Week 10: Continuous Probability Distributions

This week, we've talked about continuous probability distributions. In the Live R, we will cover the functions associated with continuous random variables and probability distributions.

First, let's load our required packages and then create some data for this Live R.

```
library(tidyverse)
library(kableExtra)
```

1 Generating the Data

Imagine that you are a statistics instructor and you're developing an assessment. You give the exam to 150 students and record the time it takes students to complete the exam, as well as their score.

We can use the `rnorm()` function to generate a set of random, continuous values that approximate a normal distribution. We'll need to pass as arguments the total `sample size`, the `mean`, and the `sd` of the data we'd like to generate. If no parameters are specified, the standard normal distribution parameters are used ($M = 0$, $SD = 1$).

```
set.seed(820)
dat <- tibble(Time = round(rnorm(150, mean = 45, sd = 10), 2),
              Score = round(rnorm(150, mean = 55, sd = 10), 2))

head(dat)
```

```
## # A tibble: 6 x 2
##    Time Score
##   <dbl> <dbl>
## 1  53.4  82.4
## 2  45.9  63.2
## 3  35.2  52.9
## 4  35.1  56.3
## 5  34.8  56.9
## 6  24.7  66.0
```

2 Describing the Data

Although we specified a mean and sd, note that the `rnorm` function will just create data that has similar parameters, but may not perfectly match. It's still a good idea to check the mean and standard deviation of your data:

```
mean(dat$Time)
```

```
## [1] 44.27613
```

```r
sd(dat$Time)
```

```
## [1] 9.107466
```

```r
mean(dat$Score)
```

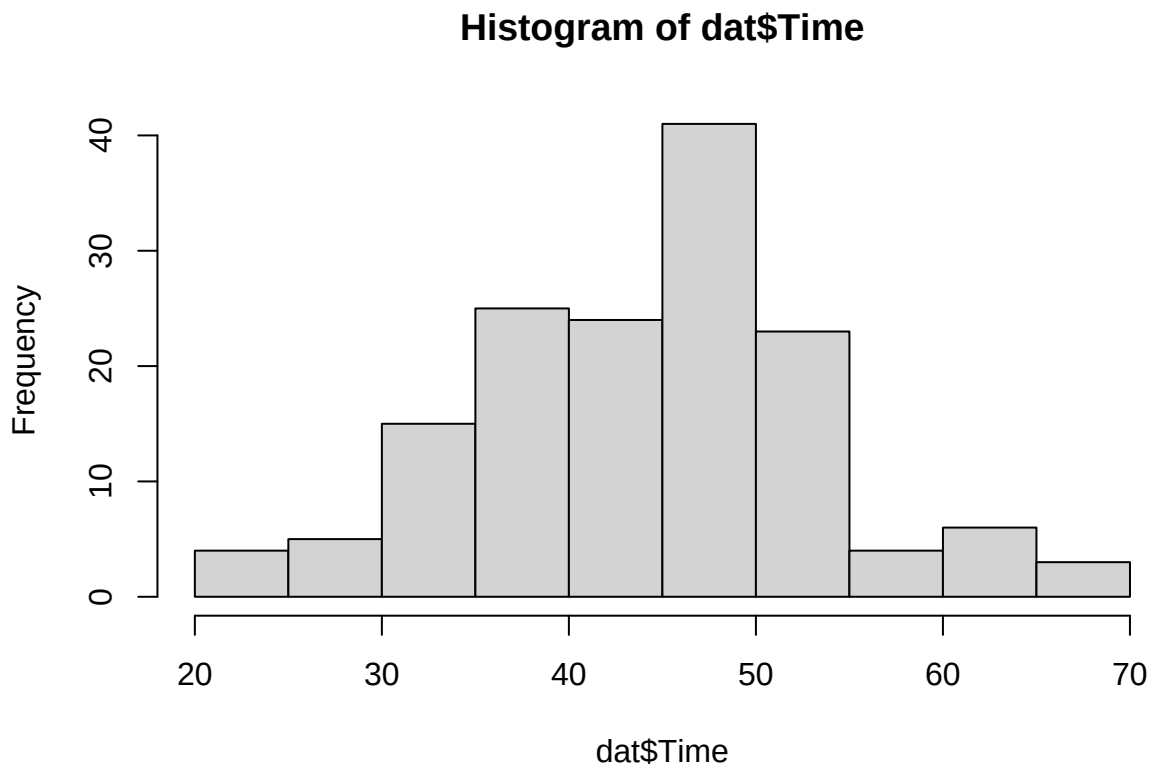
```
## [1] 54.14273
```

```r
sd(dat$Score)
```

