Lab 2: Probability

```r
library(pacman)
pacman::p_load(tidyverse, MASS, here, gtExtras, RColorBrewer)
```

Probability Distributions (7.4.1)

```r
set.seed(802)
## setting list of possible responses and prior probabilities for each response
responses <- c("Strongly disagree", "Disagree", "Neither agree nor disagree", "Agree", "Stro
prob_vector <- c(0.07, 0.13, 0.15, 0.23, 0.42)

## simulating one realization of survey data with given probabilities
simulation <- sample(responses, 80, replace = TRUE, prob = prob_vector)

## totaling up counts of each response
counts <- c(0,0,0,0,0)
for (x in simulation) {
  if (x == "Strongly disagree"){
    counts[1] = counts[1] + 1
  } else if (x == "Disagree") {
    counts[2] = counts[2] + 1
  } else if (x == "Neither agree nor disagree") {
    counts[3] = counts[3] + 1
  } else if (x == "Agree") {
    counts[4] = counts[4] + 1
  } else {
    counts[5] = counts[5] + 1
  }
}
simulation_df <- data.frame(responses, counts)
```

```r
## making a table with simulation_df
simulation_table <- simulation_df |>
  gt() |> # use 'gt' to make an awesome table...
  gt_theme_538() |>
  tab_header(title = "Counts of Survey Responses")  |>
  data_color( # Update cell colors, testing different color palettes
    columns = c(counts),
    fn = scales::col_numeric( # <- bc it's numeric
      palette = brewer.pal(5, "Blues"), # A color scheme (gradient)
      domain = c(), # Column scale endpoints
      reverse = FALSE
    )
  ) |>
  cols_label(responses = "Survey Responses", counts = "Counts") # Update labels
simulation_table
```

Counts of Survey Responses

Survey Responses	Counts
Strongly disagree	4
Disagree	9
Neither agree nor disagree	13
Agree	23
Strongly Agree	31

```r
simulation_table |>
  gtsave(
    "Lab2_Section7_Q4.png",
    path = here("Labs", "Lab2")
  )
```

Disease Prevalence (7.6)

```r
set.seed(802)
prevalence_prob_dbinom <- dbinom(4, 24, 0.12)
prevalence_prob_dbinom
```

```
[1] 0.1709024
```

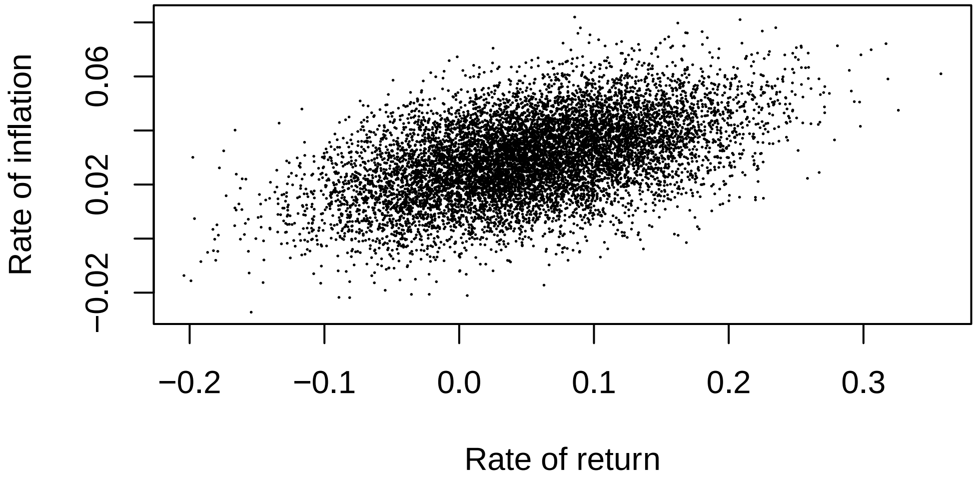
Nitrogen fixation, (7.8)

```
set.seed(802)
nitrogen_25_fix <- pnorm(0.025, 1.9, 1.4)
nitrogen_25_fix_q <- qnorm(0.025, 1.9, 1.4)
nitrogen_25_fix_q
```

```
[1] -0.8439496
```

