Programming Task 2.2

The code is based on the one explained on the book, on chapter 2.1.5 Posterior simulation

Problem 1.

This is problem 2.20

The code is as follows:

```
## Problem 2.20

# Define possible cat images
cat <- data.frame(type = c("cat", "notCat"))


# Define the prior model
prior <- c(0.08, 0.92)

# Simulate 10,000 articles
set.seed(64)
cat_sim <- sample_n(cat, size = 10000, weight = prior, replace = TRUE)

# Print the distribution of Cat NotCat images
ggplot(cat_sim, aes(x = type)) +
  geom_bar()

# Summarize the prior
cat_sim %>%
  tabyl(type) %>%
  adorn_totals("row")

# Add the likelyhood of having cat to the dataframe depending on the type of image
cat_sim <- cat_sim %>%
  mutate(data_model = case_when(type == "notCat" ~ 0.5,
                                type == "cat" ~ 0.8))

# Define whether the predictions thinks there are cats in the image
data <- c("no", "yes")

# Simulate cat detection on the image
set.seed(1)
cat_sim <- cat_sim %>%
  group_by(1:n()) %>%
  mutate(predicted = sample(data, size = 1,
                        prob = c(1-data_model, data_model)))

# Print the predicted values and the original.
cat_sim %>%
  tabyl(predicted, type) %>%
  adorn_totals(c("col","row"))

# Plot the predictions by type of image.
ggplot(cat_sim, aes(x = type, fill = predicted)) +
  geom_bar(position = "fill")

# Print the posterior approximation
cat_sim %>%
  filter(predicted == "yes") %>%
  tabyl(type) %>%
  adorn_totals("row")
```

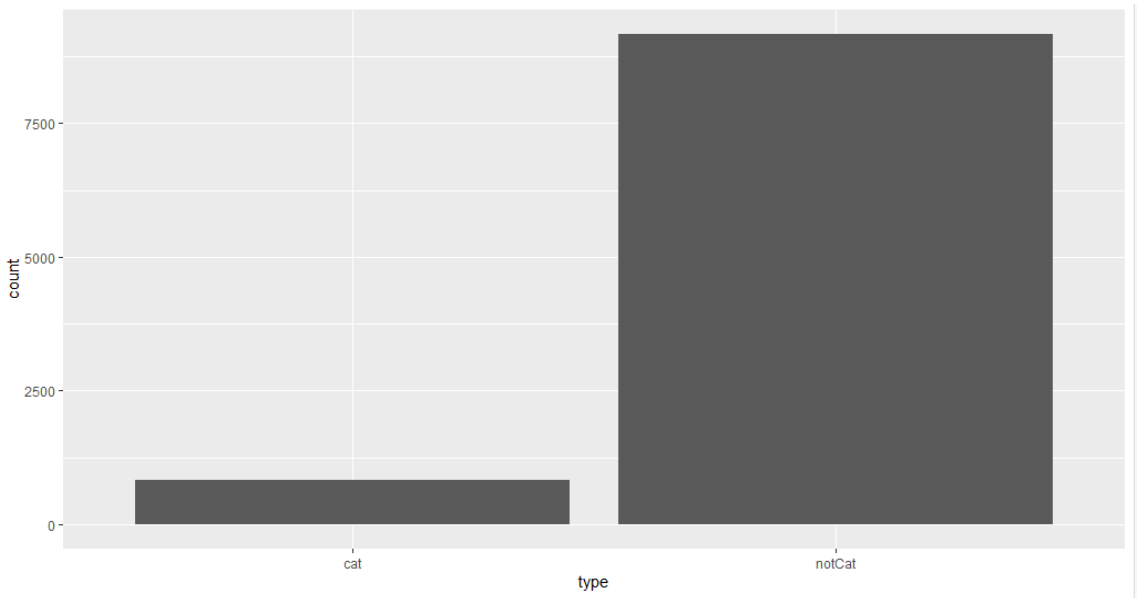
First thing we need to do is to define the prior probabilities and the variable used. For that we need to use the information provided by the problem. That info is in the next table

Alejandro C Parra García

Image	Cat	NotCat	Total
probability	0.08	0.92	1.0

The next part is to simulate 10,000 different images of cat following that distribution. We can plot it to see the results of the simulation.

```
> # Print the distribution of Cat NotCat images
> ggplot(cat_sim, aes(x = type)) +
+   geom_bar()
>
```



We can see that the number of cat image is roughly 8%.

