BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

ĐỒ ÁN MÔN HỌC

PHÂN TÍCH VÀ TRỰC QUAN HOÁ DỮ LIỆU

Ngành: **KHOA HỌC DỮ LIỆU**

Chuyên ngành: **KHOA HỌC DỮ LIỆU**

Giảng viên hướng dẫn : Ths. Lê Nhật Tùng

Sinh viên thực hiện :

2286400009 - Bùi Minh Huy

2286400042 - Trần Lê Vân

2286400028 - Nguyễn Thị Thanh Tâm

Lớp: 22DKHA1

TP. Hồ Chí Minh, 2025

LỜI CAM ĐOAN

Chúng tôi, **Bùi Minh Huy**, **Trần Lê Vân**, **Nguyễn Thị Thanh Tâm** xin cam đoan rằng:

Tất cả thông tin và phân tích trình bày trong báo cáo này được thực hiện một cách chính xác và trung thực. Mọi dữ liệu, nhận định hoặc ý kiến được trích dẫn từ các nguồn khác đều đã được nêu rõ nguồn gốc và trích dẫn đúng quy định. Chúng tôi cam đoan rằng không có bất kỳ hành vi sao chép hoặc sử dụng thông tin không hợp pháp nào từ các nguồn khác. Bài báo cáo này là kết quả của công trình nghiên cứu độc lập của chúng tôi và chưa từng được công bố tại bất kỳ nơi nào khác. Chúng tôi cam đoan đã tuân thủ nghiêm ngặt các quy tắc và quy định của môn học, bao gồm việc tham khảo và áp dụng các công cụ nghiên cứu một cách hợp lệ. Nếu phát hiện có bất kỳ sự gian lận nào, chúng tôi xin hoàn toàn chịu trách nhiệm về nội dung bài báo cáo của mình. Chúng tôi hy vọng rằng bài báo cáo này sẽ cung cấp những thông tin hữu ích cho các nhà nghiên cứu, doanh nghiệp, góp phần vào việc hiểu rõ hơn về mạng xã hội ngày nay.

TP. Hồ Chí Minh, ngày 28 tháng 3 năm 2025

Sinh viên

Contents

1 Abstract

2 Introduction

Trong thời đại của điện toán di động và thiết bị thông minh, việc theo dõi và nhận dạng hoạt động con người (Human Activity Recognition - HAR) đã trở thành một lĩnh vực nghiên cứu đóng vai trò quan trọng trong nhiều ngành như trí tuệ nhân tạo, khoa học dữ liệu, y học và công nghệ cảm biến. HAR đóng vai trò cốt lõi trong các ứng dụng như giám sát sức khỏe, phát hiện té ngã, điều khiển nhà thông minh. Ngày nay, nhu cầu càng ngày gia tăng về các thiết bị công nghệ có thể hiểu hành vi con người dẫn đến việc phát triển các mô hình HAR là chính xác, hiệu quả và có khả năng triển khai thực tế là vô cùng cần thết.

Một trong những yếu tố chính thúc đẩy sự phát triển của HẢ là sự phổ biến của các thiết bị di động thông minh và đồng hồ thông minh, vốn đọc trang bị sẵn các cảm biến quán tính bao gồm gia tốc kế (accelerometer) và con quay hồi chuyển (gyroscope). Những cảm biến này cho phép thu thập dữ liệu về chuyển động của người sử dụng với độ chính xác cao, chi phí thấp và tính khả dụng cao trong đời sống hàng ngày. Nhờ vậy, hệ thống HAR có thể được lắp đặt mà không cần sử dụng các thiết bị đắt tiền hoặc lắp đặt phức tạp.

Bên cạnh tiềm năng ứng dụng rộng rãi, việc xây dựng các mô hình HAR vẫn có nhiều khó khăn thách thức như dữ liệu cảm biến thường có số chiều lớn, có nhiều dữ liệu nhiễu và có tính biến động cao do phụ thuộc vào thói quen và hình thể của mỗi người. Bên cạnh đó, một số hoạt động có thể có mẫu tín hiệu tương tự nhau khiến cho các bài toán phân loại trở nên khó khăn hơn. Vì vậy, cần có một quy trình xử lý dữ liệu bài bản bao gồm các bước tiền xử lý dữ liệu, giảm chiều dữ liệu và huấn luyện mô hình học máy để có thể đặt được hiệu quả cao trong việc nhận dạng hoạt động con người.