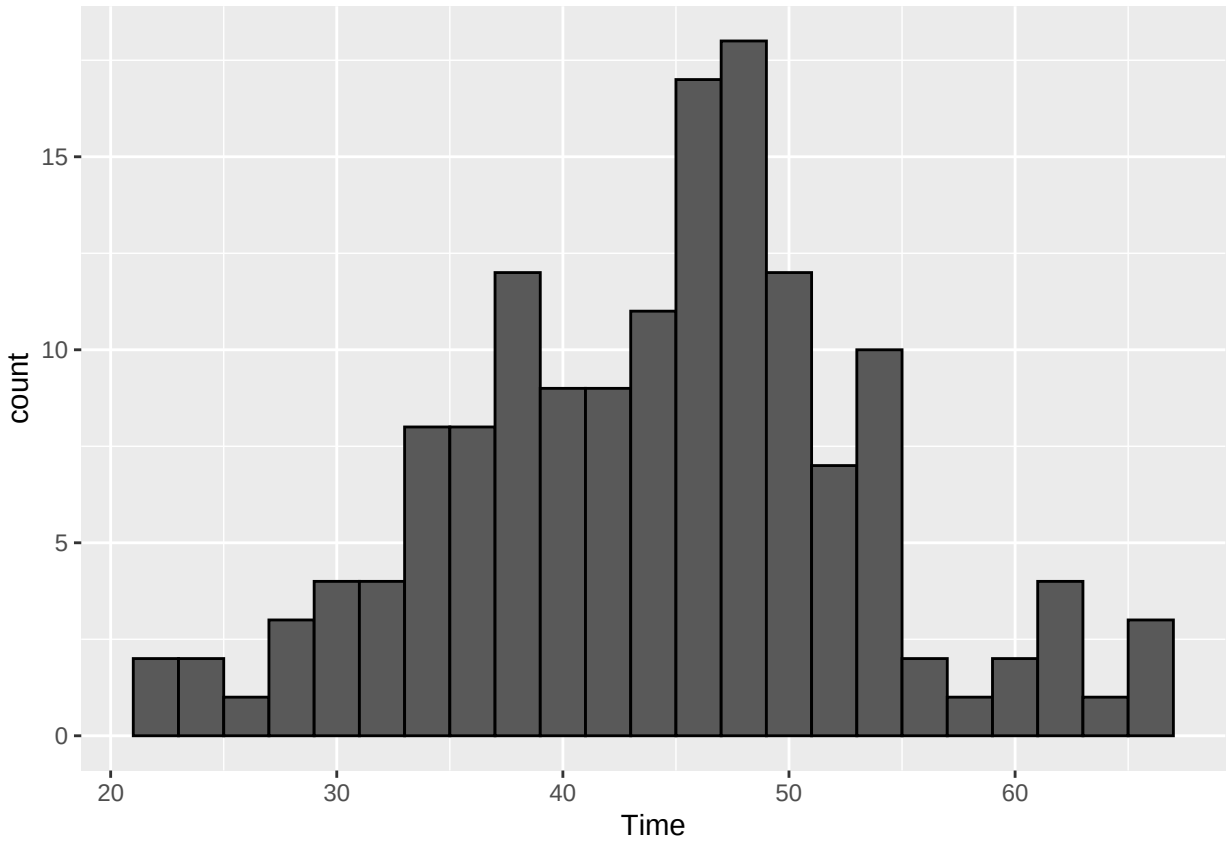
```
## [1] 10.55641
```

It's also a good idea to visualise the data. In this case, we can use a histogram to look at the frequency distribution of continuous variables. Let's first examine the `Time` variable.

```r
hist(dat$Time)
```



```r
ggplot(dat, aes(Time)) + geom_histogram(binwidth = 2, colour = 'black')
```



