Code

Stats 20, F23 – Homework 1

Jacqueline Nguyen – 606069033

04/06/24

Question 1

(a)

"**Sometimes you have to make the hardest climb to see the most beautiful sunrise.** I read that once on an old lady's decorative pillow, but it is really how I feel today. I've climbed a very weird and rocky mountain, and it was a pain in the ass, and my legs are tired, and I'm starving. But *the sun is rising over a sea of love and waffles and possibility. So I'm just gonna relax and take a deep breath and enjoy this view for as long as I possibly can."* – Leslie Knope (from ***Parks and Recreation***)

(b)

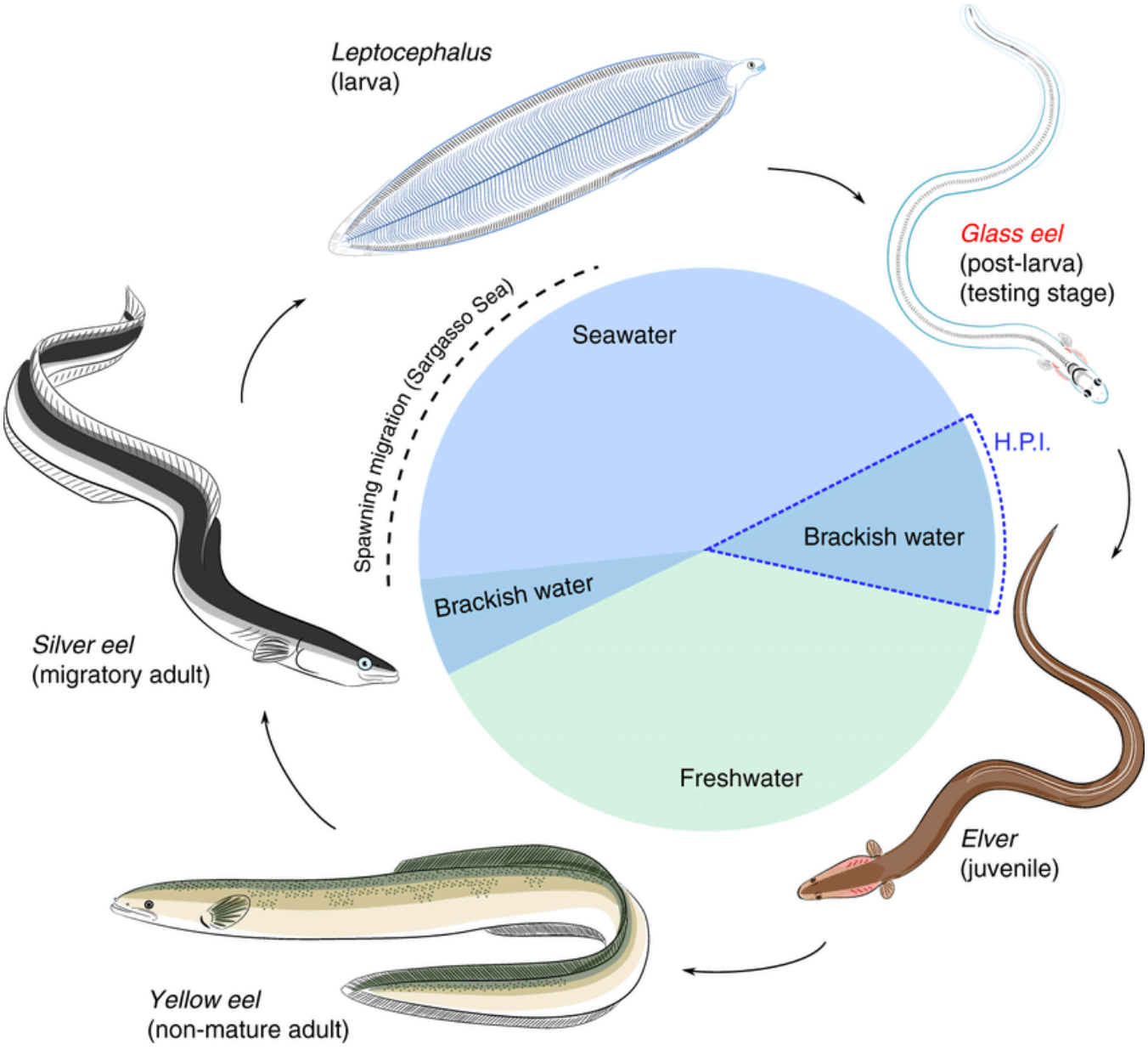
Homework	Midterm	Final Exam	Final Project
18%	30%	40%	12%

(c)

- 2022
  - Fall
    - math 31b
    - cs 31
    - cs 1: seminar
  - Winter
    - cs 32
    - math 32a
    - math 33a
    - art and arch

- Spring
  - cs 33
  - math 32b
  - math 33a

(d)



Question 3

(a)

Hide

```
(6^8)/(16^5)
```

```
## [1] 1.601807
```

Hide

```
101^(-8/7)
```

```
## [1] 0.005120908
```

Hide

```
8 + (3*pi) - 3 + (4/2)*(-1)
```

```
## [1] 12.42478
```

i. 1.601807
ii. 0.005120908
iii. 12.42478 ### (b)

Hide

```
log(54,base=3)
```

```
## [1] 3.63093
```

Hide

```
exp(1)^(exp(1)^(sqrt(2)))
```

```
## [1] 61.14514
```

Hide

```
(exp(1)^pi + exp(1)^-pi)/2
```

```
## [1] 11.59195
```

i. 3.63093
ii. 61.14514
iii. 11.59195

(c)

