

# PCA\_t-SNE.R

evegas

2022-03-15

```
#  
# Principal Component Analysis (PCA) & t-SNE  
#  
  
# load libraries  
library(ggplot2)  
library(RColorBrewer)  
library(Rtsne)  
  
## Warning: package 'Rtsne' was built under R version 4.1.3  
library(ggrepel)  
  
# Load Data: enc_output.cifra & y_test_cifra  
# from AE-MNIST.R script which do an autoencoder network  
load("Encod_0123456789.RData")  
  
# Load Data: enc_output.cifra & y_test_cifra  
# from CAE-MNIST.R script which do an autoencoder network  
# load("Conv_Encod_Flat_0123456789.RData")  
  
# Change r objects names  
#  
  
data_to_project <- enc_output_cifra  
v_labels <- as.factor(y_test_cifra)  
n_v_labels <- nlevels(v_labels)  
  
  
#cifra <- as.factor(y_test_cifra)  
  
  
# k is the number of images to project  
k <- 8000 # nrow(data_to_project)  
  
  
  
  
# Principal Component Analysis  
  
# PCA function
```

```

plotPCA3 <- function (datos, labels, factor,title,scale,colores, size = 2, glineas = 0.25) {
  data <- prcomp(datos , scale = scale)
  dataDf <- data.frame(data$x)
  Group <- factor
  loads <- round(data$sdev^2/sum(data$sdev^2)*100,1)
  # the graphic
  p1 <- ggplot(dataDf,aes(x=PC1, y=PC2)) +
    theme_classic() +
    geom_hline(yintercept = 0, color = "gray70") +
    geom_vline(xintercept = 0, color = "gray70") +
    geom_point(aes(color = Group), alpha = 0.55, size = 3) +
    coord_cartesian(xlim = c(min(data$x[,1])-5,max(data$x[,1])+5)) +
    scale_fill_discrete(name = "")
  # the graphic with ggrepel
  p1 + geom_text_repel(aes(y = PC2 + 0.25, label = labels),segment.size = 0.25, size = size) +
    labs(x = c(paste("PC1",loads[1],"%")),y=c(paste("PC2",loads[2],"%")))) +
    ggtitle(paste("PCA based on", title, sep=" ")) +
    theme(plot.title = element_text(hjust = 0.5)) +
    scale_color_manual(values=colores)
}

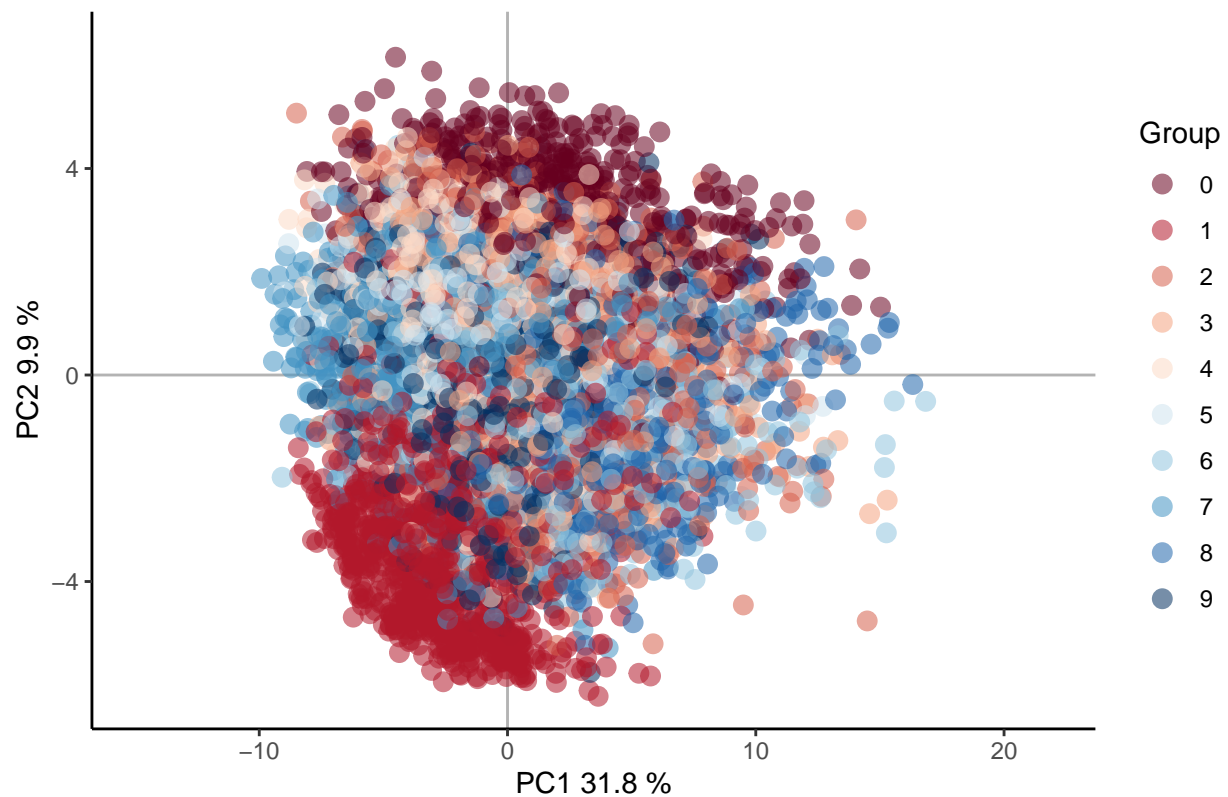
scale = FALSE

if (n_v_labels > 2) {
  colores <- brewer.pal(n = n_v_labels, name = "RdBu")
} else {
  colores <- c("red","blue")
}

plotPCA3(datos = data_to_project[1:k,],
  labels = rep("",k),
  factor = v_labels[1:k],
  scale = scale,
  title = paste ("last encode layer.", "# Samples:", k),
  colores = colores)

