

BAS 471 Spring 2023 Homework on Unit 6 - Probability Models for Continuous Quantities

Ryan Curling

Worth double and cannot be dropped; see Canvas for due date; late assignments are accepted

Note: This assignment is worth double and cannot be dropped

The homework on Unit 6 is longer than others. It will be graded out of 200 points and thus will count double the other assignments. As such, it cannot be dropped. Unlike other assignments though, it is ok to submit this up to 3 days late (at a penalty of 15% per day). “Late” starts at 7am each day so that you have the usual buffer after 11:59pm on the due date to have it turned in without penalty.

Note: these are your homework problems

Reminder about collaboration policy. You can develop a common set of R code with you and your friends. However, anything that is written interpretation, i.e., anything that follows a **Response:** needs to be written up in your own words. Homeworks that look to be near copy/pastes of each other will receive substantially reduced credit.

Note: For this assignment, you’ll be required to perform integration (finding the area under the curve) and differentiation (finding the slope of a tangent line to a curve). Show your work when possible by pasting a screenshot from Wolfram-Alpha or a photo of work you’ve done “by hand” into your knitted Word document, or by using the R chunk when running `integrate` or `optimize` as requested.

Question 1:

Chain restaurants like McDonalds and Taco Bell receive plenty of online feedback, both positive and negative. Their customer support teams attempt to respond to all emails in a timely manner. Let X be the “response time”, i.e., the amount of time (in hours) that it takes customer service to address a concern raised in an email. There’s no apparent hard cap on the minimum or maximum length of time it takes to respond, so let’s have the possible

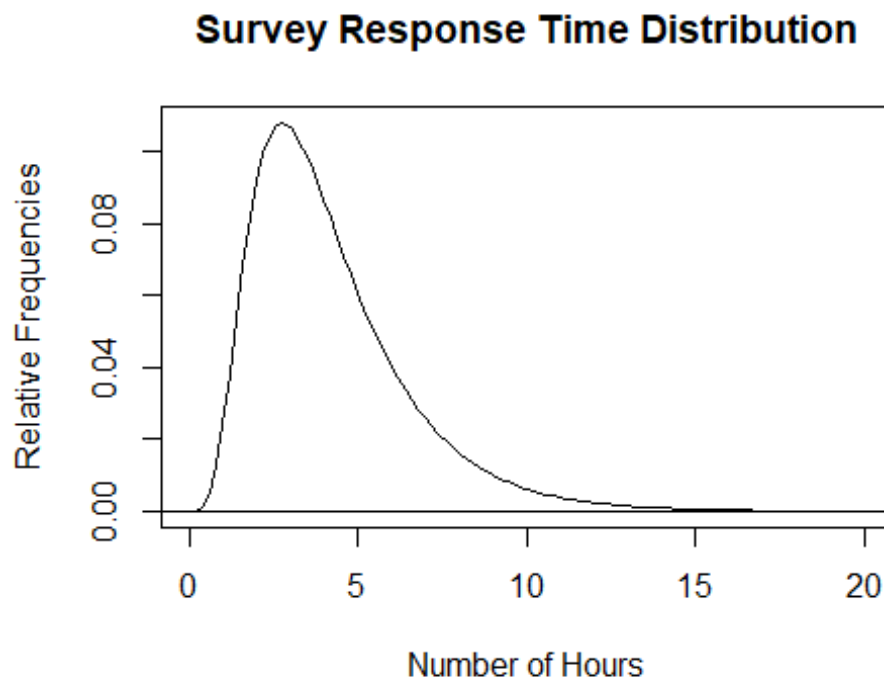
values of x be the continuum of values between 0 and infinity (any decimal number in this range, e.g. 1.42932 hours, is possible).

One shape that seems to describe the relative frequencies of the response time is the function :

$$Shape = \frac{e^{-x/2}}{\left(1 + \frac{4}{x^2}\right)^2} \quad 0 < x < \infty$$

- a) Use curve to visualize the shape from $x=0$ to $x=20$ (although there are larger possible values of x , most of the probability is contained between 0 and 20 hours). When transcribing, recall that to take e and raise it to a power, you have to use the function `exp()` rather than doing $e^$ and specifying the power. Label the x-axis “Number of Hours” and the y-axis “Relative Frequencies”. Add a horizontal line at 0 by running `abline(h=0)` as well. Note: you may need to have the curve and the `abline` command on the same line separated by a semi-colon if you have the Rmd set up to embed previews of the plots in the Rmd rather than in the plotting window.

```
curve(exp(-x/2) / (1 + 4/(x^2))^2, from = 0, to = 20, xlab = "Number of
Hours", ylab = "Relative Frequencies", main = "Survey Response Time
Distribution")
abline(h=0)
```



- b) For an equation to describe a valid probability density function (PDF), its values must all be positive (they are here). There is one more requirement on the equation.

Use an integrate command and explain why this shape doesn't describe a valid PDF. Remember that R refers to infinity as Inf.

Response: The integral of the PDF must equal 1, which it does not in this case. Thus, this shape does not describe a valid PDF.

```
shape <- function(x) exp(-x/2) / (1 + 4/(x^2))^2
tot_area <- integrate(shape, 0, Inf)
tot_area
## 0.4790291 with absolute error < 7.6e-05
```

- c) It is easy to convert the desired shape into a valid PDF by multiplying the shape equation by the reciprocal of a “well-chosen number”. You found this number in part (b), so define a function in R named PDF that takes a single argument x and returns the height of the PDF at that value of x. Include the output of `integrate(PDF, lower=0, upper=Inf)` to establish it is indeed a valid PDF (it integrates to “basically 1”), then provide a curve of the PDF from 0 to 20. Label the x-axis “Number of Minutes” and the y-axis “PDF”.

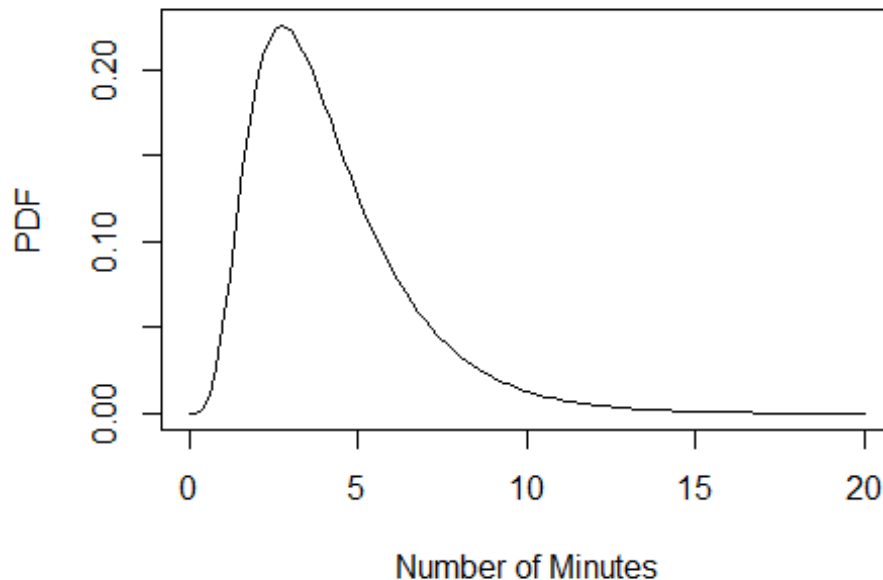
```
well_chosen_number <- 0.4790291

PDF <- function(x) {
  1/well_chosen_number * (exp(-x/2) / (1 + 4/(x^2))^2)
}

# Check if the new PDF function integrates to 1
integration_result <- integrate(PDF, lower = 0, upper = Inf)
integration_result
## 1 with absolute error < 4.9e-07

# Create a curve of the PDF from 0 to 20
curve(PDF, from = 0, to = 20, xlab = "Number of Minutes", ylab = "PDF", main = "Survey Response Time Distribution")
```

Survey Response Time Distribution



- d) Show that the PDF evaluated at $x=5$ is about 0.127 and that the PDF evaluated at $x=10$ is about 0.013. What do these numbers tell us about the probability that the response time is *exactly* 5 or *exactly* 10 hours? Provide an interpretation of these two numbers by considering their ratio.

Response: These numbers tell us nothing about the response time being exactly 5 or exactly 10. In continuous distribution models, the probability of an exact value is always zero. Rather, these numbers tell us the density of distribution at these points. In terms of the ratio between these two numbers, we can say that the distribution at $x=5$ is 9.79 times more dense than at $x=10$.

```
#Run PDF(5) and PDF(10)
```

```
pdf_5 <- PDF(5)
pdf_10 <- PDF(10)
```

```
pdf_5
## [1] 0.1273462
pdf_10
## [1] 0.01300466
```

```
#Density comparison ratio
```

```
pdf_5/pdf_10
## [1] 9.792349
```

- e) Find the mode of this distribution (using `optimize`; about 2.76), the expected value (using `integrate`; about 4.2), and the standard deviation (using `integrate`; about 2.3). Interpret the mean and standard deviation in the usual way (layman's terms, as if you were talking to the boss).

Response: For every piece of feedback sent to the fast food chain, the average response time will be 4.19 hours give or take 2.33 hours.

#Mode; from the plot you can see the peak is somewhere between 0 and 10, so use that for interval

```
mode_function <- function(x) -PDF(x) #Negate the PDF
mode_result <- optimize(mode_function, interval = c(0, 10))
mode <- mode_result$minimum

# Expected value
expected_value_function <- function(x) x * PDF(x)
expected_value_result <- integrate(expected_value_function, lower = 0, upper
= Inf)
mean <- expected_value_result$value

#standard deviation
variance_function <- function(x) (x - mean)^2 * PDF(x)
variance_result <- integrate(variance_function, lower = 0, upper = Inf)
sd <- sqrt(variance_result$value)

mode
## [1] 2.757599
mean
## [1] 4.196148
sd
## [1] 2.339075
```

- f) Bonus: find the 25th and 75th percentiles of this distribution to one decimal place. Reminder: there's a 25% chance of observing the 25th percentile response time or something smaller; there's a 75% chance of observing the 75th percentile or something smaller. Try to figure out how to find these by setting up an `optimize` command. If you can't, trial and error is fine (I could not get Wolfram to solve the integral equation).