Trong nghiên cứu này, chúng em đã tiến hành khai thác bộ dữ liệu "**Human Activity Recognition with Smartphones**" do UCI Machine Learning Repository cung cấp, một bộ dữ liệu được sử dụng rộng raix trong cộng đồng nghiên cứu HAR. Chúng em đề xuất một quy trình học máy toàn diện bao gồm phân tích đặc trưng, giảm chiều dữ liệu bằng UMAP, PCA, TSNE và huấn luyện bằng các mô hình học máy như Random Forest, Decision Tree, Logistic Regression, Support Vectot Machine (SVM) để phân loại các hoạt động với mục tiêu là nâng cao độ chính xác và hiệu quả của mô hình. Những kết quả này sẽ cung cấp cái nhìn thực nghiệm rõ ràng cho các nhà nghiên cứu, đồng thời làm nền móng cho việc triển khai các hệ thống nhận dạng hoạt động trong thế giới thực.

3 Dataset and Preprocessing (Dữ liệu và tiền xử lý)

Bộ dữ liệu **Human Activity Recognition with Smartphones** được thu thập từ 30 người tham gia (gọi là *subjects*) (15 nam và 15 nữ, độ tuổi từ 19 đến 48) thực hiện sáu hoạt động thường ngày như Đi bộ (Walking), đi lên cầu thang (Walking Upstairs), đi xuống cầu thang (WalkingDownstairs), ngồi (Sitting),đứng (Standing),nằm (Laying). Dữ liệu được ghi lại bằng một điện thoại thông minh (Samsung Galaxy S II) đeo ở thắt lưng của người dùng. Các cảm biến gồm **accelerometer** và **gyroscope** được sử dụng để ghi lại chuyển động theo 3 trục X, Y, Z với tần suất 50Hz.

Mỗi chuỗi tín hiệu được chia thành các **cửa sổ trượt** có độ dài 2.56 giây, tương ứng với 128 lần đo. Từ mỗi cửa sổ, các đặc trưng (features) đã được trích xuất từ **miền thời gian** và **miền tần số** để tạo ra một tập hợp dữ liệu có cấu trúc sẵn sàng cho mô hình học máy.

Các nhóm tín hiệu chính bao gồm:

- `tBodyAcc-XYZ`: Gia tốc cơ thể theo 3 trục trong miền thời gian
- `tGravityAcc-XYZ`: Gia tốc do trọng lực
- `tBodyGyro-XYZ`: Tốc độ quay từ con quay hồi chuyển
- `tBodyAccJerk-XYZ`, `tBodyGyroJerk-XYZ`: Jerk - đo sự thay đổi đột ngột của chuyển động

- `Mag`: Độ lớn vector gia tốc, được tính bằng chuẩn Euclidean:

$$\text{Mag} = \sqrt{X^2 + Y^2 + Z^2}$$

- `fBodyAcc-XYZ`, `fBodyGyro-XYZ`: Biến miền tần số được tạo từ FFT

Từ các tín hiệu trên, một loạt đặc trưng thống kê được tính toán như:

- `mean()`, `std()`, `mad()`, `max()`, `min()`
- `sma()`: Signal Magnitude Area
- `energy()`: Tổng bình phương chia số phần tử
- `entropy()`, `iqr()`, `arCoeff()`, `correlation()`
- `meanFreq()`, `skewness()`, `kurtosis()`, `bandsEnergy()`, `angle()`

  Toàn bộ bộ dữ liệu được chia thành hai phần: Tập huấn luyện (train.csv): bao gồm 7352 mẫu. Tập kiểm tra (test.csv): bao gồm 2947 mẫu. Mỗi mẫu tương ứng với một cửa sổ thời gian 2.56 giây, được biểu diễn bằng **561 đặc trưng đầu vào (features)**, 1mỗi mẫu còn bao gồm một mã định danh người thực hiện (subject) và một nhãn hoạt động (Activity), ác nhãn này được mã hóa từ 1 đến 6 tương ứng với:

Giá trị nhãn	Hoạt động
1	WALKING
2	WALKING_UPSTAIRS
3	WALKING_DOWNSTAIRS
4	SITTING
5	STANDING
6	LAYING

```
# install.packages("showtext")
# install.packages("ggplot2")
```

```
# train <-read.csv('/Users/huy/Documents/doanthaytung/archive/train.csv')
# train <-read.csv("D:/BT/clonegit/doanthaytung/archive/train.csv")
train <- read.csv('archive/train.csv')
# head(train)
dim(train)
```

```
## [1] 7352  563
```

```
# test <-read.csv('/Users/huy/Documents/doanthaytung/archive/test.csv')
# test <-read.csv("D:/BT/clonegit/doanthaytung/archive/test.csv")
test <- read.csv("archive/test.csv")
# head(test)
dim(test)
```

```
## [1] 2947  563
```

xem các các type có trong data

```r
table(sapply(train, class))
```

```
##
## character   integer   numeric
##         1         1       561
```

in ra cột có dạng charactor

```r
names(train)[sapply(train, class) == "character"]
```

```
## [1] "Activity"
```

```r
unique(train$Activity)
```

```
## [1] "STANDING"           "SITTING"            "LAYING"
## [4] "WALKING"            "WALKING_DOWNSTAIRS" "WALKING_UPSTAIRS"
```

chuyển thành factor

```r
train$Activity <- as.factor(train$Activity)
```

```r
cat("Giá trị thiếu ở tập train:", sum(is.na(train)), "\n")
```

```
## Giá trị thiếu ở tập train: 0
```

```r
cat("Giá trị thiếu ở tập test:", sum(is.na(test)), "\n")
```

```
## Giá trị thiếu ở tập test: 0
```

```r
cat("Số dòng bị trùng lặp trong tập train:", sum(duplicated(train)), "\n")
```

```
## Số dòng bị trùng lặp trong tập train: 0
```

```r
cat("Số dòng bị trùng lặp trong tập test :", sum(duplicated(test)), "\n")
```

```
## Số dòng bị trùng lặp trong tập test : 0
```

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

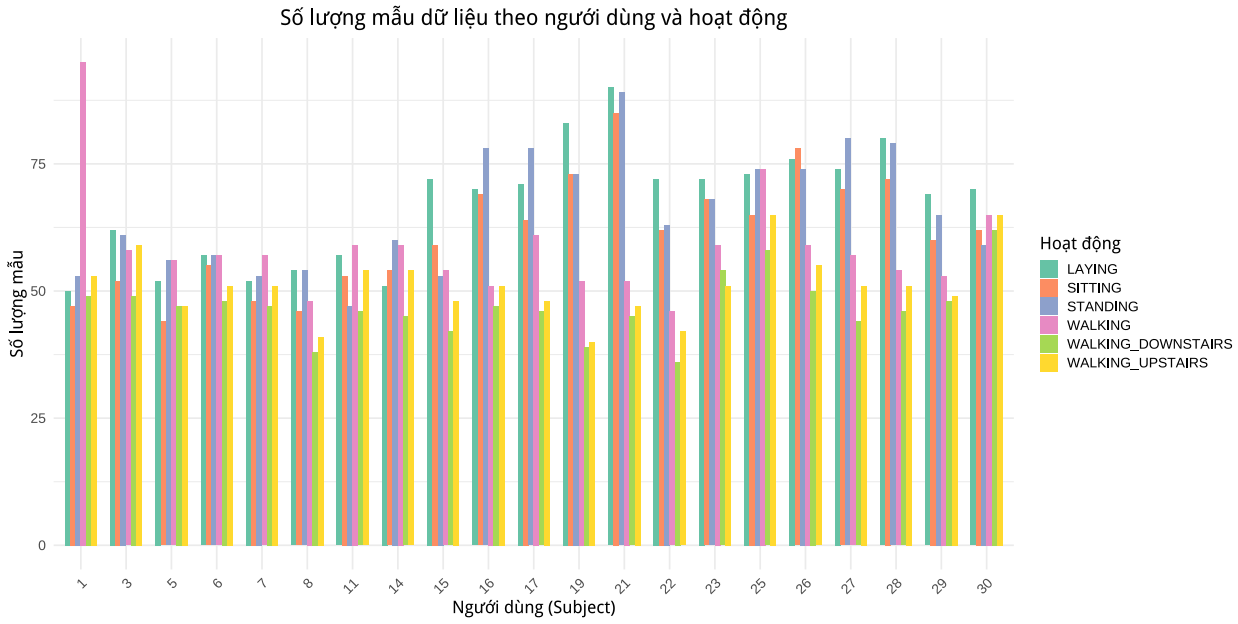
```r
library(showtext)
```

```
## Warning: package 'showtext' was built under R version 4.4.3
```

```
## Loading required package: sysfonts
```

