Introductions

- Name
- Year
- Previous stats courses

Goals

- Learn relevant R skills for the week
- See similar examples to the homework
- Learn something about the world

Linear algebra and matrices in R

Let

$$\mathbf{a} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}, \quad \boldsymbol{\mu} = \begin{bmatrix} 4 \\ 4 \\ 4 \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} 1 & 2 & 0 \\ 3 & -1 & 2 \\ -2 & 3 & -2 \end{bmatrix} \mathbf{f} = \begin{bmatrix} 3 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 3 \end{bmatrix}$$

1. How do you multiply matrices? What about scalar multiplication? Find 4**d**. Why is **ab** not defined? What should we change to be able to multiply them?

Each entry in the output is the dot product of a row in the first matrix and a column in the second; match dimensions as necessary. For scalar multiplication, multiply every element in the matrix by the scalar.

$$4\mathbf{d} = \begin{bmatrix} 4 & 8 & 0 \\ 12 & -4 & 8 \\ -8 & 12 & -8 \end{bmatrix}$$

$$\mathbf{a}'\mathbf{b} = 3$$

2. How do you find the determinant of a matrix? What is the determinant of **d**?

2x2: $(ad - bc)$

$n \times n$: Recursively: Multiply the first row's first element by the determinant of the $(n-1) \times (n-1)$ matrix remaining when you cover the first row and first column. From this, subtract the first row's second element times the determinant of the $(n-1) \times (n-1)$ matrix remaining when you cover the first row and second column. Continue alternating adding and subtracting.

$$\det(\mathbf{d}) = 0$$

3. How do you find the inverse of a matrix? When does the inverse of a matrix exist? What is the inverse of **f**?

Write the $n \times n$ matrix on the left and an $n \times n$ identity matrix to the right and row reduce the first n columns. Rearranging the rows to get the identity on the left gives you your inverse on the right. The inverse exists when the determinant exists.

$$\mathbf{f}^{-1} = \begin{bmatrix} 2/5 & -1/10 & -1/10 \\ -1/10 & 2/5 & -1/10 \\ -1/10 & -1/10 & 2/5 \end{bmatrix}$$

Some useful commands:

- `rep(k, n)` gives a vector `n` long of `k`
- `cbind(c1, c2, ...)` makes a matrix with columns `c1`, `c2`, . . .
- `rbind(r1, r2, ...)` makes a matrix with rows `r1`, `r2`, . . .
- `matrix(k, nrow, ncol)` makes a matrix with all entries `k`
- `diag(x)` gets the diagonal of a matrix
- `%*%` does matrix multiplication
- `t(x)` gives the transpose of `x`
- `det(x)` gives the determinant of `x`
- `solve(x)` gives the matrix inverse of `x`

4. What should the following code print?

```
a = cbind(c(1/3, 1/3, 1/3))
b = cbind(c(2, 3, 4))
mu = cbind(rep(4,3))
d = rbind(c(1, 2, 0), c(3, -1, 2), c(-2, 3, -2))
f = matrix(1, nrow = 3, ncol = 3)
diag(f) <- 3

d * 4 # Scalar multiplication
```

```
##      [,1] [,2] [,3]
## [1,]    4    8    0
## [2,]   12   -4    8
## [3,]   -8   12   -8
```

```
# This gives an error because of dimension mismatches
#a %*% b
t(a) %*% b
```

```
##      [,1]
## [1,]    3
```

```
sum(a * b) # Easier way if you just have vectors
```

```
## [1] 3
```

```
# This gives an error because of dimension mismatches
#a %*% d
d %*% a
```

```
##            [,1]
## [1,]  1.0000000
## [2,]  1.3333333
## [3,] -0.3333333
```

```
# Matrix inverse
det(d)
```

```
## [1] 7.771561e-16
```

```
# This gives an error because the determinant is 0
# solve(d)
solve(f)
```

```
##      [,1] [,2] [,3]
## [1,]  0.4 -0.1 -0.1
## [2,] -0.1  0.4 -0.1
## [3,] -0.1 -0.1  0.4
```

```
f %*% solve(f)
```

```
##              [,1]          [,2] [,3]
## [1,] 1.000000e+00 -2.775558e-17    0
## [2,] 1.387779e-16  1.000000e+00    0
## [3,] 5.551115e-17 -5.551115e-17    1
```

```
solve(f) %*% f
```

```
##              [,1]         [,2]         [,3]
## [1,] 1.000000e+00 4.163336e-17 5.551115e-17
## [2,] 6.938894e-17 1.000000e+00 0.000000e+00
## [3,] 0.000000e+00 0.000000e+00 1.000000e+00
```

5. What does the following code represent if **b** is a data vector, $\boldsymbol{\mu}$ is a mean vector, and **f** is a covariance matrix?

```
t(a) %*% (b-mu) %*% (t(a) %*% f %*% a)^(-1/2)
```

```
##            [,1]
## [1,] -0.7745967
```

This gives the standardized value of the statistic $(b_1 + b_2 + b_3)/3$. The **f** matrix represents the fact that the variance of a sum is the sum of the variances plus two times the sum of the covariances.