Continuous Random Variables (8.2)

```
DrawRates=function(n,int,int.sd,inf,inf.sd,rho.rates){
covar=rho.rates*int.sd*inf.sd
Sigma=matrix(c(int.sd^2,covar,covar,inf.sd^2),2,2)
mu=c(int,inf)
x=(mvrnorm(n=n,mu=mu,Sigma))
#x[x[,2]<0]=0 #do not allow for deflation
return(x)
}
mu.int=0.0531
sd.int=0.0746
mu.inf=0.03
sd.inf=0.015
rho=0.5
n=10000
x=DrawRates(n,int=mu.int,int.sd=sd.int,inf=mu.inf,inf.sd=sd.inf,rho.rates=rho)
plot(x[,1],x[,2],pch=19,cex=.05,xlab="Rate of return",ylab="Rate of inflation")
```



Moment Matching

Above Ground biomass (9.1)

```r
set.seed(802)
## probability density of 94 given mean and standard deviation
biomass_94 <- dlnorm(94, meanlog = log(103.4), sdlog = log(23.3))
biomass_94
```

```
[1] 0.001347367
```

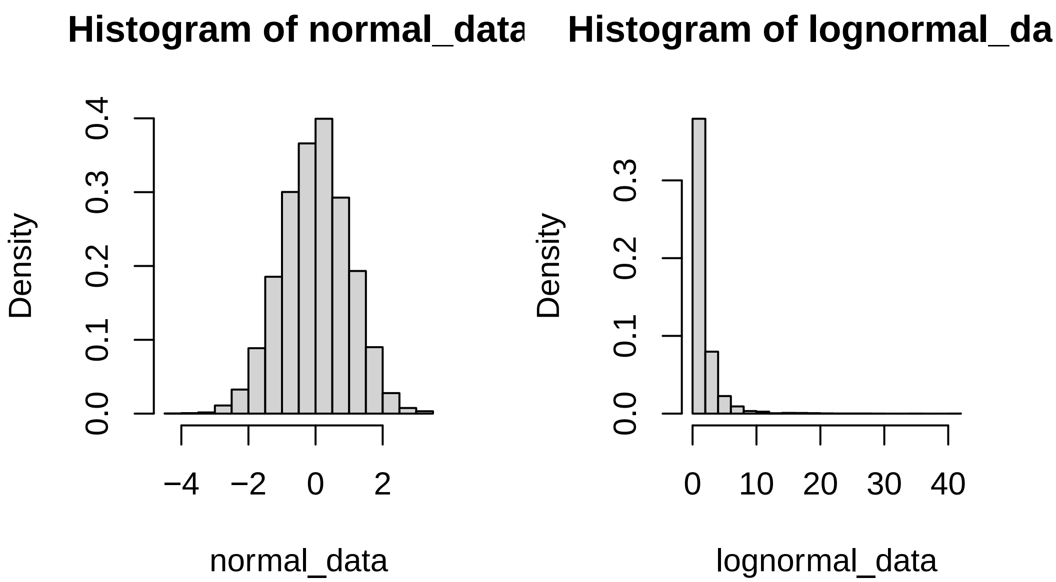
```r
## probability that plot will contain between 90 and 110 gm of biomass
biomass_90 <- plnorm(90, meanlog = log(103.4), sdlog = log(23.3))
biomass_110 <- plnorm(110, meanlog = log(103.4), sdlog = log(23.3))
biomass_90_110 <- biomass_110 - biomass_90
biomass_90_110
```

```
[1] 0.0254209
```

Simulating 10,000 data points in normal and lognormal with mean 0 and SD 1

```r
set.seed(802)
normal_data <- rnorm(10000,0,1)
lognormal_data <- rlnorm(10000,0,1)

## making histograms
par(mfrow = c(1,2))
hist(normal_data, probability = TRUE)
hist(lognormal_data, probability = TRUE)
```




```r
## finding mean and variance of lognormal dist using moment matching
```

$\mu = e\hat{}\{(\alpha + \beta/2)\}$