#Set up CDF

```
CDF <- function(x) {
  cdf_result <- integrate(PDF, lower = 0, upper = x)
  return(cdf_result$value)
}
```

Setting up a function to find the difference between the target percentile and the CDF for a given x value

```
percent_diff <- function(x, target) {  
  return(abs(target - CDF(x)))  
}
```

25th percentile

```
percentile_25_result <- optimize(percent_diff, interval = c(0, mean), target  
= 0.25)  
percentile_25 <- percentile_25_result$minimum
```

75th percentile

```
percentile_75_result <- optimize(percent_diff, interval = c(mean, 10), target  
= 0.75)  
percentile_75 <- percentile_75_result$minimum
```

#Round answer to one decimal place

```
round(percentile_25, 1)  
## [1] 2.5  
round(percentile_75, 1)  
## [1] 5.3
```

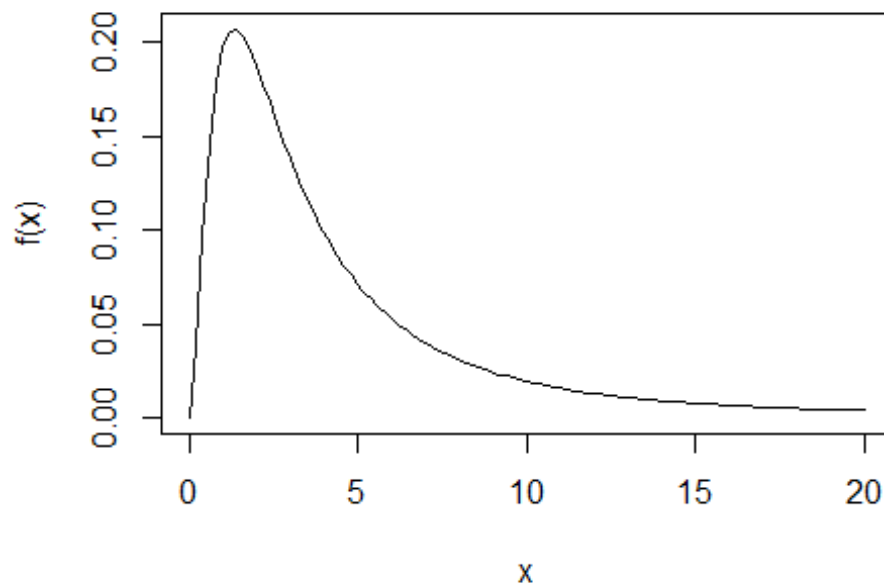
Question 2:

The customer analytics team would prefer a different equation since taking e and raising it to a power makes the model feel intimidating and less approachable. Since taking x and raising it to a power is more comfortable, the team wants to model X (the response time in hours) with the PDF:

$$f(x) = \frac{48x^2}{(2+x)^5} \quad 0 \leq x \leq \infty$$

Let's define a function named PDF that takes as its argument a vector x and that returns the corresponding values of the PDF.

```
PDF <- function(x) { 48*x^2/(2+x)^5 } #Define the PDF  
curve( PDF, from=0, to=20, ylab="f(x)") #Look at the PDF (values are all  
positive, good!)
```



```
integrate(PDF,lower=0,upper=Inf) #Establish it's a valid PDF
## 1 with absolute error < 5.3e-06
```

According to the probability model, what fraction of response times are:

- a) exactly 80 minutes (1.333 hours, which is near where the distribution peaks)
- b) less than 5 hours (about 0.68)
- c) 10 hours or more (about 0.13)
- d) between 2 and 3 hours (about 0.16)

```
#a P(X=80)
```

```
PDF_80min <- PDF(80/60)
PDF_80min
## [1] 0.20736
```

```
#b P(X<5)
```

```
p_less_than_5 <- integrate(PDF, lower = 0, upper = 5)$value
p_less_than_5
## [1] 0.6768013
```

```
#c P(X>=10)
```

```


p_greater_than_10 <- integrate(PDF, lower = 10, upper = Inf)$value
p_greater_than_10
## [1] 0.1319444

#d P(2 <= X <= 3)



p_between_2_and_3 <- integrate(PDF, lower = 2, upper = 3)$value
p_between_2_and_3
## [1] 0.1627





```

- e) Using Wolfram-Alpha, find the formula for $F(x) = P(X \leq x)$, the cumulative distribution function (CDF) for X . Recall that $F(x)$ provides a nice function for finding the probability of observing a value of “at most x ”. Include a screenshot showing the equation. Then, define a function CDF in the R chunk below that takes a single argument x and that returns the corresponding values of F . Include a curve of the CDF for values between 0 and 20. Note: you can ignore Wolfram’s “for $\text{Re}(x) > -2 \forall x \notin \mathbb{R}$ ” - that’s for picky math folk and is irrelevant to us.

 **WolframAlpha** computational intelligence

integrate $48x^2/(2+x)^5$ from $x=0$ to $x=x$

 NATURAL LANGUAGE  MATH INPUT

 EXTENDED KEYBOARD  EXAMPLES  UPLOAD  RANDOM

Definite integral

$$\int_0^x \frac{48x^2}{(2+x)^5} dx = \frac{x^3(x+8)}{(x+2)^4} \text{ for } \text{Re}(x) > -2 \forall x \notin \mathbb{R}$$

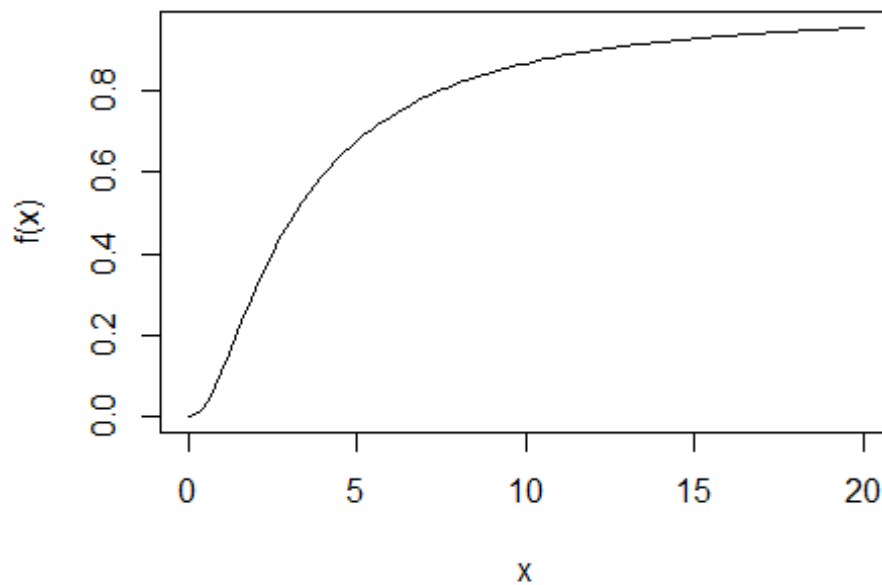
$\text{Re}(z)$ is the real part of z
 $e_1 \vee e_2 \vee \dots$ is the logical OR function
 \mathbb{R} is the set of real numbers

Show/verify that $\text{CDF}(0)$ equals 0, $\text{CDF}(999999)$ (“basically infinity”) equals 1, then use the CDF function to calculate

- $P(X \leq 5)$; about 0.68
- $P(X > 15)$; about 0.07
- $P(2 \leq X \leq 3)$, which was also calculated in part (d)

Response:

```
CDF <- function(x) {(x^3*(8+x))/(2+x)^4}
curve(CDF, from = 0, to=20, ylab="f(x)")
```



```
#Show CDF(0) and CDF(999999)
```

```
CDF(0)
## [1] 0
CDF(999999)
## [1] 1
```

```
#P(X <= 5)
```

```
CDF(5)
## [1] 0.6768013
```

```
#P(X > 15)
```

```
1-CDF(15)
```

```
## [1] 0.07059302
```

```
#P(2 <= X <= 3)
```

```
CDF(3)-CDF(2)
```

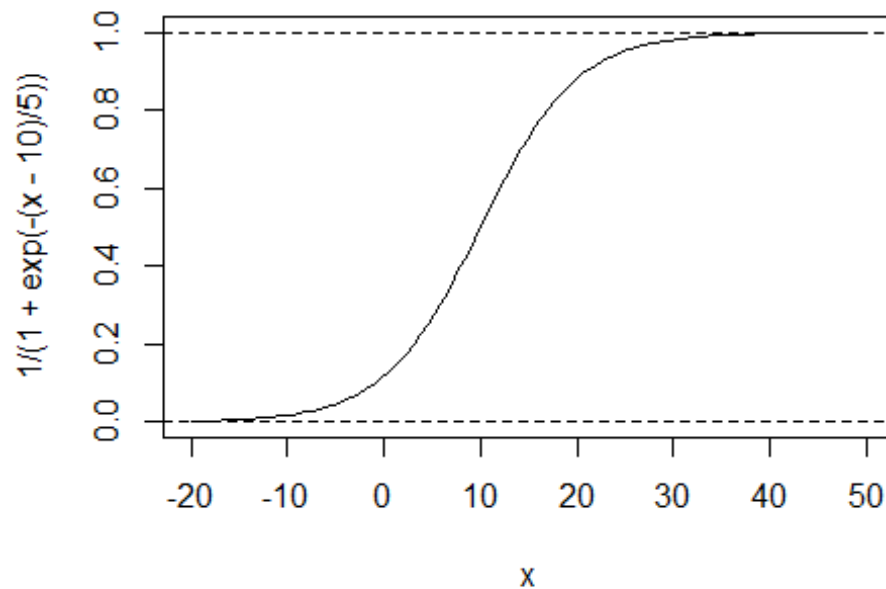
```
## [1] 0.1627
```

Question 3:

Most of the time, stores are able to sell items at a profit. Occasionally, items need to be liquidated at a loss. A CDF that provides a distribution for both positive and negative numbers is:

$$F(x) = \frac{1}{1+e^{-(x-10)/5}} \quad -\infty < x < \infty$$

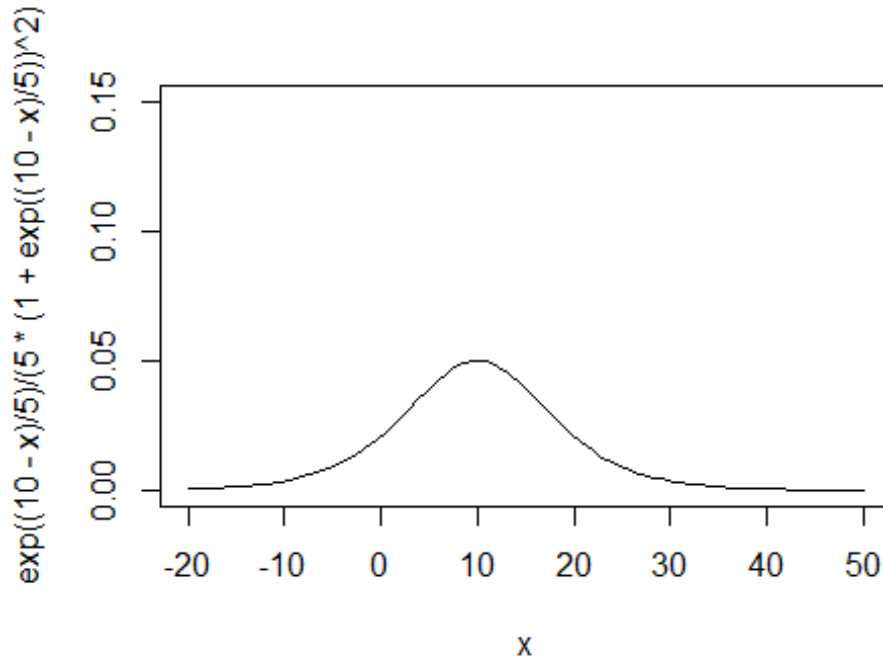
```
curve( 1/(1+exp(-(x-10)/5)) ,from=-20,to=50); abline(h=c(0,1),lty=2)
```



This distribution could be useful in modeling the amount of money gained or lost when an item is sold. Let's work with this model.

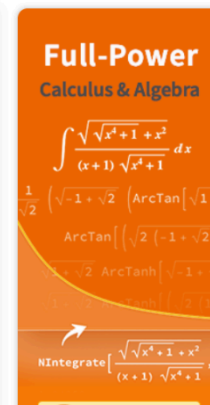
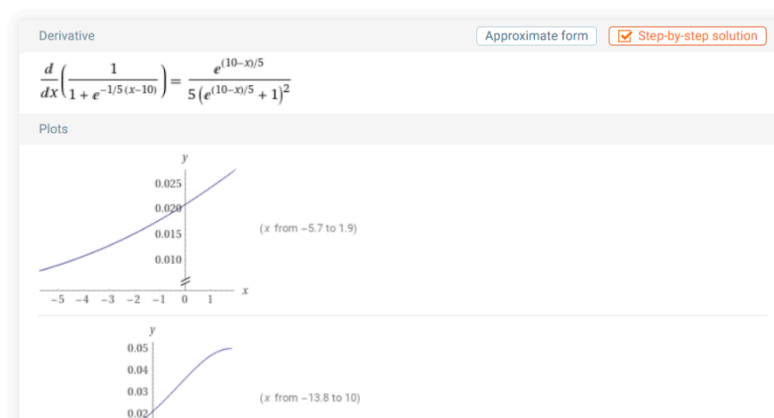
- Find the formula for the PDF $f(x)$, then provide a curve between $x=-20$ and $x=50$. A screenshot from Wolfram-Alpha will suffice for finding the formula. This is a tricky one to type in R with all the parentheses. You should see a nice bell-shaped curve peaking at 10.

```
curve(exp((10-x)/5)/(5*(1+exp((10-x)/5))^2), from=-20, to=50, ylim=c(0,0.15))
```



derivative $1 / (1 + e^{-(x-10)/5})$ with respect to x

NATURAL LANGUAGE MATH INPUT EXTENDED KEYBOARD EXAMPLES UPLOAD compute input



Response:

- b) Find the formula for the quantiles of this distribution, i.e., the equation for the quantile function $F^{-1}(x)$. A screenshot from Wolfram-Alpha will suffice. Then, write a function named QUANTILE that takes a single argument q and returns the q-th quantile (e.g., if you passed 0.60 to the function, it would give you the 0.60 quantile / 60th percentile). Show that QUANTILE(0.60) is about 12.

Note: choose the “Real solution” rather than the “Result”. The equation is pretty simple here, so if you get an extremely complex one, ensure you’re working with the CDF and not the PDF when setting up Wolfram.

The screenshot shows the WolframAlpha interface with the input: `solve (integrate exp((10-x)/5)/(5*(1+exp((10-x)/5))^2)) = q for x`. The input is interpreted as solving the equation $\int \frac{\exp(\frac{10-x}{5})}{5(1 + \exp(\frac{10-x}{5}))^2} dx = q$ for x . The result section shows a complex expression involving $2i\pi n + \log(-\frac{q+1}{q}) + 2$, with a note that $\log(x)$ is the natural logarithm and \mathbb{Z} is the set of integers. The real solution section shows $x = 5 \log\left(\frac{-q-1}{q}\right) + 10$ for $-1 < q < 0$. The indefinite integral section shows the integral of the function, which is $-\frac{5}{e^2}$.

Response:

```
#QUANTILE <- function(q) { transcribe your function with q as the variable }
```

```
QUANTILE <- function(q) {  
  if (q <= 0 || q >= 1) {  
    stop("q must be between 0 and 1")  
  }  
  result <- -5*log(1/q - 1) + 10  
}
```

```
    return(result)
}
```

```
QUANTILE(0.60)
## [1] 12.02733
```

- c) Using your QUANTILE function, find the 50th (median) and 20th percentiles (about 10 and 21, respectively). Finally, fill in the blank: 90% of individuals' values of x will be less than or equal to ____ (i.e. there's a 90% chance that x is less than or equal to ____)

Response:

#Q50/ median

```
QUANTILE(.50)
## [1] 10
```

#Q20

```
QUANTILE(.2)
## [1] 3.068528
```

Question 4:

Garbage trucks sent out to collect the trash rarely return at 100% capacity. In fact, a lot of randomness factors in to how full they are on return since the amount of trash households discard is random (and they may not even remember to put the trash out to be picked up).

Let's use the following PDF to model the "fullness" at which trucks return. We'll represent the fullness as a number between 0 (empty) and 1 (completely full).

$$f(x) = 24x(1 - x)^7 + 70x^4(1 - x)^2 \quad 0 \leq x \leq 1$$

#Valid PDF because it integrates to 1

```
integrate( function(x) 24*x*(1-x)^7 + 70*x^4*(1-x)^2, lower=0, upper=1)
## 1 with absolute error < 1.1e-14
```

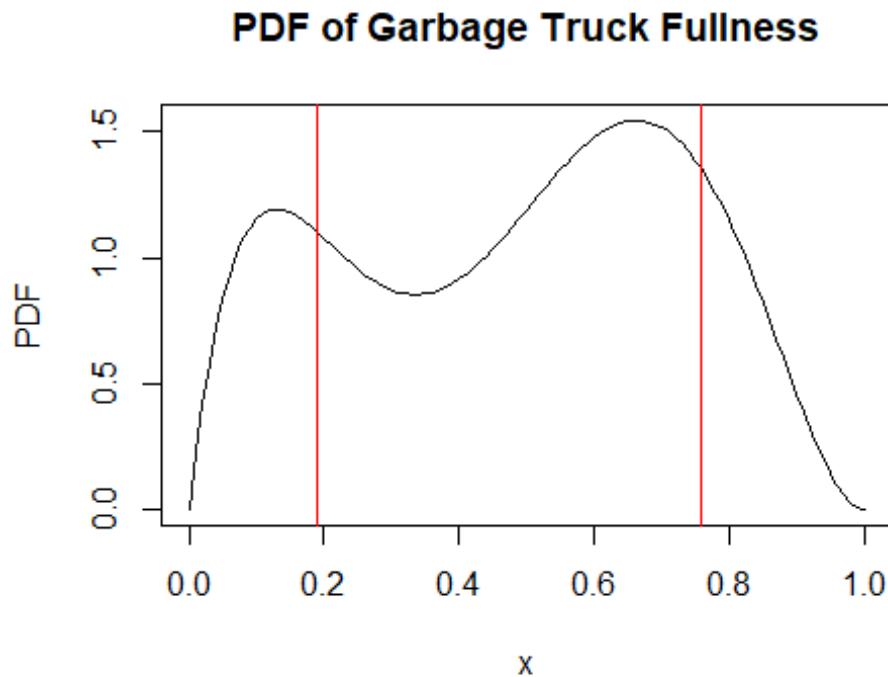
- a) Find the two modes of the distribution. Provide a curve of the PDF, then use `abline(v=?)` to add a vertical line at each mode. Note: you may need to have the curve and two `abline` commands on the same line separated by a semi-colon if you have the Rmd set up to embed previews of the plots in the Rmd rather than in the plotting window.