To be sure let's print the prior values.

```
> # Summarize the prior
> cat_sim %>%
+   tabyl(type) %>%
+   adorn_totals("row")
   type     n percent
    cat   830   0.083
 notCat  9170   0.917
  Total 10000   1.000
>
```

We can see that in fact the percentage of cat images is 8.3% roughly 8%.

Then we add the likelihood of the images getting predicted correctly or not. 80% for the cats' images, and 50% for the not cat's image. Then we need to make the predictions at random depending on the 80% or 50%.

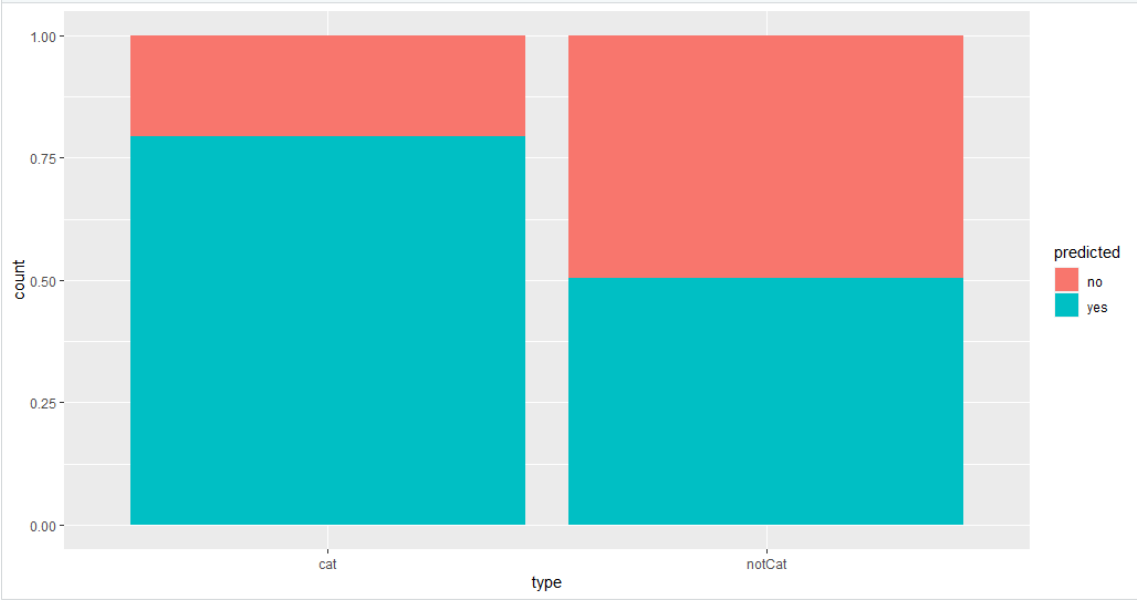
Once we have finished that step lets check if we are correct, for that we print the values broken down by the predictions.

```
> # Print the predicted values and the original.
> cat_sim %>%
+   tabyl(predicted, type) %>%
+   adorn_totals(c("col","row"))
 predicted cat notCat Total
        no 171   4548  4719
       yes 659   4622  5281
     Total 830   9170 10000
> |
```

We can see that 659/830=0.794 that is almost 80%, for the cat images. And 4622/9170=0.504 that is almost 50% for the not cat images. That means that the values are generated correctly.

We can also plot it to make it easier to see. Roughly 80% of cat images got predicted correctly and roughly 50% of not cat got predicted correctly.

```
> # Plot the predictions by type of image.
> ggplot(cat_sim, aes(x = type, fill = predicted)) +
+   geom_bar(position = "fill")
> |
```



We now can print the Data for the predicted as Cat images:

```
> cat_sim %>%
+   filter(predicted == "yes") %>%
+   tabyl(type) %>%
+   adorn_totals("row")
   type    n  percent
    cat  659 0.124787
 notCat 4622 0.875213
  Total 5281 1.000000
> |
```

From them we can see that of the images predicted as cats only 12.48% are actually cats and 87.52% are not cats.

