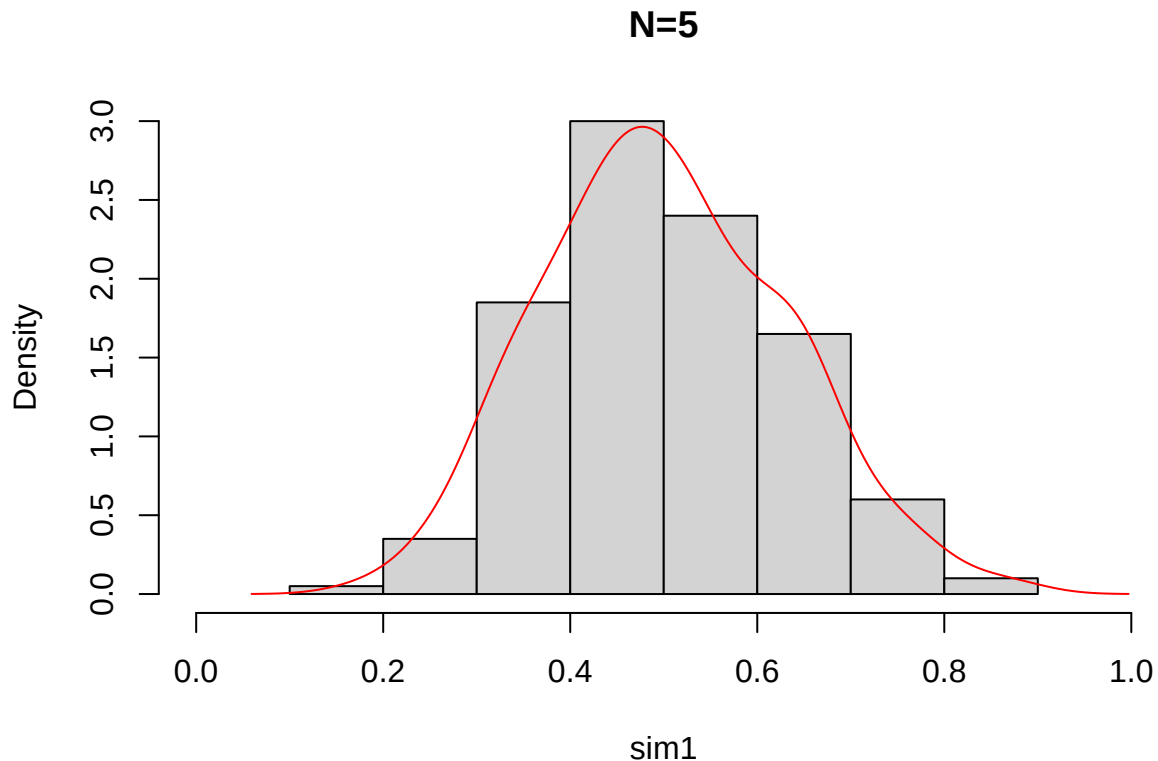


540hw4sol

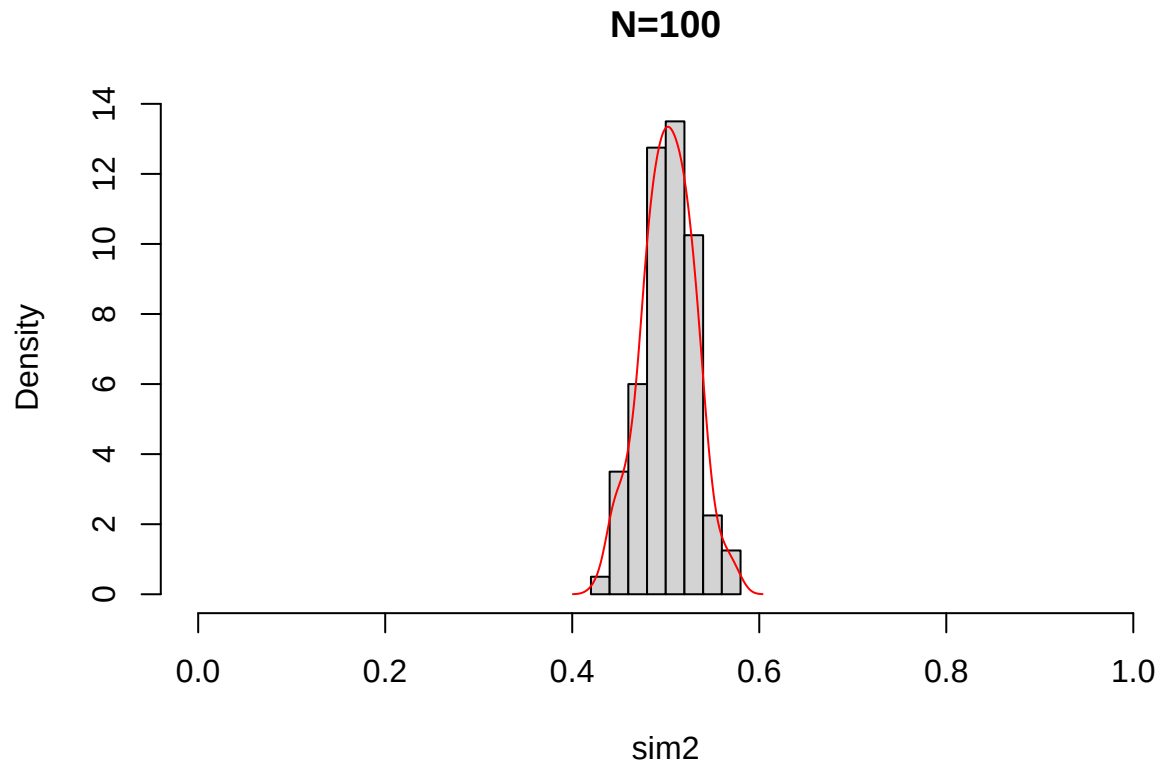
Problem 1

Both of the distributions for the sample means when $n = 5$ and $n = 100$ are bell-shaped at the center of 0.5. However, the variation of distribution for sample means when $n = 5$ is much bigger than that of the distribution when $n = 100$.

```
##draw n=5 samples from uniform distribution and repeat this