Distribution of the sample mean with and without covariance

1. If we have $X_1, X_2, \ldots, X_n \sim \mathcal{N}(0, 1)$, what is the distribution of $\bar{X}$? If $n = 50$, what is its variance?

$$\bar{X} \sim \mathcal{N}(0, 1/n)$$

which has variance 0.02 if $n = 50$.

2. If we have $X_1, X_2, \ldots, X_n \sim \mathcal{N}(0, 1)$, but each are correlated with correlation ρ with their neighbors, what is the distribution of $\bar{X}$? If $n = 50$ and $\rho = 0.5$, what is its variance? What about if $\rho = 0.2$?

The sum of normals is normal, so it's still normal. $E(\bar{X}) = 0$ by linearity of expectation. $\text{Var}(\bar{X}) = \sum_{i=1}^{n} \text{Var}(X_i/n) + \sum_{i,j} \rho\sqrt{\text{Var}(X_i/n)\text{Var}(X_j/n)} = \text{Var}(X_i)(n + \rho(2n-2))/n^2$ after rearranging the formula for correlation. The $\rho(2n-2)$ comes from the fact that we have $2n-2$ covariance terms since all x_j have two neighbors except the first and last.

$$\bar{X} \sim \mathcal{N}(0, (n + \rho(2n-2))/n^2)$$

which has variance 0.0396 for $n = 50, \rho = 0.5$ and 0.02784 for $n = 50, \rho = 0.2$.

3. If you want your sample mean to have a variance equal to the variance of the sample mean of n uncorrelated observations when you have a correlation of ρ, what n' do you need for your correlated samples?

$$1/n = 1/n' + \rho(2n'-2)/n'^2 \implies n = \frac{n'^2}{n' + \rho(2n'-2)} \implies n'^2 - n'(n + 2\rho n) + 2\rho n = 0$$

$$\implies n' = \frac{n + 2\rho n + \sqrt{(n+2\rho n)^2 - 8\rho n}}{2} \approx n + 2\rho n$$

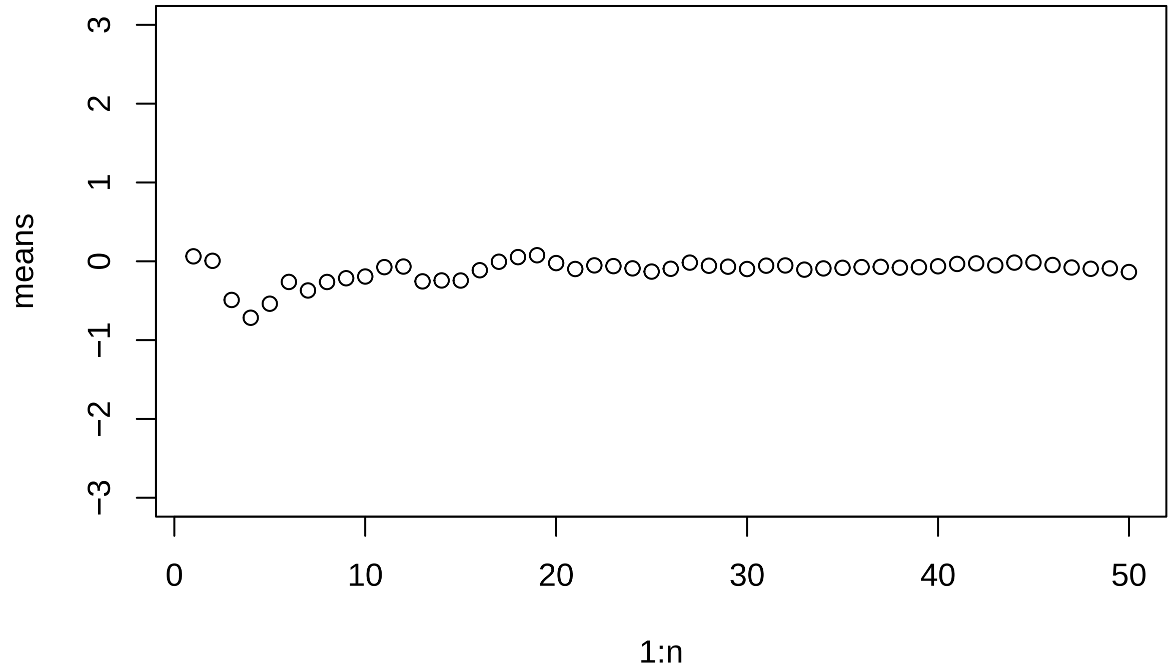
```
set.seed(139)

# Import MASS for mvrnorm
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.1.3
```

```
# Number of samples
n <- 50
```