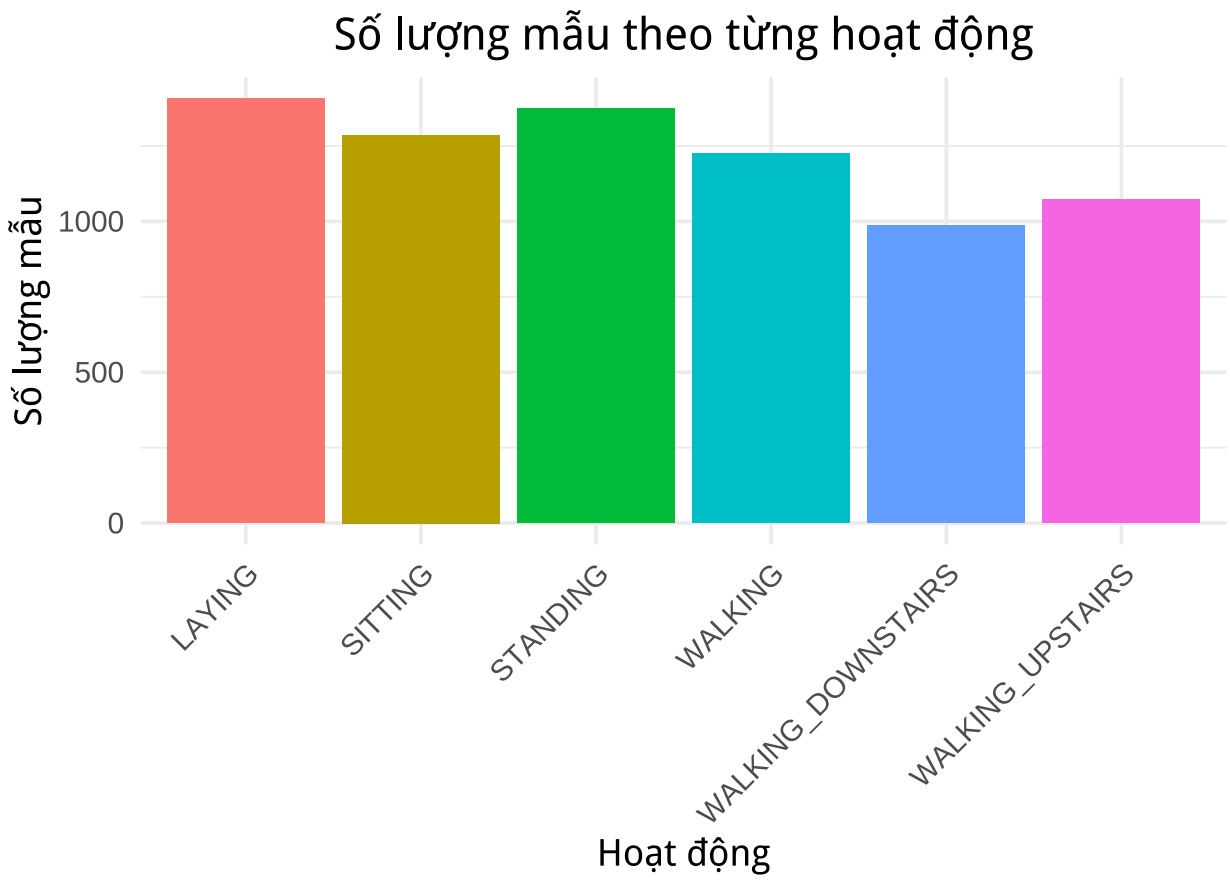
```
## Warning: package 'sysfonts' was built under R version 4.4.3

## Loading required package: showtextdb

## Warning: package 'showtextdb' was built under R version 4.4.3
```

```r
showtext_auto()

ggplot(train, aes(x = factor(subject), fill = Activity)) +
  geom_bar(position = "dodge", width = 0.7) +
  scale_fill_brewer(palette = "Set2") +
  labs(
    title = "Số lượng mẫu dữ liệu theo người dùng và hoạt động",
    x = "Người dùng (Subject)",
    y = "Số lượng mẫu",
    fill = "Hoạt động"
  ) +
  theme_minimal(base_size = 16) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 20),
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "right"
  )
```




```r
library(ggplot2)

ggplot(train, aes(x = Activity, fill = Activity)) +
  geom_bar() +
  labs(
    title = "Số lượng mẫu theo từng hoạt động",
    x = "Hoạt động",
    y = "Số lượng mẫu"
```

```
  ) +
  theme_minimal(base_size = 15) +
  theme(
    axis.text.x = element_text(angle = 45, vjust = 1, hjust=1),
    plot.title = element_text(hjust = 0.5, face = "bold")
  ) +
  guides(fill = "none")
```



