CrpSci545: Statistical Genomics
Homework 1
Matthew McGowan

Problem 1

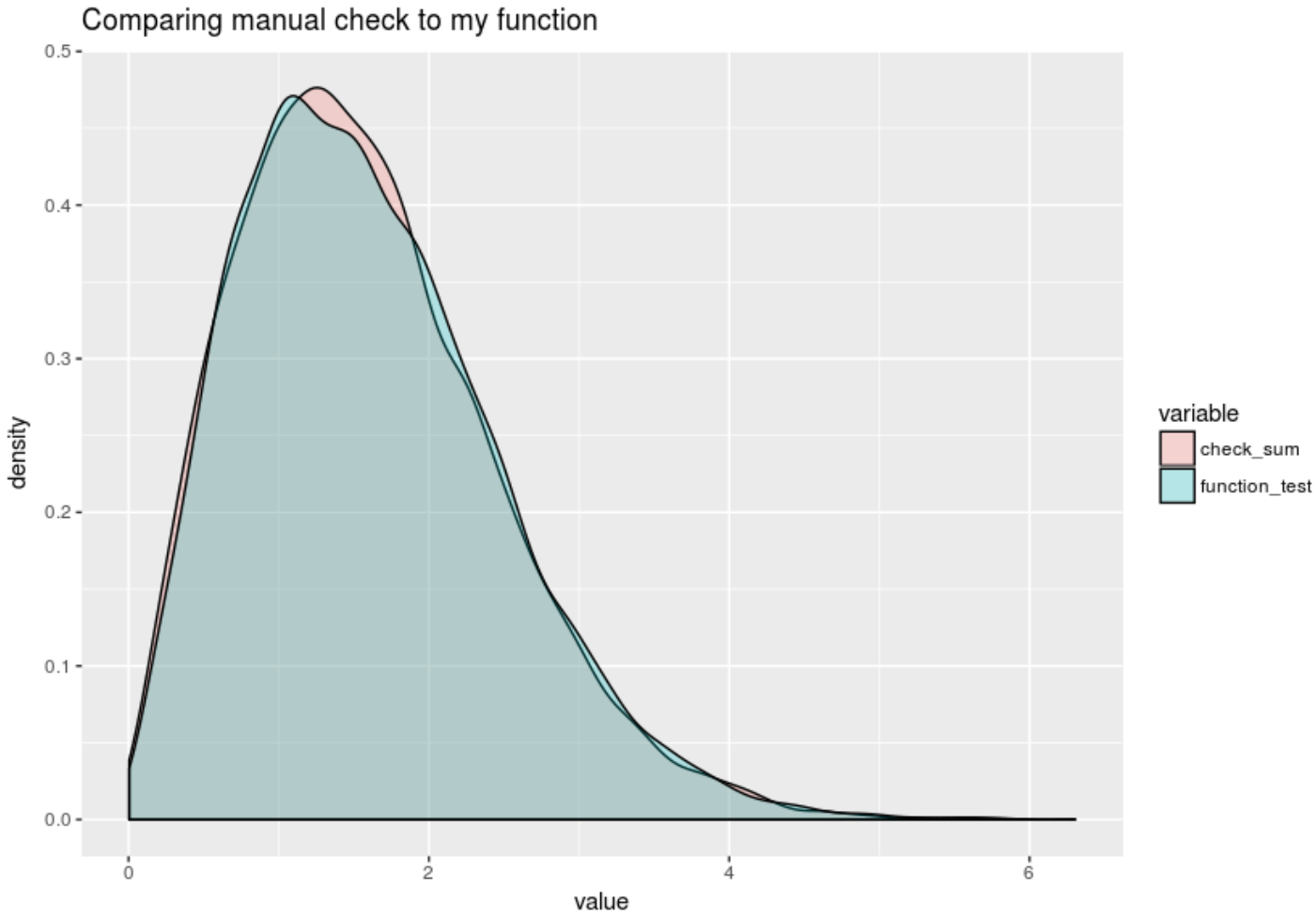
Create multiple random variables that follow the standard normal distribution and combine them to create a new distribution.
Name the random variable as your last name
Include n = the number of variables,

Explanation of the distribution:
For my novel distribution, I wanted to incorporate the absolute value function to effectively 'fold-over' the normal distribution so that all negative values are converted into positive values. Because I discovered that this distribution is already described on wikipedia (folded normal distribution), I wanted to incorporate a second element that calculates the sum distribution of these absolute values after a set number of 'events'. Therefore, I needed to draw multiple normal distributions based on the number of 'events' and sum the values across each distribution.

