BEE 550 Intro to R

Jennifer Bradley

09-24-2024

Q1

47

```
12 * 3 - 10 / 2 + 16
```

```
## [1] 47
```

Q2

-8

```
12 * (3 - 10 / 2) + 16
```

```
## [1] -8
```

Q3

4

```
2 ^ 3 - sqrt(16) + log(1)
```

```
## [1] 4
```

Q4

47

```
x = 12
y = 3
z = 10
w = 2
t = 16

x * y - z / w + t
```

```
## [1] 47
```

Q5

The result of exp(log(x)) is always x.

```
x = 6

exp(log(x))
```

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

Q6

The area is 6.

```
tri_area = function(L1, L2, L3){
  p = (L1 + L2 + L3) / 2
  area = sqrt(p * (p - L1) * (p - L2) * (p - L3))
  return(area)
}

tri_area(3, 4, 5)
```

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

Q7

-2

```
v = c(3, 5, -2, 0, 1)

v[c(3)]
```

```
## [1] -2
```

Q8

v[2:4] lists the 2nd, 3rd, and 4th elements of vector v. It seems that the : operator represents "until" or "through." In v[x:y] all elements in the vector v starting with the x'th element until (and including) the y'th element are pulled.

```
v[2:4]
```

```
## [1]  5 -2  0
```

Q9

The output of mean(v1 + v2) is -1.2 The output of sum(v1 + v2) / length(v1 + v2) is the same, -1.2

```r
v1 = v
v2 = c(-5, -1, -9, 2, 0)
print(mean(v1 + v2))
```

```
## [1] -1.2
```

```r
print(sum(v1 + v2) / length(v1 + v2))
```

```
## [1] -1.2
```

Q10

```r
my_vector = mean(v1 + v2)

sort(my_vector)
```

```
## [1] -1.2
```

Q11

```r
q11_matrix = matrix(c(4, -2, 0, 1, 1, 7), nrow = 3, ncol = 2)
q11_matrix
```

```
##      [,1] [,2]
## [1,]    4    1
## [2,]   -2    1
## [3,]    0    7
```

Q12

The str() function prints the data type of the object fed to the function, and displays the object itself.

```r
my_dogs_age = 3.5
my_dogs_name = 'Hobbes'
my_cats_name = NA
is_my_dog_a_puppy = my_dogs_age < 1

mda = str(my_dogs_age)
```

```
##  num 3.5
```

```r
mdn = str(my_dogs_name)
```

```
##  chr "Hobbes"
```

```r
mcn = str(my_cats_name)
```

```
##  logi NA
```

```r
imdap = str(is_my_dog_a_puppy)
```

```
##  logi FALSE
```

Q13

The command is attempting to multiply a non-numeric data type (string) to a numeric type (integer). As this is not possible, R gives the error.

```r
my_pet_info =
  list(
    my_dogs_age,
    my_dogs_name,
    my_cats_name,
    is_my_dog_a_puppy
  )

my_dog_info =
  list(
    age = my_dogs_age,
    name = my_dogs_name,
    puppy = is_my_dog_a_puppy
  )

2 * my_dog_info$name
```

```
## Error in 2 * my_dog_info$name: non-numeric argument to binary operator
```

Q14

20.09062 This is the average of all mpg values in the data frame; the average mpg of the cars.

```r
mean(mtcars$mpg)
```

```
## [1] 20.09062
```

Q15

```r
var(mtcars$cyl)
```

```
## [1] 3.189516
```

Q16

```
characters |>
  arrange(characters, year_created)
```

```
## # A tibble: 7 x 4
##   name         species comic_strip year_created
##   <chr>        <chr>   <lgl>              <dbl>
## 1 Bugs Bunny   rabbit  FALSE               1940
## 2 Calvin       human   TRUE                1985
## 3 Garfield     cat     TRUE                1978
## 4 Hobbes       tiger   TRUE                1985
## 5 Lisa Simpson human   FALSE               1987
## 6 Popeye       human   TRUE                1929
## 7 Woodstock    canary  TRUE                1967
```

Q17

```
my_tibble =
  mtcars |>
  rownames_to_column(var = 'model') |>
  as_tibble() |>
  select(model, mpg, cyl, hp) |>
  mutate(kml = mpg * 0.43) |>
  filter(mpg > 22)
my_tibble
```