So, the probability of an image being a cat if the algorithm says it is a cat, is only of 12.48%

Problem 2.

This is problem 2.21

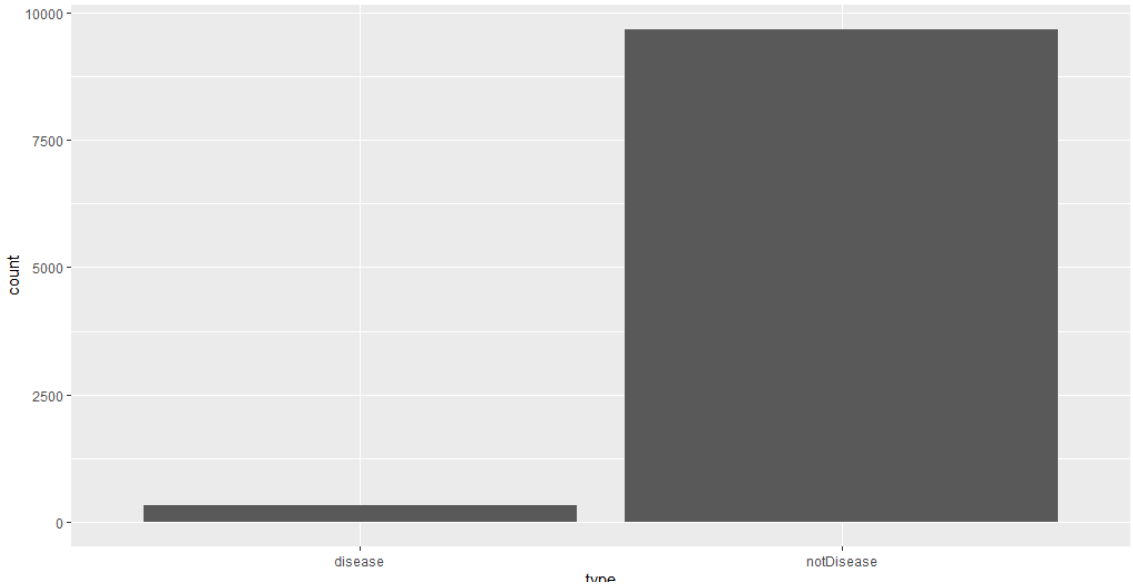
The code is as follows:

```
76  ## Problem 2.21
77
78  # Define possible condition
79  disease <- data.frame(type = c("disease", "notDisease"))
80
81
82  # Define the prior model
83  prior <- c(0.03, 0.97)
84
85  # Simulate 10,000 persons
86  set.seed(64)
87  disease_sim <- sample_n(disease, size = 10000, weight = prior, replace = TRUE)
88
89  # Print the distribution of people with and without disease
90  ggplot(disease_sim, aes(x = type)) +
91    geom_bar()
92
93  # Summarize the prior
94  disease_sim %>%
95    tabyl(type) %>%
96    adorn_totals("row")
97
98  # Add the likelyhood of having the disease to the dataframe
99  disease_sim <- disease_sim %>%
100   mutate(data_model = case_when(type == "notDisease" ~ 0.07,
101                                 type == "disease" ~ 0.93))
102
103 # Define whether the test predict the disease or not
104 data <- c("no", "yes")
105
106 # Simulate the test
107 set.seed(1)
108 disease_sim <- disease_sim %>%
109   group_by(1:n()) %>%
110   mutate(predicted = sample(data, size = 1,
111                             prob = c(1-data_model, data_model)))
112
113 # Print the predicted values and the original.
114 disease_sim %>%
115   tabyl(predicted, type) %>%
116   adorn_totals(c("col","row"))
117
118 # Plot the predictions by disease or not disease
119 ggplot(disease_sim, aes(x = type, fill = predicted)) +
120   geom_bar(position = "fill")
121
122 # Print the posterior approximation
123 disease_sim %>%
124   filter(predicted == "yes") %>%
125   tabyl(type) %>%
126   adorn_totals("row")
127
```

The code is the same as for the previous problem. The only difference is in the numbers used, they got change in the line 83 and in the 100 and 101. The first change is to represent the new distribution of population with disease, while the second value is the likelihood of getting a positive in the test depending on if you have or not the disease.

