ES 220 Lab 2

Linda Dominguez

Jan 31, 2024

(1a) How much usable wood (m^3) is available on Water Tower Hill right now?

There is between 25,402,976,396 m^3 and 37,023,585,129 m^3 of usable wood is available on Water Tower Hill right now.

```
# After every calculation, be sure to show the resulting low and high values
# e.g. print(My_Variable) or round(My_Variable)
```

#volume of hole code:

volumeHole<- function(a,b,c){ (((4*pi*) ((((a*b)$^{1.6)+((ac)\hat{}1.6)+\ ((bc)}$1.6)) /3)^(1/1.6))/2) } #this formula divides by 2 bc we only have half of the surface area for the ellipsoid.

volumeHigh <- volumeHole(405, 220, 48) volumeLow <- volumeHole(390,200,20) print(paste("Volume of the hole high:", round(volumeHigh), "meters cubed.")) print(paste("Volume of the hole low:", round(volumeLow), "meters cubed."))

#surface of tress code: aka number of trees surfaceTrees <- function(a, b) (a*b*pi) surfaceMax = surfaceTrees(405, 220) surfaceMin = surfaceTrees(390,200)

print(paste("Surface trees max", round(surfaceMax), "meters squared.")) print(paste("Surface trees min", round(surfaceMin), "meters squared."))

#ratio of amount of trees surfaceTreesPerMeter <- function(l, w, ratio) pi*ratio*l*w treesPerMeterMax <- surfaceTreesPerMeter(405, 220, .3) treesPerMeterMin <- surfaceTreesPerMeter(390,200, .6)

print(paste("Amount of trees max:", round(treesPerMeterMax), "trees")) print(paste("Amount of trees min:", round(treesPerMeterMin), "trees"))

#wood usable

#use prior functions for volume and surface and multiply them woodUsable <- function(l, w, h, ratio)volumeHole(l,w,h)*surfaceTreesPerMeter(l,w, ratio) woodUsableMax <- woodUsable(405, 220, 48, .3) woodUsableMin <- woodUsable(390,200, 20, .6)

print(paste("Wood usable max:", round(woodUsableMax), "trees in m^3")) print(paste("Wood usable min:", round(woodUsableMin), "trees in m^3"))

print(paste("There is between", round(woodUsableMax), 'and', round(woodUsableMin), "usable wood (m^3) is available on Water Tower Hill right now."))

#max length * max width * pi for surface

(1b) How much sand (m^3) is available to excavate from Water Tower Hill?

There's between 27.72 m^3 and 83.15 m^3 of sand available to excavate from Water Tower Hill.

```
# After every calculation, be sure to show the resulting low and high values
# e.g. print(My_Variable) or round(My_Variable)
```

#volume of hole code: #From before volume of the hole function: surfaceHole<- function(a,b,c){ (((4*pi)* ((((a*b)$^{1.6)+((ac)\hat{}1.6)+ ((bc)}$1.6)) /3)^(1/1.6))/2) }

surfaceHigh <- surfaceHole(405, 220, 48) surfaceLow <- surfaceHole(390,200,20)

print(paste("surface of the hole high:", round(surfaceHigh), "meters cubed.")) print(paste("surface of the hole low:", round(surfaceLow), "meters cubed."))

averageSurface <- (surfaceHigh+surfaceLow)/2

print(round(averageSurface)) # (range: (251816), 302507 and average = 277,161)

#now that we have our range of the estimated surface area, take the thickness of the sand and multiply # I found the type of sand usually found in MA; and found thickness => 0.1-0.3 mm totalSand<- function(surface, thickness){ surface * (thickness/1000) }

sandVolumeHigh = round(totalSand(averageSurface, .3), 2)

sandVolumeLow = round(totalSand(averageSurface, .1), 2)

print(paste("Sand volume ranges:", sandVolumeLow,",", sandVolumeHigh))

(1c) How many Deluxe Zen Gardens could you make from resources extracted from Water Tower Hill?

