HW3 Solutions

J. Lucas McKay

1/30/2023

---

1. Suppose that `X` is a discrete random variable with `P(X = 0) = 0.25`, `P(X = 1) = 0.125`, `P(X = 2) = 0.125`, and `P(X = 3) = 0.5`. Use `ggplot2` to graph the frequency function *(1 point)* and the cumulative distribution function *(1 point)* of `X`.

---

2. The following table shows the cumulative distribution of a discrete random variable `x`. Calculate and graph the frequency function. *(1 point)* (NB: try `diff`)

X	Cumulative Frequency
0	0
1	20
2	60
3	140
4	160
5	200

---

3. The Boolean expression `x==y` does not work well for floating-point numbers in that rounding errors may produce a FALSE answer. Compare it with the functions `all.equal` and `identical`.

- Examine the behavior when you compare the approximation to π `pi_user = 3.14159` to the constant `pi` built in to R using the `identical` function. *(1/3 point)*
- Examine the behavior when you use the `all.equal` function. *(1/3 point)*
- Examine the behavior when you use the `all.equal` function, specifying the tolerance as `1e-5`. *(1/3 point)*

---

4. Show that, when `a` is a scalar and `x` is a vector (say, `1:12`), `match(a,x)` is equivalent to `min(which(x == a))`. *(1/3 point)* Then try the infix operator `%in%`. What happens when you say `x %in% 4`? *(1/3 point)* How about `x %in% c(5,10)`? *(1/3 point)*

---

5. We did an earlier problem to show that approximately normally distributed numbers could be produced from those on the unit interval. A more modern transformation is the Box-Muller transformation. The Box-Muller transformation takes two samples from the uniform distribution on the interval [0, 1] and maps them to two standard, normally distributed samples. Create a function `BoxMuller(n)` that creates `n` samples

from the standard normal distribution, *(1/2 point)* and show that `mean(BoxMuller(500)) == 0` and `var(BoxMuller(500)) == 1`. *(1/2 point)*

---

6. Use simulation and `ggplot2` (with `geom_density`) to plot the probability density function of the output of your `BoxMuller()` function. *(1/2 point)* (Use, say, 10,000 random samples.) Plot the output of `rnorm()` with the same sample size for comparison.*(1/2 point)*

- *EXTRA 1 POINT:* Plot both of these on the same graph. (You may have to `pivot_longer`, which we will see soon.)

---

7. Use simulation and `ggplot2` to plot the cumulative distribution function of the output of your `BoxMuller()` function. *(1 point)*

---

8. There are many different approximations for the normal distribution. Create a function `Unif12(n)` that generates 12 uniform random variables on the interval [-0.5, 0.5], and calculates their sum. Compare the densities obtained with `rnorm(n)`, `BoxMuller(n)`, *(1/2 point)* and `Unif12(n)`. *(1/2 point)*

---

9. Plot a histogram of Z^2, where $Z \approx N(0, 1)$ This is the chi-squared distribution with one degree of freedom. Compare this to a plot using `geom_density()`. What do you notice about the lower limits? *(1/2 point)* What happens when you modulate the parameter `adjust` in geom_density? *(1/2 point)*

---

10. The function `ecdf()` produces an "empirical" cumulative distribution function. Simulate 200 samples of data on $N(0, 1)$, and compare the results of `ecdf()` to `pnorm()`. *(1/2 point)* Then compare when you simulate 10,000 samples to start with. *(1/2 point)*