Project 3

Justin Williams

2024-10-12

Data, packages, etc.

```
library(tidyverse)
library(dplyr)
```

```
#define the scores as a vector
scores <- c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

#dataframe
scores_df <- data.frame(Scores = scores)
print(scores_df)
```

```
##    Scores
## 1       0
## 2       1
## 3       2
## 4       3
## 5       4
## 6       5
## 7       6
## 8       7
## 9       8
## 10      9
```

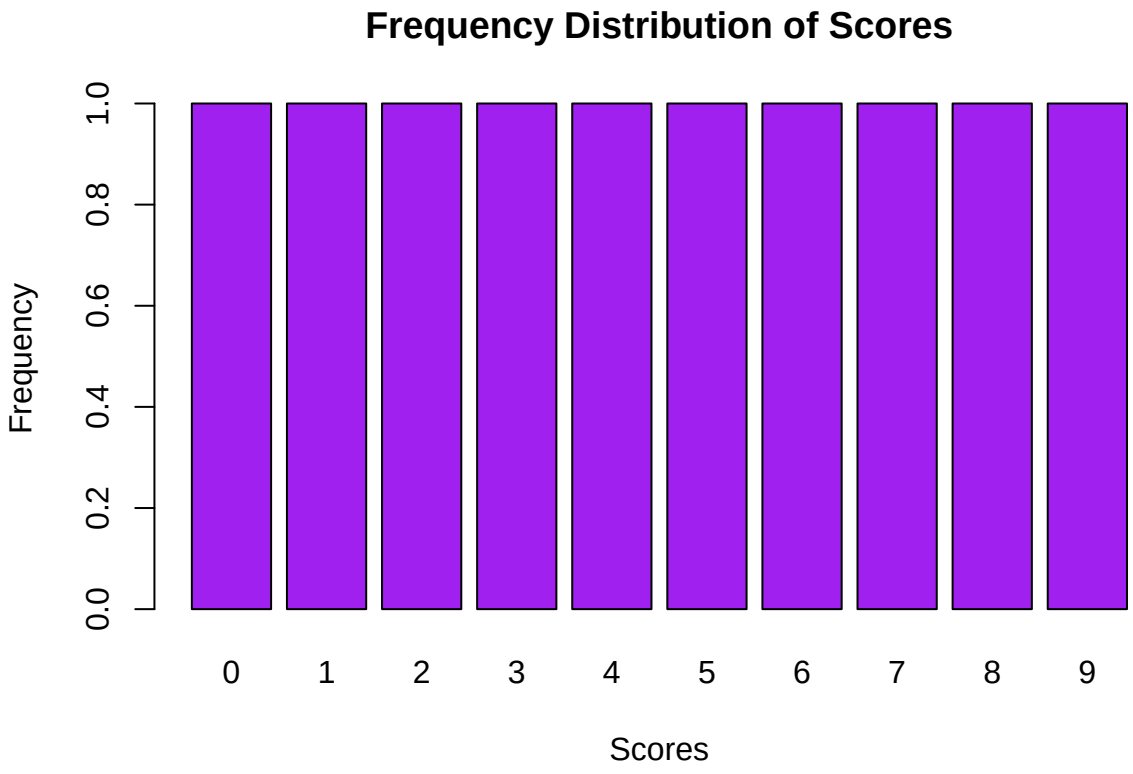
1. Produce a frequency distribution. Describe the shape.

```
freq_dist <- table(scores_df$Scores)
print(freq_dist)
```

```
##
## 0 1 2 3 4 5 6 7 8 9
## 1 1 1 1 1 1 1 1 1 1
```

Barplot

```r
barplot(freq_dist, main="Frequency Distribution of Scores", xlab="Scores", ylab="Frequency", col="purple
```



Shape description

The shape of the distribution is uniform because each score from 0 to 9 only appears once.