4. Write a function called `moving_mean` that returns the mean of the first n elements of x.

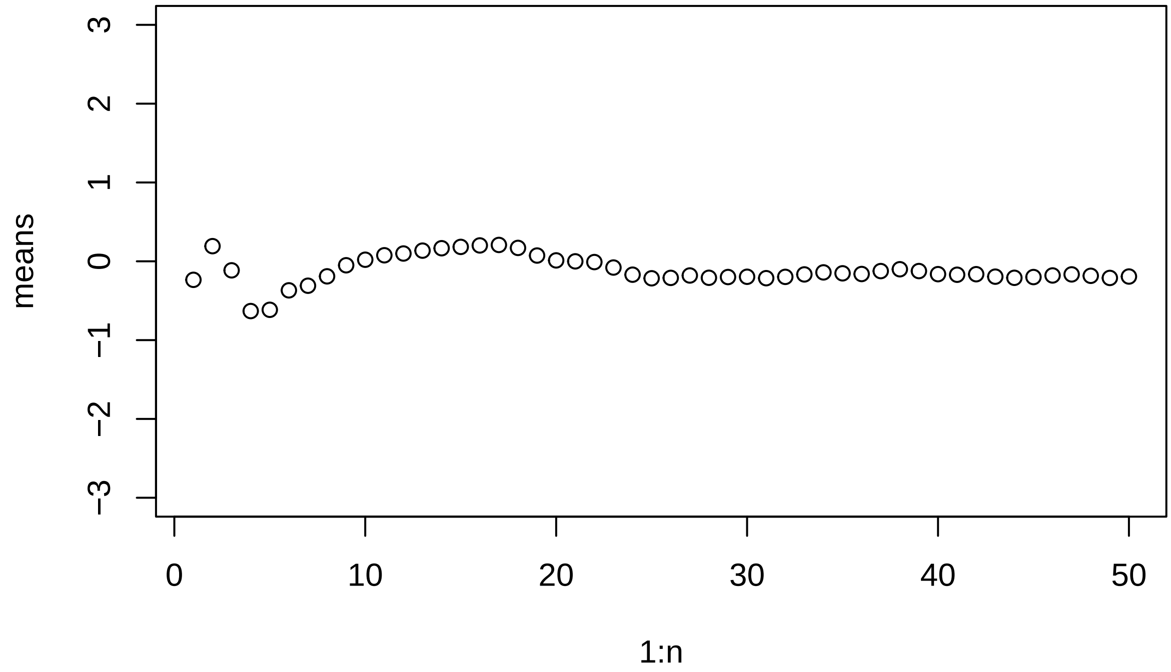
```
# Gets the mean of the vector x through index n
moving_mean <- function(x, n) {
  return (mean(x[1:n]))
}

# Show convergence with no correlation
x <- rnorm(n, 0, 1)
means = vector(length = n)
for (i in 1:n) {
  means[i] <- moving_mean(x, i)
}
plot(1:n, means, ylim = c(-3,3))
```



```r
# Correlation (Why could this not be above 0.5? E.g. what would be wrong if it was 1?)
p = 0.5
```