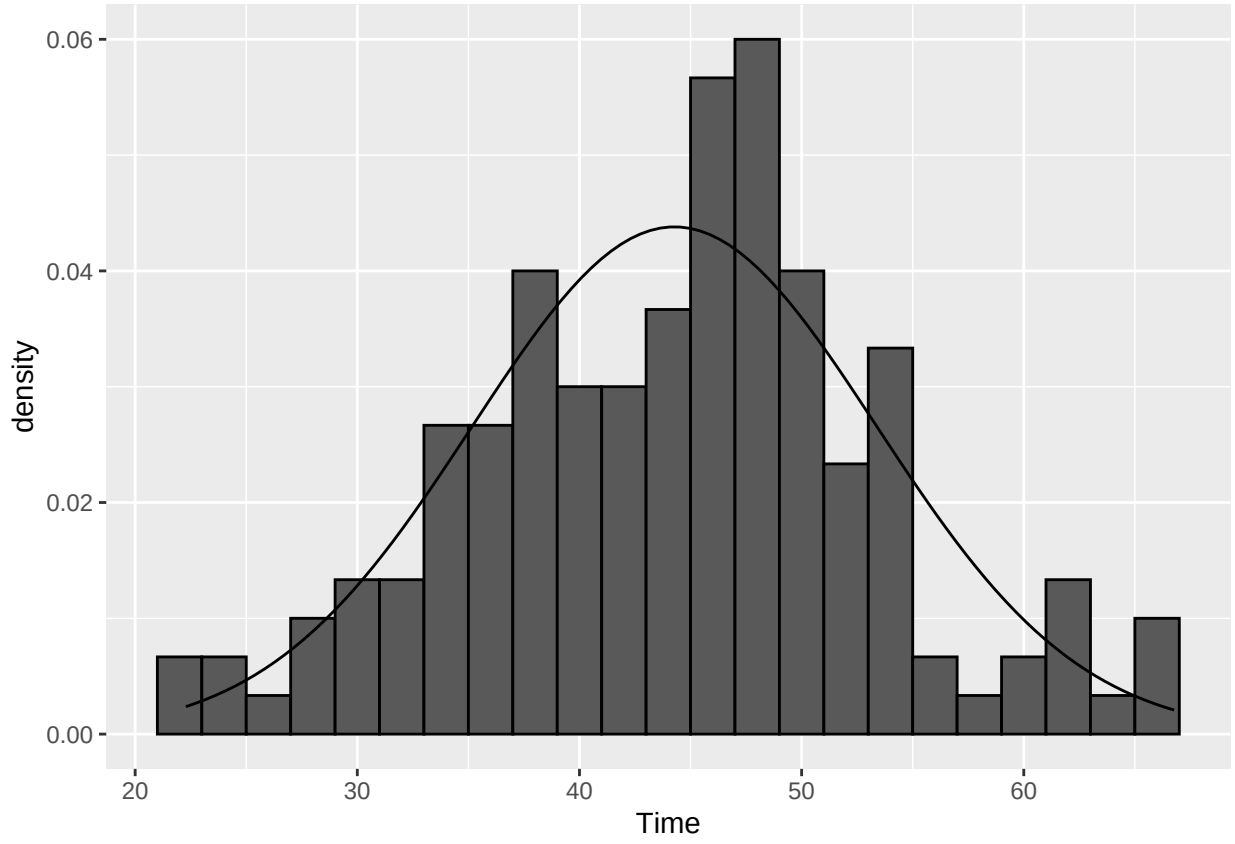
We can see that the data seem to be shaped like a normal distribution.

We can also overlay a normal curve to check how well the data follow the shape of the curve. We can do this using the `dnorm` function, which produces the probability density of a normal distribution with a given mean and standard deviation.