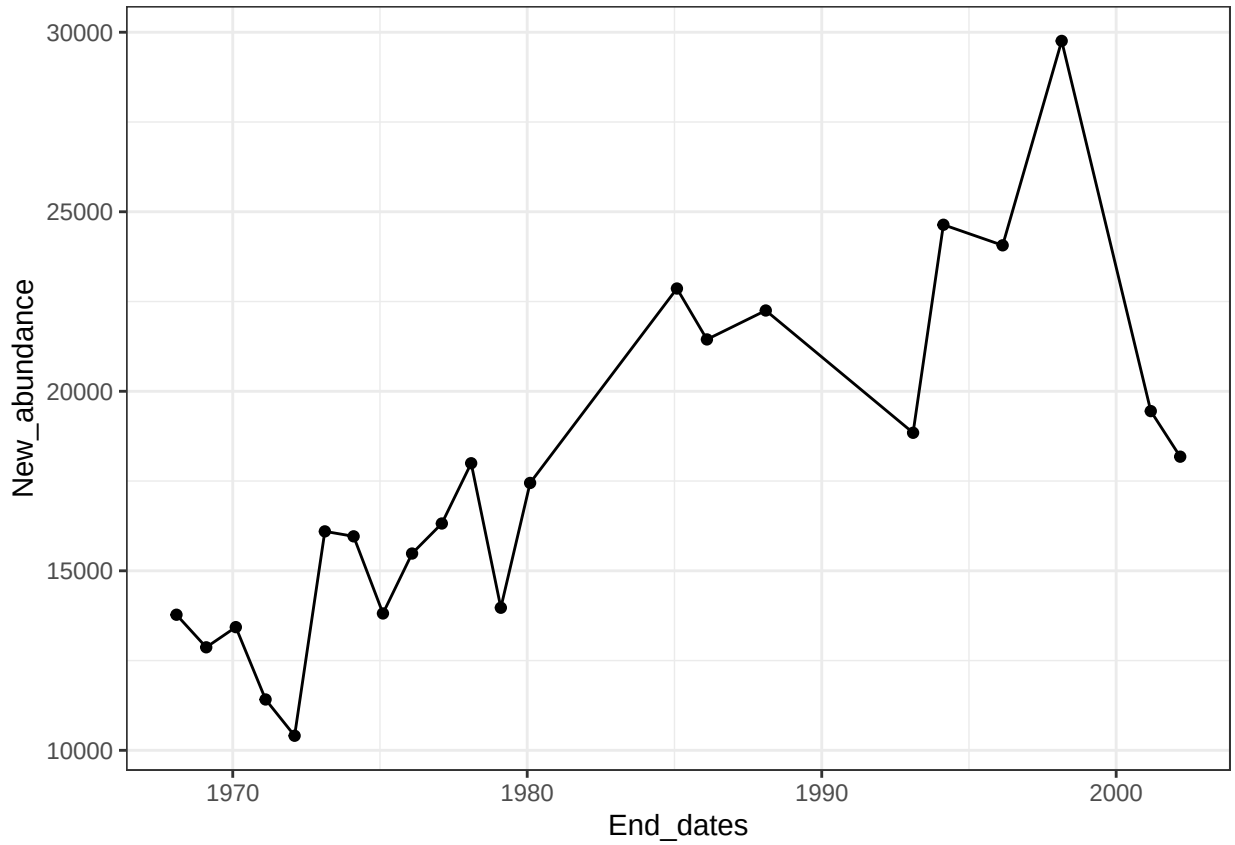
```
## # A tibble: 9 x 5
##   model            mpg   cyl    hp   kml
##   <chr>          <dbl> <dbl> <dbl> <dbl>
## 1 Datsun 710      22.8     4    93  9.80
## 2 Merc 240D       24.4     4    62 10.5
## 3 Merc 230        22.8     4    95  9.80
## 4 Fiat 128        32.4     4    66 13.9
## 5 Honda Civic     30.4     4    52 13.1
## 6 Toyota Corolla  33.9     4    65 14.6
## 7 Fiat X1-9       27.3     4    66 11.7
## 8 Porsche 914-2   26       4    91 11.2
## 9 Lotus Europa    30.4     4   113 13.1
```