```

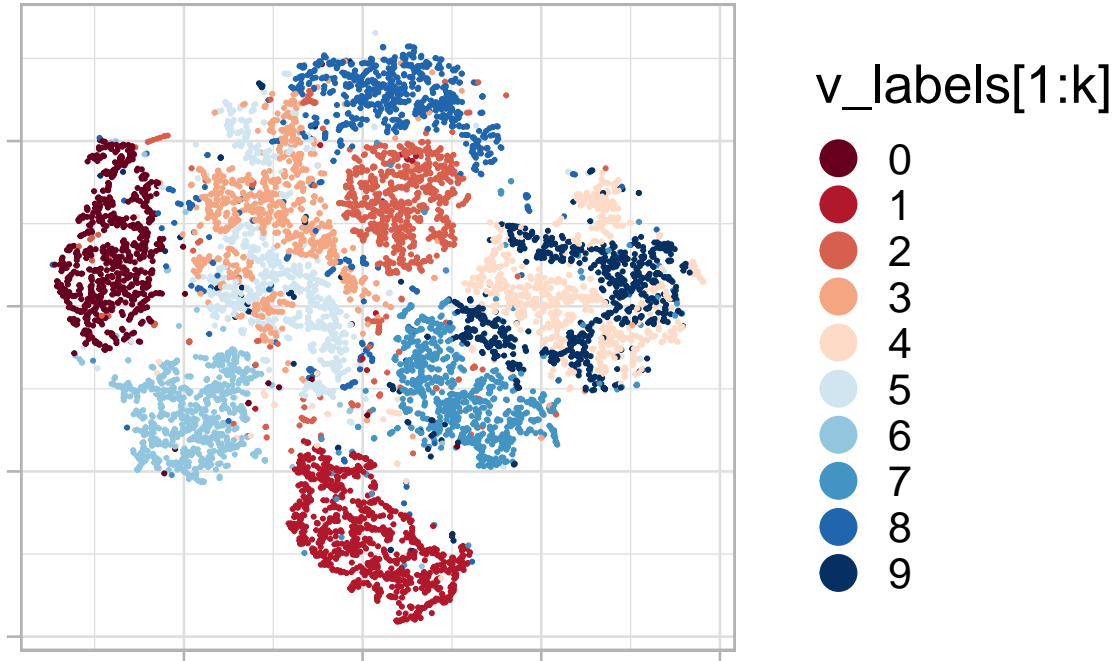
PCA based on last encode layer. # Samples: 8000



```
#  
# t-SNE Representation  
#  
# The next code of t-SNE reduction dimension is based on  
# https://www.r-bloggers.com/playing-with-dimensions-from-clustering-pca-t-sne-to-carl-sagan/  
  
## Rtsne function may take some minutes to complete...  
set.seed(123456)  
  
tsne_model_1 = Rtsne(data_to_project[1:k,],  
                      check_duplicates=FALSE,  
                      pca=TRUE,  
                      perplexity=30,  
                      theta=0.5,  
                      dims=2)  
  
## getting the two dimension matrix  
d_tsne_1 = as.data.frame(tsne_model_1$Y)  
  
#str(d_tsne_1)  
  
## plotting the results without clustering  
ggplot(d_tsne_1, aes(x=V1, y=V2, colour=v_labels[1:k])) +  
  geom_point(size=0.40) +
```

```
guides(colour=guide_legend(override.aes=list(size=6))) +
xlab("") + ylab("") +
ggtitle("t-SNE") +
theme_light(base_size=20) +
theme(axis.text.x=element_blank(),
      axis.text.y=element_blank()) +
scale_color_manual(values=colores)
```

## t-SNE



```
#+ scale_colour_brewer(palette = "RdBu") # create a custom color scale(>2 colours)
```