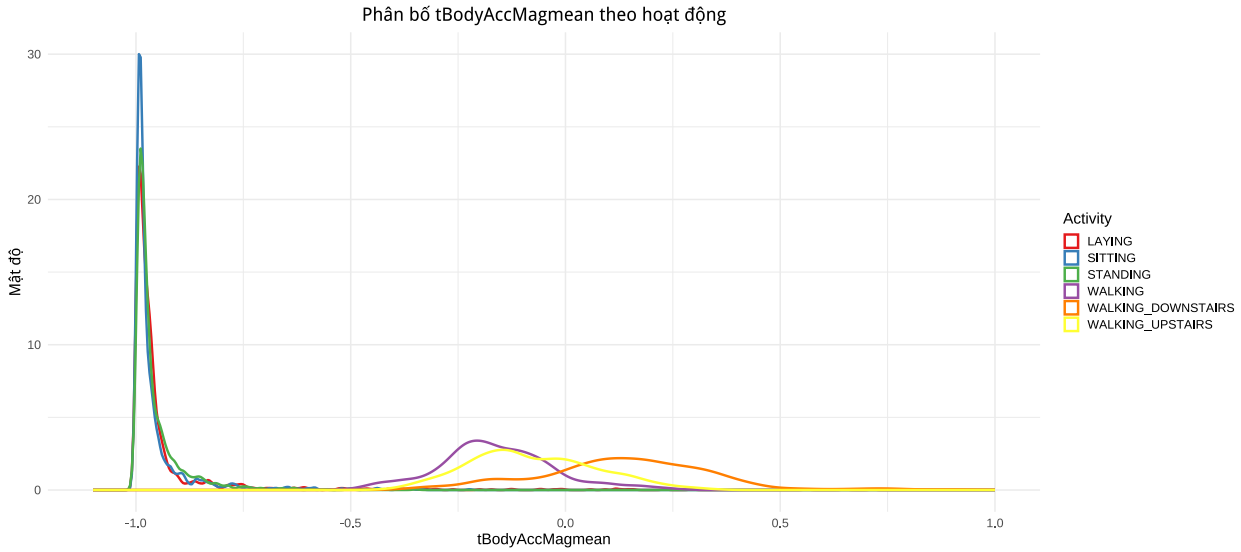
Số lượng mẫu cho mỗi hoạt động dao động trong khoảng 1000 mẫu đến 1200 mẫu. Các hoạt động tĩnh như laying, sitting, standing có xu hướng chiếm tỷ lệ cao hơn một chút so với các hoạt động di chuyển như walking, walking_upstairs, walking_downstairs. Mức độ chênh lệch không qua lớn giữa các nhóm, là điểm thuận lợi cho các mô hình học máy tránh bị lệch nhãn và đảm bảo khả năng học đều giữa các lớp.

```
columns <- colnames(train)
columns <- gsub("\\.", "", columns)
colnames(train) <- columns
colnames(test) <- columns
# colnames(train)
```

```
library(ggplot2)

ggplot(train, aes(x = tBodyAccMagmean, color = Activity)) +
  geom_density(size = 1.2) +
  scale_color_brewer(palette = "Set1") +
  scale_x_continuous(limits = c(-1.1, 1)) +
```

```
  labs(
    title = "Phân bố tBodyAccMagmean theo hoạt động",
    x = "tBodyAccMagmean",
    y = "Mật độ"
  ) +
  theme_minimal(base_size = 16) +
  theme(
    plot.title = element_text(hjust = 0.5)
  )
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
class(train$Activity)
```

```
## [1] "factor"
```

```
unique(train$Activity)
```

```
## [1] STANDING           SITTING            LAYING             WALKING
## [5] WALKING_DOWNSTAIRS WALKING_UPSTAIRS
## 6 Levels: LAYING SITTING STANDING WALKING ... WALKING_UPSTAIRS
```

```
library(ggplot2)
library(dplyr)
```