2. Identify all possible samples of size 2 that could be selected WITH REPLACEMENT.

```r
#generate all possible samples of size 2 with replacement
samples <- expand.grid(scores, scores)
```

Possible samples

```r
#print samples
print(samples)
```

```
##     Var1 Var2
```

```
## 1      0    0
## 2      1    0
## 3      2    0
## 4      3    0
## 5      4    0
## 6      5    0
## 7      6    0
## 8      7    0
## 9      8    0
## 10     9    0
## 11     0    1
## 12     1    1
## 13     2    1
## 14     3    1
## 15     4    1
## 16     5    1
## 17     6    1
## 18     7    1
## 19     8    1
## 20     9    1
## 21     0    2
## 22     1    2
## 23     2    2
## 24     3    2
## 25     4    2
## 26     5    2
## 27     6    2
## 28     7    2
## 29     8    2
## 30     9    2
## 31     0    3
## 32     1    3
## 33     2    3
## 34     3    3
## 35     4    3
## 36     5    3
## 37     6    3
## 38     7    3
## 39     8    3
## 40     9    3
## 41     0    4
## 42     1    4
## 43     2    4
## 44     3    4
## 45     4    4
## 46     5    4
## 47     6    4
## 48     7    4
## 49     8    4
## 50     9    4
## 51     0    5
## 52     1    5
## 53     2    5
## 54     3    5
```

```
## 55     4    5
## 56     5    5
## 57     6    5
## 58     7    5
## 59     8    5
## 60     9    5
## 61     0    6
## 62     1    6
## 63     2    6
## 64     3    6
## 65     4    6
## 66     5    6
## 67     6    6
## 68     7    6
## 69     8    6
## 70     9    6
## 71     0    7
## 72     1    7
## 73     2    7
## 74     3    7
## 75     4    7
## 76     5    7
## 77     6    7
## 78     7    7
## 79     8    7
## 80     9    7
## 81     0    8
## 82     1    8
## 83     2    8
## 84     3    8
## 85     4    8
## 86     5    8
## 87     6    8
## 88     7    8
## 89     8    8
## 90     9    8
## 91     0    9
## 92     1    9
## 93     2    9
## 94     3    9
## 95     4    9
## 96     5    9
## 97     6    9
## 98     7    9
## 99     8    9
## 100    9    9
```

3. Enter the list of the means of the samples you identified for the previous question into a new R data frame. Create a frequency distribution. Describe the shape of this distribution

Means of the Samples Dataframe

```
sample_means <- rowMeans(samples)
```

```
## Sample Dataframe
sample_means_df <- data.frame(SampleMeans = sample_means)
```

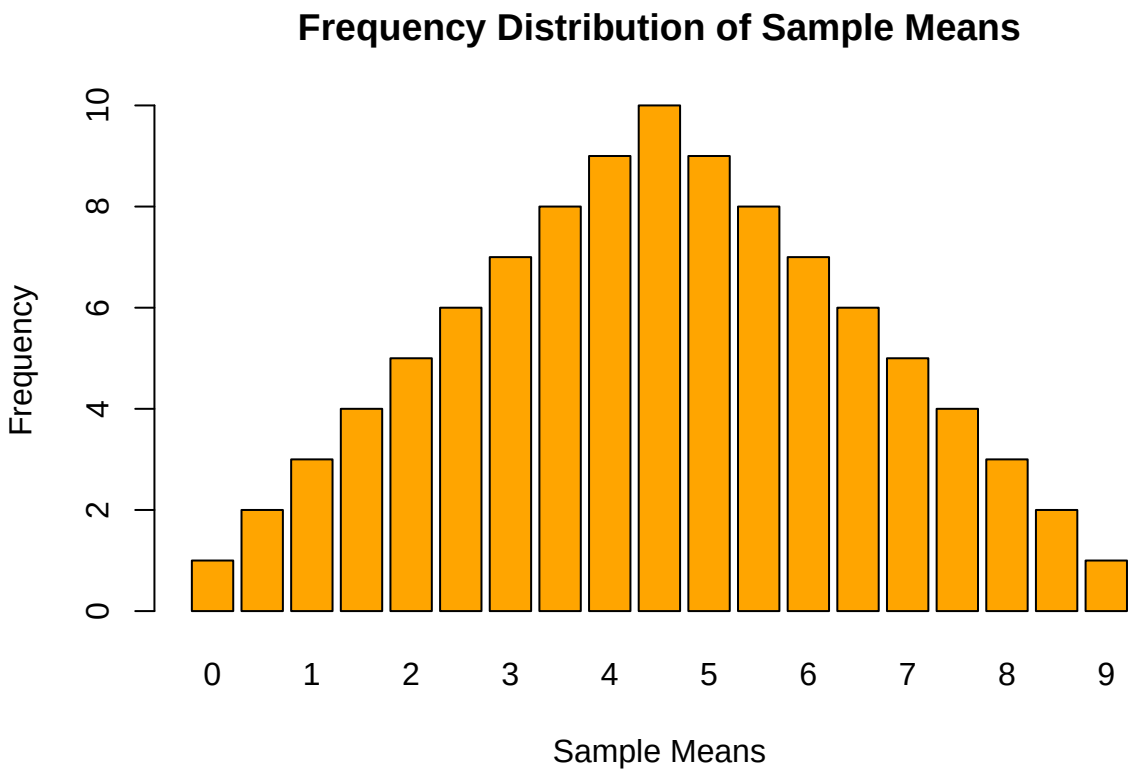
Frequency Distribution

```
freq_dist_means <- table(sample_means_df$SampleMeans)
print(freq_dist_means)
```

```
##
##   0 0.5   1 1.5   2 2.5   3 3.5   4 4.5   5 5.5   6 6.5   7 7.5   8 8.5   9
##   1   2   3   4   5   6   7   8   9  10   9   8   7   6   5   4   3   2   1
```