The input of this R function is:

- **n = the sample size of the final distribution.**
- **v = the number of 'events', which is also the number of normal distributions to sum together.**
- **f = the mean value for each normal distributions drawn.**

To verify that this function is working correctly, I manually simulated the same situation by independently creating two standard normal distributions, applying the absolute value function to them and summing them together. Then I plotted the cumulative density of the check against my function with the same properties. My sample size for this test was 10,000.



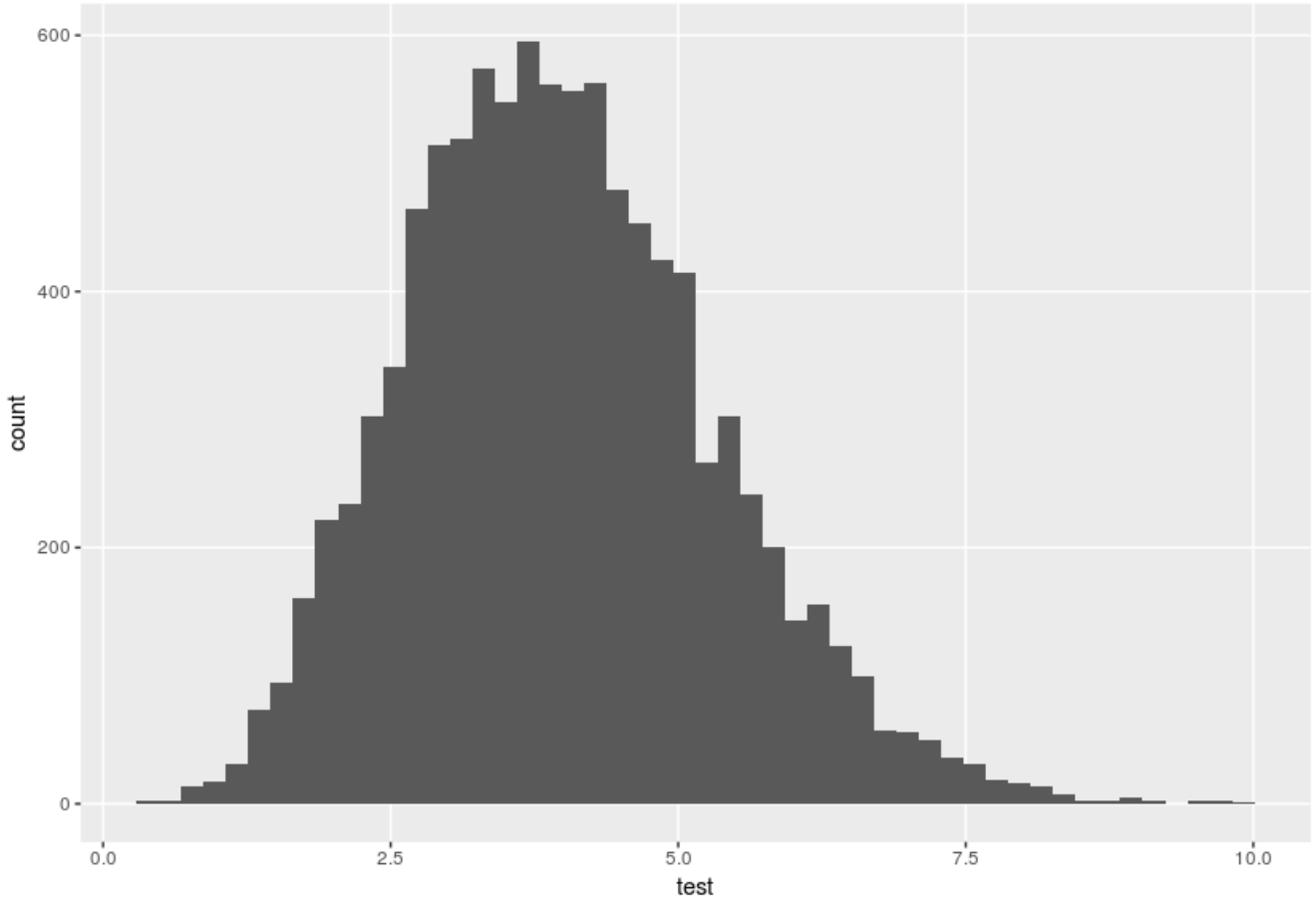
The two distributions almost perfect overlap, suggesting that my function is working correctly.

