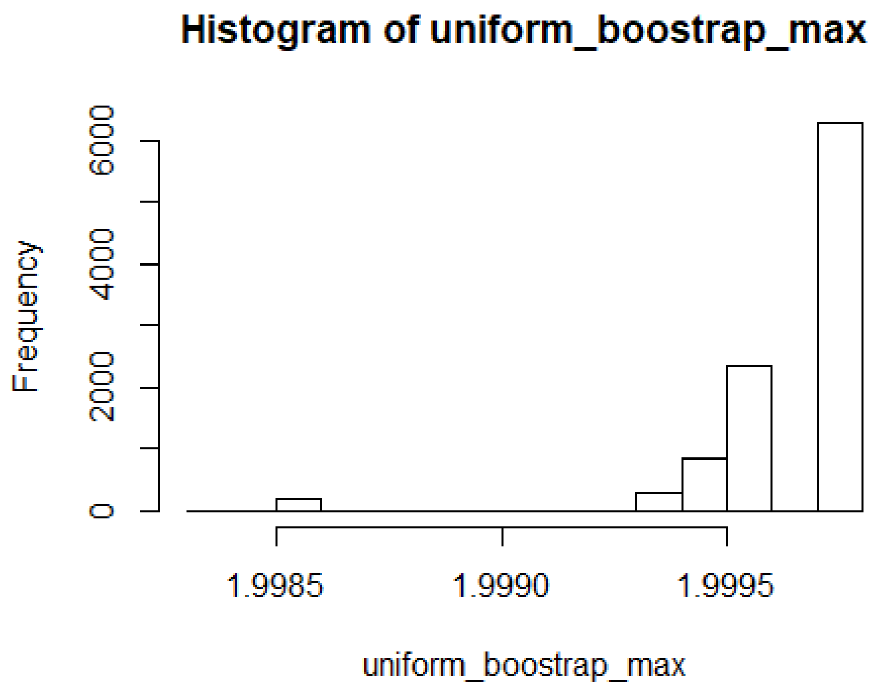



```r
mean(cauchy_bootstrap_means)

## [1] -10.33495

#testing a smaller bootstrap
cauchy = rcauchy(100)
mean(cauchy)

## [1] -5.394489

b = 100
cauchy_bootstrap_means = NULL
for(i in 1:b){
```

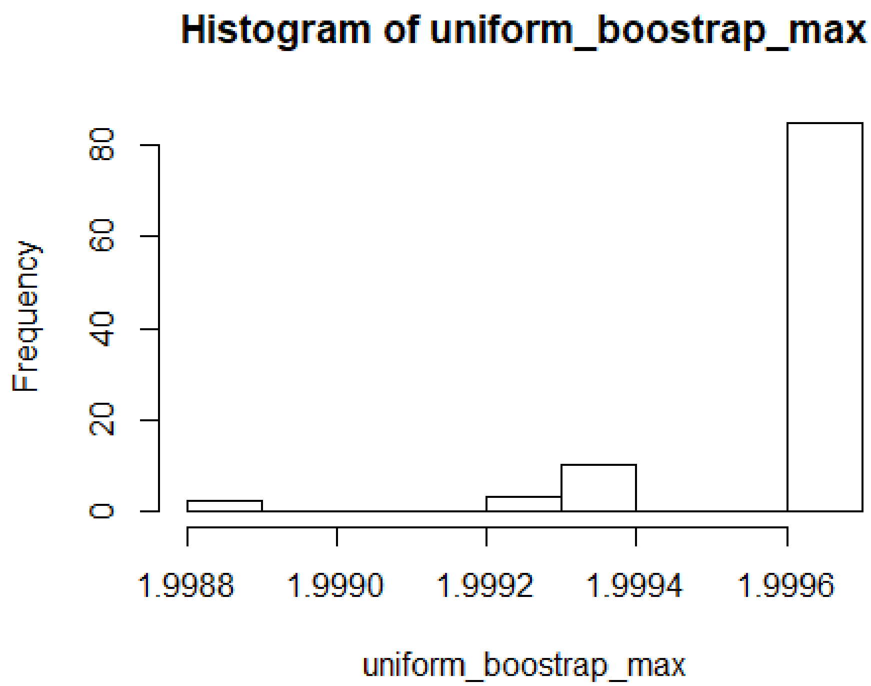
```r
  n = length(cauchy)
  bootstrap_cauchy = sample(cauchy,size = n,replace = TRUE)
  cauchy_bootstrap_means[i] = mean(bootstrap_cauchy)
}
hist(cauchy_bootstrap_means)
```




```r
mean(cauchy_bootstrap_means)

## [1] -5.215819

# for estimating max
uniform = runif(10000,0,2)
max(uniform)

## [1] 1.99973

b = 10000
uniform_boostrap_max = NULL
for(i in 1:b){
  n = length(uniform)
  bootstrap_uniform = sample(uniform,size=n,replace= TRUE)
  uniform_boostrap_max[i] = max(bootstrap_uniform)
}
hist(uniform_boostrap_max)
```




```
max(uniform_boostrap_max)

## [1] 1.99973

#testing a smaller bootstrap
uniform = runif(10000,0,2)
max(uniform)

## [1] 1.999659

b = 100
uniform_boostrap_max = NULL
for(i in 1:b){
  n = length(uniform)
  bootstrap_uniform = sample(uniform,size=n,replace= TRUE)
  uniform_boostrap_max[i] = max(bootstrap_uniform)
}
hist(uniform_boostrap_max)
```




```
max(uniform_boostrap_max)

## [1] 1.999659
```

I'm honestly not too sure about this. I feel like they are predicting pretty accurately for the max and the mean of the cauchy. Maybe it's because the histogram does not look centered around a single value, so that the variance of a small number of bootstraps could be very high?