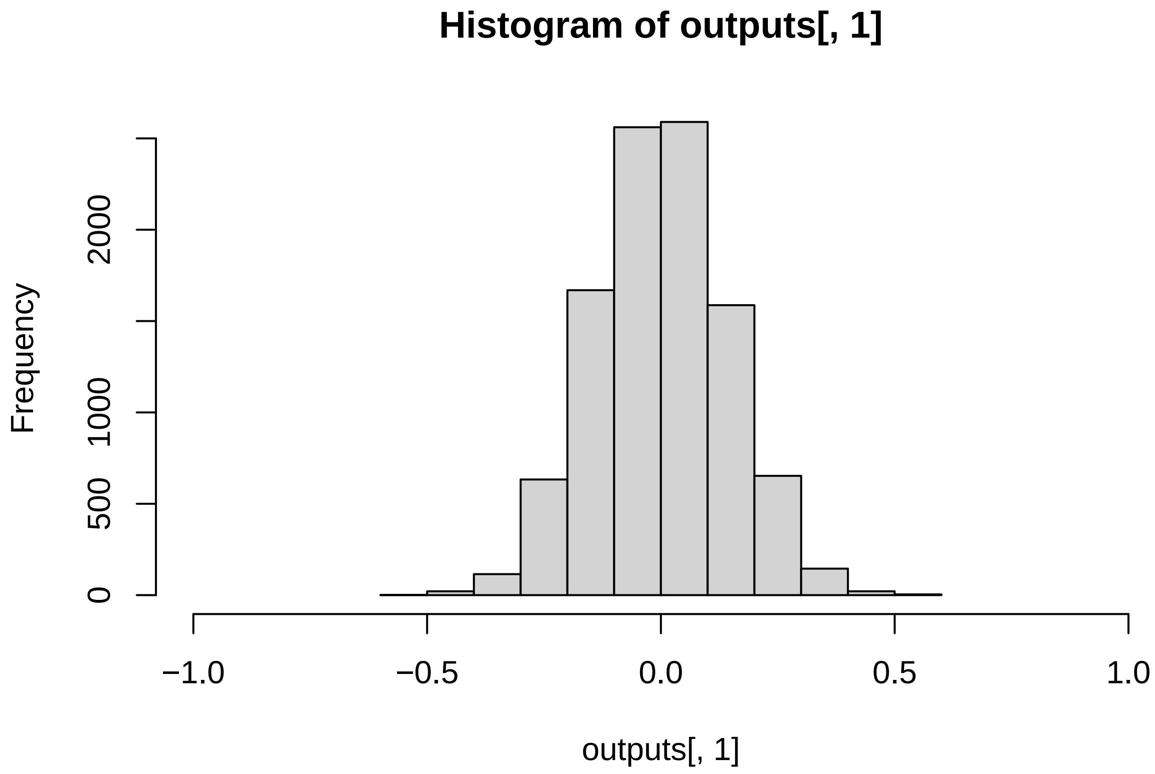
5. Create a matrix filled with 0s except for a diagonal of 1 and 1-off diagonals of ρ.

```r
# Creates covariance matrix with correlation between adjacent samples
Sigma = matrix(0, nrow = n, ncol = n)
diag(Sigma) <- 1
for (i in 2:n) {
  Sigma[i, i-1] <- p
  Sigma[i-1, i] <- p
}

# Show convergence with correlation
x <- mvrnorm(n = 1, rep(0, n), Sigma)
means = vector(length = n)
for (i in 1:n) {
  means[i] <- moving_mean(x, i)
}
plot(1:n, means, ylim = c(-3,3))
```



6. Fill in the outputs table so the first column is uncorrelated means and the second is correlated means.

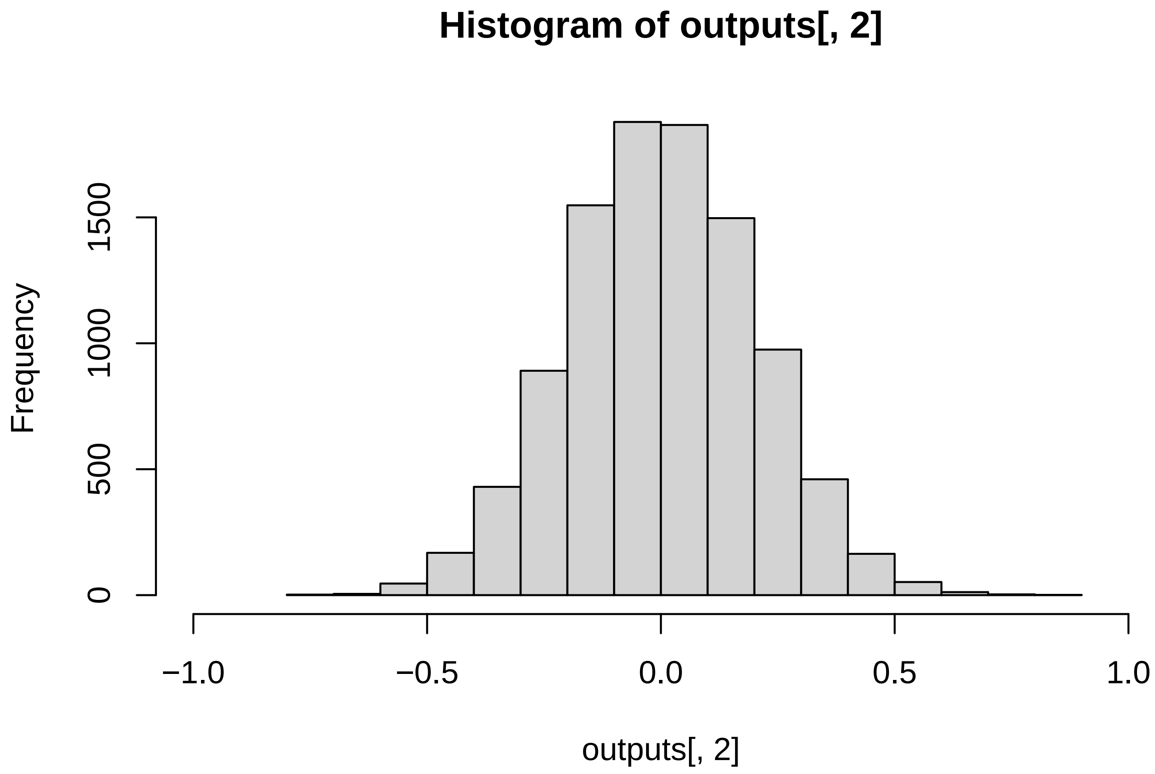
```
# Simulate many times to get the variance of the sample mean
nsim = 10000
outputs <- matrix(nrow = nsim, ncol = 2)
for (i in 1:nsim) {
  x <- rnorm(n, 0, 1)
  outputs[i,1] <- mean(x)

  x <- mvrnorm(n = 1, rep(0, n), Sigma)
  outputs[i,2] <- mean(x)
  #print(paste0("Finished ", i, " out of ", nsim))
}

# No correlation
hist(outputs[,1], xlim = c(-1,1))
```



```
var(outputs[,1])
```

```
## [1] 0.01988117
```

```
# Correlation
hist(outputs[,2], xlim = c(-1,1))
```



```
var(outputs[,2])
```

```
## [1] 0.04037917
```