Problem 2

Sample 10,000 observations from the distribution you defined. Graph their properties and describe the potential application of your distribution in nature.

test <- mcgowan(10000, 5, 0)

I plotted my custom distribution (n = 10000, v = 5, f = 0).



This distribution could have potential application to situations where a polarized input (either positive or negative) pulse is always converted into a positive response (thus the use of the 'abs' function). While imperfect, one real-world example of this could be for modeling damage caused by folding a piece of metal back and forth. Regardless of whether the metal is folded forward or backwards, the metal is weakened according to the sum of the distance it has been folded. Therefore, if each fold follows a normal distribution, by knowing how much damage required to break it, one could test how many times it would have to be folded to break within a specified confidence interval.

Problem 3

Develop a custom function that creates an F distribution with n,df1,df2 arguments using the normal distribution function built into R.

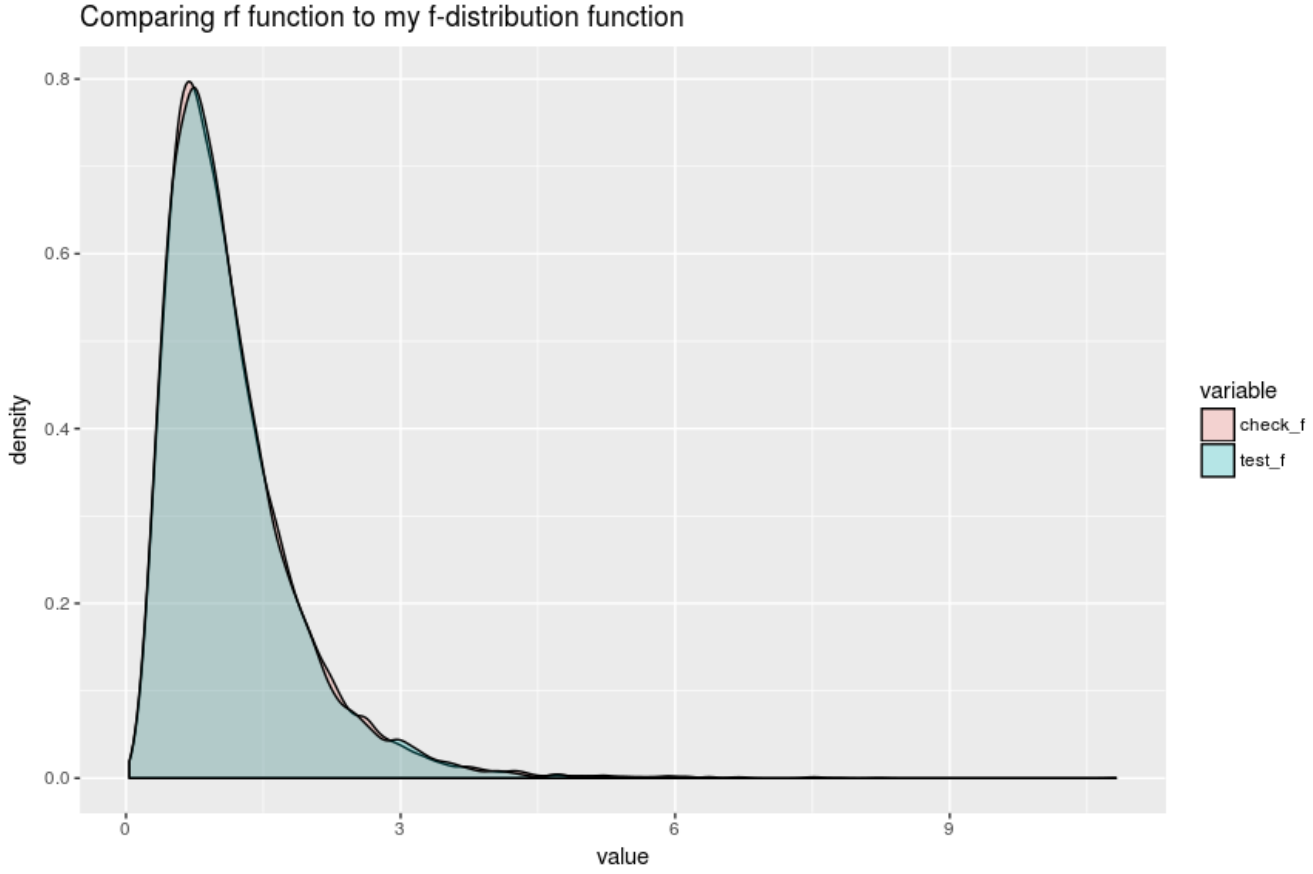
My function has the same arguments as the built in function.