Q18

New_abundance is 16098, the uncertainty is 834

Q19

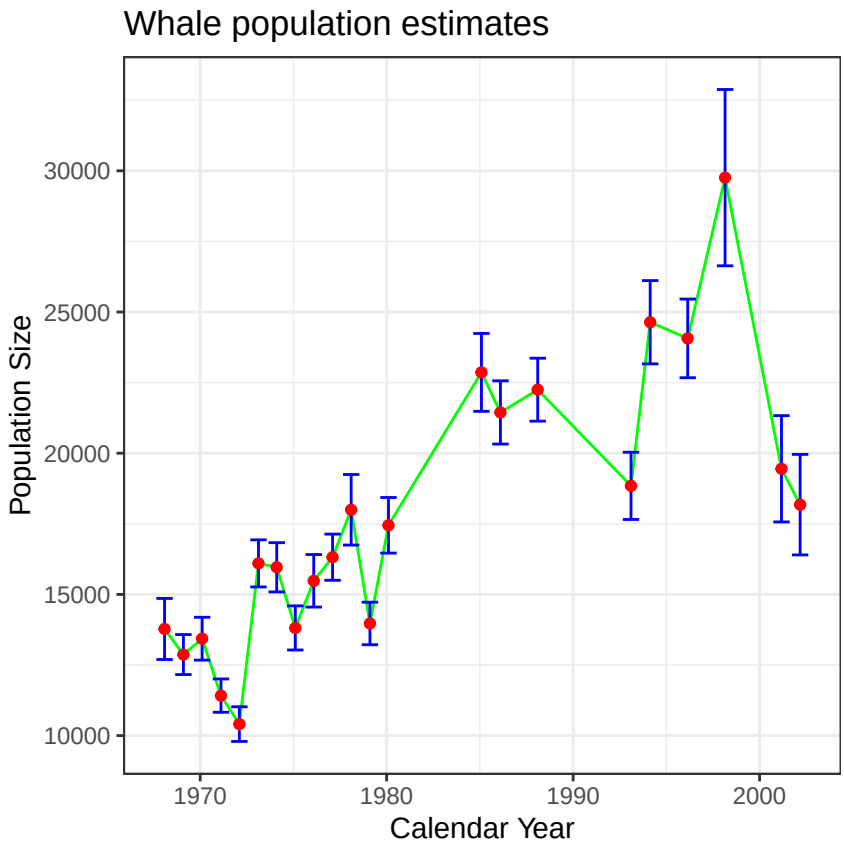
```
plot_whales =
  whale_dataset |>
  ggplot(aes(End_dates, New_abundance)) +
  geom_point() +
  geom_line()
plot_whales
```



Q20

```
plot_whales =
  whale_dataset |>
  ggplot(aes(End_dates, New_abundance)) +
  geom_line(color = 'green') +
  geom_errorbar(
    aes(
      x = End_dates,
      ymin = New_abundance + New_SE,
      ymax = New_abundance - New_SE
    ),
    color = 'blue'
  ) +
  geom_point(color = 'red') +
  labs(
```

```
    x = 'Calendar Year',
    y = 'Population Size'
  ) +
  theme(aspect.ratio = 1) +
  ggtitle('Whale population estimates')
plot_whales
```



Q21

"Error: unexpected '=' in:" print('Two is greater than three') } else if(2 =" "

```
if(2 > 3){
  print('Two is greater than three')
} else if(2 = 3){
  print('Two is equal to three')
} else {
  print('Two is less than three')
}
```

```
## Error: <text>:3:13: unexpected '='
## 2:   print('Two is greater than three')
## 3: } else if(2 =
##                ^
```

Q22

```
for(i in 1:5){
  if(i%%2 != 0){
    print(i)
  }
}
```

```
## [1] 1
## [1] 3
## [1] 5
```