Data exploration for country demographics

This section will deal with a data set of country-level statistics from this source. We'll go over the following things:

- Summary statistics
- Overlaid histogram
- Box plot
- Scatter plot
- Two-way table

```
# Read in the data
countries <- read.csv("data/countries.csv", check.names = F)
```

1. Calculate the following summary statistics for the `Population` variable: sample mean, sample standard deviation, min, median, max, and the 1st and 3rd quartiles. Also calculate the proportion of countries with less than 10 million people.

```
# Summary statistics
c(summary(countries$Population), "SD" = sd(countries$Population))
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.         SD
##       7026     437624    4786994   28740284   17497773 1313973713  117891327
```

```
# Proportion of countries with less than 10 million people
mean(countries$Population < 10000000)
```

```
## [1] 0.6519824
```

2. Split the countries into two groups: those with less than 10 million people and those with more than 10 million people. Use summary statistics and graphics to explore whether there is evidence of a difference in land area between the two groups. Comment on the results without performing a formal hypothesis test. (Because the data are very right skewed, it will help to take the log of both the population and the area; just make sure to set your 10 million threshold before taking the log!)

```
# Summary statistics
under10 <- countries[countries$Population < 10000000,]
over10 <- countries[countries$Population >= 10000000,]
c(summary(under10$`Area (sq. mi.)`), "SD" = sd(under10$`Area (sq. mi.)`))
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.         SD
##       2.00     652.75   19666.50  133831.20   91325.00 2166086.00  322087.09
```

```
c(summary(over10$`Area (sq. mi.)`), "SD" = sd(over10$`Area (sq. mi.)`))
```