Define the PDF

```
f <- function(x) 24*x*(1-x)^7 + 70*x^4*(1-x)^2
```

```
# Plot the PDF
curve(f, from=0, to=1, ylab="PDF", main="PDF of Garbage Truck Fullness")

# Add vertical lines at the modes
abline(v=c(0.189, 0.757), col="red")
```



- b) Find the expected value μ (near 0.48) and standard deviation σ (near 0.25) of this distribution.

```
#E[X]; expected value,
mu <- integrate(function(x) x*f(x), lower=0, upper=1)$value
mu
## [1] 0.4833333

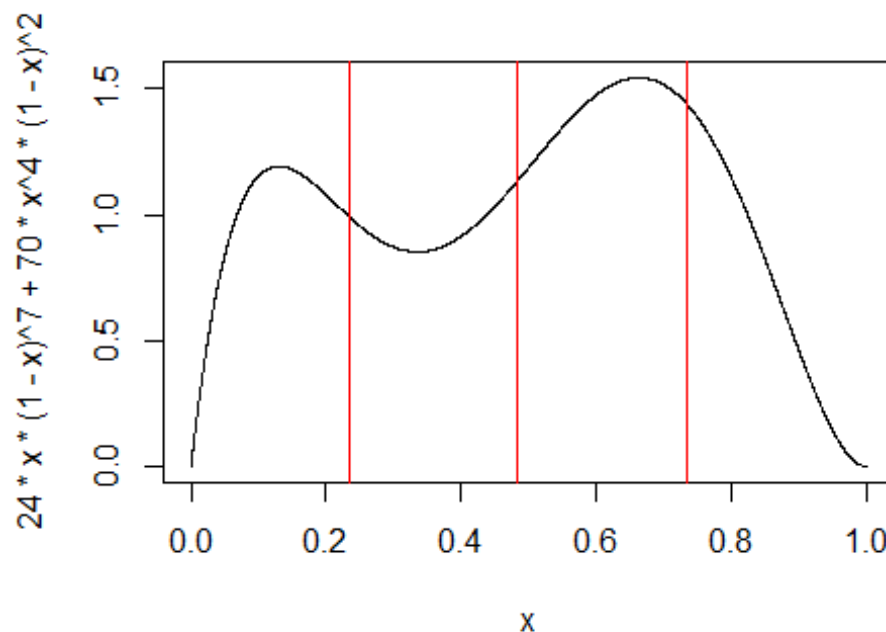
#SD; you can do two lines of code to calculate this
variance <- integrate(function(x) (x-mu)^2 * f(x), lower=0, upper=1)$value
sd <- sqrt(variance)
sd
## [1] 0.2496968
```

- c) The following R chunk re-makes the curve. It then adds vertical lines at the mean and at the mean plus/minus one standard deviation (replace the ? and @ in the lines of code with the mean and standard deviation, respectively, then uncomment and run them).

Explain why the expected value and standard deviation are “lousy” at summarizing this distribution. Note: you may need to have the curve and two abline commands on the same line separated by a semi-colon if you have the Rmd set up to embed previews of the plots in the Rmd rather than in the plotting window.

Response: This is a bimodal distribution, meaning the mean and sd do not provide an accurate calculation for the true center and spread of the distribution, because they are influenced by both modes.

```
curve( 24*x*(1-x)^7 + 70*x^4*(1-x)^2 ,from=0,to=1, n=5000)
abline(v=mu,col="red") #Replace the ? with mu
abline(v=mu + c(-1,1)* sd ,col="red" ) #Replace the ? with mu and the @ with the SD
```



- d) The time it takes to fill a truck to a certain fullness depends on the number of trash bins out for the day as well as how much trash they contain. Suppose we can convert X (Fullness) to Time (in minutes) using the following equation:

$$Time = 58(1 + X^{1/4})$$

Find the expected value of the time it takes to fill the truck (about 105 minutes). Note: this equation is *NOT* describing the shape of the PDF of the Time. We don't even need to find the PDF of Time to find its expected value!

```
# Define the PDF
g <- function(x) 24*x*(1-x)^7 + 70*x^4*(1-x)^2
```

```
# Define the time function
T <- function(x) 58*(1+x^(1/4))

# Calculate the expected value
mu <- integrate(function(x) T(x)*g(x), lower=0, upper=1)$value
mu
## [1] 104.5682
```

- e) A central theme of probabilistic modeling is that averages are not good enough to characterize a complex random process (anything that's not a simple weighted sum of its random components). You *cannot* just plug in the average value of X in the equation for the *Time* to get the average time! Go ahead and plug in the expected value of X (part b) for the value of X in the equation for *Time*, and take note how the “wrong” way of calculating the average *Time* is an overestimate of the actual average *Time* (though luckily, in this example, not by much).

```
#Define x from part b
x <- 0.4833333

# Define the time function
T <- function(x) 58*(1+x^(1/4))
T(x)
## [1] 106.3604
```

Question 5: Notorious Distribution Identification

We have studied the following notorious continuous distributions:

Normal Beta Weibull Exponential Lognormal Uniform (continuous)

Based on the following scenarios and descriptions (i.e., in the absence of any data), name which distribution would be your “go-to” for modeling the following quantities. Remember that the Weibull is a “generalization” of the Exponential, and the Beta is a “generalization” of the Uniform. Don’t list Weibull (or Beta) when the Exponential (or Uniform) will do!

Each distribution will be used at least once. *Provide a very brief justification for your choice.*

- a) The length of the longer of two pieces of a ruler when it is randomly split in two.

Response: Beta distribution. A ruler represents a bounded set of measurements, and the beta distribution has the ability to model proportions within a fixed range.

- b) The length of time between earthquakes in Knoxville
(<https://earthquaketrack.com/us-tn-knoxville/recent>; small ones happen more often than you think). Assume that the probability of an earthquake per unit

time is relatively constant, so the expected time until the next quake is independent of how long it's been since the last quake.

Response: Uniform Distribution. Independent events with a constant rate, indicating the uniform distribution would be best.

- c) The distance Cannon and Jackson will walk in a park before getting distracted and stopping to sniff the grass. Skewed left with a peak at 20 ft.

Response: Weibull distribution. Since the data is skewed, weibull distribution is appropriate.

- d) The amount by which the measured weight deviates from the true weight when putting produce on various scales across grocery stores in the US (roughly symmetric, positive and negative values, peaks near 0).

Response: Normal distribution. Symmetric data, indicates normal distribution is appropriate.

- e) The fraction of a peach's weight that is the pit (between 0 and 1; peak near 0.06).

Response: Beta distribution. Appropriate for modeling proportions within a fixed range (0-1 in this case).

- f) The time that elapses between when a dish has completed preparation in a restaurant kitchen and when it is delivered to the person who ordered it. Skewed left with a peak around 5.2 minutes.

Response: Weibull distribution. Can model skewed data.

- g) The daily percentage changes in the exchange rate between US Dollars and Euros (roughly symmetric, can be positive or negative)

Response: Normal distribution. Symmetric data.

- h) The amount of time between placing your order at the drive-thru speaker at Taco Bell and picking your order up from the pickup window (skewed to the right, peaking around 4.2 minutes).

Response: Weibull distribution. Models skewed data well.

- i) The proportion (0-1) of a pan of baked lasagna that gets eaten by a family of four before being thrown away. Skewed left with a peak at 0.85.

Response: Beta distribution. Measures proportions within a fixed range (0-1) in this case.

- j) The exit velocity (how fast, in miles per hour, a ball was hit by a batter) of home runs hit by Aaron Judge (New York Yankees), which appears to have a left-skew (mostly due to popups; overall peak is near 109.1 mph)

Response: Lognormal distribution. Good for modeling positively skewed data with a lower bound of 0.

- k) The loan amounts requested at a “payday loan” center (e.g., Check Into Cash). Amounts are skewed to the right, peaking at around 742.67.

Response: Lognormal distribution. Good for modeling positively skewed data with a lower bound of 0.

Question 6: Practice using the notorious distributions.

For each part, you’ll see a curve of the distribution that uses R’s `d` version of the PDF. In R (no Wolfram), use the `p` version of the PDF to answer questions about the probabilities, and use the `q` version of the PDF to answer questions about percentile or quantiles.

- a) Consider the length of time (in *seconds*) someone spends on TikTok each day (among a certain group of daily users of the app). A reasonable distribution to model this quantity could be the lognormal since all values are positive numbers, the times are probably skewed to the right, and it’s generally the go-to for many quantities in analytics when human behavior is involved.

Data suggests that the `meanlog` and `sdlog` of this distribution are 8.3 and 1.2, respectively. Provided is code that finds the expected value, median, and standard deviation of the distribution. A curve of this model from $x=0$ to $x=20000$ is also provided. Use this model to find:

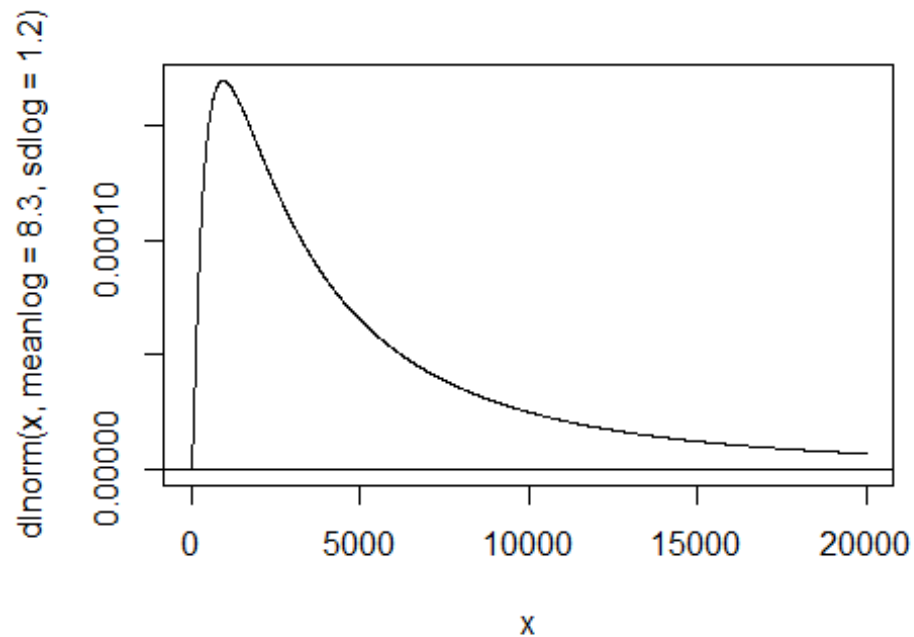
- the probability that someone spends between 30 minutes and 1 hour on Tik Tok? (about 21%)
- the probability that someone spends at least 2 hours on Tik Tok? (about 31%)
- the 60th percentile of time spent on Tik Tok (around 5450 seconds)
- the expected increase in a user’s engagement when a user spends X hours on TikTok. Mouse over the equation to see how time spent translates to increase in engagement:

$$Engagement = \sqrt{x}(1 + e^{-x/3600})$$

In other words, the increase in engagement equals $\sqrt{x} \cdot (1 - \exp(-x/3600))$. You will have to use the `integrate` function, multiplying the formula for the engagement by the `dlnorm(x, meanlog=8.3, sdlog=1.2)` function (R’s shorthand of the equation for the lognormal so we don’t have to type it out). Sanity: basically 60.

```
meanlog <- 8.3; sdlog <- 1.2
#Expected Value
exp(meanlog + sdlog^2/2)
## [1] 8266.777
#Median
exp(meanlog)
```

```
## [1] 4023.872
#Standard Deviation
sqrt( (exp(sdlog^2)-1)*exp(2*meanlog+sdlog^2) )
## [1] 14835.8
#Curve
curve(dlnorm(x,meanlog=8.3,sdlog=1.2), from=0, to=20000,n=2000 ); abline(h=0)
```



```
#Answer questions here (60 seconds in a minute; 3600 seconds in an hour)
```

```
#P(30 min < X < 1 hour)
```

```
plnorm(3600, meanlog = 8.3, sdlog = 1.2) - plnorm(1800, meanlog = 8.3, sdlog
= 1.2)
## [1] 0.2117402
```

```
#P(X >= 2 hours)
```

```
1 - plnorm(7200, meanlog = 8.3, sdlog = 1.2)
## [1] 0.3138866
```

```
#60th percentile
```

```
qlnorm(0.6, meanlog = 8.3, sdlog = 1.2)
```

```
## [1] 5453.52
```

```
#E[Engagement]
```

```
engagement <- function(x) {  
  sqrt(x) * (1 - exp(-x / 3600)) * dlnorm(x, meanlog = 8.3, sdlog = 1.2)  
}  
integrate(engagement, lower = 0, upper = Inf)$value  
## [1] 60.00091
```

