Statistical Packages

HW9

Mohamed Salem

```r
# we begin by loading our required libraries
library(reticulate)
library(tensorflow)
library(keras)
```

Due to issues and incompatibilities faced when attempting to install the latest version of Anaconda on a windows 8 machine, and incompatibilities between the latest version of Python (3.7) and Tensorflow, we have manually installed an older version of Anaconda (3.5.1) which comes with Python 3.6. We've also manually installed tensorflow through Anaconda via the pip commands:

pip install tensorflow conda create -n tensorflow_env tensorflow

Next we load both keras and tensorflow through the set up conda environment:

```r
# Note, we've excluded this chunk from being evaluated to avoid
# running an installation every time we knit
install_tensorflow(method = "conda", conda = "C:/ProgramData/Anaconda3/Scripts/conda.exe",
    envname = "tensorflow_env")
install_keras(method = "conda", conda = "C:/ProgramData/Anaconda3/Scripts/conda.exe")
```

```r
Sys.setenv(TENSORFLOW_PYTHON = "C:/ProgramData/Anaconda3/envs/tensorflow_env/python.exe")
Sys.setenv(KERAS_PYTHON = "C:/ProgramData/Anaconda3/envs/tensorflow_env/python.exe")
tensorflow::use_condaenv("tensorflow_env")
keras::use_condaenv("tensorflow_env")
use_python("C:/ProgramData/Anaconda3/envs/tensorflow_env/python.exe")
use_condaenv(conda = "C:/ProgramData/Anaconda3/Scripts/conda.exe")
is_keras_available()
```

```
## [1] TRUE
```

```r
use_condaenv(conda = "C:/ProgramData/Anaconda3/Scripts/conda.exe")
# Finally, we check if keras is available for use
is_keras_available()
```

```
## [1] TRUE
```

```r
# and we check that Tensorflow is active and visible to our system
reticulate::py_config()
```

```
## python:         C:/ProgramData/Anaconda3/envs/tensorflow_env/python.exe
## libpython:      C:/ProgramData/Anaconda3/envs/tensorflow_env/python36.dll
## pythonhome:     C:\PROGRA~3\ANACON~1\envs\TENSOR~1
## version:        3.6.9 |Anaconda, Inc.| (default, Jul 30 2019, 14:00:49) [MSC v.1915 64 bit (AMD64)]
## Architecture:   64bit
## numpy:          C:\PROGRA~3\ANACON~1\envs\TENSOR~1\lib\site-packages\numpy
## numpy_version:  1.17.4
## tensorflow:     C:\PROGRA~3\ANACON~1\envs\TENSOR~1\lib\site-packages\tensorflow\__init__.p
##
## NOTE: Python version was forced by RETICULATE_PYTHON
```

We've also faced issues loading the MNIST dataset directly from keras. So, we've opted for importing the same dataset from another package as displayed below:

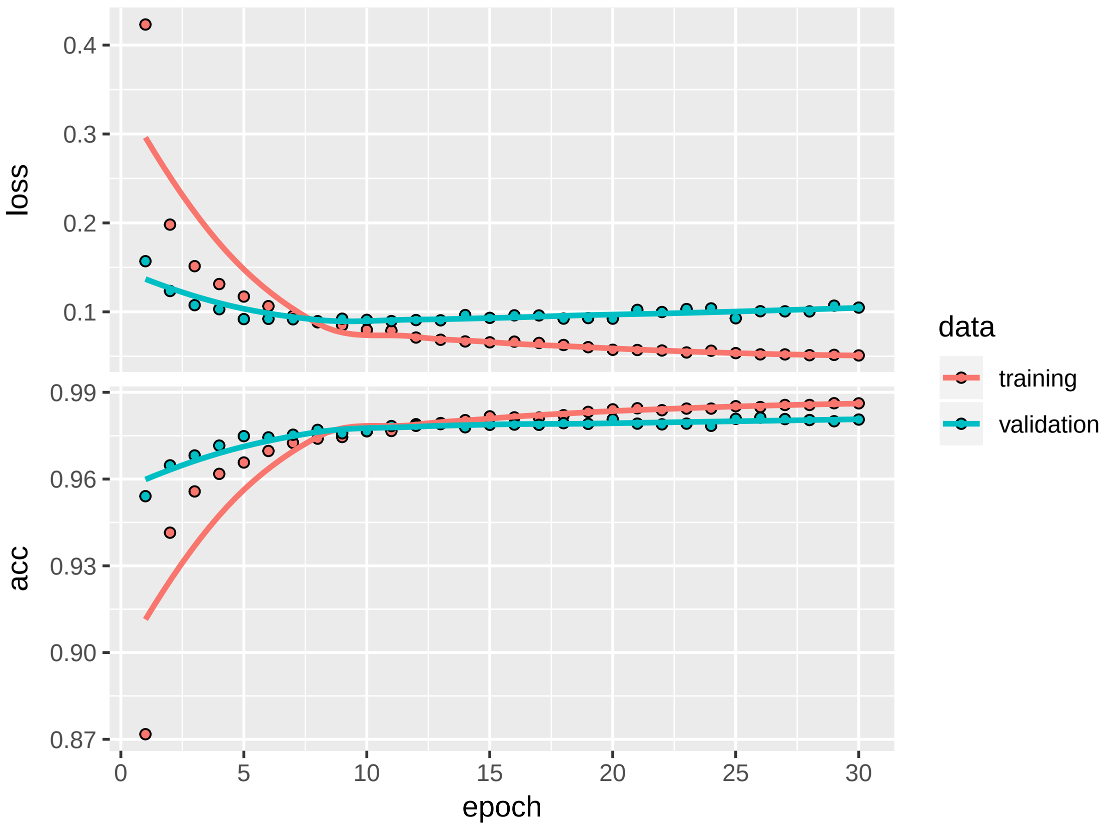
```r
library(dslabs)
mnist <- read_mnist()
```