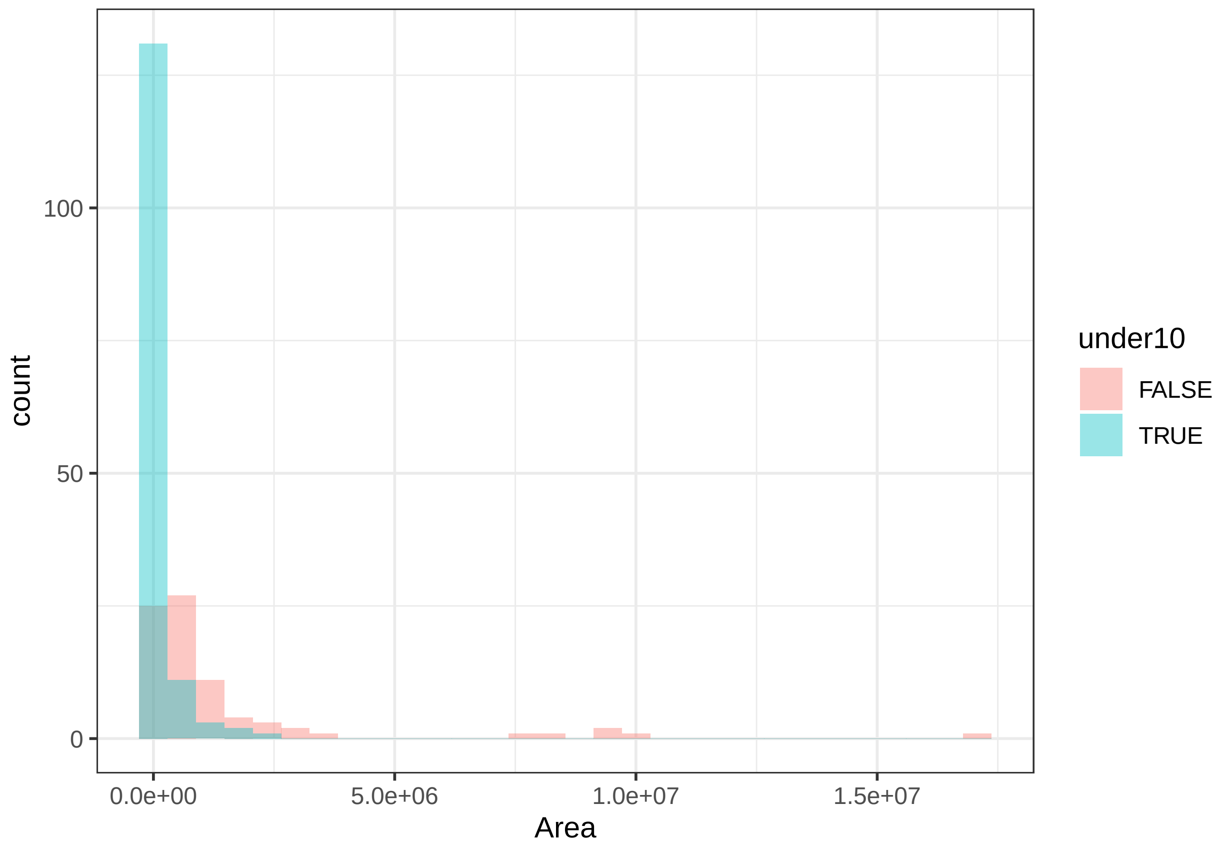
```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.       SD
##    30528   236770   514000  1468234  1229956 17075200  2813409
```

```
countries$under10 <- countries$Population < 10000000

library(ggplot2)
```

```
## Warning: package ’ggplot2’ was built under R version 4.1.3
```

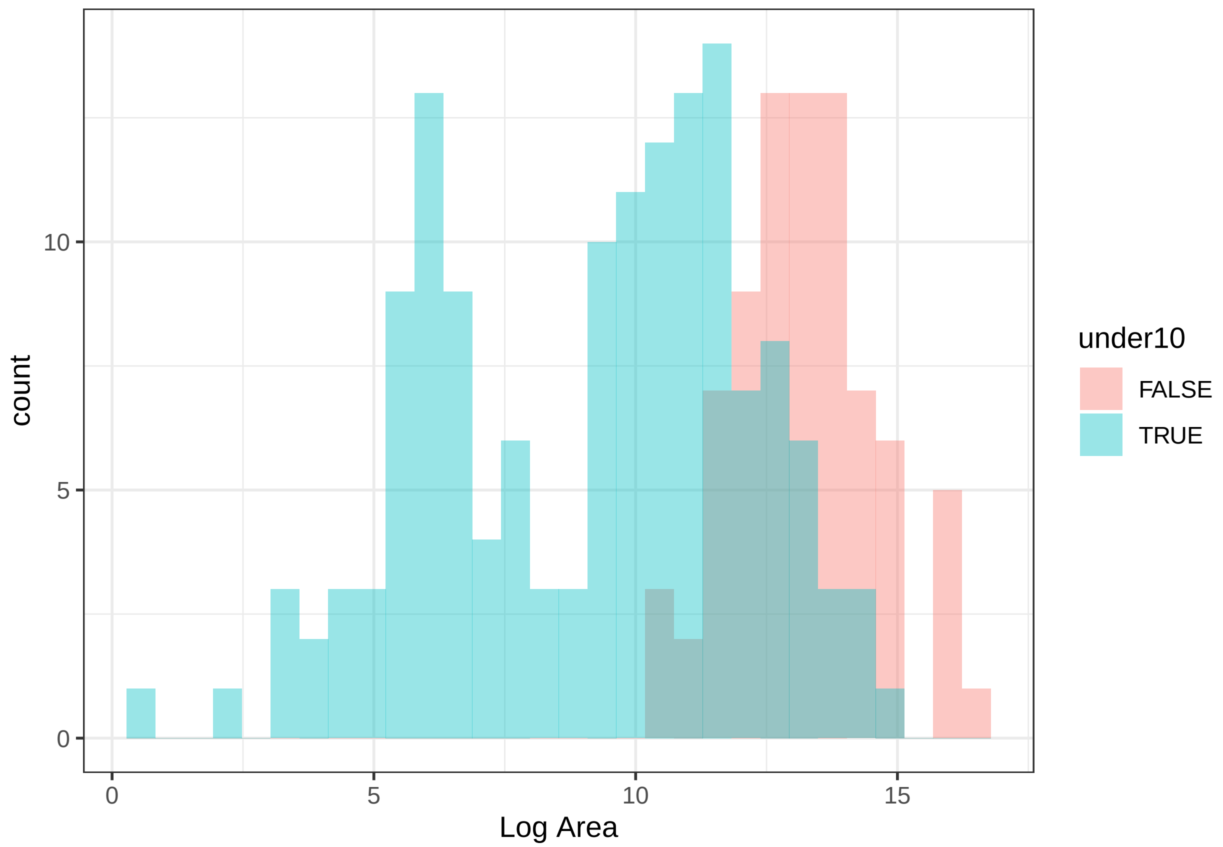
```
ggplot(countries, aes(x=`Area (sq. mi.)`, fill=under10)) +
  geom_histogram(alpha=0.4, position="identity") +
  xlab("Area") +
  theme_bw()
```