- b) Consider the length of time that elapses between sequential views of a trailer for Stanley Kubrick's masterpiece "The Shining" but recut as a romantic comedy (https://www.youtube.com/watch?v=KmkVWuP_s00 if you're curious; there's a Scary Mary Poppins trailer too at https://www.youtube.com/watch?v=2T5_0AGdFic). Let's model the elapsed time (during the 7-8am hour) by an exponential distribution with a rate (λ) parameter of $1/0.2$ (i.e., 1 every 0.2 minutes, or one every 12 seconds). The units of X are minutes.

Sidenote: the formulas for the mean and standard deviation of an exponential are identical:

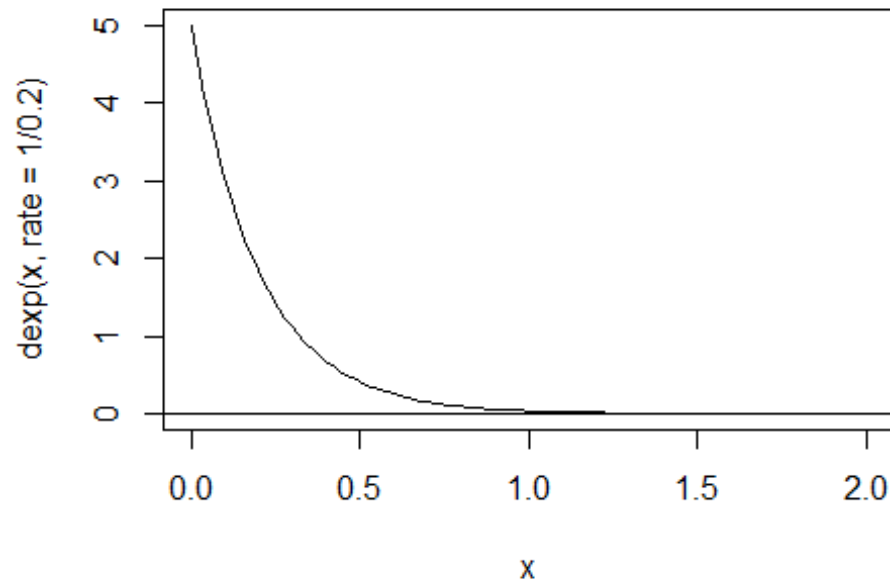
$$\mu = \sigma = \frac{1}{\lambda} = \frac{1}{1/0.2} = 0.2$$

Provided is a curve of this model from $x=0$ to $x=2$. Use this model to find:

- the probability that the next view happens 10-20 seconds ($1/6$ to $1/3$ of a minute) after the last view? (around 25%)
- the probability that the next view happens in the next 30 seconds (0.5 minutes), given that it's been over 1 minute since the last view? (around 8%)
- the median time between views (about 8.3 seconds)

```
#Curve
```

```
curve( dexp(x, rate=1/0.2), from=0, to=2); abline(h=0)
```



#Answer questions here

#P(between 1/6 to 1/3 minutes)

```
pexp(1/3, rate = 1/0.2) - pexp(1/6, rate = 1/0.2)
## [1] 0.2457226
```

#P(more than 0.5 minutes)

```
1-(pexp(1.5, rate = 1/0.2) - pexp(1, rate = 1/0.2)) / (1 - pexp(1, rate =
1/0.2))
## [1] 0.082085
```

#Median

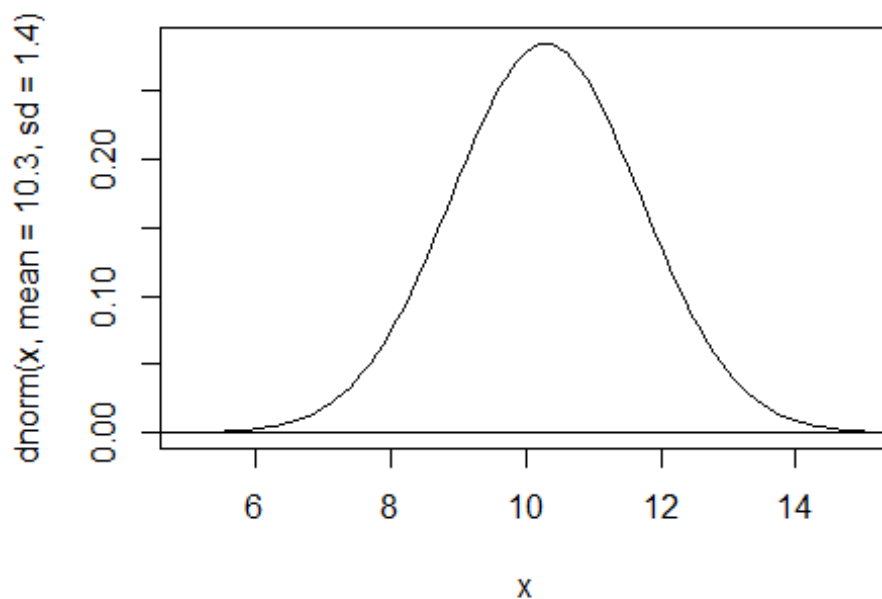
```
60 * qexp(0.5, rate = 1/0.2)
## [1] 8.317766
```

- c) Let's model the length of time it takes a computer to complete a trial of a sophisticated Monte Carlo simulation with a Normal distribution that has a mean (mean) of 10.3 seconds and a standard deviation (sd) of 1.4 seconds. Some trials take longer than others since some scenarios are more difficult to compute.

Provided is a curve of this model from $x=5$ to $x=15$. Use this model to find:

- the time at the 80th percentile of the distribution (about 11.5)
- the probability that the time is negative (after all, the Normal model allows negative values and we're using it to model a strictly positive quantity; it's necessary to know if we can "get away" with this by ensuring the probability of an impossible value is very low)
- the probability that the time for a trial is between 12 and 13 seconds (about 8.5%)

```
curve(dnorm(x, mean=10.3, sd=1.4), from=5, to=15); abline(h=0)
```



#80th percentile

```
qnorm(0.8, mean = 10.3, sd = 1.4)
## [1] 11.47827
```

$P(X < 0)$

```
pnorm(0, mean = 10.3, sd = 1.4)
## [1] 9.394412e-14
```

$P(12 \leq X \leq 13)$

```
pnorm(13, mean = 10.3, sd = 1.4) - pnorm(12, mean = 10.3, sd = 1.4)
## [1] 0.08542728
```

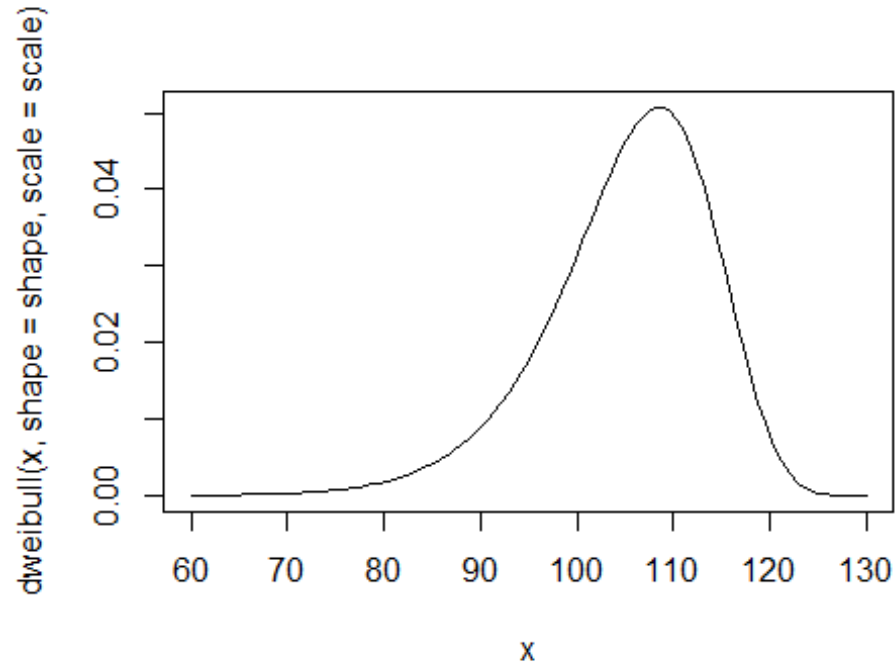
- d) Bonus. Consider the exit velocity of home runs hit by Aaron Judge (New York Yankees). The exit velocity is the speed of the ball after it is hit. Let's model the exit velocity with a Weibull distribution with a shape parameter of 15 and a scale parameter of 109.
- Provide a curve of the PDF between 60 and 130
 - Find the mode of the PDF
 - Report the probability that the exit velocity is between 105 and 110 mph
 - Report the median and 10th percentile exit velocities
 - Look up the formulas and report the mean and standard deviation of this Weibull distribution. Confirm your answers by running `v <- rweibull(1e6, shape=15, scale=109)` and getting the mean and standard deviation of the numbers in that vector (the values will be "close").
 - Interpret the mean and standard deviation in the usual layman-friendly way.