Now that we have our MNIST dataset loaded and functional, we begin by fitting our neural networks:

```r
# Splitting the data into training and testing sets
x_train <- mnist$train$images
y_train <- mnist$train$labels
x_test <- mnist$test$images
y_test <- mnist$test$labels
# Converting the data from 3D arrays into matrices
x_train <- array_reshape(x_train, c(nrow(x_train), 784))
x_test <- array_reshape(x_test, c(nrow(x_test), 784))
# converting grayscale values into integers
x_train <- x_train/255
x_test <- x_test/255
# Creating dummy variables for the response
y_train <- to_categorical(y_train, 10)
y_test <- to_categorical(y_test, 10)
# Setting up the keras Sequential model
model <- keras_model_sequential()
model %>% layer_dense(units = 256, activation = "relu", input_shape = c(784)) %>%
    layer_dropout(rate = 0.4) %>% layer_dense(units = 128, activation = "relu") %>%
    layer_dropout(rate = 0.3) %>% layer_dense(units = 10, activation = "softmax")
# A summary of the model
summary(model)
```

```
## ____________________________________________________________________________
## Layer (type)                     Output Shape                  Param #
## ============================================================================
## dense (Dense)                    (None, 256)                   200960
## ____________________________________________________________________________
## dropout (Dropout)                (None, 256)                   0
## ____________________________________________________________________________
## dense_1 (Dense)                  (None, 128)                   32896
## ____________________________________________________________________________
## dropout_1 (Dropout)              (None, 128)                   0
## ____________________________________________________________________________
## dense_2 (Dense)                  (None, 10)                    1290
## ============================================================================
## Total params: 235,146
## Trainable params: 235,146
## Non-trainable params: 0
## ____________________________________________________________________________
```

```r
# Compiling the model with a loss function and other metrics
model %>% compile(loss = "categorical_crossentropy", optimizer = optimizer_rmsprop(),
    metrics = c("accuracy"))
# Training the model 30 times using 128 images per trial
history <- model %>% fit(x_train, y_train, epochs = 30, batch_size = 128,
    validation_split = 0.2)
# A plot of the model's development and improvement
plot(history)
```




```r
# Evaluating the model's performannce on the test set
model %>% evaluate(x_test, y_test)
```

```
## $loss
## [1] 0.1040854
##
## $acc
## [1] 0.9811
```

```r
# Generating predictions
model %>% predict_classes(x_test)
```

```
##     [1] 7 2 1 0 4 1 4 9 5 9 0 6 9 0 1 5 9 7 3 4 9 6 6 5 4 0 7 4 0 1 3 1 3 4
##    [35] 7 2 7 1 2 1 1 7 4 2 3 5 1 2 4 4 6 3 5 5 6 0 4 1 9 5 7 8 9 3 7 4 6 4
##    [69] 3 0 7 0 2 9 1 7 3 2 9 7 7 6 2 7 8 4 7 3 6 1 3 6 9 3 1 4 1 7 6 9 6 0
##   [103] 5 4 9 9 2 1 9 4 8 7 3 9 7 4 4 4 9 2 5 4 7 6 7 9 0 5 8 5 6 6 5 7 8 1
##   [137] 0 1 6 4 6 7 3 1 7 1 8 2 0 3 9 9 5 5 1 5 6 0 3 4 4 6 5 4 6 5 4 5 1 4
##   [171] 4 7 2 3 2 7 1 8 1 8 1 8 5 0 8 9 2 5 0 1 1 1 0 9 0 3 1 6 4 2 3 6 1 1
##   [205] 1 3 9 5 2 9 4 5 9 3 9 0 3 5 5 5 7 2 2 7 1 2 8 4 1 7 3 3 8 8 7 9 2 2
##   [239] 4 1 5 9 8 7 2 3 0 2 4 2 4 1 9 5 7 7 2 8 2 6 8 5 7 7 9 1 8 1 8 0 3 0
##   [273] 1 9 9 4 1 8 2 1 2 9 7 5 9 2 6 4 1 5 8 2 9 2 0 4 0 0 2 8 4 7 1 2 4 0
##   [307] 2 7 4 3 3 0 0 3 1 9 6 5 2 5 9 7 9 3 0 4 2 0 7 1 1 2 1 5 3 3 9 7 8 6
##   [341] 3 6 1 3 8 1 0 5 1 3 1 5 5 6 1 8 5 1 7 9 4 6 2 2 5 0 6 5 6 3 7 2 0 8
##   [375] 8 5 4 1 1 4 0 7 3 7 6 1 6 2 1 9 2 8 6 1 9 5 2 5 4 4 2 8 3 8 2 4 5 0
##   [409] 3 1 7 7 5 7 9 7 1 9 2 1 4 2 9 2 0 4 9 1 4 8 1 8 4 5 9 8 8 3 7 6 0 0
##   [443] 3 0 2 0 6 9 9 3 3 3 2 3 9 1 2 6 8 0 5 6 6 6 3 8 8 2 7 5 8 9 6 1 8 4
##   [477] 1 2 5 9 1 9 7 5 4 0 8 9 9 1 0 5 2 3 7 0 9 4 0 6 3 9 5 2 1 3 1 3 6 5
##   [511] 7 4 2 2 6 3 2 6 5 4 8 9 7 1 3 0 3 8 3 1 9 3 4 4 6 4 2 1 8 2 5 4 8 8
```

```
##   [545] 4 0 0 2 3 2 7 7 0 8 7 4 4 7 9 6 9 0 9 8 0 4 6 0 6 3 5 4 8 3 3 9 3 3
##   [579] 3 7 8 0 2 2 1 7 0 6 5 4 3 8 0 9 6 3 8 0 9 9 6 8 6 8 5 7 8 6 0 2 4 0
##   [613] 2 2 3 1 9 7 5 8 0 8 4 6 2 6 7 9 3 2 9 8 2 2 9 2 7 3 5 9 1 8 0 2 0 5
##   [647] 2 1 3 7 6 7 1 2 5 8 0 3 7 8 4 0 9 1 8 6 7 7 4 3 4 9 1 9 5 1 7 3 9 7
##   [681] 6 9 1 3 3 8 3 3 6 7 2 4 5 8 5 1 1 4 4 3 1 0 7 7 0 7 9 4 4 8 5 5 4 0
##   [715] 8 2 1 0 8 4 8 0 4 0 6 1 9 3 2 6 7 2 6 9 3 1 4 6 2 5 9 2 0 6 2 1 7 3
##   [749] 4 1 0 5 4 3 1 1 7 4 9 9 4 8 4 0 2 4 5 1 1 6 4 7 1 9 4 2 4 1 5 5 3 8
##   [783] 3 1 4 5 6 8 9 4 1 5 3 8 0 3 2 5 1 2 8 3 4 4 0 8 8 3 3 1 7 3 5 9 6 3
##   [817] 2 6 1 3 6 0 7 2 1 7 1 4 2 4 2 1 7 9 6 1 1 2 4 3 1 7 7 4 8 0 7 3 1 3
##   [851] 1 0 7 7 0 3 5 5 2 7 6 6 9 2 8 3 5 2 2 5 6 0 8 2 9 2 8 8 8 8 7 4 9 3
##   [885] 0 6 6 3 2 1 3 2 2 9 3 0 0 5 7 8 3 4 4 6 0 2 9 1 4 7 4 7 3 9 8 8 4 7
##   [919] 1 2 1 2 2 3 2 3 2 3 9 1 7 4 0 3 5 5 8 6 3 2 6 7 6 6 3 2 7 8 1 1 7 4
##   [953] 6 4 9 5 2 3 3 4 7 8 9 1 1 0 9 1 4 4 5 4 0 6 2 2 3 1 5 1 2 0 3 8 1 2
##   [987] 6 7 1 6 2 3 9 0 1 2 2 0 8 9 9 0 2 5 1 9 7 8 1 0 4 1 7 9 5 4 2 6 8 1
##  [1021] 3 7 5 4 4 1 8 1 3 8 1 2 5 8 0 6 2 1 1 7 1 5 3 4 6 9 5 0 9 2 2 4 8 2
##  [1055] 1 7 2 4 9 4 4 0 3 9 2 2 3 3 8 3 5 7 3 5 8 1 2 4 4 6 4 9 5 1 0 6 9 5
##  [1089] 9 5 9 7 3 8 0 3 7 1 3 6 7 8 5 9 7 9 6 9 6 3 7 4 6 5 3 5 4 7 8 7 8 0
##  [1123] 7 6 8 8 7 3 3 1 9 5 2 7 3 5 1 1 2 1 4 7 4 7 5 4 5 4 0 8 3 6 9 6 0 2
##  [1157] 8 4 4 4 4 6 6 4 7 9 3 4 5 5 8 7 3 7 2 7 0 2 4 1 1 6 8 9 2 8 7 2 0 1
##  [1191] 5 0 4 1 7 0 6 0 8 6 8 1 8 0 3 3 7 2 3 6 2 1 6 1 1 3 7 9 0 8 0 5 4 0
##  [1225] 2 8 2 2 9 8 4 0 8 5 8 5 1 2 1 3 1 7 9 5 7 2 0 3 8 8 6 2 5 4 1 9 2 1
##  [1259] 5 8 1 0 2 4 4 3 6 8 8 2 4 0 5 0 4 4 7 9 3 4 1 5 9 7 3 5 8 8 0 9 3 3
##  [1293] 6 6 0 1 6 0 3 7 4 4 1 2 9 1 4 6 9 9 3 9 8 4 4 3 1 3 1 3 8 7 9 4 8 8
##  [1327] 7 9 9 1 4 5 6 0 5 2 2 2 1 5 5 2 4 9 6 2 7 7 2 2 1 1 2 8 3 7 2 4 1 7
##  [1361] 1 7 6 7 8 2 7 3 1 7 5 8 2 6 2 2 5 6 5 0 9 2 4 3 3 9 7 6 6 8 0 4 1 3
##  [1395] 8 3 9 1 8 0 6 7 2 1 0 5 5 2 0 2 2 0 2 4 9 8 0 9 9 4 6 5 4 9 1 8 3 4
##  [1429] 9 9 1 2 2 8 1 9 6 4 0 9 4 8 3 8 6 0 2 5 1 9 6 2 9 4 0 9 6 0 6 2 5 4
##  [1463] 2 3 8 4 5 5 0 3 8 5 3 5 8 6 5 7 6 3 3 9 6 1 1 2 9 0 4 3 3 6 9 5 0 3
##  [1497] 7 7 7 8 1 9 8 3 0 7 2 7 9 4 5 4 9 3 2 1 4 0 2 3 7 5 9 8 8 5 0 5 1 4
##  [1531] 7 3 9 0 0 0 6 6 2 3 7 8 4 7 7 9 2 4 1 6 5 2 4 3 9 1 8 4 0 9 8 4 8 7
##  [1565] 7 0 7 8 8 6 0 4 8 8 2 4 7 6 6 6 4 7 1 8 8 2 3 6 3 0 0 3 7 6 9 7 9 9
##  [1599] 5 4 3 3 6 1 2 3 