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

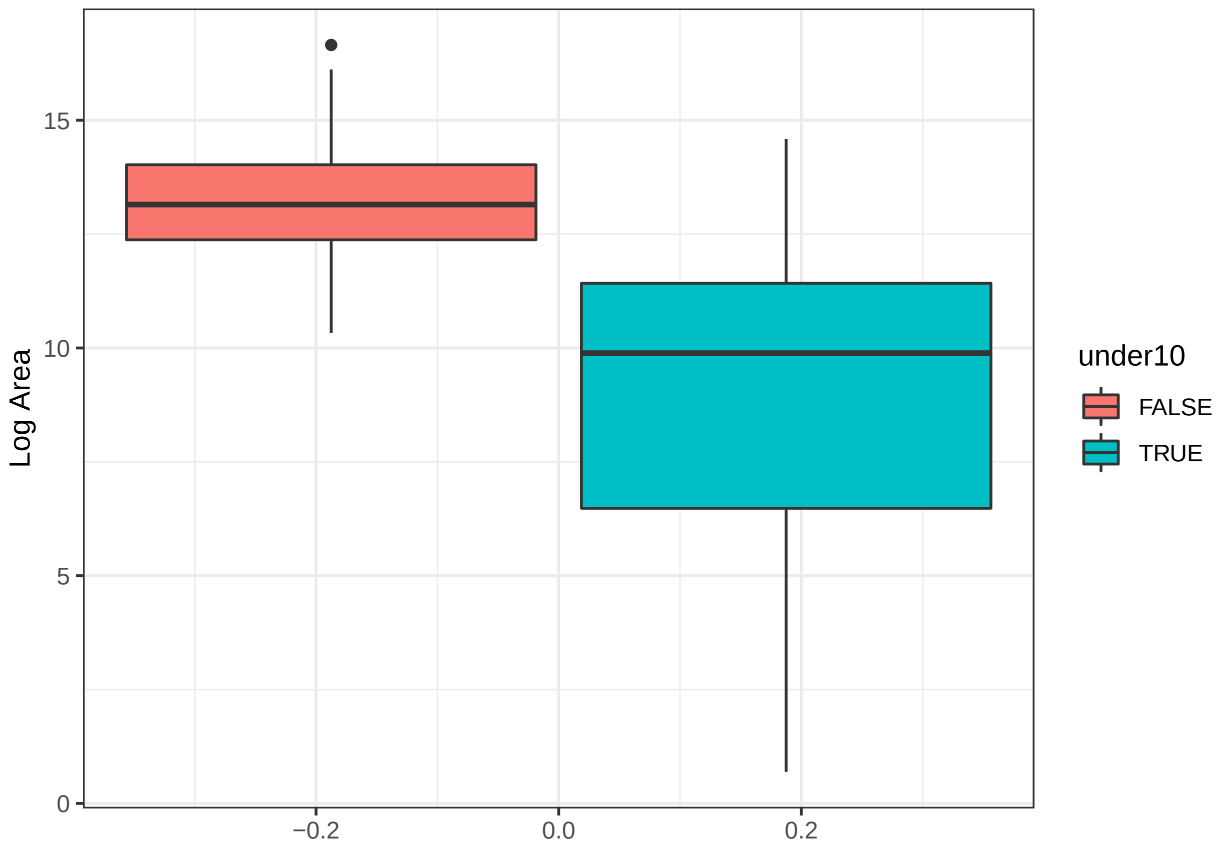


```
countries$logArea <- log(countries$`Area (sq. mi.)`)
ggplot(countries, aes(x=logArea, fill=under10)) +
  geom_histogram(alpha=0.4, position="identity") +
  xlab("Log Area") +
  theme_bw()
```