Note: if you look up the formulas on Wikipedia, the λ parameter is the scale parameter (15) and the k parameter is the shape parameter (109). Note: the Γ symbol is the "gamma" function (like the factorial function but it even works for decimals). You can evaluate it in R by running `gamma`. For example $\Gamma(1 + 1/1.5)$ is `gamma(1+1/1.5)`

Response: For every home run Aaron Judge, the average exit velocity of the baseball will be 105.26 mph give or take 8.61 mph. (rounded to 2 decimal places)

#Curve

```
shape <- 15
scale <- 109
curve(dweibull(x, shape = shape, scale = scale), from = 60, to = 130)
```



#Mode

```
mode <- (scale * ((shape - 1) / shape)^(1 / shape))  
mode  
## [1] 108.4998
```

#Between 105 and 110

```
pweibull(110, shape = shape, scale = scale) - pweibull(105, shape = shape,  
scale = scale)  
## [1] 0.247457
```

#Median and 10th percentile

```
median <- qweibull(0.5, shape = shape, scale = scale)  
tenth_percentile <- qweibull(0.1, shape = shape, scale = scale)  
median  
## [1] 106.3689  
tenth_percentile  
## [1] 93.81487
```

*#Mean: scale*gamma(1+1/shape)*


```

mean <- scale * gamma(1 + 1 / shape)
mean
## [1] 105.2574

#SD: sqrt( scale^2*( gamma(1+2/shape) - gamma(1+1/shape)^2 ) )

sd <- sqrt(scale^2 * (gamma(1 + 2 / shape) - gamma(1 + 1 / shape)^2))
sd
## [1] 8.608109

#Confirm (these numbers are close to the answers above)
v <- rweibull(1e6, shape = shape, scale = scale)
mean(v)
## [1] 105.2463
sd(v)
## [1] 8.610998

```

Question 7: fitting distributions

The following chunk of code reads in two datasets stored on Google sheets.

```

gsheets_url
<- "https://docs.google.com/spreadsheets/d/1rFHBmzf0nrNXw8dyMVIHORWkAbRIjtV2Er
034AQd0Q0/export?format=csv"
MLB <- read.csv(gsheets_url)
str(MLB)
## 'data.frame':    1983 obs. of  20 variables:
## $ id                : int  1 2 3 4 5 6 7 8 9 10 ...
## $ rank              : int  1 2 3 4 5 6 7 8 9 10 ...
## $ year              : int  2022 2022 2022 2022 2022 2022
2022 2022 2022 2022 ...
## $ player            : chr  "Judge, Aaron" "Alvarez,
Yordan" "Trout, Mike" "Schwarber, Kyle" ...
## $ batted_ball_events : int  341 304 234 312 228 354 386
241 225 350 ...
## $ launch_angle      : num  14.6 12.1 24.7 18.9 10.2 12.6
13.9 14.6 21.4 15.7 ...
## $ sweet_spot_percentage : num  37.8 40.1 37.2 34.6 26.3 34.7
39.1 34 34.2 38.6 ...
## $ max_ev            : num  118 117 114 115 120 ...
## $ average_ev        : num  95.8 95.5 91.7 93.3 94.6 92.6
92.8 93.3 92.9 91.1 ...
## $ fly_ball_line_drive_ev : num  100.2 98.2 94.9 99.7 98.2 ...
## $ ground_ball_ev    : num  89.1 92.7 87.7 87.2 94.1 87.7
87.5 91.2 88.3 87.8 ...
## $ max_distance      : int  465 469 472 468 445 462 441
441 469 437 ...

```

```

## $ average_distance      : int  205 193 218 197 160 180 190
191 194 186 ...
## $ average_homerun      : int  413 403 407 415 400 408 411
410 410 408 ...
## $ hard_hit_95mph       : int  210 186 120 170 117 168 202
130 113 161 ...
## $ hard_hit_percentage  : num  61.6 61.2 51.3 54.5 51.3 47.5
52.3 54.2 50.2 46 ...
## $ hard_hit_swing_percentage : num  20.3 23.7 16.5 17.6 17.8 16.2
17.4 20 15.8 14.3 ...
## $ total_barrels        : int  91 59 46 64 42 62 64 38 37 49
...
## $ barrels_batted_balls_percentage : num  26.7 19.4 19.7 20.5 18.4 17.5
16.6 15.8 16.4 14 ...
## $ barrels_plate_appearance_percentage: num  15.8 12.7 11.6 11.5 11.3 11.1
10.9 10.6 9.7 9.6 ...
gsheets_url <-
"https://docs.google.com/spreadsheets/d/1UEws8XIAUIp-sGbit45lbKkUeFCvRzn1CWlF
qh50uaw/export?format=csv"
SONGS <- read.csv(gsheets_url)
str(SONGS)
## 'data.frame': 848 obs. of 18 variables:
## $ popularity      : int  73 62 55 60 76 55 29 54 47 61 ...
## $ track_name      : chr  "Unconditionally" "IN MY REMAINS" "If I Ruled
the World" "Bravado" ...
## $ artist_name     : chr  "Katy Perry" "Linkin Park" "BTS" "Lorde" ...
## $ duration_ms     : int  228879 200693 247450 221409 226867 321827 176118
215947 234185 438493 ...
## $ explicit        : int  0 0 0 0 0 1 1 0 0 1 ...
## $ danceability     : num  0.432 0.553 0.744 0.489 0.506 0.501 0.593 0.47
0.475 0.443 ...
## $ energy          : num  0.725 0.907 0.725 0.535 0.886 0.301 0.914 0.792
0.554 0.576 ...
## $ key             : int  7 9 7 7 11 3 1 0 9 1 ...
## $ loudness        : num  -4.86 -5.62 -6.29 -10.34 -3.23 ...
## $ mode            : int  1 0 1 1 1 0 1 1 0 0 ...
## $ speechiness     : num  0.0431 0.0484 0.112 0.0933 0.0655 0.0366 0.0363
0.26 0.0288 0.201 ...
## $ acousticness    : num  0.00273 0.00401 0.0434 0.539 0.0916 0.758
0.00137 0.171 0.552 0.0709 ...
## $ instrumentalness: num  0.00 0.00 0.00 2.66e-02 0.00 7.93e-03 4.45e-01
1.28e-06 0.00 1.97e-02 ...
## $ liveness        : num  0.208 0.266 0.296 0.166 0.0734 0.107 0.0714
0.313 0.212 0.41 ...
## $ valence         : num  0.353 0.451 0.73 0.04 0.595 0.179 0.0381 0.214
0.331 0.127 ...
## $ tempo           : num  129 101 96 176 144 ...
## $ time_signature  : int  4 4 4 4 4 4 4 4 4 4 ...
## $ followers       : int  18021998 18611017 31623813 7474152 6325985
4418323 14588353 27048881 9339948 6122508 ...

```

The first dataset contains information on 596 players in Major League Baseball (2015-2022 seasons), specifically the pitches they've hit at bat (mainly exit velocity and related information). The second dataset comes from Spotify and contains information on 848 songs from popular artists. Spotify provides numerical summaries of songs like energy, acousticness, danceability, loudness, etc. See <https://rpubs.com/PeterDola/SpotifyTracks> for a nice writeup.

- a) Consider the `hard_hit_95mph` column in `MLB`, which gives the number of hits where the exit velocity of the ball is at least 95 mph.
 - Look at a histogram (remember to extract out a column of a dataframe you'll refer to it as `MLB$hard_hit_95mph`). You don't need to include this initial histogram in the writeup.
 - Include the QQ plots for fits to a Normal, lognormal, exponential, and Weibull model.
 - Based on the QQ plots, which two of these probability models look the most reasonable (explain why you feel this way)?
 - Look at the relevant metric as a tie-breaker, and report what model is the "best" and why.
 - Make a histogram (add `freq=FALSE` as an argument), then use `curve` (add `add=TRUE` as an argument) to superimpose this "best" fit, and use it along with the Q-Q plot to comment on whether ultimately the "best" fit provides a reasonable fit to the data.

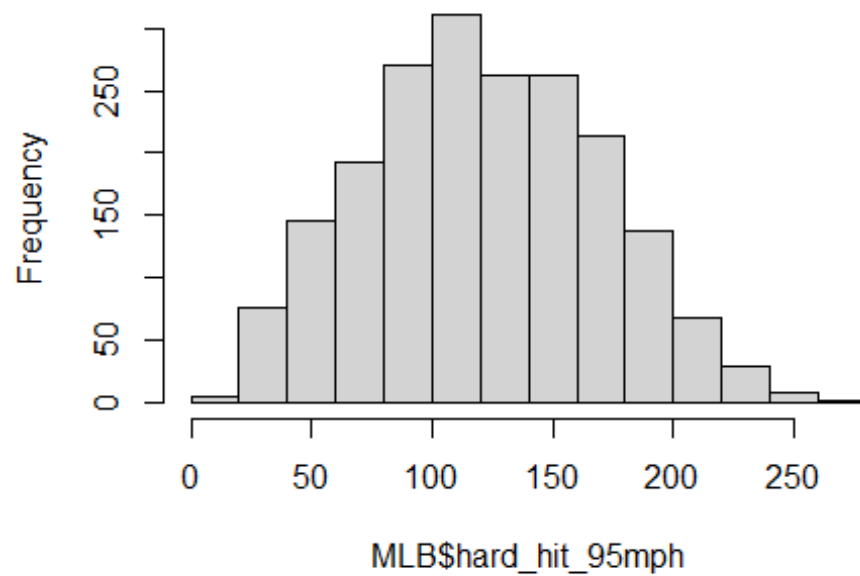
Response: Upon analyzing the QQ plots for these distributions, we can conclude that the Normal and Weibull distributions provide the best fit for this data.

The Normal and Weibull distributions have the best looking Q-Q plots, with most of the points close to the diagonal line. Comparing their AICs, the Weibull emerges as the "best" since its AIC is way more than 4 below the AIC of the Normal fit. Making the histogram and superimposing the curve of the Weibull, it looks alright. One thing I don't like in the QQ plot is the dip below the black line at around 50 and the streak of points slightly above the line at 150. However, since the red dots encompass the black line throughout, we say it's acceptable. The values we see in the data match up reasonably well with what they should be had the data been coming from a Weibull.

