Solutions to exercises

Listed below are the solutions to the exercises.

All solutions are found using RStudio, though **you should only do the exercises in RStudio if indicated in the list of exercises.** This may result in slight differences in numerical answers, which is due to rounding errors.

The solutions may often be computed in different ways and when two solutions are given it does not necessarily mean that more solutions does not exist. However, when two solutions are given we encourage you to think about why these two solutions are equivalent.

```
library(mosaic)
```

5.7:

a)

```
n <- 817
sample_prop <- 0.022
sample_prop_se <- sqrt(sample_prop * (1 - sample_prop) / n)
sample_prop_se
```

```
## [1] 0.0051318
```

b)

```
sample_prop + c(-1, 1) * qdist("norm",1 - 0.05 / 2, mean = 0, sd = 1,plot=FALSE) * sample_prop_se
```

```
## [1] 0.01194186 0.03205814
```

The true proportion (population proportion) is with 95% confidence (**not** probability) included in the interval above. More precisely, after repeated trails (that is if we collect multiple samples and compute a confidence interval for each sample) 95% of the confidence intervals will include the true proportion. The interpretation is difficult and due to the fact that the true proportion (population proportion) is considered fixed while the confidence interval is stochastic.

5.17:

a)

```
sample_prop_A <- 0.62
n <- 800
sample_prop_A_se <- sqrt(sample_prop_A * (1 - sample_prop_A) / n)
sample_prop_A + c(-1, 1) * qdist("norm",1 - 0.005, mean = 0, sd = 1,plot=FALSE) * sample_prop_A_se
```

```
## [1] 0.5757962 0.6642038
```

We are quite confident, that A will be the winner.

b)

```
n <- 80
sample_prop_A_se <- sqrt(sample_prop_A * (1 - sample_prop_A) / n)
sample_prop_A + c(-1, 1) * qdist("norm", 0.995, mean = 0, sd = 1,plot=FALSE) * sample_prop_A_se
```