Similar to the `rnorm` function, the default `mean` and `sd` values are 0 and 1, respectively.

Note that if we want to overlay a normal curve, we have to specify that the y-axis of the histogram should show density rather than raw count data.

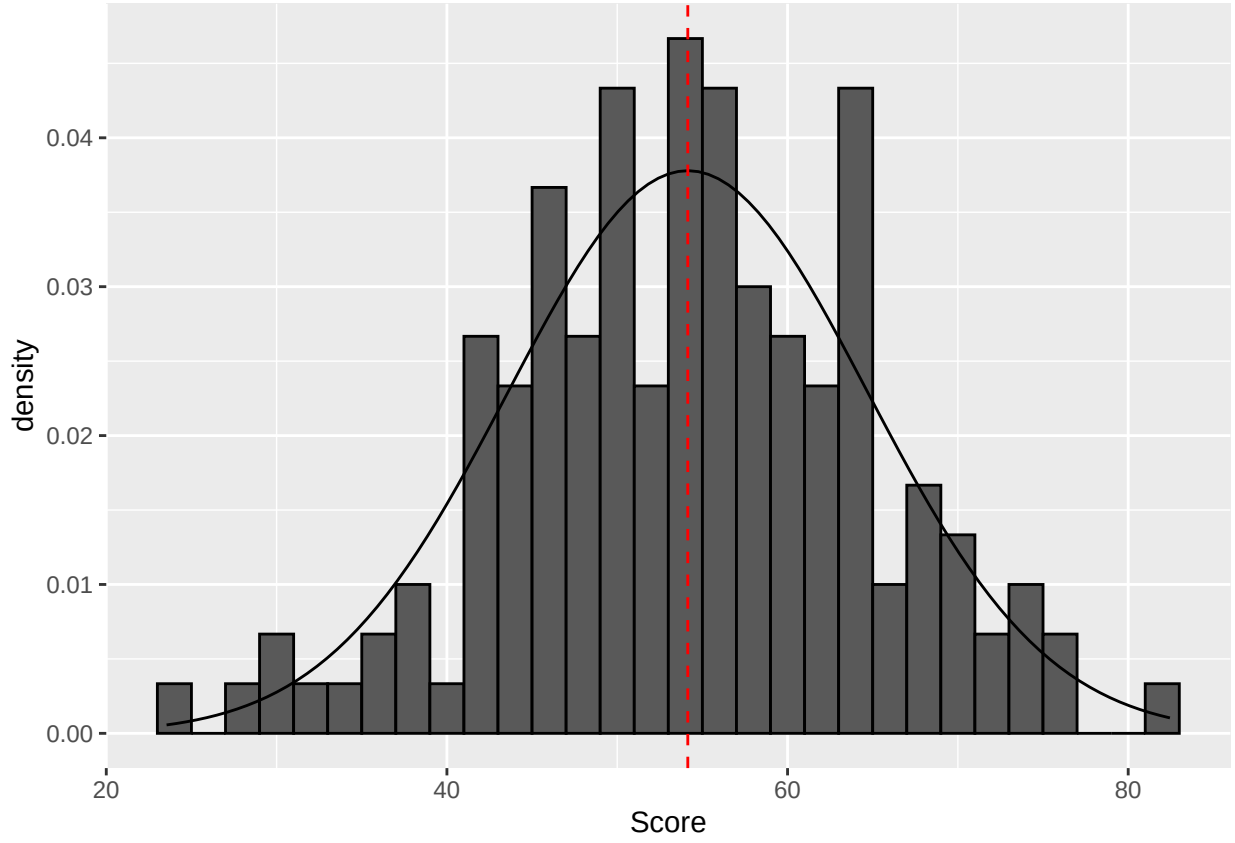
```
ggplot(dat, aes(Time)) +
  geom_histogram(aes(y=after_stat(density)), binwidth = 2, colour = 'black') +
  stat_function(fun = dnorm, args = list(mean = mean(dat$Time), sd = sd(dat$Time)))
```