Q23

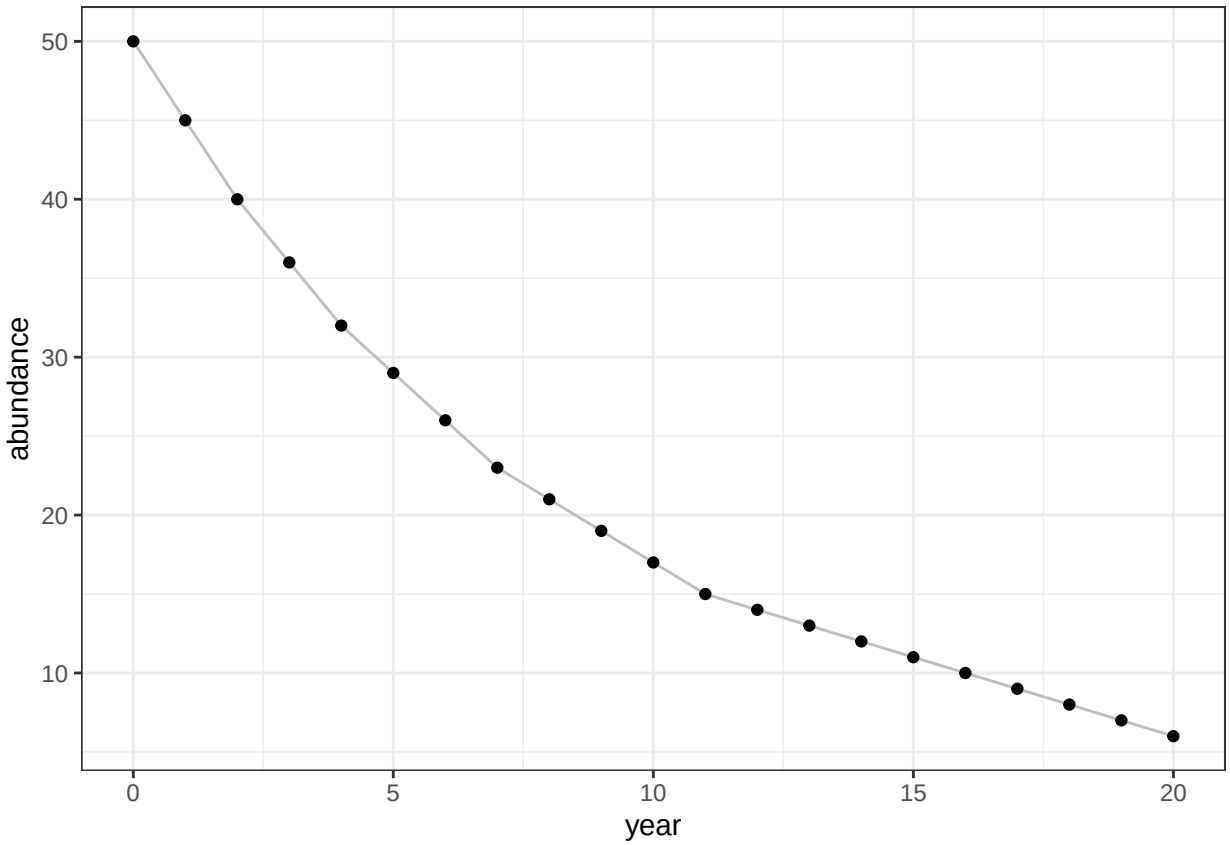
The growth rate is less than 1, so the population declines.

```
lambda = 0.9
N = 50
N_vector = N
for(year in 1:20){
  N = round(lambda * N)
  N_vector = c(N_vector, N)
}

data =
  tibble(
    year = 0:20,
    abundance = N_vector
  )

plot =
  data |>
  ggplot(aes(year, abundance)) +
  geom_line(color = 'grey') +
  geom_point()

plot
```



Q24

The range of the plot with a lower standard deviation is smaller, and the data is more centered around the mean.

```
set.seed(1)

random_numbers1 = rnorm(n = 1000, mean = 0, sd = 1)

random_data1 =
  tibble(
    x = random_numbers1
  )

plot_data1 =
  random_data1 |>
  ggplot(aes(x)) +
  geom_histogram(bins = 30, fill = 'darkred') +
  xlim(c(-5, 5))

random_numbers2 = rnorm(n = 1000, mean = 0, sd = 0.4)

random_data2 =
  tibble(
    x = random_numbers2
```

```
  )

plot_data2 =
  random_data2 |>
  ggplot(aes(x)) +
  geom_histogram(bins = 30, fill = 'darkblue') +
  xlim(c(-5, 5))

grid.arrange(plot_data1, plot_data2)
```