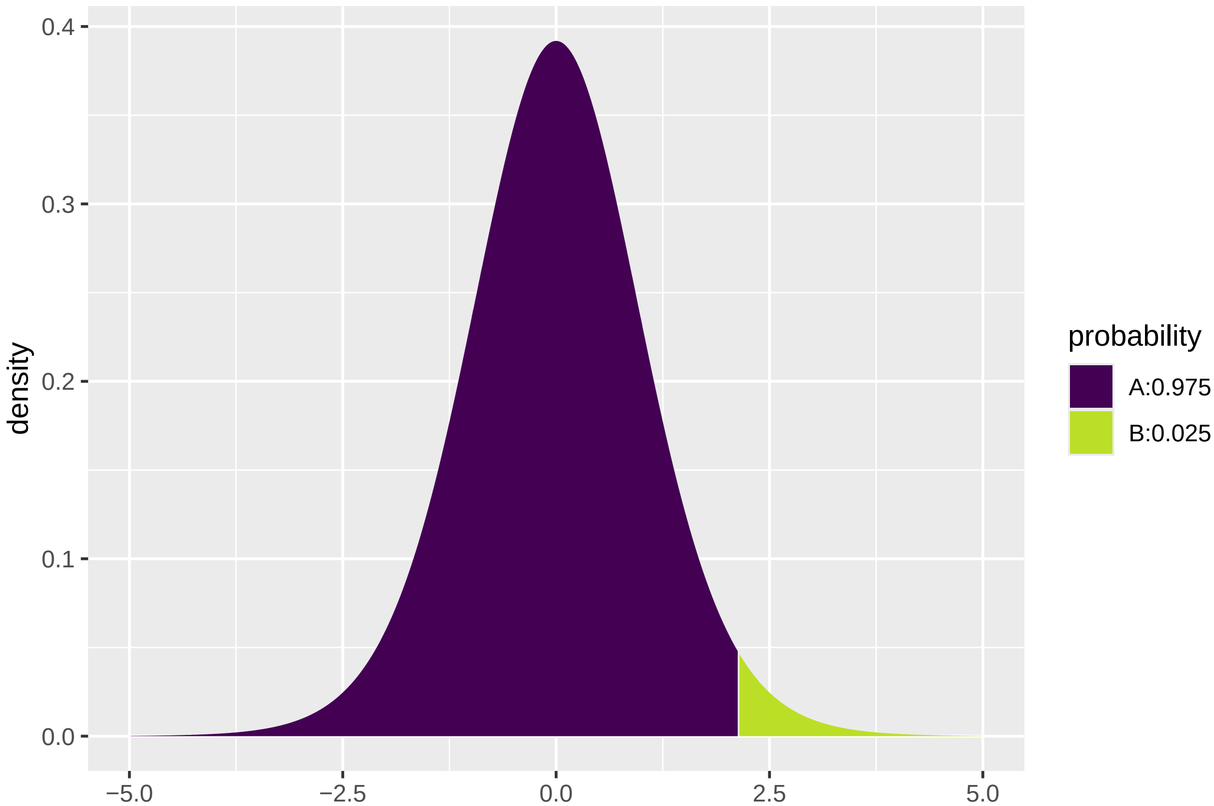
```
## [1] 0.4802153 0.7597847
```

We cannot (with 99% confidence) conclude anything regarding the outcome, since 0.5 (equal probability of both outcomes) is included in the confidence interval.

5.19:

a)

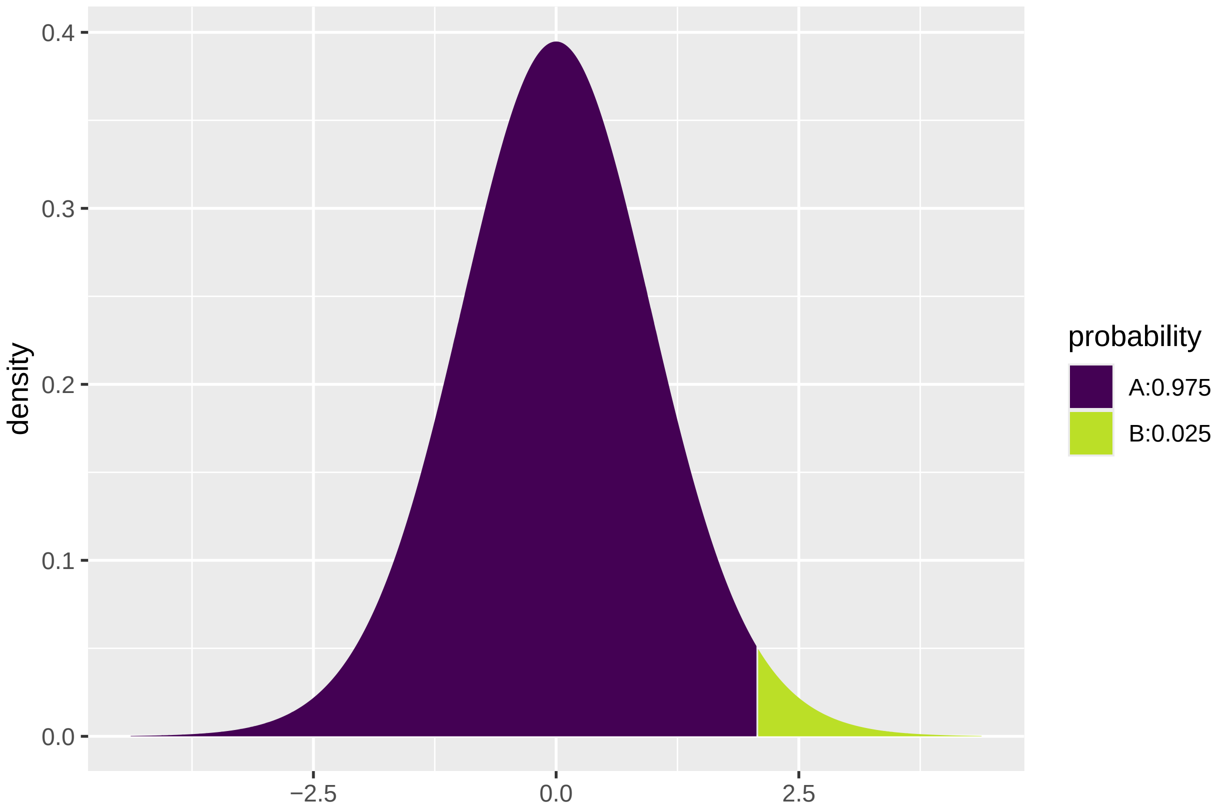
```r
df <- 15 - 1
qdist("t", 1 - (1 - 0.95)/2, df = df)
```




```
## [1] 2.144787
```

b)