Hide

```
119/25
```

```
## [1] 4.76
```

Hide

```
119%%25
```

```
## [1] 19
```

i. 4.76

ii. 19

Question 4

(a)

Hide

```
vol_1 <- (4/3)*pi*1^3
vol_4 <- (4/3)*pi*4^3

vol_1
```

```
## [1] 4.18879
```

Hide

```
vol_4
```

```
## [1] 268.0826
```

i. 4.18879

ii. 268.0826

(b)

Hide

```
vol_sphere <- function (r=1) {
  vol<-(4/3)*pi*r^3
  vol
}
```

(c)

Hide

```
vol_sphere()
```

```
## [1] 4.18879
```

Hide

```
vol_sphere(4)
```

```
## [1] 268.0826
```

i. 4.18879

ii. 268.0826

Question 5

(a)

Hide

```
z_prop <- function(x,n,p0) {
  p_hat <- x/n
  denom <- sqrt((p0*(1-p0))/n)
  z <- (p_hat - p0)/denom
  z
}
```

(b)

Hide

```
z_prop(10,13,.6)
```

```
## [1] 1.245505
```

(c)

the z-statistic is 1.245505 which indicates that the trial is within 1.245505 sd of the mean of other data points. Since it is less than two, we can assume that this is not an extreme outlier, though it is slightly above the mean.

(d)

Hide

```
z_prop(30,39,.6)
```

```
## [1] 2.157277
```

despite the ratio of the prawnee residents and the los angeles residents being proportional, the sample size is larger allowing for a lower standard deviation. This means that since the standard score is the same, but the standard error is lower for prawnee, the z-statistic for prawnee is greater.

Question 6

(a)

Hide

```
lease_calc<- function(msrp, price,down,n=36,res=.6, mf=.001,tax=.095) {

  capitalized_cost <- price - down
  residual_value <- msrp * res
  monthly_depreciation <- (capitalized_cost - residual_value)/n
  monthly_finance_charge <- (capitalized_cost + residual_value)*mf
  subtotal <- monthly_depreciation + monthly_finance_charge
  total <- subtotal + tax

}
```

(b)

Hide

```
result <- lease_calc(msrp = 31495, price = 29895, down = 2500, n = 36, res = 0.52,
mf = 0.0016, tax = 0.095)
print(result)
```

```
## [1] 376.1753
```

Question 7

(a)

the minimum amount of coins that can be used to make 47 cents is 5. The process in which I used to come up with this answer is looking at the largest possible coin which as 25 cents. We can use and subtract as 1 time from 47 before exceeding 47 cents. Then I repeat it with dimes which allows me to subtract 2 dimes from the total. I cannot subtract any nickels but can subtract 2 pennies. Using this process, I get 1 quarter, 2 dimes, and 2 pennies to make 47 cents.

(b)

The process in which I used to come up with this answer is looking at the largest possible coin value we can use and subtract as many times from the calue before exceeding the number. Then I repeat it with the second largest coin, 3rd largest, and fourth largest.

(c)

Hide

```
get_minimum_coins <- function(coins) {
  coin_count <- 0

  while(coins >= 25) {
    coin_count <- coin_count + 1
    coins <- coins - 25
  }
  while(coins >= 10) {
    coin_count <- coin_count + 1
    coins <- coins - 10
  }
  while(coins >= 5) {
    coin_count <- coin_count + 1
    coins <- coins - 5
  }
  while(coins > 0) {
    coin_count <- coin_count + 1
    coins <- coins - 1
  }
  coin_count
}
```

(d)

Hide

```
get_minimum_coins(31)
```

```
## [1] 3
```

Hide

```
get_minimum_coins(48)
```

```
## [1] 6
```

Hide

```
get_minimum_coins(139)
```

```
## [1] 10
```

i. 3

ii. 6

iii. 10

(e)

Hide

```
get_minimum_coins(94)
```

```
## [1] 9
```

Hide

```
get_minimum_coins(99)
```

```
## [1] 9
```

Either 99 or 94. We want to maximize the amount of quarters, dimes, nickels, and pennies that can be used. The most amount of quarters we can have is 3 which makes 75. Then we can split into two different ways of adding up to get 9 coins total. To get 9 coins from 99, we look at the maximum number of dimes we can make given than we have 24 coins left. That gives us two dimes, the remaining coins are used in pennies. To get to 94, we can use 1 dime, 1 nickel, and four pennies.

Question 8

Hide

```
x <- 3
y <- 4
exp1 <- function() {
  x^y
}
exp2 <- function(x = 2) {
  x^y
}
exp_y <- function(x) {
  function(y) {
    x^y
  }
}
exp1()
```

```
## [1] 81
```

Hide

```
exp2()
```

```
## [1] 16
```

Hide

```
exp_y()
```

```
## function(y) {
##     x^y
##   }
## <environment: 0x116080c50>
```

(a)

The reason why exp1 and exp2 output different answers is because of the fact that exp2 has a local default x value which takes precedent over the gobal x variable. ### (b) the command doesn't throw an error because it does not make any calculations, it instead calls another function inside of it to make a calculation. The function calls another function which is supposed to take in a y value which is not provided which is why it does not calculate any values and output a number. Instead of providing a number as an output, it provides the content of the function. ### (c)

Hide

```
exp_y(4)(15)
```

```
## [1] 1073741824
```