MNIST Dataset

1 Introduction

This dataset contains 60,000 28x28 grayscale photos of the ten digits, as well as a test set of 10,000 images. More information is available at the MNIST homepage.

5
9
1
3
0
2
4
6
4
1
3
1
1
3
5
7

MNIST dataset can be easily retrieved using `dataset_MNIST()` function from library `keras`. Because this is a grayscale image dataset, every data point is an image (2D matrix of pixels) and each pixel of an image has value within the range from 0 (black) to 255 (white). Therefore, for ease of handling, we can reshape it to lay out all the pixels as features.

```
mnist <- dataset_mnist()$train
x <- mnist$x
y <- mnist$y
# reshape
dim(x) <- c(nrow(x), 784)
# convert to data.frame
x <- as.data.frame(x)
y <- as.data.frame(y) %>% rename(label = y)
```

```r
mnist_features <- x %>% as_tibble()
mnist_df <- cbind(y, x) %>% as_tibble()
skimmed_mnist_10_features <- skimr::skim(mnist_df[,1:11])
print(mnist_df)
```

```
## # A tibble: 60,000 x 785
##    label    V1    V2    V3    V4    V5    V6    V7    V8    V9   V10   V11   V12
##    <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
##  1     5     0     0     0     0     0     0     0     0     0     0     0     0
##  2     0     0     0     0     0     0     0     0     0     0     0     0     0
##  3     4     0     0     0     0     0     0     0     0     0     0     0     0
##  4     1     0     0     0     0     0     0     0     0     0     0     0     0
##  5     9     0     0     0     0     0     0     0     0     0     0     0     0
##  6     2     0     0     0     0     0     0     0     0     0     0     0     0
##  7     1     0     0     0     0     0     0     0     0     0     0     0     0
##  8     3     0     0     0     0     0     0     0     0     0     0     0     0
##  9     1     0     0     0     0     0     0     0     0     0     0     0     0
## 10     4     0     0     0     0     0     0     0     0     0     0     0     0
## # ... with 59,990 more rows, and 772 more variables: V13 <int>, V14 <int>,
## #   V15 <int>, V16 <int>, V17 <int>, V18 <int>, V19 <int>, V20 <int>,
## #   V21 <int>, V22 <int>, V23 <int>, V24 <int>, V25 <int>, V26 <int>,
## #   V27 <int>, V28 <int>, V29 <int>, V30 <int>, V31 <int>, V32 <int>,
## #   V33 <int>, V34 <int>, V35 <int>, V36 <int>, V37 <int>, V38 <int>,
## #   V39 <int>, V40 <int>, V41 <int>, V42 <int>, V43 <int>, V44 <int>,
## #   V45 <int>, V46 <int>, V47 <int>, V48 <int>, V49 <int>, V50 <int>, ...
```

2 Exploratory Data Analysis (EDA)

In this part, we will walk through a few characteristics of MNIST dataset.

2.1 Numerical variables

First let's check if the MNIST dataset has any missing values.

```r
mnist_numerical <- mnist_df %>% select(where(is.numeric))
# Number of numerical features
ncol(mnist_numerical)
```

```
## [1] 785
```

```r
# List any numerical features having more than one NA value
names(which(colSums(is.na(mnist_numerical))>0))
```

```
## character(0)
```

As can be seen, there is no missing value found in the features (image pixels) or the labels.

General statistics about the first 10 numerical variables from MNIST dataset:

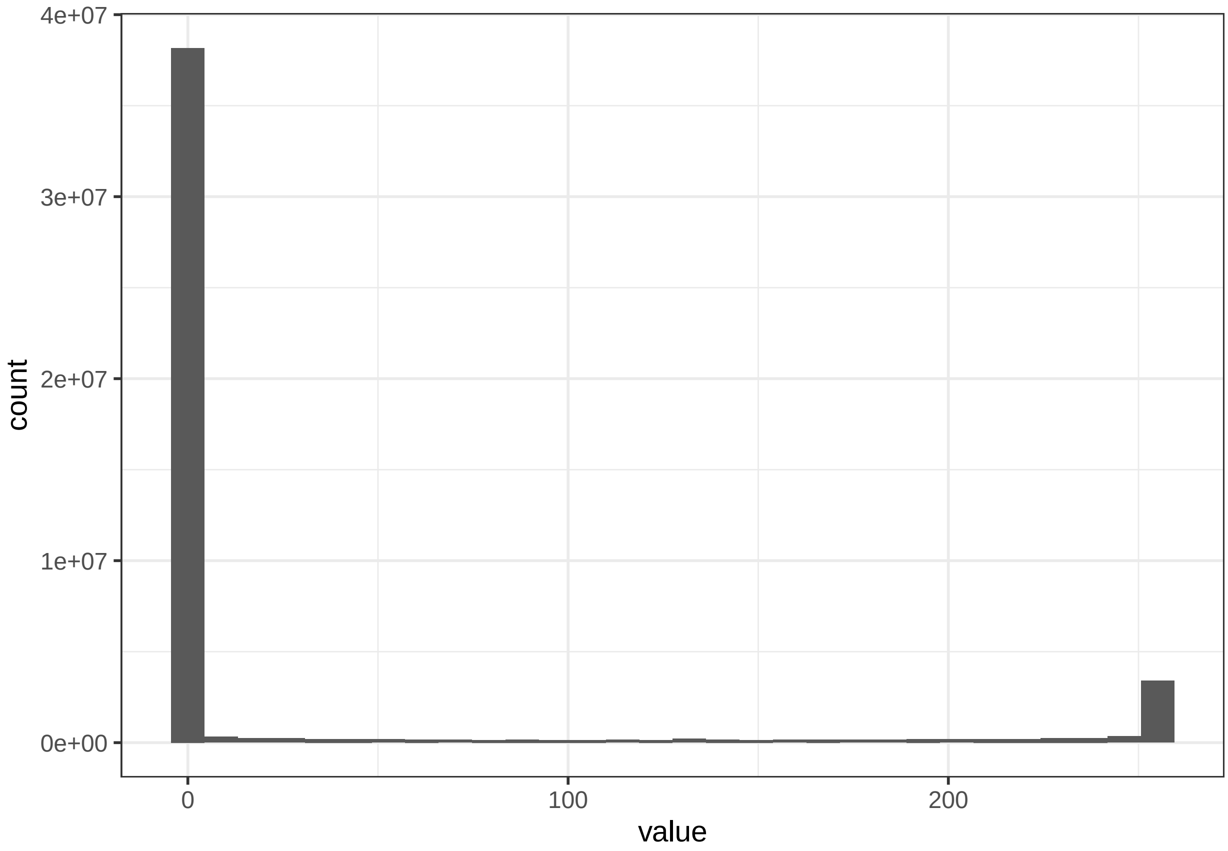
```
skimr::partition(skimmed_mnist_10_features)$numeric %>%
        knitr::kable(format = 'latex', booktabs = TRUE) %>%
        kableExtra::kable_styling(latex_options = 'HOLD_position')
```

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
label	0	1	4.4539333	2.8892704	0	2	4	7	9	
V1	0	1	0.0000000	0.0000000	0	0	0	0	0	
V2	0	1	0.0000000	0.0000000	0	0	0	0	0	
V3	0	1	0.0000000	0.0000000	0	0	0	0	0	
V4	0	1	0.0000000	0.0000000	0	0	0	0	0	
V5	0	1	0.0000000	0.0000000	0	0	0	0	0	
V6	0	1	0.0000000	0.0000000	0	0	0	0	0	
V7	0	1	0.0000000	0.0000000	0	0	0	0	0	
V8	0	1	0.0007833	0.1918767	0	0	0	0	47	
V9	0	1	0.0040667	0.8092143	0	0	0	0	191	
V10	0	1	0.0048667	0.8240677	0	0	0	0	184	

From the first 10 pixels, we can see that the majority of values is 0.

We can also take a look into the distribution of values of all the pixels in the dataset:

```
mnist_features %>%
  melt() %>%
  ggplot() + geom_histogram(aes(x=value))
```



Similar to the observation from the first 10 pixels, the majority of the pixels has value 0 (black) and the second-highest frequency value is 255 (white). This corresponds to the fact that the images in this dataset have black background and white numbers on them.