Between 25,402,976,396 m^3 and 37,023,585,129 m^3 of usable wood is available on Water Tower Hill right now. We need to consider the amount of wood needed for one box(which we estimated would be .000702m^2/box) and the of sand needed for one box, (which we estimated would be .0010764 m^2). I then wrote the function totalGardens, which adds one garden until the sand or wood quantity run out. Finally, I averaged everything (min and max for all measurements) to give the best estimate. I added the min and max and divided by two for all measurements, which gave me 51,498 or nearly 51,500 gardens.

FINAL: 51,498 gardens

```
# After every calculation, be sure to show the resulting low and high values
# e.g. print(My_Variable) or round(My_Variable)
```

#sand use calculated to be .0010764 m #wood use calculated to be .000702 m

totalGardens <- function(wood, sand) { count <- 0

while (sand > 0 & wood > 0) { count <- count + 1 sand <- sand - 0.0010764 wood <- wood - 0.000702 } print("The expected number of zen gardens:") return(count) }

avgsand <-.2 woodFinal <- (woodUsableMax+woodUsableMin)/2 sandFinal<- (totalSand(averageSurface, avgsand)) allGardens<- totalGardens(woodFinal, sandFinal)

print (allGardens) ***

(2a) What is the expected population size after 42 years given a starting population of 5,700 and a constant annual growth 2.3%?

The expected population 42 year(s) after is 244,906.

```
#The expected population 42 year(s) after is 244,906.
```

popGrowth <- function(yearToCalculate){ startPop <- 5700 rate <- 1.023 expectedPop <- startPop * rate * yearToCalculate print(paste("Expected pop", yearToCalculate, "year(s) after:", round(expectedPop))) return (expectedPop) }

#to find the population after 42 years, insert 42 into our function:

growth42 <- popGrowth(42) print(growth42)

(2b) What per capita birth rate would be required to achieve a lambda value of 1.022, given a per capita death rate of 0.08?

```
# After every calculation, be sure to show the resulting low and high values
# e.g. print(My_Variable) or round(My_Variable)
```

capitaBirthRate<- function(lambda, deathRate){ return (lambda + deathRate) }

desiredLambda <- 1.022 deathRate <- .08

requiredBirthRate <- capitaBirthRate(desiredLambda, deathRate) print(requiredBirthRate) ### (2c) If human birth and death rates remain constant (assume average values from 2020), what would the total world population be in the year 2100? With a steady rate of births and deaths, the estimated world population in 2100 is 203.3306 billion people.

```
# After every calculation, be sure to show the resulting low and high values
# e.g. print(My_Variable) or round(My_Variable)
```

estimatedWorldPop<- function(year, birthrate, deathrate){ year <- year-2020

initialPop <- 7830000000 differencePop <- birthrate-deathrate print("estimated population in billions:") return ((initialPop * (1+ differencePop)^year)/100000000) }

birthRate <- .0194 deathRate <- .0074

estimateFuturePop <- estimatedWorldPop(2100, birthRate, deathRate) print(estimateFuturePop)

---

(3) How many people have ever existed?

Take average life expectancy throughout time Throughout history, average couple has 5 children. The average generation is 25 years. Each person produces 2.5 people 25 years later. Divide 200000/25 to get number of generations and multiply by 2.5. Get a final rough estimate of 79,991,000 people. This doesn't account for different growth rates.

totalGenerations <- 200000/25 print(totalGenerations) 100 people is initial pop at

startingPop + 2.5(popAll) 100+2.5(year measuring)

#how many years have ppl been around => 200,000 #take average num of ppl born per year =>

estimatedWorldPop<- function(){ total<- 1000 totalGenerations <- 200000/25 count <- 0 while (count < totalGenerations) { popThisYear <- 2.5*(count) total <- total + popThisYear count <- count + 1

} return (total) }

allPop <- estimatedWorldPop() print(allPop)