QSCI 482 Lab 2

Simon Hans Edasi

2023-10-06

Exercise 1: Rounding Data

Round operated the way I expected, in that it rounded to the nearest integer. I expected this because I have used it before.

```
library(glue)

newdata <- c(1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0)

glue('Ceiling = {ceiling(newdata)}, Floor = {floor(newdata)}, Rounded = {round(newdata)}')
```

```
## Ceiling = 1, Floor = 1, Rounded = 1
## Ceiling = 2, Floor = 1, Rounded = 2
## Ceiling = 2, Floor = 2, Rounded = 2
## Ceiling = 3, Floor = 2, Rounded = 2
## Ceiling = 3, Floor = 3, Rounded = 3
## Ceiling = 4, Floor = 3, Rounded = 4
## Ceiling = 4, Floor = 4, Rounded = 4
```

Setting digits to a negative value rounds to the nearest power of ten, while positive values determines the number of decimal points to round

```
round(3.456, digits = -2)
```

```
## [1] 0
```

```
round(3456, digits = -1)
```

```
## [1] 3460
```

```
round(3456, digits = -3)
```

```
## [1] 3000
```

```
round(3456.789, digits = 2)
```

```
## [1] 3456.79
```

```
round(3456.789, digits = 1)
```

```
## [1] 3456.8
```

Exercise 2

Probability of two or more people having the same birthday in our class

```
pbirthday(n = 19, classes = 365, coincident = 2)
```

```
## [1] 0.3791185
```

How many people are needed to have a 50% probability?

```
qbirthday(prob = .50, classes = 365, coincident = 2)
```

```
## [1] 23
```

How many people are needed to have a 99.9% probability?

```
qbirthday(prob = .999, classes = 365, coincident = 2)
```

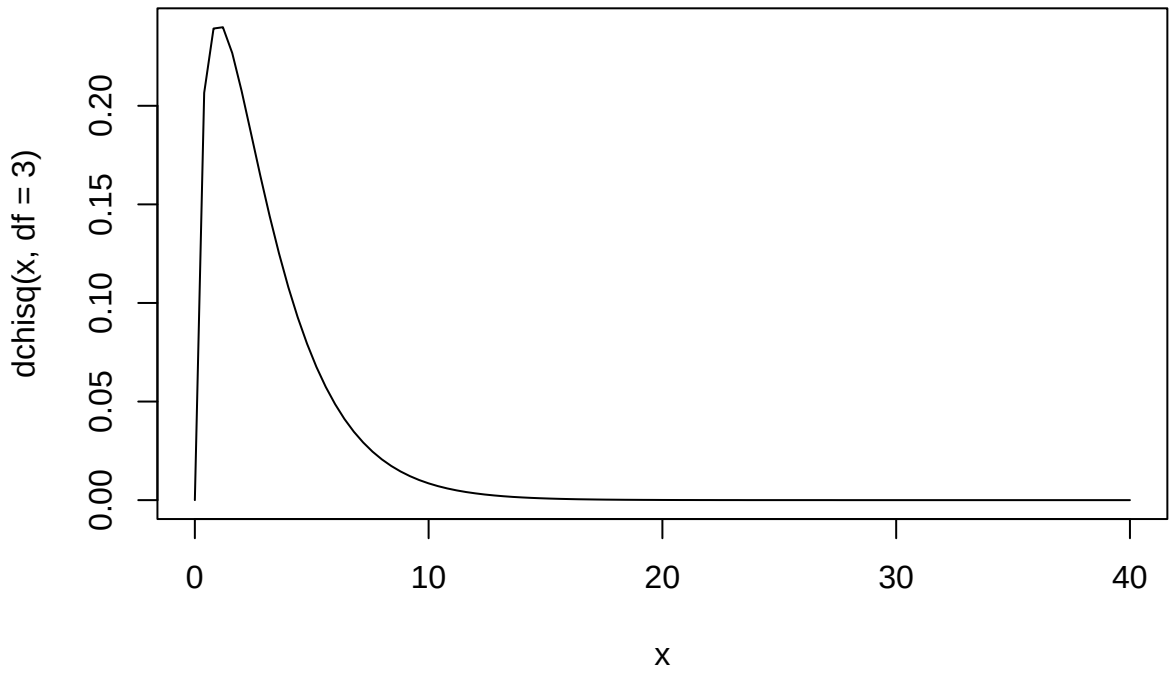
```
## [1] 70
```

We need at least 70 people to have a 99.99% probability of two people sharing a birthday.