So, lets only explore the different outputs of the code. The first one is to check if the 10,000 sample is generated correctly.

```
> # Print the distribution of people with and without disease
> ggplot(disease_sim, aes(x = type)) +
+   geom_bar()
> |
```



As we can see roughly 3% of the population have the disease. We can check it looking at the prior.

```
> # Summarize the prior
> disease_sim %>%
+   tabyl(type) %>%
+   adorn_totals("row")
       type     n percent
    disease   328  0.0328
 notDisease  9672  0.9672
      Total 10000  1.0000
> |
```

As we can see 3.28% of the population have the disease and 96.72% don't have the disease.

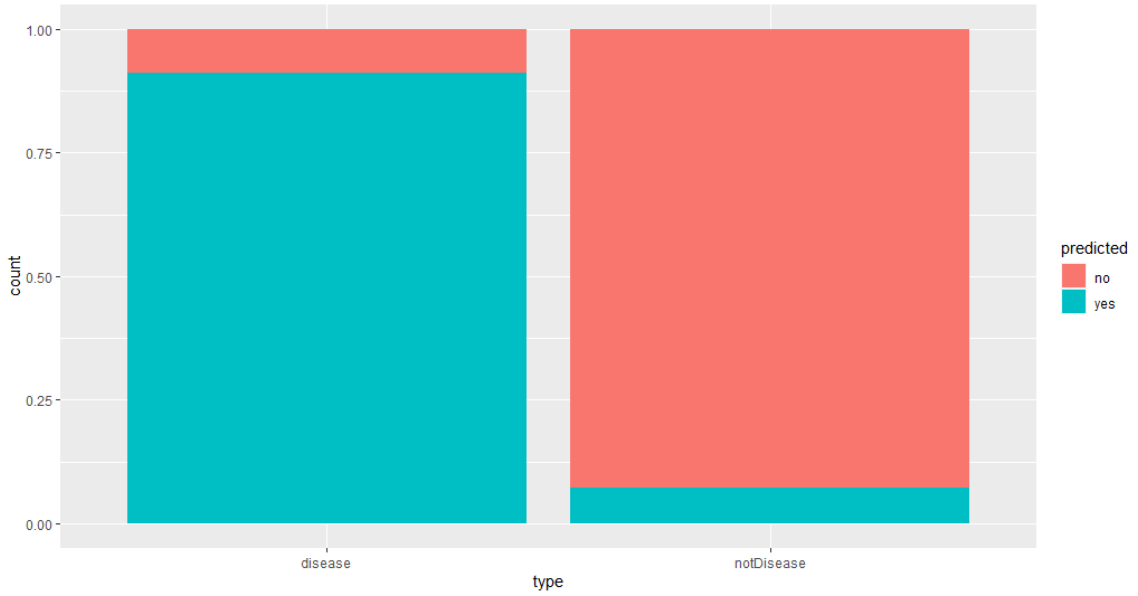
Now we generate the predicted values of the test depending on the different likelihood.

```
> # Print the predicted values and the original.
> disease_sim %>%
+   tabyl(predicted, type) %>%
+   adorn_totals(c("col","row"))
 predicted disease notDisease Total
        no      29       8966  8995
       yes     299        706  1005
     Total     328       9672 10000
> |
```

We can see that 299/328=0.9115 that is almost 93%, for having the disease. And 706/9672=0.07299 that is almost 7% for the not having the disease. That means that the values are generated correctly.

We can also plot it to make it easier to see. Roughly 93% of the people with the disease got tested correctly and roughly 7% of the people without the disease got tested incorrectly.

```
> # Plot the predictions by disease or not disease
> ggplot(disease_sim, aes(x = type, fill = predicted)) +
+   geom_bar(position = "fill")
> |
```



We now can print the Data for the predicted as disease cases:

```
> disease_sim %>%
+   filter(predicted == "yes") %>%
+   tabyl(type) %>%
+   adorn_totals("row")
       type    n   percent
    disease  299 0.2975124
 notDisease  706 0.7024876
      Total 1005 1.0000000
> |
```

From them we can see that of the people tested as positive only 29.75% have actually the disease, and 70.24% don't have the disease.

So, the probability of a person having the disease if they tested positive is only 29.75%