7 3 3 2 0 3 3 8 4 3 6 3 5 0 2 0 9 0 7 4 6 9 3 5 1 9
##  [1633] 6 1 4 5 4 5 0 5 9 5 2 1 2 9 1 9 9 4 0 8 4 5 2 9 2 1 2 1 7 3 6 8 8 4
##  [1667] 9 1 9 8 5 7 5 1 1 8 6 5 2 4 4 7 2 3 5 6 8 8 6 2 3 1 0 5 8 9 2 9 6 7
##  [1701] 0 4 8 7 1 7 4 1 0 3 7 2 0 0 9 1 7 0 7 8 4 7 2 0 4 6 0 3 1 1 3 3 9 6
##  [1735] 7 4 1 5 3 0 8 7 3 9 6 9 3 5 0 2 7 4 5 1 2 5 8 0 8 8 1 5 0 3 0 3 1 4
##  [1769] 0 3 7 2 7 1 8 0 7 0 4 3 1 9 8 7 7 1 4 9 9 3 7 1 7 9 0 2 0 3 3 7 6 9
##  [1803] 2 3 3 7 7 0 0 7 5 2 9 8 7 4 4 2 6 6 1 9 6 8 2 9 0 8 3 1 1 6 3 5 1 1
##  [1837] 1 3 1 2 3 0 2 0 1 3 5 5 7 4 8 9 6 9 6 8 3 6 6 8 5 1 4 2 4 4 5 1 1 9
##  [1871] 0 2 4 9 5 7 1 8 8 5 6 9 8 7 1 1 6 7 6 3 2 2 0 8 9 2 5 1 0 8 1 4 5 7
##  [1905] 9 6 9 0 6 1 5 5 8 3 8 2 6 5 0 7 4 6 1 3 4 7 3 2 3 4 2 5 2 7 1 7 2 6
##  [1939] 4 1 5 7 8 6 0 1 8 2 5 7 7 6 5 3 5 8 4 2 4 0 8 8 3 4 9 2 7 5 8 6 5 6
##  [1973] 0 8 6 7 3 6 4 9 4 6 6 3 2 4 1 0 1 4 6 2 9 1 1 0 6 3 9 5 6 5 6 5 9 4
##  [2007] 6 4 3 9 1 3 4 1 9 1 7 1 1 9 3 5 4 0 9 3 6 1 7 5 5 3 3 0 1 5 7 5 8 6
##  [2041] 5 1 0 8 2 3 4 6 7 9 8 1 4 9 9 2 8 6 2 7 0 0 6 7 5 8 6 0 9 3 7 1 3 5
##  [2075] 4 3 3 5 5 6 3 0 2 3 4 2 3 0 9 9 4 7 2 1 4 7 0 6 2 8 5 2 8 5 7 3 0 8
##  [2109] 2 7 2 8 2 5 5 7 6 4 0 8 4 8 2 7 4 5 2 0 3 9 9 6 7 2 5 1 1 1 2 3 6 7
##  [2143] 8 7 6 4 8 9 4 8 6 3 8 3 1 0 6 2 2 5 6 9 5 8 1 4 1 7 8 4 6 1 8 4 3 1
##  [2177] 2 8 0 8 5 9 2 4 2 0 2 7 0 1 0 2 5 7 6 7 9 4 2 6 2 4 4 8 0 4 4 5 8 0
##  [2211] 6 8 9 8 5 6 9 0 4 8 7 1 3 4 5 8 0 9 1 3 3 6 9 8 7 1 0 5 7 1 7 5 2 7
##  [2245] 9 1 8 5 2 4 9 4 7 2 2 3 4 9 1 9 2 1 7 9 4 4 1 6 7 2 7 8 8 1 9 7 1 1
##  [2279] 7 5 3 3 5 1 3 7 6 1 3 8 7 5 9 0 0 0 2 8 8 2 3 7 1 3 0 3 4 4 3 8 9 2
##  [2313] 3 9 7 1 1 7 0 4 9 6 5 9 1 7 0 2 0 2 4 6 7 0 7 1 4 6 4 5 4 9 9 1 7 9
##  [2347] 5 3 3 8 2 3 6 2 2 1 1 1 1 1 6 9 8 4 3 7 1 6 4 8 0 4 7 4 2 4 0 7 0 1
```

```
## [2381] 9 8 8 6 0 0 4 1 6 8 2 2 3 8 4 8 2 2 1 7 5 4 4 0 4 3 4 7 3 1 0 1 2 5
## [2415] 9 2 1 0 1 8 9 1 4 8 3 8 9 3 6 2 8 3 2 2 1 0 4 2 9 2 4 3 7 9 1 5 2 4
## [2449] 9 0 3 8 5 3 6 0 9 4 6 2 5 0 2 7 4 6 6 8 6 6 8 6 9 1 7 2 5 9 9 0 7 2
## [2483] 7 6 7 0 6 5 4 4 7 2 0 9 9 2 2 9 4 4 2 3 3 2 1 7 0 7 6 4 1 3 8 7 4 5
## [2517] 9 2 5 1 8 7 3 7 1 5 5 0 9 1 4 0 6 3 3 6 0 4 9 7 5 1 6 8 9 5 5 7 9 3
## [2551] 8 3 8 1 5 3 5 0 5 5 3 8 6 7 7 7 3 7 0 5 9 0 2 5 5 3 1 7 7 8 6 5 9 3
## [2585] 8 9 5 3 7 9 1 7 0 0 3 7 2 3 8 1 8 6 2 9 5 7 5 7 8 6 2 5 1 4 8 4 5 8
## [2619] 3 0 6 2 7 3 3 2 1 0 7 3 4 0 3 9 3 2 8 9 0 3 8 0 7 6 5 4 7 3 5 0 8 6
## [2653] 2 5 1 1 0 0 4 4 0 1 2 3 2 7 7 8 5 2 5 7 6 9 1 4 1 6 4 2 4 3 5 4 3 9
## [2687] 5 0 1 5 3 8 9 1 9 7 9 5 5 2 7 4 6 0 1 1 1 0 4 4 7 6 3 0 0 4 3 0 6 1
## [2721] 9 6 1 3 8 1 2 5 6 2 7 3 6 0 1 9 7 6 6 8 9 2 9 5 8 3 1 0 0 7 6 6 2 1
## [2755] 6 9 3 1 8 6 9 0 6 0 0 0 6 3 5 9 3 4 5 5 8 5 3 0 4 0 2 9 6 8 2 3 1 2
## [2789] 1 1 5 6 9 8 0 6 6 5 5 3 8 6 2 1 4 5 4 3 7 8 5 0 9 3 5 1 1 0 4 4 7 0
## [2823] 1 7 0 1 6 1 4 5 6 6 5 7 8 4 4 7 2 5 3 7 0 7 7 9 6 4 2 8 5 7 8 3 9 5
## [2857] 8 9 9 8 6 2 8 9 2 3 6 1 1 8 9 3 4 0 7 9 6 7 1 4 1 3 4 9 3 1 4 7 7 4
## [2891] 7 2 9 3 0 8 8 8 4 0 4 4 1 5 2 8 3 4 9 5 2 8 1 5 3 7 9 4 2 5 6 2 5 9
## [2925] 3 5 9 2 1 9 5 3 0 6 9 8 4 0 4 7 2 9 0 1 0 3 1 6 5 8 1 5 3 5 0 3 5 5
## [2959] 9 2 8 7 0 4 9 1 9 7 7 5 3 2 0 9 1 8 6 2 3 9 6 2 1 9 1 3 5 5 0 3 8 3
## [2993] 3 7 6 6 0 1 4 0 6 9 8 1 2 9 9 5 9 7 3 7 8 0 1 3 0 4 6 1 0 2 5 8 4 4
## [3027] 1 1 5 4 6 6 0 6 9 2 6 2 7 1 7 9 4 0 0 3 8 2 2 3 1 6 0 5 7 7 9 2 6 7
## [3061] 7 7 8 6 8 8 4 6 8 4 1 2 8 2 3 9 4 0 3 7 3 2 3 3 7 3 4 0 6 2 0 8 1 5
## [3095] 3 5 4 1 7 1 5 7 5 7 3 2 2 7 3 7 3 7 8 5 4 5 2 9 6 5 3 6 7 4 1 7 1 5
## [3129] 2 3 6 3 1 4 2 6 7 4 3 8 0 6 2 1 6 5 3 9 1 9 3 2 1 8 4 4 6 5 8 6 9 7
## [3163] 7 8 6 9 7 3 9 4 0 5 4 6 4 1 2 3 0 0 2 6 6 5 7 0 8 6 4 7 9 0 7 3 4 2
## [3197] 1 8 8 5 9 2 7 1 8 8 3 2 7 6 0 1 2 7 1 0 8 3 6 0 5 3 6 2 8 7 0 1 4 2
## [3231] 1 1 4 4 4 4 7 1 6 2 9 9 0 0 1 8 8 4 3 4 2 0 6 1 6 1 2 2 2 1 2 3 7 8
## [3265] 1 0 0 2 1 6 6 0 1 6 2 5 1 7 4 8 2 1 4 3 8 3 9 9 4 9 3 4 7 2 7 5 7 0
## [3299] 4 3 3 2 6 7 6 0 0 6 7 7 0 5 5 8 1 0 7 0 2 8 1 5 0 8 8 0 3 2 7 7 2 6
## [3333] 4 7 5 5 9 2 9 2 8 4 6 8 6 5 0 0 8 7 6 1 7 1 1 2 7 4 0 0 7 7 6 3 8 6
## [3367] 4 2 0 9 4 0 5 7 8 2 7 4 7 1 1 3 6 6 2 9 1 9 4 8 3 6 9 5 9 6 2 4 6 7
## [3401] 7 0 6 6 9 4 8 3 5 3 4 9 0 0 5 2 5 0 7 1 1 1 6 7 6 7 9 6 6 4 1 4 3 1
## [3435] 1 2 2 4 1 0 8 7 6 3 4 0 0 6 3 3 0 7 1 7 1 1 3 1 0 9 9 7 5 4 1 4 8 9
## [3469] 5 3 5 1 9 8 2 7 3 9 9 0 1 0 2 9 3 9 3 3 6 2 4 9 8 3 7 4 0 4 7 8 4 9
## [3503] 8 1 9 7 5 9 2 8 2 2 0 2 2 3 8 4 6 8 4 8 2 4 6 7 9 3 3 9 4 3 1 9 4 7
## [3537] 0 5 9 6 0 4 4 4 4 6 1 2 3 2 6 4 5 9 6 8 5 6 0 8 6 4 1 8 6 5 2 8 4 5
## [3571] 5 4 7 7 0 7 8 2 2 3 7 0 1 8 0 7 1 9 8 7 5 5 9 1 7 5 4 3 1 2 2 1 6 6
## [3605] 7 1 1 4 0 7 4 2 4 0 6 4 7 6 9 5 3 4 6 5 0 1 8 8 2 8 3 5 7 8 0 8 5 7
## [3639] 1 1 0 1 3 7 8 5 0 7 1 1 0 1 1 4 5 2 7 6 2 3 0 2 8 5 9 6 9 7 2 1 3 6
## [3673] 4 1 8 2 4 0 5 1 0 2 2 6 4 4 3 9 6 1 6 5 7 9 2 0 2 6 0 1 4 3 5 2 8 8
## [3707] 0 8 8 9 0 9 6 7 6 3 9 3 4 7 7 7 4 9 0 6 4 4 4 2 7 2 8 1 0 0 7 8 3 3
## [3741] 3 1 3 7 6 1 3 1 6 6 5 2 4 7 5 9 5 8 4 9 9 1 6 5 0 1 3 7 0 3 4 8 2 2
## [3775] 0 2 8 1 5 1 6 8 8 9 1 2 1 3 5 1 0 9 4 4 8 3 2 5 9 7 6 6 2 0 0 0 5 8
## [3809] 3 1 5 3 3 8 5 1 8 2 0 4 9 9 6 2 3 3 5 6 4 8 0 9 2 8 3 6 7 5 1 2 9 4
## [3843] 9 1 2 8 6 0 7 0 4 1 1 5 7 5 9 9 1 9 5 9 2 5 0 4 1 0 8 4 0 8 9 8 9 4
## [3877] 2 5 7 9 8 9 8 0 9 9 6 8 9 9 5 9 8 5 1 0 3 3 5 2 1 6 5 0 2 8 2 5 6 2
## [3911] 3 0 2 2 6 4 3 5 5 1 7 2 1 6 9 1 9 9 5 5 1 6 2 2 8 6 7 1 4 6 0 2 0 5
## [3945] 3 2 2 3 6 8 9 8 5 3 8 5 4 5 2 0 5 6 3 2 8 3 9 9 5 7 9 4 6 7 1 3 7 3
## [3979] 6 6 0 9 0 1 9 9 2 8 8 0 1 6 9 7 5 3 4 7 4 9 4 4 3 6 3 1 1 7 6 9 1 8
## [4013] 4 1 1 9 9 4 3 6 8 1 6 0 4 1 3 7 7 4 9 5 1 0 0 1 1 6 2 1 9 8 4 0 3 6
## [4047] 4 9 0 7 1 6 5 7 5 2 5 1 8 5 4 7 0 6 7 3 2 5 8 1 0 4 5 7 1 3 5 1 9 0
## [4081] 0 6 0 7 3 1 8 3 9 7 0 0 8 9 5 9 8 3 2 7 2 9 7 2 1 1 3 7 5 3 1 9 8 2
## [4115] 2 2 8 8 5 7 3 8 9 8 8 6 8 2 3 9 7 5 6 2 9 2 8 8 1 6 2 8 7 9 1 8 0 1
## [4149] 7 2 0 7 5 1 9 0 2 0 9 8 6 2 3 0 3 8 0 2 1 1 1 1 4 2 9 7 7 5 1 1 2 1
## [4183] 9 9 9 1 0 2 0 2 1 1 4 6 4 1 5 4 9 7 7 7 5 6 2 2 2 8 0 6 9 6 1 9 7 7
```

```