Problem 3

Problem 3

$E_{\hat{p}}(\bar{Y}^*) = \bar{Y}$ $\qquad \bar{Y} = \dfrac{\sum_{i=1}^{n} Y_i}{n}$

$$\bar{Y}^* = \frac{\sum_{i=1}^{n} Y_i^*}{n}$$

Since it is drawn with replacement, each Y_i^* has an equal chance to be each Y_i. Thus, $E_{\hat{p}}(Y_i^*) = \dfrac{\sum_{i=1}^{n} Y_i}{n}$

$E_{\hat{p}}\left(\dfrac{\sum_{i=1}^{n} Y_i^*}{n}\right) = \bar{Y}$ Expected value of sums is the sum of expected values

$$\frac{1}{n} \cdot \sum_{i=1}^{n} E_{\hat{p}}(Y_i^*) = \frac{1}{n} \cdot \sum_{i=1}^{n} \left(\frac{\sum_{i=1}^{n} Y_i}{n}\right)$$

$$= \frac{1}{n} \cdot n\left(\frac{\sum_{i=1}^{n} Y_i}{n}\right) = \frac{\sum_{i=1}^{n} Y_i}{n} = \bar{Y}$$

Thus we show that $E_{\hat{p}}(\bar{Y}^*) = \bar{Y}$

$E_p(\bar{Y}^*) = \mu$

We know that $\bar{Y} = \frac{\sum_{i=1}^{n} Y_i}{n}$. Then, $E_p(\bar{Y}) = E_p\left(\frac{\sum_{i=1}^{n} Y_i}{n}\right)$

$= \frac{1}{n} \cdot \sum_{i=1}^{n} E_p(Y_i)$

Since we are using the population distribution, each $E_p(Y_i) = \mu$. The

$E(\bar{Y}) = \frac{1}{n} \cdot \left(\sum_{i=1}^{n} \mu\right) = \frac{1}{n} \cdot n\mu = \mu$

We also know that $E_{\hat{p}}(\bar{Y}^*) = \bar{Y}$. Then,

Also, since each Y_i^* is drawn from the population distribution,

$E(Y_i^*) = E(Y_i) = \mu$

Thus

$E_p(\bar{Y}^*) = E_p\left(\frac{\sum_{i=1}^{n} Y_i^*}{n}\right) = \frac{1}{n} \cdot \sum_{i=1}^{n} E(Y_i^*) = \frac{1}{n} \cdot n\mu = \mu$

Problem 4

```
p4_rnorm = rnorm(100,0,1)
mean(p4_rnorm)

## [1] -0.1307336

##standard bootstrap
b = 10
p4_boot_mean = NULL
for(i in 1:b){
  n = length(p4_rnorm)
  bootstrap_norm = sample(p4_rnorm,size=n,replace=TRUE)
  p4_boot_mean[i] = mean(bootstrap_norm)
}
p4_theta_bar_star = mean(p4_boot_mean)
bias_corrected = 2*mean(p4_rnorm) - mean(p4_boot_mean)
p4_b_est_variance = 1/(b-1)*sum((p4_boot_mean - p4_theta_bar_star)^2)
p4_theta_bar_star

## [1] -0.07590083
```

```
bias_corrected

## [1] -0.1855664

p4_b_est_variance

## [1] 0.007953236

##balanced bootstrap
b=10
balance_p4_rnorm = NULL
for(i in 1:b){
  balance_p4_rnorm = c(balance_p4_rnorm,p4_rnorm)
}
balance_p4_rnorm_permute = sample(balance_p4_rnorm, length(balance_p4_rnorm))
balance_p4_means = NULL
for(i in 1:b){
  start = i*100 -99
  end = i*100
  balance_p4_means[i] = mean(balance_p4_rnorm_permute[start:end])
}
p4_balance_theta_bar_star = mean(balance_p4_means)
balance_bias_corrected = 2*mean(p4_rnorm) - mean(balance_p4_means)
p4_balance_variance = 1/(b-1)*sum((balance_p4_means-p4_balance_theta_bar_star
)^2)
mean(balance_p4_means)

## [1] -0.1307336

balance_bias_corrected

## [1] -0.1307336

p4_balance_variance

## [1] 0.005972435
```

We can see that the balanced bootstrap gave us an answer that was a lot closer to the real mean, with a smaller variance than the original bootstrap.