Barplot

```
barplot(freq_dist_means, main="Frequency Distribution of Sample Means", xlab="Sample Means", ylab="Freq
```



Shape description

The shape of the distribution of sample means is normal, unimodal, and leptokurtic.

4. Calculate the mean and standard deviation of the sample means. Compare these values to the population mean and standard deviation.

Population Scores

```
population_scores <- c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
```

Population Mean and Standard Deviation

```
population_mean <- mean(population_scores)
population_sd <- sd(population_scores)
cat("Population Mean:", population_mean, "\n")
```

```
## Population Mean: 4.5
```

```r
cat("Population Standard Deviation:", population_sd, "\n")
```

```
## Population Standard Deviation: 3.02765
```

Sample Distribution Mean and Standard Deviation

```r
sampling_mean <- mean(sample_means_df$SampleMeans)
sampling_sd <- sd(sample_means_df$SampleMeans)
cat("Sampling Distribution Mean:", sampling_mean, "\n")
```

```
## Sampling Distribution Mean: 4.5
```

```r
cat("Sampling Distribution Standard Deviation:", sampling_sd, "\n")
```

```
## Sampling Distribution Standard Deviation: 2.041241
```

Explanation of Results

The mean of the sample means from all possible samples of size 2 should closely match the population mean due to the Central Limit Theorem which states that the mean of the sampling distribution will approach a normal distribution. The standard deviation of these sample means, also called the standard error, is smaller than the population standard deviation. It's calculated by dividing the population standard deviation by the square root of the sample size. This indicates that while the sample mean is close to the population mean, there's less variability among sample means than in the population itself.

5. If we randomly selected a sample, what is the probability of getting a mean within 2 points of the population value?

Probability of Sample Mean within 2 Points of Population Mean

```r
# population scores
population_scores <- c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

#population mean and standard deviation
population_mean <- mean(population_scores)
population_sd <- sd(population_scores)

#sample size
sample_size <- 2

#standard error
standard_error <- population_sd / sqrt(sample_size)

#zscores for the sample mean being within 2 points of the population mean
z_lower <- (population_mean - 2 - population_mean) / standard_error
```

```r
z_upper <- (population_mean + 2 - population_mean) / standard_error

#probability for zscores
probability <- pnorm(z_upper) - pnorm(z_lower)
cat("Probability of getting a sample mean within 2 points of the population mean:", probability, "\n")
```

```
## Probability of getting a sample mean within 2 points of the population mean: 0.6497986
```