##experiment 200 times
sim1=replicate(200,mean(runif(5)))
hist(sim1,xlim=c(0,1),freq = F, main="N=5")
lines(density(sim1),col="red")
```

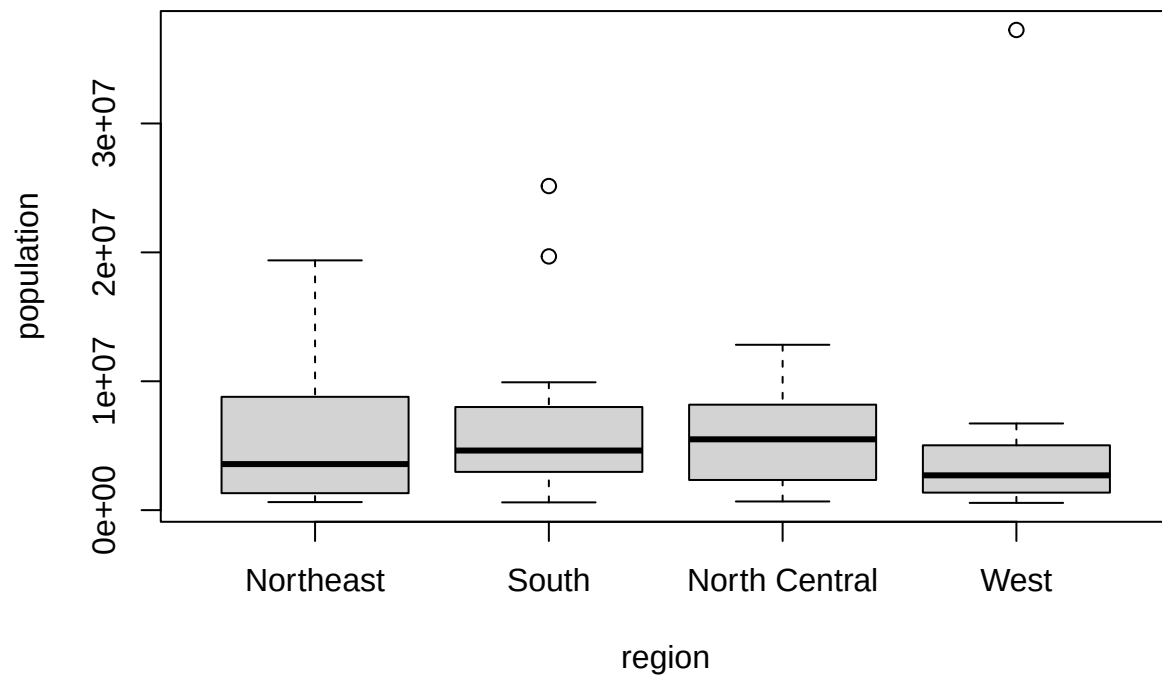


```
##draw n=100 samples from uniform distribution and repeat this
##experiment 200 times
sim2=replicate(200,mean(runif(100)))
hist(sim2,freq = F, main="N=100", xlim=c(0, 1))
lines(density(sim2),col="red")
```