## [4217] 1 4 8 5 3 4 3 4 7 7 5 0 7 4 8 8 1 5 3 9 5 9 7 6 9 0 3 6 3 9 8 2 8 1
## [4251] 2 8 6 8 5 5 3 9 4 9 2 5 1 5 1 1 4 1 4 4 3 5 9 1 2 2 3 3 0 2 9 0 0 9
## [4285] 9 6 0 9 3 2 8 4 1 9 9 7 2 7 9 9 5 9 5 1 1 8 3 5 1 9 5 3 5 4 9 5 9 3
## [4319] 1 9 0 9 7 5 4 9 2 0 1 0 5 1 4 9 3 3 6 1 5 2 5 2 2 0 9 2 6 6 0 1 2 0
## [4353] 3 0 2 5 5 7 9 5 3 0 8 9 5 0 3 2 5 4 0 8 8 4 5 8 8 4 5 4 8 5 4 9 2 2
## [4387] 1 2 6 8 8 7 0 3 6 6 4 3 8 8 7 2 2 0 0 9 3 9 9 1 9 8 6 6 4 2 6 9 2 8
## [4421] 5 4 5 7 9 9 9 2 1 8 3 4 0 7 8 3 9 2 4 6 5 6 2 3 9 2 6 0 0 6 1 2 8 7
## [4455] 9 8 2 0 4 7 7 5 0 5 6 4 6 7 4 3 0 7 5 0 7 4 2 0 8 9 9 4 2 4 6 7 8 7
## [4489] 6 9 4 1 3 7 3 0 8 7 7 6 9 3 9 2 2 9 2 1 8 3 2 9 6 8 4 0 1 2 8 4 5 2
## [4523] 7 8 1 1 3 0 3 5 7 0 3 1 9 3 5 3 1 7 7 3 0 8 4 8 2 6 6 2 9 7 3 9 0 9
## [4557] 9 6 4 2 9 7 2 1 1 6 7 4 7 5 9 6 8 2 1 4 4 5 7 6 1 3 2 5 9 9 3 6 1 1
## [4591] 4 6 9 7 2 1 5 1 4 6 3 8 1 1 0 3 1 6 8 4 9 0 7 3 0 2 9 0 6 6 6 3 6 7
## [4625] 7 2 8 6 0 8 3 0 2 9 8 3 2 5 3 8 8 0 0 1 9 5 1 3 9 6 0 1 4 1 7 1 2 3
## [4659] 7 9 7 4 9 9 3 9 2 8 2 7 1 8 0 9 1 0 1 7 7 9 6 9 9 9 2 1 6 1 3 5 7 1
## [4693] 9 7 6 4 5 7 6 6 9 9 6 3 6 2 9 8 1 2 2 5 5 2 3 7 2 1 0 1 0 4 5 2 8 2
## [4727] 8 3 5 1 7 8 1 1 2 9 7 8 4 0 5 0 7 8 8 4 7 7 8 5 8 4 9 8 1 3 8 0 3 1
## [4761] 7 1 5 5 1 6 5 7 4 9 3 5 4 7 1 2 0 8 1 6 0 7 3 4 7 3 9 6 0 8 6 4 8 7
## [4795] 7 9 3 8 6 9 7 2 3 4 0 2 1 0 3 5 5 7 2 4 6 7 2 8 3 0 8 7 8 4 0 8 4 4
## [4829] 5 8 5 6 6 2 0 9 3 7 6 8 9 3 4 9 5 8 9 1 2 8 8 6 8 1 3 7 9 0 1 1 4 7
## [4863] 0 8 1 7 4 5 7 1 2 1 1 3 9 6 2 1 2 8 8 7 6 6 9 3 7 0 5 2 8 0 5 4 3 8
## [4897] 4 6 6 2 7 9 5 1 3 2 4 3 6 1 9 4 4 7 6 5 4 1 9 9 2 7 8 0 1 3 6 1 3 4
## [4931] 1 1 1 5 6 0 7 0 7 2 3 2 5 2 2 9 4 9 8 1 2 1 6 1 2 7 8 0 0 0 8 2 2 9
## [4965] 2 2 1 9 9 2 7 5 1 3 4 9 4 1 8 5 6 2 8 3 1 2 8 4 9 9 3 7 0 7 7 2 3 2
## [4999] 4 0 3 9 9 8 4 1 0 6 0 9 6 8 6 1 1 9 8 9 2 3 5 5 9 4 2 1 9 4 3 9 6 0
## [5033] 4 0 6 0 1 2 3 4 7 8 9 0 1 2 3 4 7 8 9 0 1 2 3 4 5 6 7 8 9 8 3 4 7 8
## [5067] 6 3 4 0 9 7 1 9 3 8 4 7 3 0 9 1 4 5 4 6 2 0 6 2 1 1 1 1 7 2 4 7 5 2
## [5101] 9 4 5 8 4 2 9 7 0 0 7 5 1 1 7 6 6 6 8 2 2 7 7 4 0 2 4 2 1 8 9 6 1 0
## [5135] 5 9 6 9 5 0 3 0 8 3 9 6 3 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 8 9 0 1 2
## [5169] 3 4 5 6 7 8 5 4 8 7 4 7 7 3 9 8 8 3 1 5 8 2 7 4 2 1 5 4 5 5 8 6 4 4
## [5203] 4 1 8 7 5 5 1 8 9 1 3 6 3 3 2 2 6 9 9 6 5 5 3 3 8 1 6 5 6 8 1 9 7 6
## [5237] 8 3 7 4 7 0 9 0 0 3 7 9 3 0 2 0 1 0 1 0 4 0 1 0 4 7 9 6 2 6 2 2 9 9
## [5271] 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 8 0 5 6
## [5305] 6 0 8 0 2 3 7 9 4 7 1 9 1 7 1 4 0 0 4 1 7 5 7 1 3 3 3 6 6 9 7 4 3 0
## [5339] 2 5 2 6 0 8 9 4 3 5 4 8 1 5 9 0 6 4 3 6 3 3 8 1 4 7 5 7 2 2 0 0 1 7
## [5373] 7 9 5 9 8 9 6 8 8 2 3 6 1 2 9 8 9 5 2 6 2 4 8 4 6 5 0 1 5 6 7 8 9 0
## [5407] 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 7 4 2 0 9 0 1 5 8 8 0 2 7 8 4
## [5441] 4 6 1 0 4 5 3 9 4 2 0 5 0 1 3 2 9 8 6 0 1 1 8 0 4 7 7 6 3 6 0 7 3 5
## [5475] 4 2 4 1 8 3 5 6 7 0 6 7 1 2 5 8 1 9 3 8 2 8 7 6 7 1 4 6 2 9 3 0 1 2
## [5509] 3 4 5 6 7 0 1 2 3 4 5 0 1 2 8 9 1 4 0 9 5 0 8 0 7 7 1 1 2 9 3 6 7 2
## [5543] 3 8 1 2 9 8 8 7 1 7 1 1 0 3 4 2 6 4 7 4 2 7 4 9 1 0 6 8 5 5 5 3 5 9
## [5577] 7 4 8 5 9 6 9 3 0 3 8 9 1 8 1 6 0 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
## [5611] 7 8 9 0 1 2 3 4 5 6 7 8 9 3 5 3 2 9 3 2 1 4 5 5 2 3 2 1 3 9 7 2 8 2
## [5645] 8 9 1 8 8 7 8 1 0 0 7 7 8 7 5 0 6 1 5 7 4 6 1 2 5 0 7 9 9 0 3 8 3 4
## [5679] 8 1 8 6 5 9 0 0 0 3 7 1 6 4 2 6 6 0 4 5 4 1 3 8 6 3 9 9 5 9 3 7 8 5
## [5713] 6 4 7 6 2 2 0 9 4 0 1 2 3 4 5 6 7 8 9 0 1 2 7 5 6 0 1 2 3 4 5 6 8 7
## [5747] 1 3 2 8 0 7 5 9 9 6 0 9 4 1 3 2 1 2 3 8 3 2 6 5 6 8 2 7 4 8 1 8 0 5
## [5781] 3 9 4 1 9 2 1 9 6 7 9 0 4 6 1 7 3 8 7 2 9 6 5 8 3 9 0 5 7 1 6 1 0 9
## [5815] 3 3 4 4 0 6 2 5 4 2 3 4 6 0 0 2 0 1 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 0
## [5849] 1 2 3 4 5 6 7 8 9 8 7 1 3 7 5 2 8 0 7 5 9 9 0 9 1 1 5 8 8 6 3 2 1 8
## [5883] 3 2 6 5 6 0 4 1 0 5 3 1 9 2 1 9 6 0 4 6 1 7 3 8 7 2 9 6 5 8 3 5 7 1
## [5917] 6 1 0 9 6 2 5 4 2 3 4 4 6 0 0 2 0 1 2 3 9 5 6 7 8 9 0 1 2 3 4 5 6 7
## [5951] 8 9 0 1 2 8 4 5 6 7 8 9 8 6 5 0 6 8 9 4 1 9 5 9 0 4 8 9 1 4 0 5 5 2
## [5985] 1 5 4 0 7 6 0 1 7 0 6 8 9 5 1 7 9 8 6 0 8 1 7 7 1 3 2 5 1 4 2 0 0 7
## [6019] 8 4 6 4 9 3 8 4 7 2 5 6 3 6 9 6 3 2 2 4 6 9 0 2 5 5 1 3 3 9 7 8 7 2
```

```
## [6053] 2 5 7 9 8 2 1 3 1 3 0 1 2 8 4 5 6 7 8 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3
## [6087] 4 5 6 7 8 9 1 2 6 5 3 0 7 0 4 1 4 3 6 7 2 3 1 2 1 2 9 6 0 1 3 0 2 7
## [6121] 5 7 6 2 9 1 9 0 6 0 6 0 2 0 6 1 5 8 4 3 0 1 5 4 4 8 5 7 5 7 8 3 4 8
## [6155] 8 5 2 9 7 1 3 8 1 0 7 5 3 6 3 4 7 7 9 9 3 4 4 3 8 6 2 0 1 2 3 4 5 6
## [6189] 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 8 3 9 5 5 2 6 8 4 9
## [6223] 1 7 1 2 3 5 9 6 9 1 1 1 2 9 5 6 8 1 2 0 7 7 5 8 2 9 8 9 0 4 6 7 1 3
## [6257] 4 5 6 0 3 6 8 7 0 4 2 7 4 7 5 4 3 4 2 8 1 5 1 2 0 2 5 6 4 3 0 0 0 3
## [6291] 3 5 7 0 6 4 8 8 6 3 4 6 9 9 8 2 7 7 1 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4
## [6325] 5 6 7 8 0 1 2 3 4 5 6 7 8 2 1 7 2 5 0 8 0 2 7 8 8 3 6 0 2 7 6 6 1 2
## [6359] 8 8 7 7 4 7 7 3 7 4 5 4 3 3 8 4 1 1 9 7 4 3 7 3 3 0 2 5 5 6 6 3 5 2
## [6393] 5 9 9 8 4 1 0 6 0 9 6 8 8 5 6 1 1 9 8 9 2 3 5 5 9 4 2 1 9 3 9 2 0 6
## [6427] 0 4 0 0 1 2 3 4 7 8 9 0 1 2 3 7 8 9 0 1 2 3 4 7 8 9 7 3 0 3 1 8 7 6
## [6461] 4 0 2 6 8 3 2 8 1 2 0 7 1 0 4 4 5 8 0 6 2 3 1 5 1 8 5 9 4 0 7 5 8 8
## [6495] 3 8 9 2 6 2 5 3 1 7 3 0 1 9 9 6 0 3 9 2 8 1 4 3 5 2 9 2 5 8 9 5 0 1
## [6529] 2 4 5 6 0 1 2 3 4 5 6 7 1 2 3 4 5 1 0 4 5 6 6 3 4 4 2 9 1 0 6 4 9 7
## [6563] 2 3 3 9 2 0 9 3 3 7 1 5 6 3 1 1 8 4 0 2 4 0 2 4 7 8 0 7 0 6 9 3 2 8
## [6597] 6 7 5 7 5 1 0 8 1 6 7 2 9 7 9 5 8 6 2 6 2 8 1 7 5 0 1 1 3 7 4 9 1 8
## [6631] 6 8 9 0 1 2 3 4 5 6 7 5 9 0 1 2 3 4 7 8 9 5 1 7 8 9 9 8 9 8 4 1 7 7
## [6665] 3 3 7 6 6 6 1 9 0 1 7 6 3 2 1 7 1 3 9 1 7 6 8 4 1 4 3 6 9 6 1 4 4 7
## [6699] 2 4 4 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 9 0 1 2 3 4 7 8 1 3 5 1 7 7
## [6733] 2 1 4 8 3 4 4 3 9 7 4 1 2 3 5 9 1 6 0 1 0 0 2 7 7 1 1 4 0 4 7 3 6 8
## [6767] 0 3 7 4 0 6 9 2 6 5 8 6 9 0 4 0 6 6 9 2 0 9 5 1 3 7 6 9 3 0 2 2 0 1
## [6801] 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 2 1 7 2 5 0
## [6835] 8 0 2 7 8 8 3 0 6 0 2 7 6 6 1 2 8 8 7 7 4 7 7 3 7 4 5 4 3 3 8 4 5 4
## [6869] 1 1 9 7 4 3 7 3 3 0 2 5 5 6 3 1 5 2 5 9 9 8 4 1 0 6 0 9 6 8 8 5 6 1
## [6903] 1 9 8 9 2 3 5 5 9 4 2 1 9 4 9 1 3 9 2 0 6 0 4 0 6 0 1 2 3 4 5 6 7 8
## [6937] 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 3 8 0 7 1 0 7 5 5 6 9 0 1
## [6971] 0 0 8 3 4 3 1 5 0 0 9 5 3 4 9 3 7 6 9 2 4 5 7 2 6 4 9 4 9 4 1 2 2 5
## [7005] 8 1 3 2 9 4 3 8 2 2 1 2 8 6 5 1 6 7 2 1 3 9 3 8 7 5 7 0 7 4 8 8 5 0
## [7039] 6 6 3 7 6 9 9 4 8 4 1 0 6 6 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9
## [7073] 0 1 2 3 4 5 6 7 8 9 7 4 0 4 0 1 7 9 5 1 4 2 8 9 4 3 7 8 2 4 4 3 3 6
## [7107] 9 9 5 8 6 7 0 6 8 2 6 3 9 3 2 8 6 1 7 4 8 8 9 0 3 3 9 0 5 2 9 4 1 0
## [7141] 3 7 5 8 7 7 8 2 9 7 1 2 6 4 2 5 2 3 6 6 5 0 0 2 8 1 6 1 0 4 3 1 6 1
## [7175] 9 0 1 4 5 6 7 8 9 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 8 9 8 4 0 0 7 2 4 3
## [7209] 8 6 6 3 2 6 3 3 6 