- **n = the sample number for the final distribution**
- **df1 = the degrees of freedom in the first chi-square distribution**
- **df2 = the degrees of freedom in the second chi-square distribution**

Methods:

- **First I created two chi-squared distributions from df1 and df2 numbers of normal distributions with sample size n.**
- **Because a chi-squared distribution is the sum of df number of squared normal distributions, I created the two chi-squared distributions by calculating the sum squared value of df1 or df2 normal distributions with a sample size of n.**
- **Then I calculated the final f-distribution from the two chi-square distributions using the equation for a f-distribution (chi1_value / df1) / (chi2_value / df2).**

I then verified my function against the built-in t-distribution function by plotting superimposed density distributions (n = 10000, df1 = 10, df2 = 15).



Problem 4

Sample 10, 100, 1000, and 100,000 F distributed variables using the custom function. Calculate means and test them against the null hypothesis that samples have expected mean of df2/(df2-2)

Because the problem did not stipulate what df1 and df2 parameters should be, I decided to set my df1 to 100 and my df2 to 150.

I then created a loop that will create my 4 f-distributions for the different n values, calculated the mean and variance of them and then performed a t-test. The means, variances, t-values, and p-values for these tests were then stored in a data table.

Here are my t.test results for my 4 variables:

nsamp	variance	mean	t-value	p-value1
10	1.023579	0.009029978	0.3349647	0.7453266
100	1.021713	0.034186734	0.4434502	0.6584078
1000	1.008422	0.034426407	-0.8678348	0.3856932
10000	1.015199	0.034386883	0.9089842	0.3633804
500000	1.013327	0.034825155	-0.7080294	0.4789273

These test results indicate that I should accept the null hypothesis that my distributions are not significantly different from a distribution with a mean of `df2/(df2-2).`

Problem 5

Sample 10, 100, 1000, and 100,000 F distributed variables using the R function previously developed.

This was done already for problem 4.

Calculate the variance of these samples and test them on the null hypothesis that the samples have expected variance of (2n[2]^2(n[1] + n[2] - 2))/(n[1](n[2]-2)^2 * (n[2]-4))

For this test, I created a function with two parameters:

- **dist = the distribution to test**
- **var_expect = an expected variance (in this case, the variance calculated with the provided equation)**

I performed this test on each of my f-distributions and added the test value and p-value to the the table. Here are the results:

nsamp	variance	mean	t-value	p-value1	chi-sq	p-value2
10	1.023579	0.009029978	0.3349647	0.7453266	2.328849	0.9851377
100	1.021713	0.034186734	0.4434502	0.6584078	96.985075	0.5385162
1000	1.008422	0.034426407	-0.8678348	0.3856932	985.528716	0.6132543
10000	1.015199	0.034386883	0.9089842	0.3633804	9852.840874	0.8494114
500000	1.013327	0.034825155	-0.7080294	0.4789273	500623.935795	0.2658730

Based on these results and assuming a threshold cutoff at 5%, I can conclude that my distributions have variances that do not significantly differ from the expected variance of an f-distribution.