Problem 2

```
library(dslabs)
data(murders)
boxplot(population~region, data=murders)
```



Problem 3

The answer is (b) ChickWeight is tidy: each observation (a weight) is represented by one row. The chick from which this measurement came is one of the variables.

```
data("ChickWeight")
str(ChickWeight)
```

```
## Classes 'nfnGroupedData', 'nfGroupedData', 'groupedData' and 'data.frame':  578 obs. of  4 variables:
## $ weight: num  42 51 59 64 76 93 106 125 149 171 ...
## $ Time : num  0 2 4 6 8 10 12 14 16 18 ...
## $ Chick : Ord.factor w/ 50 levels "18"<"16"<"15"<...: 15 15 15 15 15 15 15 15 15 15 ...
## $ Diet : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
## - attr(*, "formula")=Class 'formula' language weight ~ Time | Chick
## .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>
## - attr(*, "outer")=Class 'formula' language ~Diet
## .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>
## - attr(*, "labels")=List of 2
## ..$ x: chr "Time"
## ..$ y: chr "Body weight"
## - attr(*, "units")=List of 2
## ..$ x: chr "(days)"
## ..$ y: chr "(gm)"
```

```
ChickWeight[1:5,]
```

```
##   weight Time Chick Diet
## 1     42    0     1     1
## 2     51    2     1     1
## 3     59    4     1     1
## 4     64    6     1     1
## 5     76    8     1     1
```

Problem 4

```
library(NHANES)
data(NHANES)
subset_data <- subset(NHANES, Gender == "female" & AgeDecade == " 20-29")
mean_bp <- mean(subset_data$BPSysAve, na.rm = TRUE) # Remove missing values
sd_bp <- sd(subset_data$BPSysAve, na.rm = TRUE)     # Remove missing values
ref <- c(mean = mean_bp, sd = sd_bp)

print(ref)
```

```
##      mean      sd
## 108.42243 10.14668
```

Problem 5

(a)

```
library(tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

```

## Warning: package 'dplyr' was built under R version 4.4.3

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.2.0      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2     4.0.2      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.1
## v purrr      1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

url <- "https://people.stat.sc.edu/hoyen/STAT540/Data/baby.txt"
babiesraw <- read.table(file = url, header = TRUE)
babies <- filter(babiesraw, smoke != 9)
str(babies)