Now let's check the `Score` variable, against the normal curve. This time, let's also add a red dashed line to show the mean.

```
p <- ggplot(dat, aes(Score)) +
  geom_histogram(aes(y=after_stat(density)), binwidth = 2, colour = 'black') +
  stat_function(fun = dnorm, args = list(mean = mean(dat$Score), sd = sd(dat$Score))) +
  geom_vline(xintercept = mean(dat$Score), colour = 'red', linetype = 'dashed')

p
```



3 Assessing the Assessment

As the instructor, you want the test to be challenging, but not excessively difficult. You want to compute the likelihood that a student will fail the current version of this test. To do this, you can calculate the probability of a student failing this assessment. To do this, you can use the `pnorm` function:

```
pnorm(40, mean=mean(dat$Score), sd=sd(dat$Score))
```

```
## [1] 0.09016667
```

You also want to know the probability of students scoring within each grade band (A, B, C, D). To calculate the likelihood of a student getting a B on the exam, you can use the `pnorm` function as well. In this case, you will be the probability of scores falling within a specified interval:

$$P(score = 60) \leq P(X) \leq P(score = 69.99)$$

```
pnorm(69.99, mean=mean(dat$Score), sd=sd(dat$Score))
```

```
## [1] 0.9333479
```

```
pnorm(60, mean=mean(dat$Score), sd=sd(dat$Score))
```

```
## [1] 0.7105027
```

Because the output reflects the probability of all scores below 69.99 (0.93) and all scores below 60 (0.71), you'll have to subtract the two to get the probability of scoring between 60-69.99:

```
pnorm(69.99, mean=mean(dat$Score), sd=sd(dat$Score))-pnorm(60, mean=mean(dat$Score), sd=sd(dat$Score))
```

```
## [1] 0.2228451
```

The probability of getting a B on the exam is 22%.

You can repeat this for each grade band:

```
pnorm(59.99, mean=mean(dat$Score), sd=sd(dat$Score))-pnorm(50, mean=mean(dat$Score), sd=sd(dat$Score))
```

```
## [1] 0.3628112
```

```
pnorm(49.99, mean=mean(dat$Score), sd=sd(dat$Score))-pnorm(40, mean=mean(dat$Score), sd=sd(dat$Score))
```

```
## [1] 0.2568509
```

You can also look at the probability of a student getting an A. There are two options by which you can do this. Note the `lower.tail` argument in the second option:

```
1- pnorm(70, mean=mean(dat$Score), sd=sd(dat$Score))
```

```
## [1] 0.06652974
```

```
pnorm(70, mean=mean(dat$Score), sd=sd(dat$Score), lower.tail = F)
```

```
## [1] 0.06652974
```

Now, imagine that you want to get a sense of whether the majority of students will have enough time to complete the test. For future assessments, you want to set the time limit as the point by which 95% of the students will have completed the exam. To compute this, you can use the `qnorm()` function, which takes a quantile (your probability metric) and then produces the corresponding value of x given a normal distribution with specific `mean` and `sd` parameters:

```
qnorm(.95, mean=mean(dat$Time), sd=sd(dat$Time))
```

```
## [1] 59.25658
```

Here, I'll create a dataset that summarises our mark band results that I can use to create a table for our write-up:

```
mScore <- mean(dat$Score)
sdScore <- sd(dat$Score)

markBand <- tibble(Mark = c('A', 'B', 'C', 'D', 'F'),
                   Probability = round(c(pnorm(70, mean=mScore, sd=sdScore, lower.tail = F),
                            pnorm(69, mean=mScore, sd=sdScore)-pnorm(60, mean=mScore, sd=sdScore),
                            pnorm(59, mean=mScore, sd=sdScore)-pnorm(50, mean=mScore, sd=sdScore),
                            pnorm(49, mean=mScore, sd=sdScore)-pnorm(40, mean=mScore, sd=sdScore),
                            pnorm(40, mean=mScore, sd=sdScore)), 2))

markBand
```

```
## # A tibble: 5 x 2
##   Mark  Probability
##   <chr>       <dbl>
## 1 A            0.07
## 2 B            0.21
## 3 C            0.33
## 4 D            0.22
## 5 F            0.09
```

4 Write-Up Example

We developed a statistics assessment and administered it to 150 students. Students scored between 23.54 and 82.45 points on the exam ($M = 54.14$, $SD = 10.56$; see Figure 1).