1 4 7 8 0 3 1 9 0 1 9 1 2 7 0 1 3 8 2 9 2 7 6 5 5
## [7243] 9 9 8 2 9 1 3 2 3 4 3 1 9 0 9 3 6 8 7 0 1 0 5 8 2 7 7 0 1 2 3 4 5 6
## [7277] 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 1 7 4 8 1 5 6 5 7 2 8
## [7311] 6 3 3 8 6 5 4 0 9 1 7 2 9 1 5 1 3 2 2 3 0 6 4 3 7 6 9 0 4 8 1 4 0 6
## [7345] 1 2 6 9 2 2 3 5 5 1 0 7 7 9 6 2 9 4 7 0 2 3 4 0 0 8 8 8 5 1 3 7 4 9
## [7379] 8 8 9 0 9 8 9 0 2 6 5 6 7 4 7 5 4 1 3 5 3 1 2 3 4 5 6 1 2 3 4 6 0 1
## [7413] 2 4 5 6 7 8 1 7 2 4 1 4 1 4 9 6 8 4 5 3 7 8 8 3 3 5 6 7 0 6 1 6 8 7
## [7447] 0 1 5 0 8 5 0 1 5 8 4 2 3 9 7 6 9 1 9 0 6 7 1 2 3 9 2 4 5 5 3 7 5 3
## [7481] 1 8 2 2 3 0 2 9 4 9 7 0 2 7 4 9 9 2 5 9 8 3 8 6 7 0 0 1 2 3 4 5 6 7
## [7515] 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 0 7 2 6 5 5 3 7 8 6 6
## [7549] 6 6 4 3 8 8 3 0 1 9 0 5 4 1 9 1 2 7 0 1 3 8 2 9 2 7 4 2 6 5 5 9 9 1
## [7583] 1 5 7 6 8 2 9 4 3 1 9 0 9 3 6 8 7 0 1 0 5 8 2 7 7 0 1 2 3 4 5 6 7 8
## [7617] 9 0 1 2 3 4 5 8 9 0 1 2 3 4 5 6 7 8 9 2 1 2 1 3 9 9 8 5 3 7 0 7 7 5
## [7651] 7 9 9 4 7 0 3 4 1 5 8 1 4 8 4 1 8 6 6 4 6 0 5 5 3 3 5 7 2 5 9 6 9 2
## [7685] 6 2 1 2 0 8 3 8 3 0 8 7 4 9 5 0 9 7 0 0 4 6 0 9 1 6 2 7 6 8 3 5 2 1
## [7719] 8 3 8 6 1 0 2 1 4 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4
## [7753] 5 6 7 8 9 7 6 4 7 6 2 3 4 8 7 8 6 9 8 3 2 2 8 4 8 5 6 5 0 2 0 1 1 2
## [7787] 9 6 8 2 1 0 6 5 2 9 7 5 3 9 3 7 1 8 3 8 1 9 5 5 0 1 1 9 8 2 6 0 4 5
## [7821] 0 3 1 8 6 7 5 9 9 3 0 3 1 4 4 0 4 9 0 1 2 3 5 6 7 8 0 1 2 3 5 6 7 8
## [7855] 9 0 1 2 3 5 6 7 8 9 9 7 0 9 0 1 5 8 8 0 9 3 2 7 8 4 6 1 0 4 9 4 2 0
```

```
## [7889] 5 0 1 6 9 3 2 9 1 6 0 8 1 8 7 7 6 3 6 0 7 2 4 1 7 0 6 7 1 2 5 8 1 2
## [7923] 2 8 7 6 8 7 1 6 2 9 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
## [7957] 3 4 5 6 7 8 9 8 9 5 7 0 3 1 6 8 4 1 5 6 4 2 7 8 1 3 4 3 4 7 2 0 5 0
## [7991] 1 9 2 3 2 3 5 5 7 8 4 9 9 7 1 1 9 0 7 8 3 4 8 6 3 8 0 9 6 2 1 0 1 0
## [8025] 6 2 3 8 9 0 7 2 3 4 5 5 2 8 5 4 6 6 6 7 9 1 8 2 1 5 3 4 7 9 4 0 0 0
## [8059] 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 9 0 1 3 1 5 1 2
## [8093] 4 9 8 4 6 8 0 1 1 9 2 6 6 8 7 4 2 9 7 0 2 1 0 3 6 0 1 2 3 4 5 6 7 8
## [8127] 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 8 6 5 9 7 0 2 3 4 3 8 5 1
## [8161] 5 2 3 0 1 2 1 3 2 6 5 3 0 7 2 7 4 6 4 0 5 9 9 5 9 5 3 1 7 4 7 6 5 4
## [8195] 0 0 6 6 2 0 6 3 7 7 4 4 3 9 2 8 9 6 0 9 5 3 8 8 7 1 4 0 4 8 5 2 3 9
## [8229] 0 1 9 1 5 1 7 4 8 6 2 1 6 8 8 0 1 2 3 4 7 8 9 0 1 2 3 4 6 7 8 9 0 1
## [8263] 2 3 4 7 8 9 1 4 5 3 3 0 9 5 4 3 0 8 4 6 7 0 7 7 1 6 9 1 3 6 2 3 8 2
## [8297] 3 8 9 5 8 8 7 1 7 1 1 0 3 4 2 6 4 7 4 2 7 4 2 9 2 7 9 2 1 6 6 5 3 4
## [8331] 8 5 9 6 9 0 6 3 0 8 1 6 0 0 1 2 3 4 5 6 7 0 1 2 3 4 7 8 9 0 1 2 3 4
## [8365] 7 2 5 1 6 4 3 9 9 0 9 7 1 6 4 3 6 2 0 9 8 6 5 7 0 0 1 7 4 3 2 4 1 3
## [8399] 7 6 4 7 7 7 9 8 4 3 6 2 8 3 5 8 0 5 4 7 1 3 1 7 9 6 2 0 9 1 7 3 3 9
## [8433] 1 6 4 3 9 8 2 1 8 6 4 1 5 5 6 5 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7
## [8467] 8 9 0 1 2 3 4 5 6 7 8 9 6 9 7 0 2 3 4 3 8 5 1 3 0 1 2 1 3 2 0 7 2 6
## [8501] 4 0 5 9 9 8 9 5 3 1 7 4 7 0 0 6 6 6 3 7 9 2 8 9 8 7 1 8 0 4 8 5 2 3
## [8535] 9 0 1 9 1 5 1 7 6 1 2 1 6 8 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 0
## [8569] 1 2 3 5 6 7 8 1 0 4 5 6 6 3 4 4 2 8 1 0 6 4 9 7 2 9 2 0 9 3 3 9 1 5
## [8603] 2 3 1 6 7 3 7 8 4 0 2 4 0 2 4 7 8 0 7 0 6 9 3 2 4 8 6 0 5 7 5 1 0 8
## [8637] 1 6 7 2 9 7 9 5 6 5 2 6 2 8 1 7 5 5 7 3 5 0 1 1 3 8 4 9 4 5 1 8 6 8
## [8671] 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 3 5 3
## [8705] 2 9 3 2 1 4 5 5 2 3 2 1 3 9 7 2 1 2 8 9 1 8 8 7 8 1 0 0 6 7 7 8 7 5
## [8739] 0 6 1 5 7 4 6 1 2 5 0 7 9 9 0 3 4 4 8 4 1 8 6 5 9 0 0 0 3 7 1 6 4 6
## [8773] 0 4 5 4 1 3 8 6 3 9 9 5 9 3 7 8 5 6 4 7 6 2 2 0 9 4 0 1 2 3 4 5 6 7
## [8807] 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 6 4 2 6 4 7 5 5 4 7 2 9
## [8841] 3 9 3 8 2 0 9 5 6 0 1 0 6 5 3 5 3 8 0 0 3 4 1 5 3 0 8 3 0 6 2 7 8 1
## [8875] 7 1 3 8 5 4 2 0 9 7 6 7 4 1 6 2 6 7 1 9 8 0 6 9 4 9 9 6 2 3 7 1 9 2
## [8909] 2 5 3 7 8 0 1 2 3 4 7 8 9 0 1 2 3 4 7 8 9 0 1 7 8 9 8 9 2 6 1 3 5 4
## [8943] 8 2 6 4 3 4 5 9 2 0 3 9 4 9 7 3 8 7 4 4 9 8 5 8 2 6 6 2 3 1 3 2 7 3
## [8977] 1 9 0 1 1 3 5 0 7 8 1 5 1 4 6 0 0 4 9 1 6 6 9 0 7 6 1 1 0 1 2 3 4 2
## [9011] 2 3 4 5 6 2 0 1 2 2 8 6 3 9 2 1 9 3 9 6 1 7 2 4 4 5 7 0 0 1 6 6 8 2
## [9045] 7 7 2 4 2 1 6 1 0 6 9 8 3 9 6 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7
## [9079] 8 9 0 1 2 3 4 5 6 7 8 9 1 6 8 9 9 0 1 2 4 4 3 7 4 4 4 0 3 8 7 5 8 2
## [9113] 1 7 5 3 8 5 2 5 1 1 6 2 1 3 8 6 4 2 6 2 5 5 0 2 8 0 6 8 1 7 9 1 9 2
## [9147] 6 7 6 6 8 7 4 9 2 1 3 3 0 5 5 8 0 3 7 9 7 0 2 7 9 1 7 8 0 3 5 3 6 0
## [9181] 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 7 8 9 6 4 2 6 4 7 8
## [9215] 9 2 9 3 9 3 0 0 1 0 4 2 6 3 5 3 0 3 4 1 5 3 0 8 3 0 6 1 7 8 0 9 2 6
## [9249] 7 1 9 6 9 4 9 9 6 7 1 2 5 3 7 8 0 1 2 4 5 6 7 8 9 0 1 3 4 5 6 7 5 0
## [9283] 1 3 4 7 8 9 7 5 5 1 9 9 7 1 0 0 5 9 7 1 7 2 2 3 6 8 3 2 0 0 6 1 7 5
## [9317] 8 6 2 9 4 8 8 7 1 0 8 7 7 5 8 5 3 4 6 1 1 5 5 0 7 2 3 6 4 1 2 4 1 5
## [9351] 4 2 0 4 8 6 1 9 0 2 5 6 9 3 6 3 6 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
## [9385] 7 8 9 0 1 2 3 5 6 7 8 1 0 9 5 7 5 1 8 6 9 0 4 1 9 3 8 4 4 7 0 1 9 2
## [9419] 8 7 8 2 5 9 6 0 6 5 5 3 3 3 9 8 1 1 0 6 1 0 0 6 2 1 1 3 2 7 7 8 8 7
## [9453] 8 4 6 0 2 0 7 0 3 6 8 7 1 5 9 9 3 7 2 4 9 4 3 6 2 2 5 3 2 5 5 9 4 1
## [9487] 7 2 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 1 0
## [9521] 1 2 7 5 3 4 4 0 0 6 9 6 6 5 7 2 3 4 4 9 1 4 0 7 9 5 7 2 3 1 4 4 0 9
## [9555] 9 6 1 8 3 3 7 3 9 8 8 4 7 7 6 2 1 9 8 7 8 8 7 2 2 3 9 3 3 5 5 0 7 4
## [9589] 5 6 5 1 4 1 1 2 8 2 6 1 5 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
## [9623] 1 2 3 4 5 6 7 8 8 0 6 0 1 2 3 7 9 4 7 1 7 1 7 1 4 0 0 1 7 5 7 1 3 3
## [9657] 3 1 6 9 7 1 3 0 7 6 0 8 9 7 3 5 4 8 1 5 9 0 6 5 3 8 1 4 7 5 2 0 0 1
## [9691] 7 8 7 6 8 8 2 3 6 1 2 9 5 2 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9
```

```
##  [9725] 0 1 2 3 4 6 6 7 8 9 7 4 6 1 4 0 4 9 3 7 8 0 7 5 8 5 3 2 2 0 5 8 6 0
##  [9759] 3 8 1 0 3 0 4 7 4 9 0 9 0 7 1 7 1 6 6 5 6 2 8 7 6 4 9 9 5 3 7 4 3 0
##  [9793] 7 6 6 1 1 3 2 1 0 0 1 2 3 4 7 8 9 0 1 2 3 4 5 6 7 8 0 1 2 3 4 7 8 9
##  [9827] 0 8 3 9 5 5 2 6 8 4 1 7 1 7 3 5 6 9 1 1 1 2 1 2 0 7 7 5 8 2 9 8 6 7
##  [9861] 3 4 6 8 7 0 4 2 7 7 5 4 3 4 2 8 1 5 1 0 2 3 3 5 7 0 6 8 6 3 9 9 8 2
##  [9895] 7 7 1 0 1 7 8 9 0 1 0 3 4 5 6 7 8 0 1 2 3 4 7 8 9 7 8 6 4 1 9 3 1 4
##  [9929] 4 7 0 1 9 2 8 7 8 2 6 0 6 5 3 3 3 9 1 4 0 6 1 0 0 6 2 1 1 7 7 8 4 6
##  [9963] 0 7 0 3 6 8 7 1 5 2 4 9 4 3 6 4 1 7 3 6 6 0 1 2 3 4 5 6 7 8 9 0 1 2
##  [9997] 3 4 5 6
```