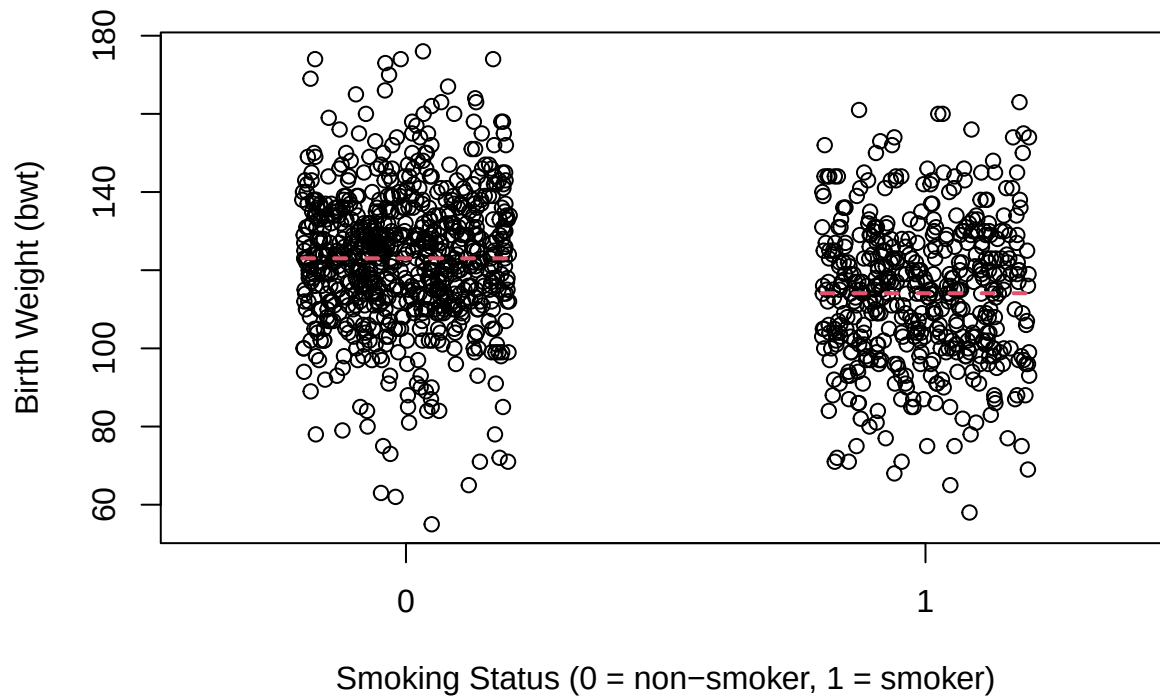
## 'data.frame':    1226 obs. of  7 variables:
## $ bwt      : int  120 113 128 123 108 136 138 132 120 143 ...
## $ gestation: int  284 282 279 999 282 286 244 245 289 299 ...
## $ parity   : int   0  0  0  0  0  0  0  0  0  0 ...
## $ age      : int   27 33 28 36 23 25 33 23 25 30 ...
## $ height   : int   62 64 64 69 67 62 62 65 62 66 ...
## $ weight   : int  100 135 115 190 125 93 178 140 125 136 ...
## $ smoke    : int   0  0  1  0  1  0  0  0  1 ...

stripchart(bwt ~ smoke, data = babies, method = "jitter", jitter = 0.2, vertical = TRUE,
  pch = 1, xlab = "Smoking Status (0 = non-smoker, 1 = smoker)",
  ylab = "Birth Weight (bwt)", main = "Scatter plots for babies' birth weight")

x1 <- 0.9
x2 <- 2.1
y1 <- mean(babies$bwt[babies$smoke == 0], na.rm = TRUE)
y2 <- mean(babies$bwt[babies$smoke == 1], na.rm = TRUE)
lines(c(0.8, 1.2), rep(y1, 2), lty=2, col=2, lwd=2)
lines(c(1.8, 2.2), rep(y2, 2), lty=2, col=2, lwd=2)

```

Scatter plots for babies' birth weight



b

```
# Filter the data to include only rows where smoke is 0 or 1
t_test <- t.test(bwt ~ smoke, data = babies)
print(t_test)