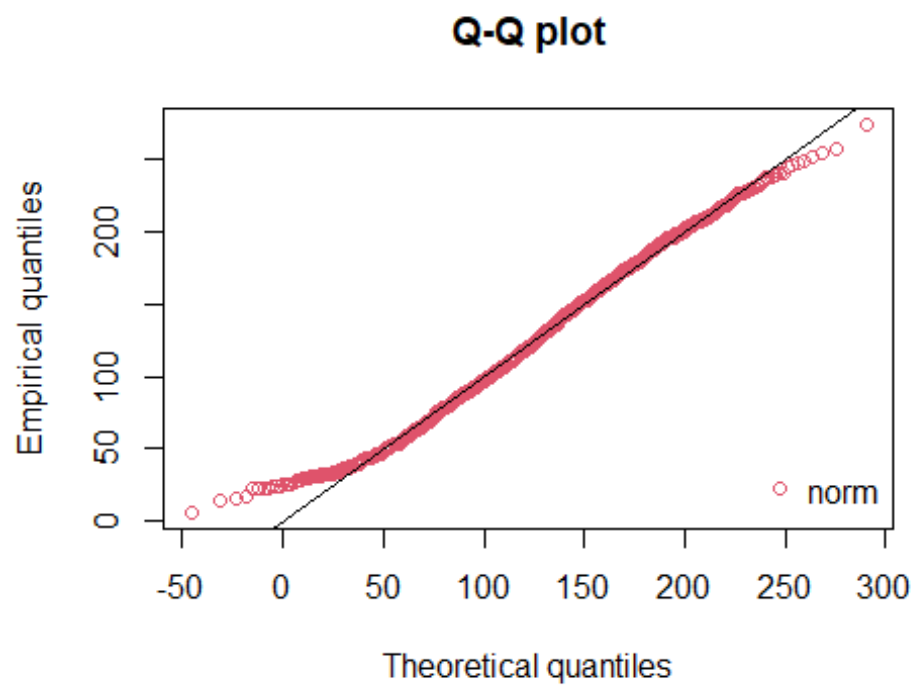
```
# Load necessary Libraries
library(ggplot2)
library(fitdistrplus)

# Histogram
hist(MLB$hard_hit_95mph)
```

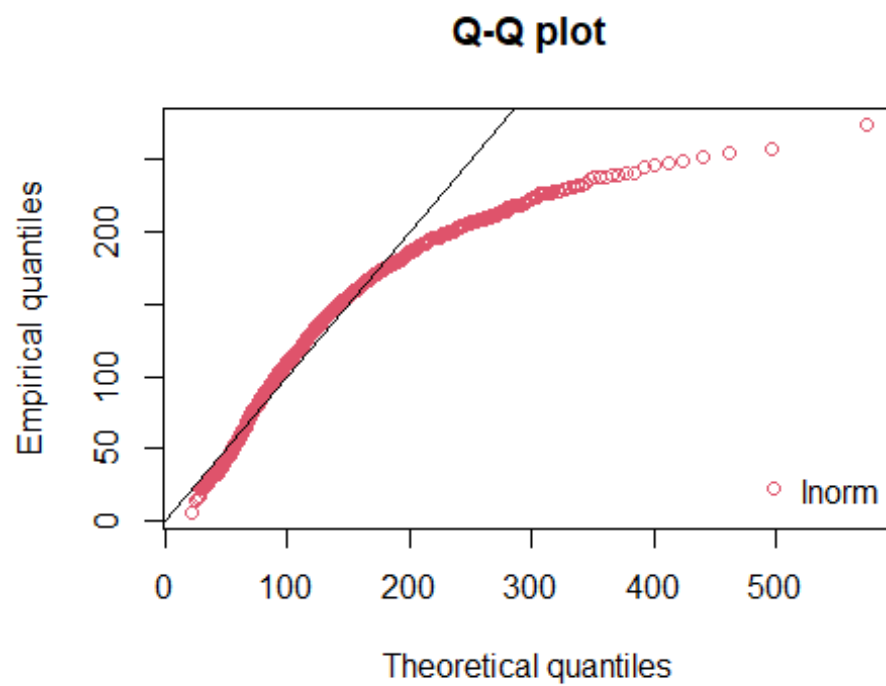
Histogram of MLB\$hard_hit_95mph



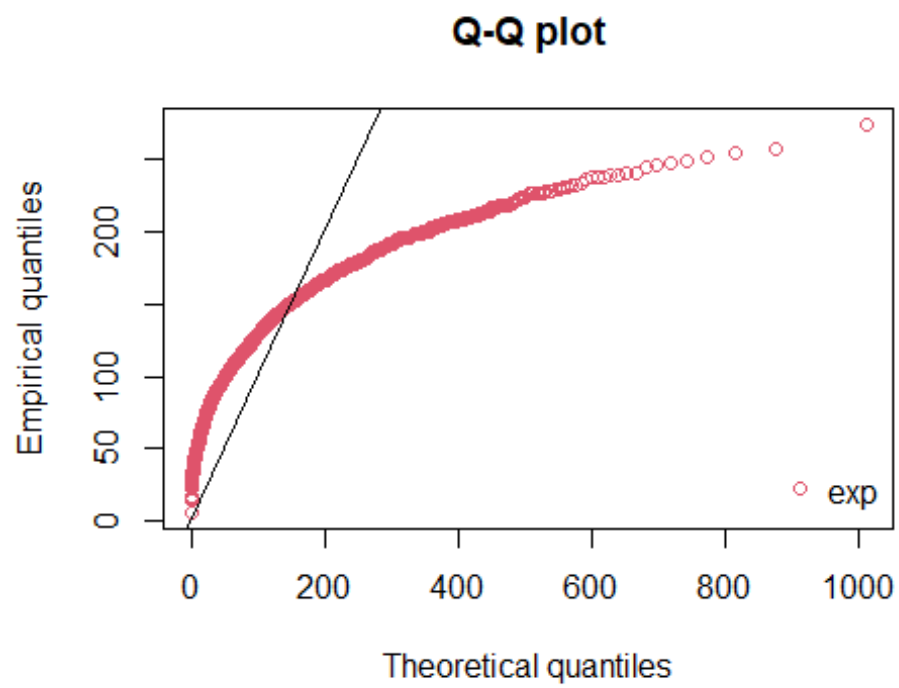
```
# Normal fit
normal_fit <- fitdist(MLB$hard_hit_95mph, "norm")
qqcomp(normal_fit)
```



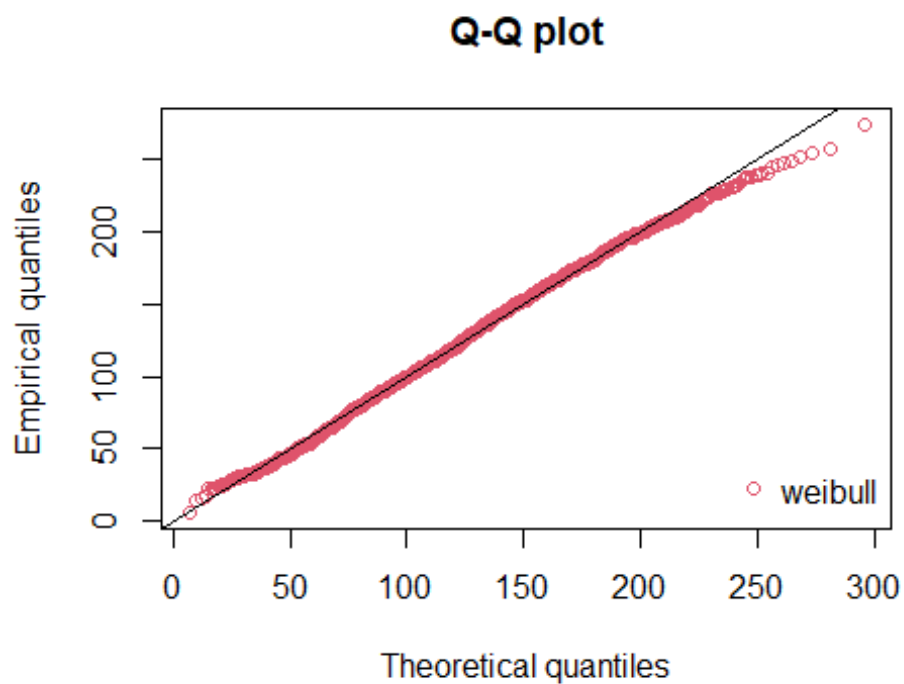
```
# Lognormal fit  
lognormal_fit <- fitdist(MLB$hard_hit_95mph, "lnorm")  
qqcomp(lognormal_fit)
```



```
# Exponential fit  
exponential_fit <- fitdist(MLB$hard_hit_95mph, "exp")  
qqcomp(exponential_fit)
```