Now that we've got our tools up and running on the MNIST dataset, we'll move on to the more interesting Fashion MNIST dataset.

```r
# Importing the Fashion MNIST dataset
fashion_mnist <- dataset_fashion_mnist()
# Splitting the data into testing and training sets
c(train_images, train_labels) %<-% fashion_mnist$train
c(test_images, test_labels) %<-% fashion_mnist$test
# Creating a vector of class names
class_names = c("T-shirt/top", "Trouser", "Pullover", "Dress", "Coat",
    "Sandal", "Shirt", "Sneaker", "Bag", "Ankle boot")
# Exploring the dimensions and characteristics of the data
dim(train_images)
```

```
## [1] 60000    28    28
```

```r
dim(train_labels)
```

```
## [1] 60000
```

```r
train_labels[1:20]
```

```
##  [1] 9 0 0 3 0 2 7 2 5 5 0 9 5 5 7 9 1 0 6 4
```

```r
dim(test_images)
```

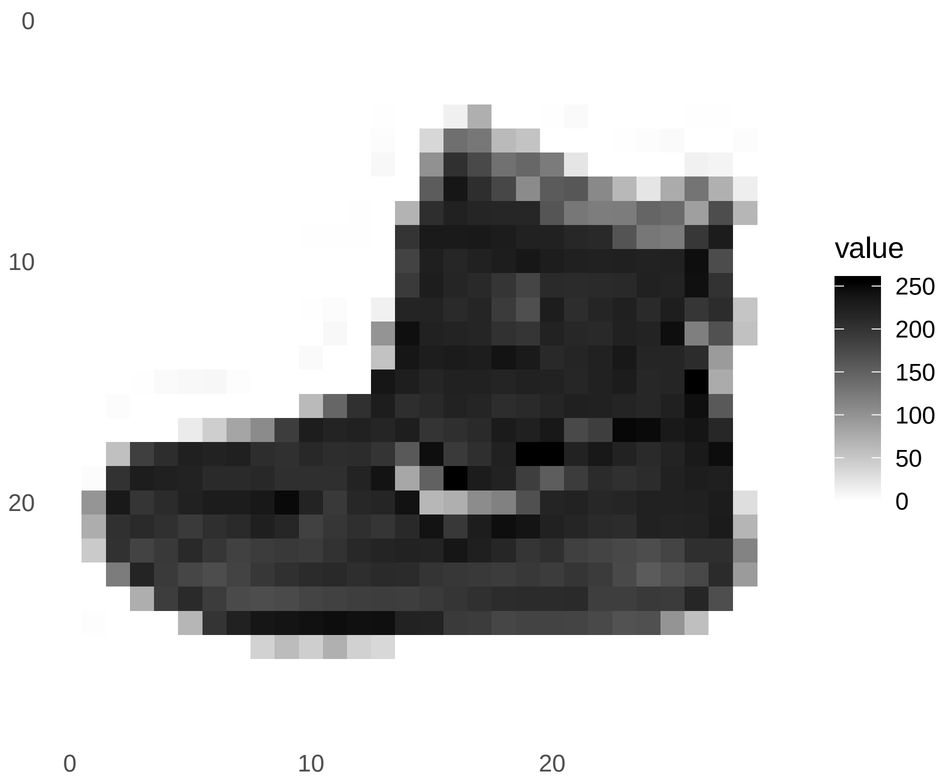
```
## [1] 10000    28    28
```

```r
dim(test_labels)
```