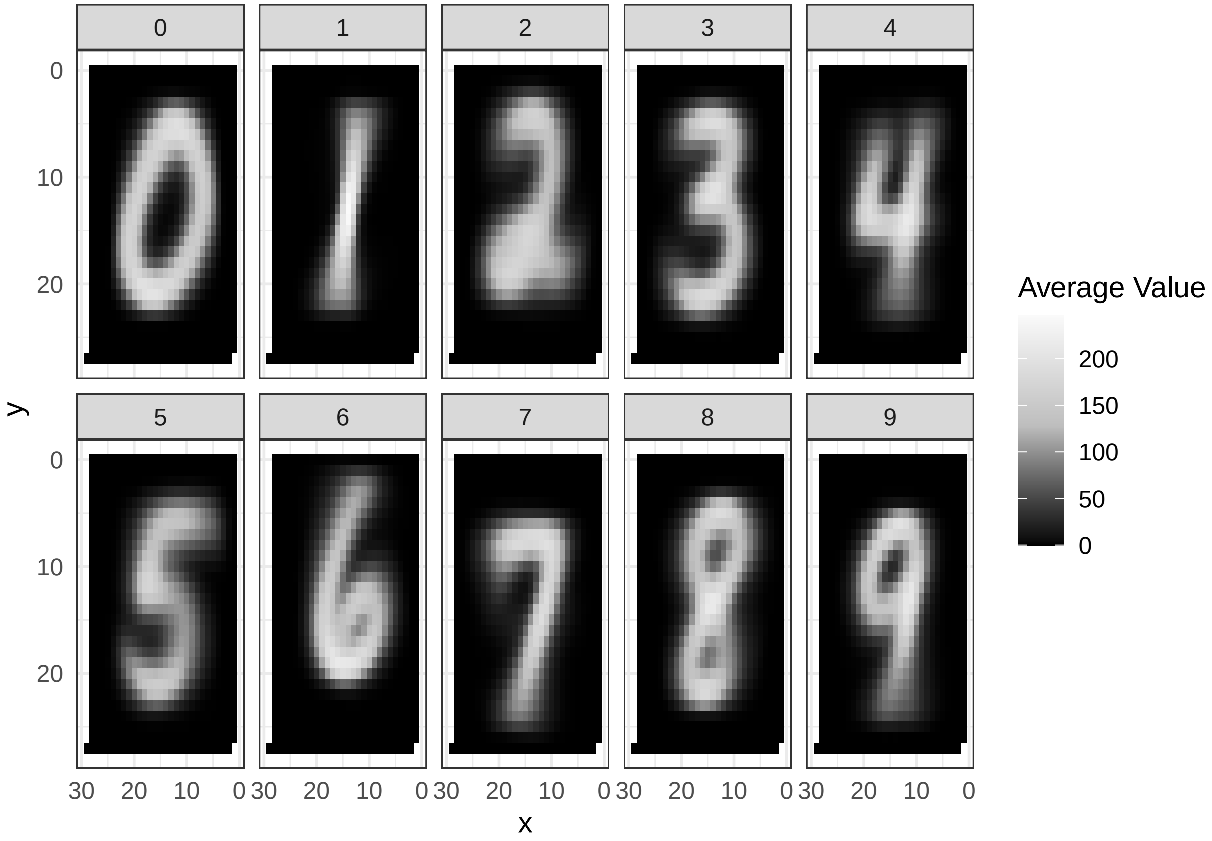
We are also interested in whether the images within the same class share similar characteristics if we average out their corresponding pixels. To do that, we create an average summary for pixels as follows:

```
pixels_summary <- mnist_df %>%
  tidyr::gather(pixel, value, -label) %>%
  tidyr::extract(pixel, "pixel", "(\\d+)", convert = TRUE) %>%
  mutate(pixel = pixel - 2,
         y = pixel %% 28,
         x = 28 - pixel %/% 28) %>%
  group_by(x, y, label) %>%
  summarize(mean_value = mean(value)) %>%
  ungroup()
print(pixels_summary)
```

```
## # A tibble: 7,840 x 4
##        x     y label mean_value
##    <dbl> <dbl> <int>      <dbl>
##  1     1     0     0          0
##  2     1     0     1          0
##  3     1     0     2          0
##  4     1     0     3          0
##  5     1     0     4          0
##   6     1     0     5           0
##   7     1     0     6           0
##   8     1     0     7           0
##   9     1     0     8           0
## 10     1     0     9           0
## # ... with 7,830 more rows
```

And then use `geom_tile` from `ggplot` for visualization:

```
pixels_summary %>%
  ggplot(aes(x, y, fill = mean_value)) +
  geom_tile() +
  scale_fill_gradient2(low = "black", high = "white", mid = "gray", midpoint = 127.5) +
  facet_wrap(~ label, nrow = 2) +
  scale_x_reverse() +
  scale_y_reverse() +
  labs(fill="Average Value") +
  theme(axis.ticks.x = element_blank(),
        axis.ticks.y = element_blank())
```



From the plot, we can see that using our human eyes, it is still easy to distinguish numbers within these averaged-out images. This indicates that there is not much variability between images from the same class.