##
## Welch Two Sample t-test
##
## data:  bwt by smoke
## t = 8.5813, df = 1003.2, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##   6.89385 10.98148
## sample estimates:
## mean in group 0 mean in group 1
##      123.0472      114.1095
```

c

```
set.seed(123)
permute_t_stat <- function(data) {
  perm_smoke <- sample(data$smoke)
  perm_t_test <- t.test(data$bwt ~ perm_smoke)
  return(perm_t_test$statistic)
}
perm_t_stat <- permute_t_stat(babies)
```

```
print(perm_t_stat)
```

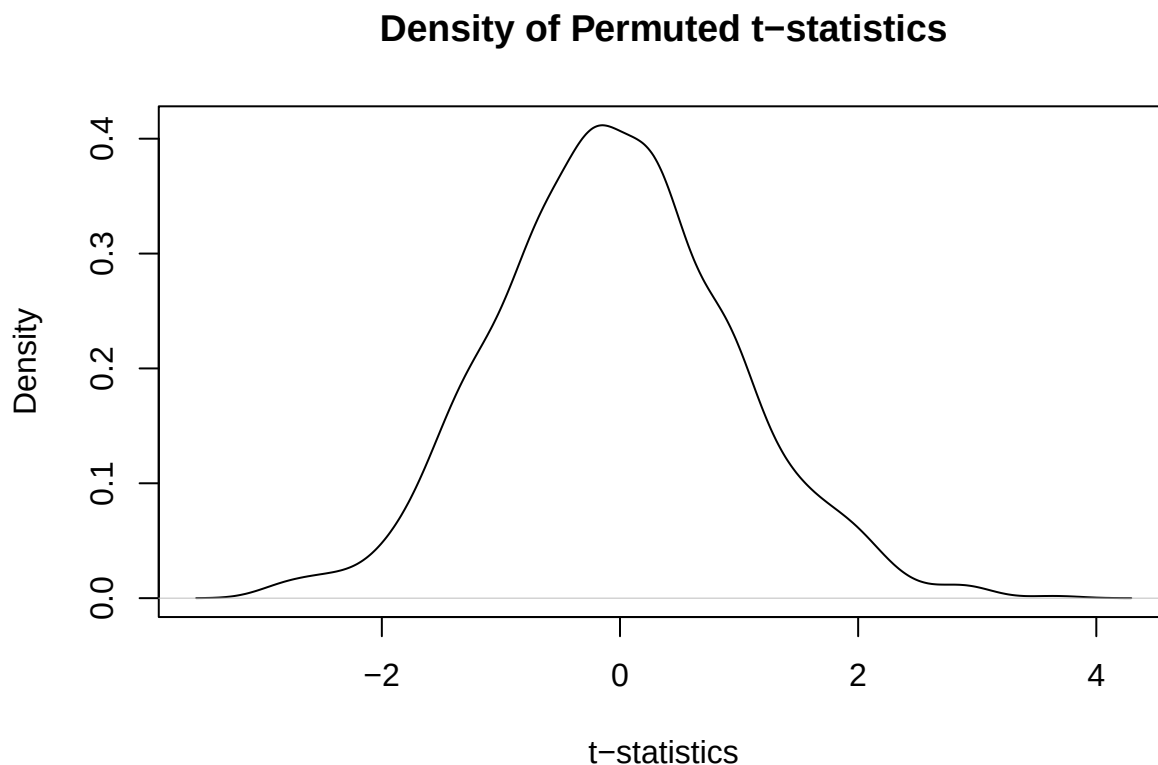
```
##      t  
## 0.38099
```

d

```
n_permutations <- 1000  
perm_t_stats <- replicate(n_permutations, permute_t_stat(babies))  
obs_t_stat <- t_test$statistic  
p_value <- mean(abs(perm_t_stats) >= abs(obs_t_stat))  
print(paste("P-value:", p_value))
```