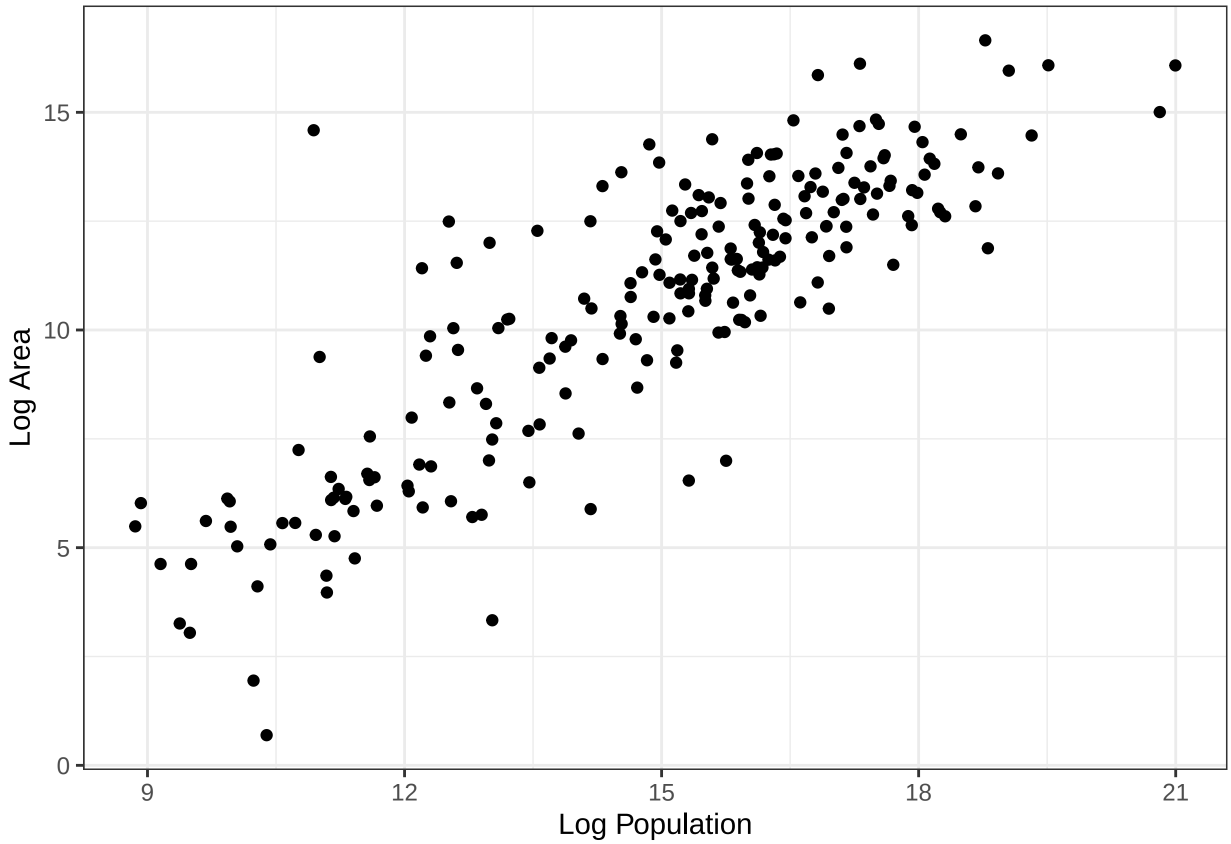
```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



```
ggplot(countries, aes(x=logArea, fill=under10)) +
  geom_boxplot() +
  xlab("Log Area") +
  theme_bw() +
  coord_flip()
```



```
ggplot(countries, aes(x=log(Population), y=logArea)) +
  geom_point() +
  xlab("Log Population") +
  ylab("Log Area") +
  theme_bw()
```



As seen in the histogram and box plots, there is a clear difference between the land areas of countries with over and under 10 million people. The scatter plot also shows a strong correlation between the number of citizens and the area of the country.

3. Find the number of countries with under and over 10 million people by region. Does there seem to be a difference between regions?

```
table(countries$Region, countries$under10)
```

```
##
##                                       FALSE TRUE
##   ASIA (EX. NEAR EAST)                   19    9
##   BALTICS                                 0    3
##   C.W. OF IND. STATES                     5    7
##   EASTERN EUROPE                          3    9
##   LATIN AMER. & CARIB                    10   35
##   NEAR EAST                               5   11
##   NORTHERN AFRICA                         4    2
##   NORTHERN AMERICA                        2    3
##   OCEANIA                                 1   20
##   SUB-SAHARAN AFRICA                     21   30
##   WESTERN EUROPE                          9   19
```

In many regions, there are too few countries to confidently decide one way or the other, but Asia seems to have an abnormally high proportion of countries with over 10 million people compared to other regions like Latin America, sub-Saharan Africa, and West Europe.