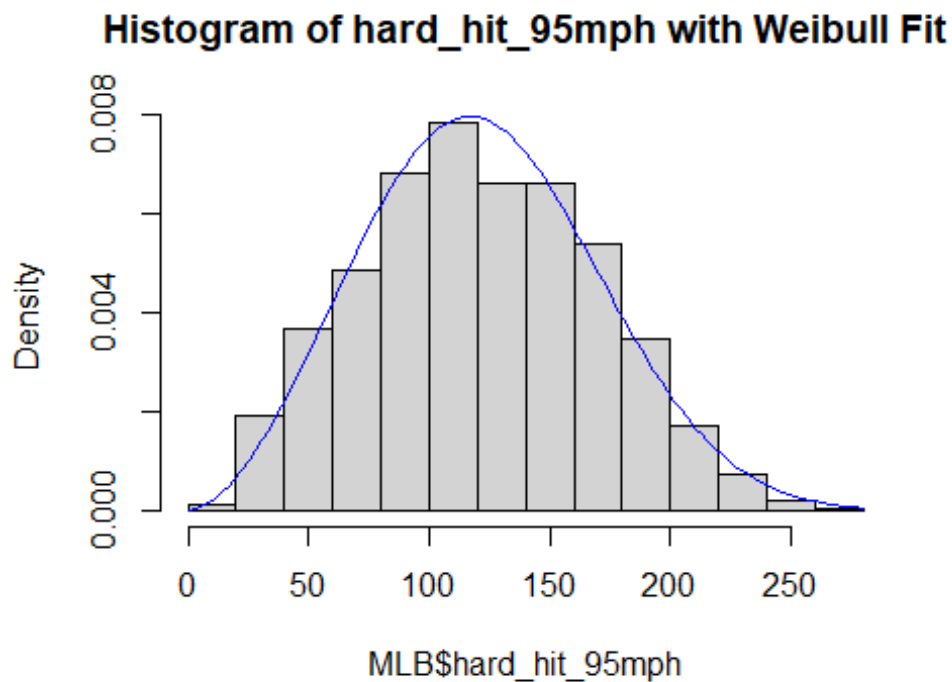
```
# Weibull fit  
weibull_fit <- fitdist(MLB$hard_hit_95mph, "weibull")  
qqcomp(weibull_fit)
```



```
# AIC values
normal_aic <- normal_fit$aic
weibull_aic <- weibull_fit$aic
normal_aic
## [1] 21011.16
weibull_aic
## [1] 20956.93

#Weibull has the lower AIC, so it is a better fit

#histogram
hist(MLB$hard_hit_95mph, freq=FALSE, main="Histogram of hard_hit_95mph with
Weibull Fit")
curve(dweibull(x, shape=weibull_fit$estimate["shape"],
scale=weibull_fit$estimate["scale"]), add=TRUE, col="blue")
```

- b) Consider the acousticness column in SONGS. This gives a number (0-1) quantifying Spotify's "confidence" that the track is acoustic. Many songs blur the line between being acoustic vs. not, so you can think of this as a measure of just how much of the song is acoustic. For example, "Stay" by Rihanna consists of her singing with a piano in the background (it gets a score of 0.945). "Animals" by Martin Garrix (a prime example of "electronic" music) gets a score of 0.00107.

Fit the data to a Normal, Weibull, exponential, and beta distribution. Include a QQ-plot of a "bad" fit and a QQ-plot of the "best" fit. In your response, identify what makes the "bad" fit so bad. Also comment on whether the "best" fit actually provides a reasonable fit to the data (and why).

Response: The best fit here is the beta distribution. Upon examining the QQ-plot for this distribution, we actually see that the data points follow the linear slope quite well. This is a strong indication that the beta distribution provides a reasonable fit to the data.

Normal fit

```
normal_fit_songs <- fitdist(SONGS$acousticness, "norm")
normal_aic_songs <- normal_fit_songs$aic
```

Weibull fit

```
weibull_fit_songs <- fitdist(SONGS$acousticness, "weibull")
weibull_aic_songs <- weibull_fit_songs$aic
```

```

# Exponential fit
exponential_fit_songs <- fitdist(SONGS$acousticness, "exp")
exponential_aic_songs <- exponential_fit_songs$aic

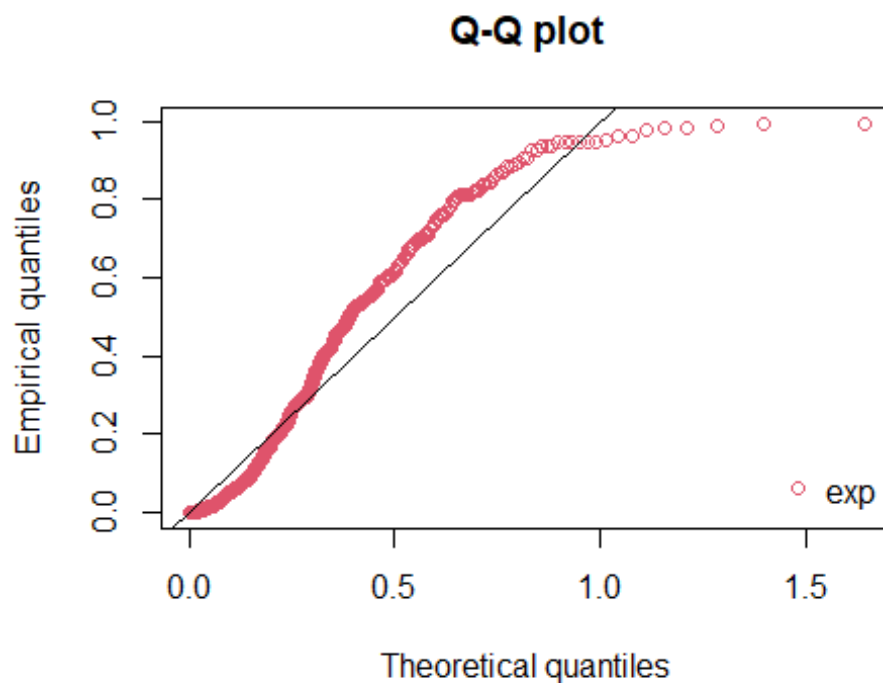
# Beta fit
beta_fit_songs <- fitdist(SONGS$acousticness, "beta")
beta_aic_songs <- beta_fit_songs$aic

# AIC values
normal_aic_songs
## [1] 118.4014
weibull_aic_songs
## [1] -1145.352
exponential_aic_songs
## [1] -864.7132
beta_aic_songs
## [1] -1244.761

#qqcomp(normal_fit_songs)
#qqcomp(weibull_fit_songs)

qqcomp(exponential_fit_songs) #bad fit

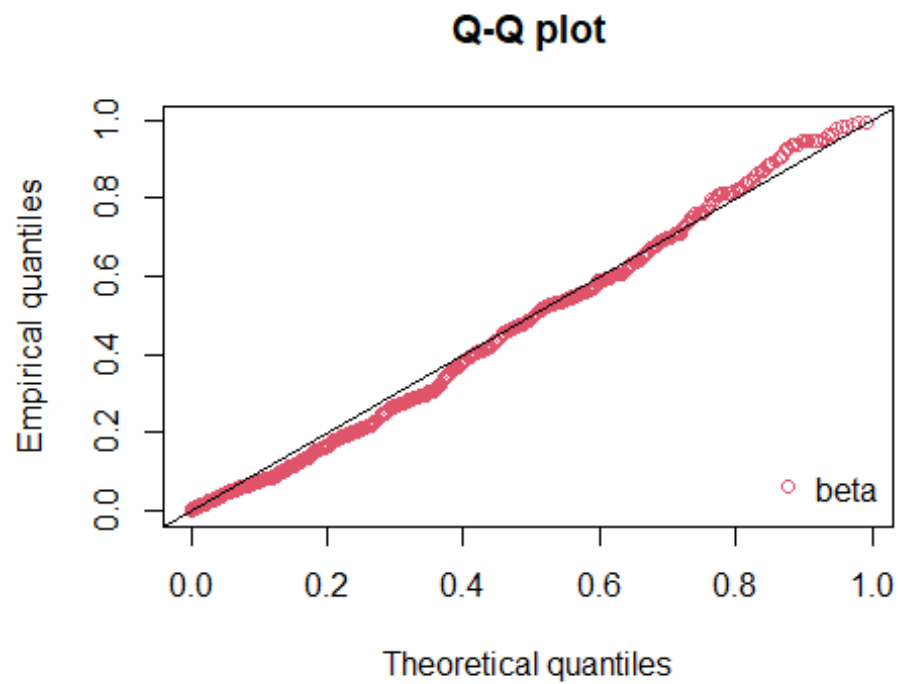
```



```

qqcomp(beta_fit_songs) #best fit

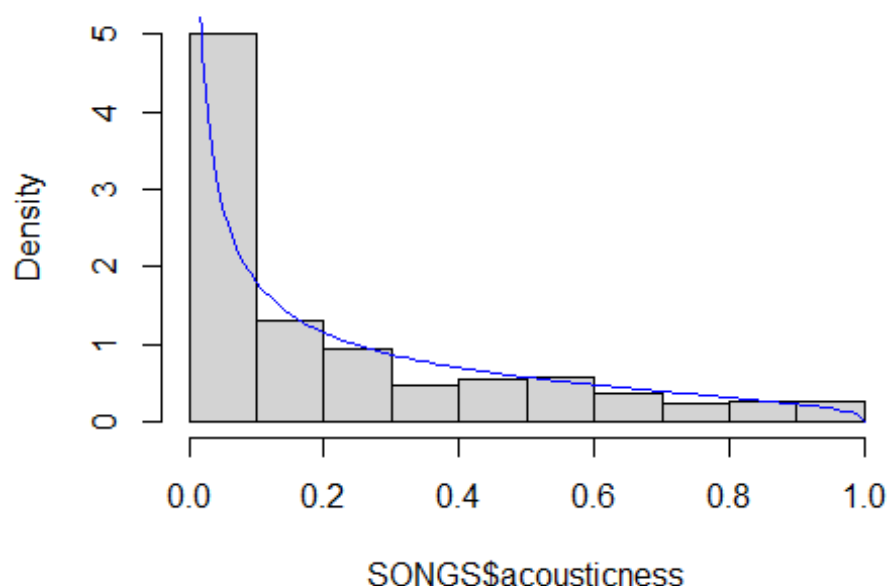
```



```
#best fit (beta) histogram
```

```
hist(SONGS$acousticness, freq=FALSE, main="Histogram of acousticness with  
Beta Fit")  
curve(dbeta(x, shape1=beta_fit_songs$estimate["shape1"],  
shape2=beta_fit_songs$estimate["shape2"]), add=TRUE, col="blue")
```

Histogram of acoustiness with Beta Fit



- c) The `fitdist` command is great, but it can sometimes yield error messages. Try fitting the `launch_angle` column in `MLB` to a lognormal distribution and the `loudness` column in `SONGS` to a Weibull. Both fail. Make a histogram of both quantities (they have straight-forward shapes to model), then explain why `fitdist` isn't able to fit those distributions

Response: Both distributions are designed for positive values only, however the data in both columns contain negative values. This is why we are unable to fit the lognormal distribution to these columns.

#Note: this code/output will not be included in your knitted document and that's fine

```
hist(MLB$launch_angle, main="Histogram of launch_angle")
hist(SONGS$loudness, main="Histogram of loudness")
```