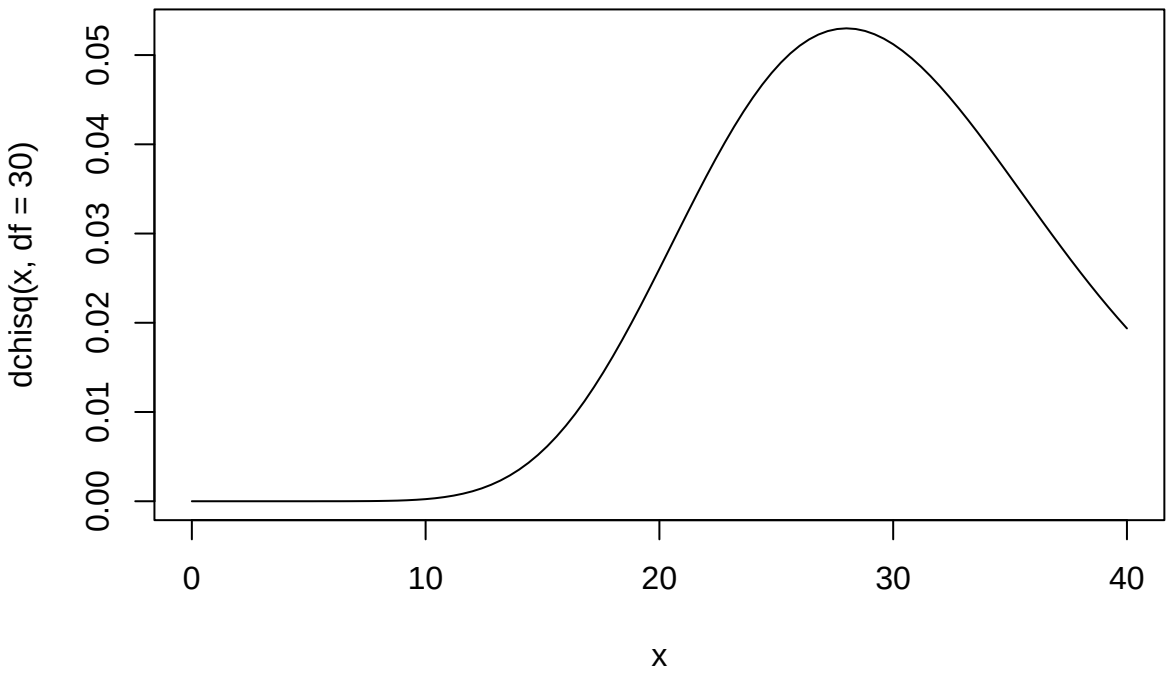
Exercise 3

```
?dchisq()
```

```
x = newdata
curve(dchisq(x, df = 3),from = 0, to = 40)
```




```
curve(dchisq(x, df = 30), from = 0, to = 40)
```



Exercise 4

Sampling 5 values 10 times from standard normal

```
samps <- c()
for (i in 1:10){
  x = rnorm(n = 5, mean = 0, sd = 1)
  samps[i] <- mean(x)
}
glue('Mean: {mean(samps[1])}, Variance: {var(samps)}')
```

```
## Mean: 0.480595631482048, Variance: 0.112780336294677
```

Sampling 1000 values 5 times from standard normal

```
print('sampling 1000 values from standard normal')
```

```
## [1] "sampling 1000 values from standard normal"
```

```
samps <- c()
for (i in 1:5){
  x = rnorm(n = 1000, mean = 0, sd = 1)
  samps[i] <- mean(x)
}
glue('Mean: {mean(samps[1])}, Variance: {var(samps)}')
```

```
## Mean: 0.0186854012367459, Variance: 4.1869556055408e-05
```