```
## [1] "P-value: 0"
```

```
plot(density(perm_t_stats), main = "Density of Permuted t-statistics", xlab =  
     "t-statistics")  
abline(v = obs_t_stat, col = "red", lwd = 2)
```

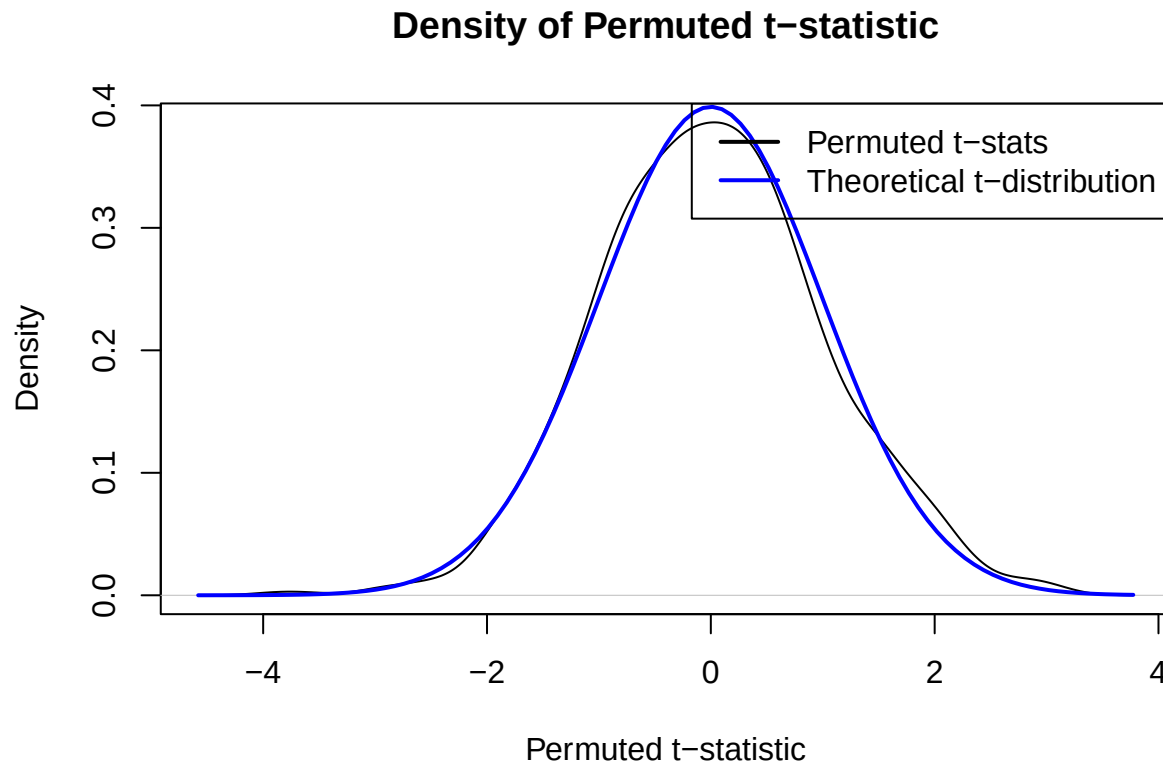


e

```
n_permutations <- 1000  
perm_t_stats <- replicate(n_permutations, permute_t_stat(babies))  
obs_t_stat <- t_test$statistic  
p_value <- mean(abs(perm_t_stats) >= abs(obs_t_stat))  
print(paste("P-value:", p_value))
```

```
## [1] "P-value: 0"
```

```
plot(density(perm_t_stats), main = "Density of Permuted t-statistic", xlab = "Permuted  
↪ t-statistic")  
abline(v = obs_t_stat, col = "red", lwd = 2)  
curve(dt(x, df = t_test$parameter), add = TRUE, col = "blue", lwd = 2)  
legend("topright", legend = c("Permuted t-stats", "Theoretical t-distribution"),  
      col = c("black", "blue"), lwd = 2)
```



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.