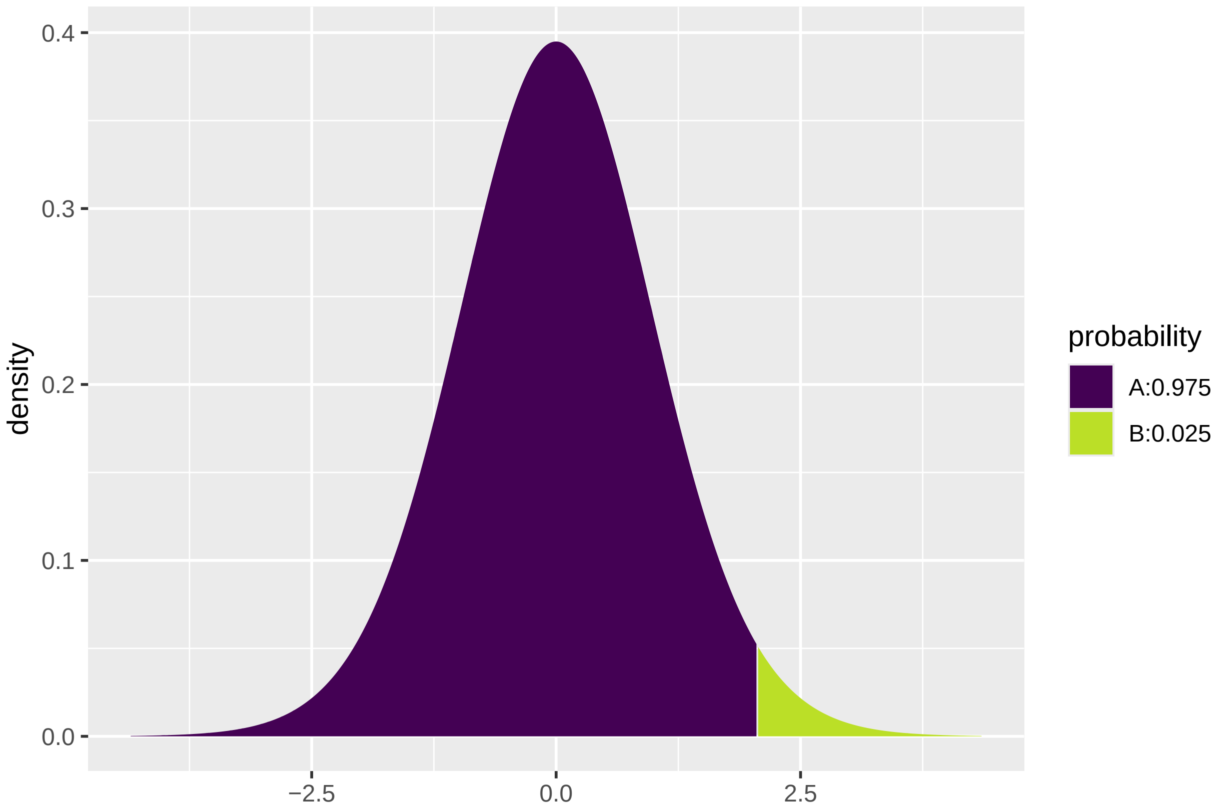
```r
df <- 25 - 1
qdist("t", 1 - (1 - 0.95)/2, df = df)
```




```
## [1] 2.063899
```

c)