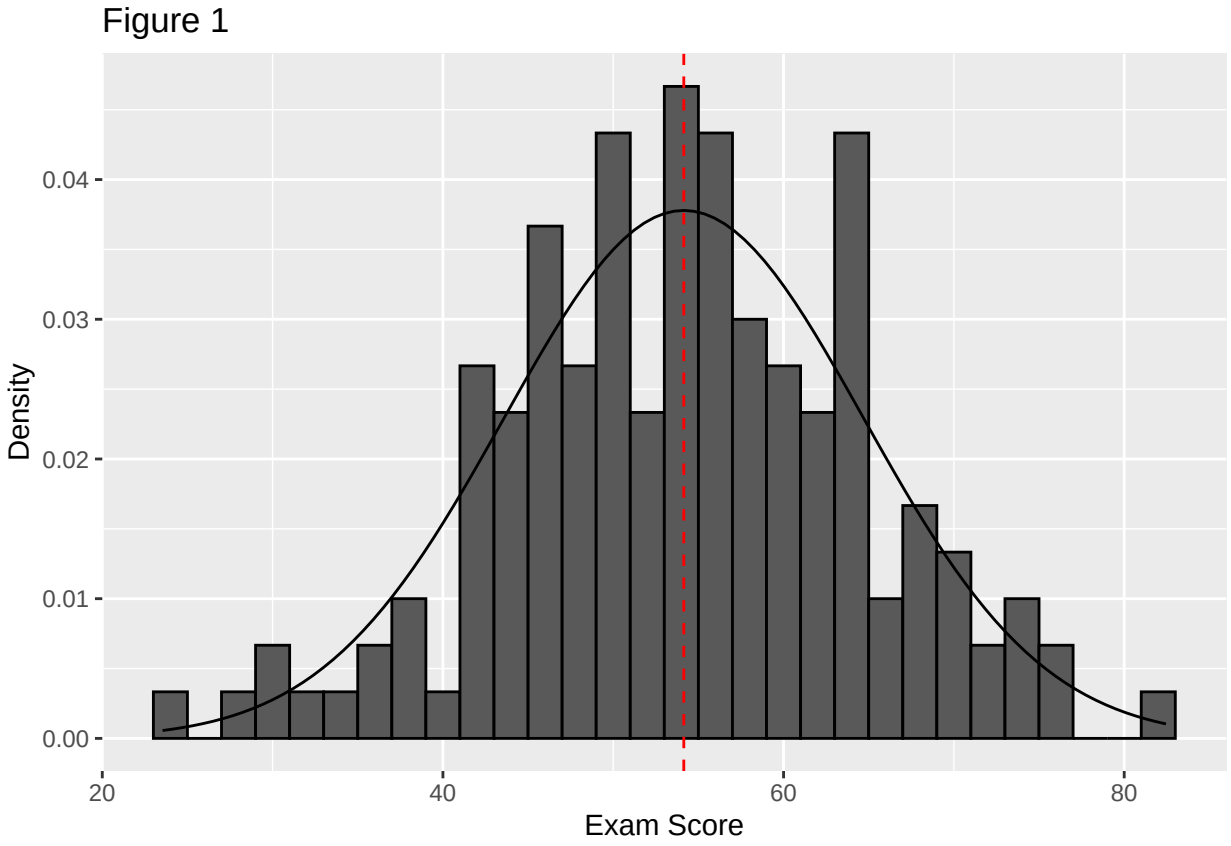


Figure 1: Probability Distribution of Exam Scores

The estimated probability of students getting each letter mark can be found in Table 1.

Table 1: Probability Distribution of Exam Scores

Mark	Probability
A	0.07
B	0.21
C	0.33
D	0.22
F	0.09

95% of the students are expected to be able to complete the exam in 59.26 minutes.