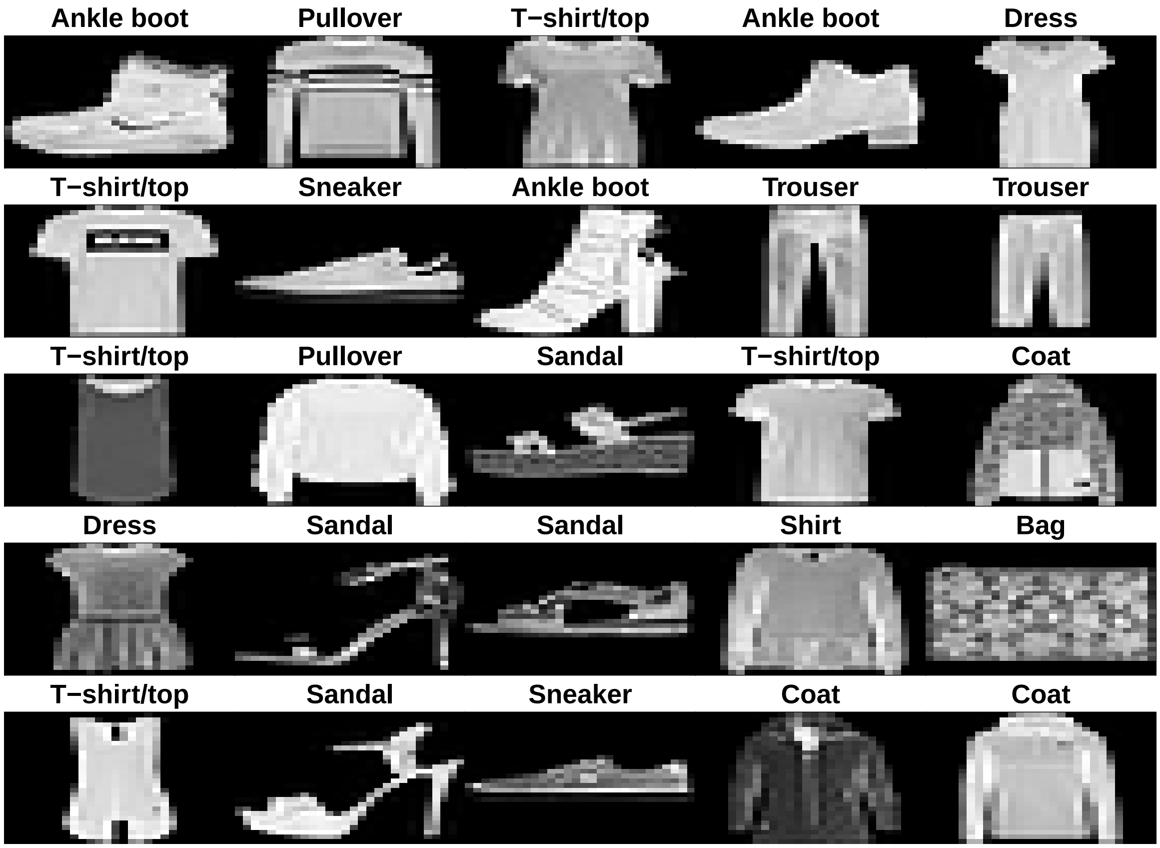
```
## [1] 10000
```

```r
# Examining a sample of the data
image_1 <- as.data.frame(train_images[1, , ])
colnames(image_1) <- seq_len(ncol(image_1))
image_1$y <- seq_len(nrow(image_1))
image_1 <- gather(image_1, "x", "value", -y)
image_1$x <- as.integer(image_1$x)

ggplot(image_1, aes(x = x, y = y, fill = value)) + geom_tile() + scale_fill_gradient(low = "white",
    high = "black", na.value = NA) + scale_y_reverse() + theme_minimal() +
    theme(panel.grid = element_blank()) + theme(aspect.ratio = 1) +
    xlab("") + ylab("")
```




```
# Converting grayscale into integer values
train_images <- train_images/255
test_images <- test_images/255
# Visually examining the training set images
par(mfcol = c(5, 5))
par(mar = c(0, 0, 1.5, 0), xaxs = "i", yaxs = "i")
for (i in 1:25) {
    img <- train_images[i, , ]
    img <- t(apply(img, 2, rev))
    image(1:28, 1:28, img, col = gray((0:255)/255), xaxt = "n", yaxt = "n",
        main = paste(class_names[train_labels[i] + 1]))
}
```




```
# Setting up the keras sequential model
model <- keras_model_sequential()
model %>% layer_flatten(input_shape = c(28, 28)) %>% layer_dense(units = 128,
    activation = "relu") %>% layer_dense(units = 10, activation = "softmax")
# Setting up the model's loss function and other metrics
model %>% compile(optimizer = "adam", loss = "sparse_categorical_crossentropy",
    metrics = c("accuracy"))
# training the model
model %>% fit(train_images, train_labels, epochs = 10)
# Evaluating and displaying model accuracy
score <- model %>% evaluate(test_images, test_labels)

cat("Test loss:", score$loss, "\n")
```

```
## Test loss: 0.3392665
```

```
cat("Test accuracy:", score$acc, "\n")
```

```
## Test accuracy: 0.8821
```

```
# Making predictions on the test data
predictions <- model %>% predict(test_images)
# Looking at first prediction
predictions[1, ]
```

```
##  [1] 3.023626e-05 1.135101e-08 6.872975e-07 2.134584e-08 3.637361e-06
##  [6] 2.566708e-02 6.530133e-06 3.310594e-02 9.912669e-07 9.411848e-01
```

```r
# the label associated with the previous prediction (1-10 scale)
which.max(predictions[1, ])
```