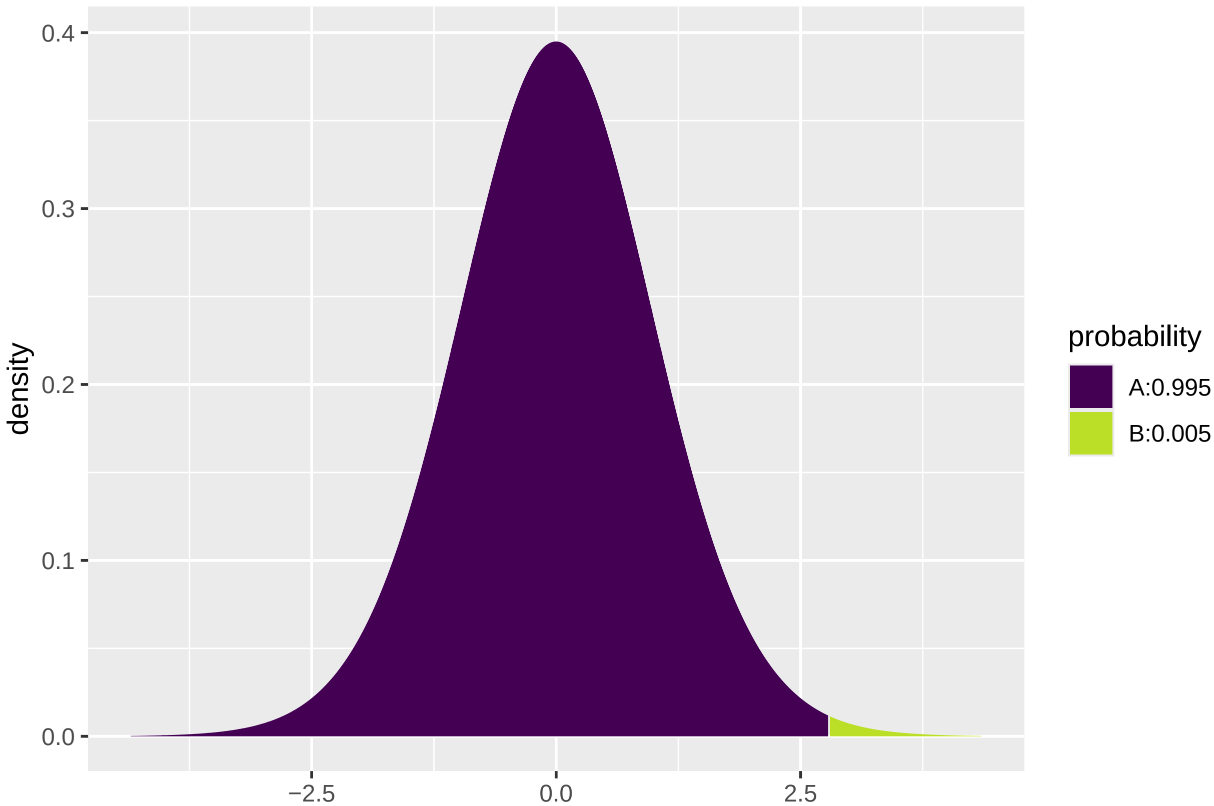
```r
df <- 25
qdist("t", 1 - (1 - 0.95) / 2, df = df)
```




```
## [1] 2.059539
```

d)

```r
df <- 25
qdist("t", 1 - (1 - 0.99) / 2, df = df)
```




```
## [1] 2.787436
```

5.31:

a)

```r
mean <- 3.23
se <- 0.005
df <- 36957 - 1
mean + c(-1, 1) * qdist("t", 1 - (1 - 0.99)/2, df = df,plot=FALSE) * se
```

```
## [1] 3.21712 3.24288
```

b.i)

Wider, since more certainty is required.

b.ii)

```r
0.946 / sqrt(550)
```

```
## [1] 0.04033758
```

Wider, since the standard error is larger for this group of people.

c)

We assume that the data are on interval scale (numerical measurements where differences are sensible) and that the spacing between the answers are equal.

5.37:

a)

```
prop_guess <- 0.1
sig_lvl <- 1 - 0.95
margin_of_error <- 0.02
n <- prop_guess * (1 - prop_guess) * (qdist("norm", 1 - sig_lvl / 2, mean = 0, sd = 1,plot=FALSE) / marg
ceiling(n) # Rounded up (towards the ceiling)
```

```
## [1] 865
```

b)

```
prop_guess <- 0.5
n <- prop_guess * (1 - prop_guess) * (qdist("norm",1 - sig_lvl / 2, mean = 0, sd = 1,plot=FALSE) / margi
ceiling(n)
```

```
## [1] 2401
```

5.41:

We Assume that 0 and 18 encompasses the mean plus and minus about 3 standard deviations, then the standard deviation is approximately 3, and we need:

```
sig_lvl <- 1 - 0.95
margin_of_error <- 1
std <- 3
n <- (std * qdist("norm", 1 - sig_lvl / 2, mean = 0, sd = 1,plot=FALSE) / margin_of_error)^2
ceiling(n)
```

```
## [1] 35
```