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

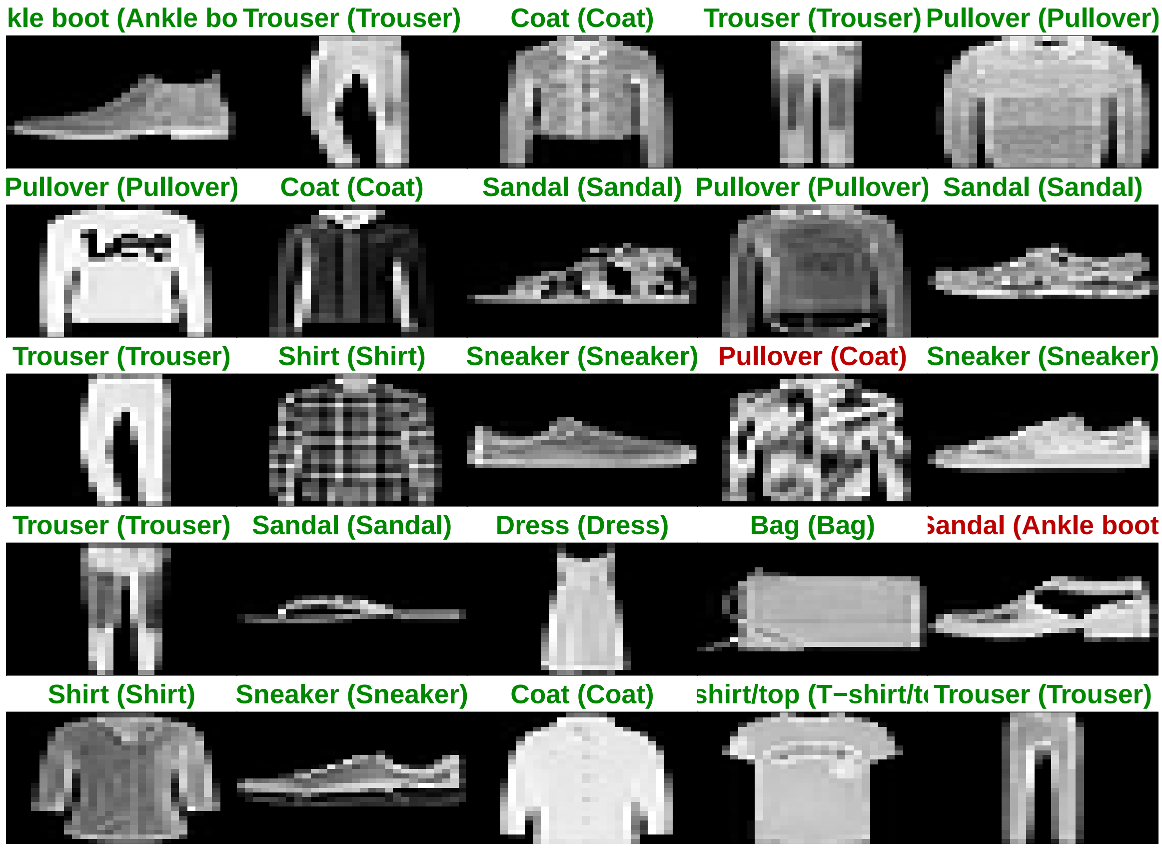
```r
# Another way to retrieve the first twenty predictions
class_pred <- model %>% predict_classes(test_images)
class_pred[1:20]
```

```
##  [1] 9 2 1 1 6 1 4 6 5 7 4 5 7 3 4 1 2 2 8 0
```

```r
# Retrieving first prediction label again (0-9) scale
test_labels[1]
```

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

```r
# Plotting sample of first 25 predictions
par(mfcol = c(5, 5))
par(mar = c(0, 0, 1.5, 0), xaxs = "i", yaxs = "i")
for (i in 1:25) {
    img <- test_images[i, , ]
    img <- t(apply(img, 2, rev))
    # subtract 1 as labels go from 0 to 9
    predicted_label <- which.max(predictions[i, ]) - 1
    true_label <- test_labels[i]
    if (predicted_label == true_label) {
        color <- "#008800"
    } else {
        color <- "#bb0000"
    }
    image(1:28, 1:28, img, col = gray((0:255)/255), xaxt = "n", yaxt = "n",
        main = paste0(class_names[predicted_label + 1], " (", class_names[true_label +
            1], ")"), col.main = color)
}
```




```r
# Making a single prediction
img <- test_images[1, , , drop = FALSE]
dim(img)
```

```
## [1]  1 28 28
```

```r
predictions <- model %>% predict(img)
predictions
```

```
##              [,1]         [,2]         [,3]         [,4]         [,5]
## [1,] 3.023629e-05 1.135103e-08 6.872995e-07 2.134576e-08 3.637361e-06
##            [,6]         [,7]       [,8]         [,9]     [,10]
## [1,] 0.02566708 6.530126e-06 0.03310595 9.912669e-07 0.9411848
```

```r
# Prediction label (1-10 scale)
prediction <- predictions[1, ] - 1
which.max(prediction)
```

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

```r
# Prediction label (0-9 scale)
class_pred <- model %>% predict_classes(img)
class_pred
```

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