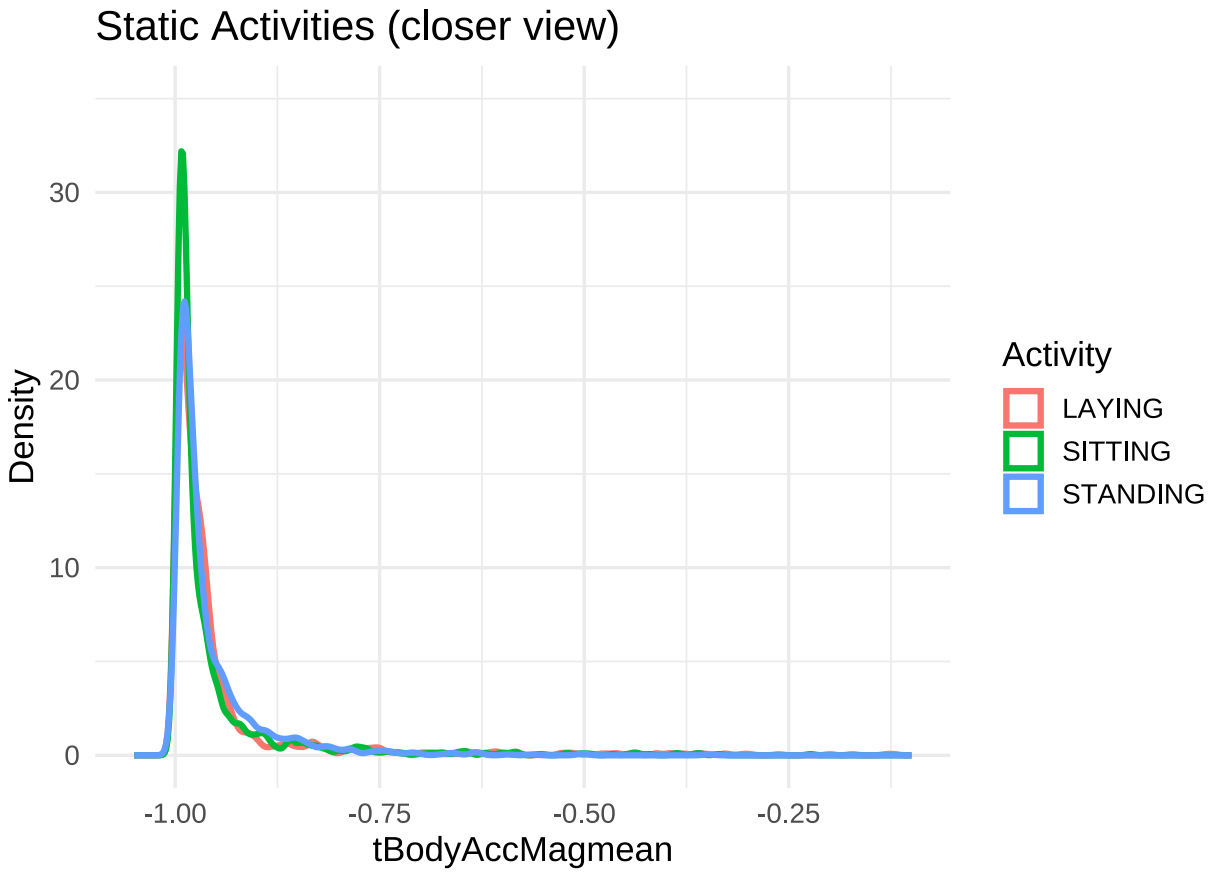
```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

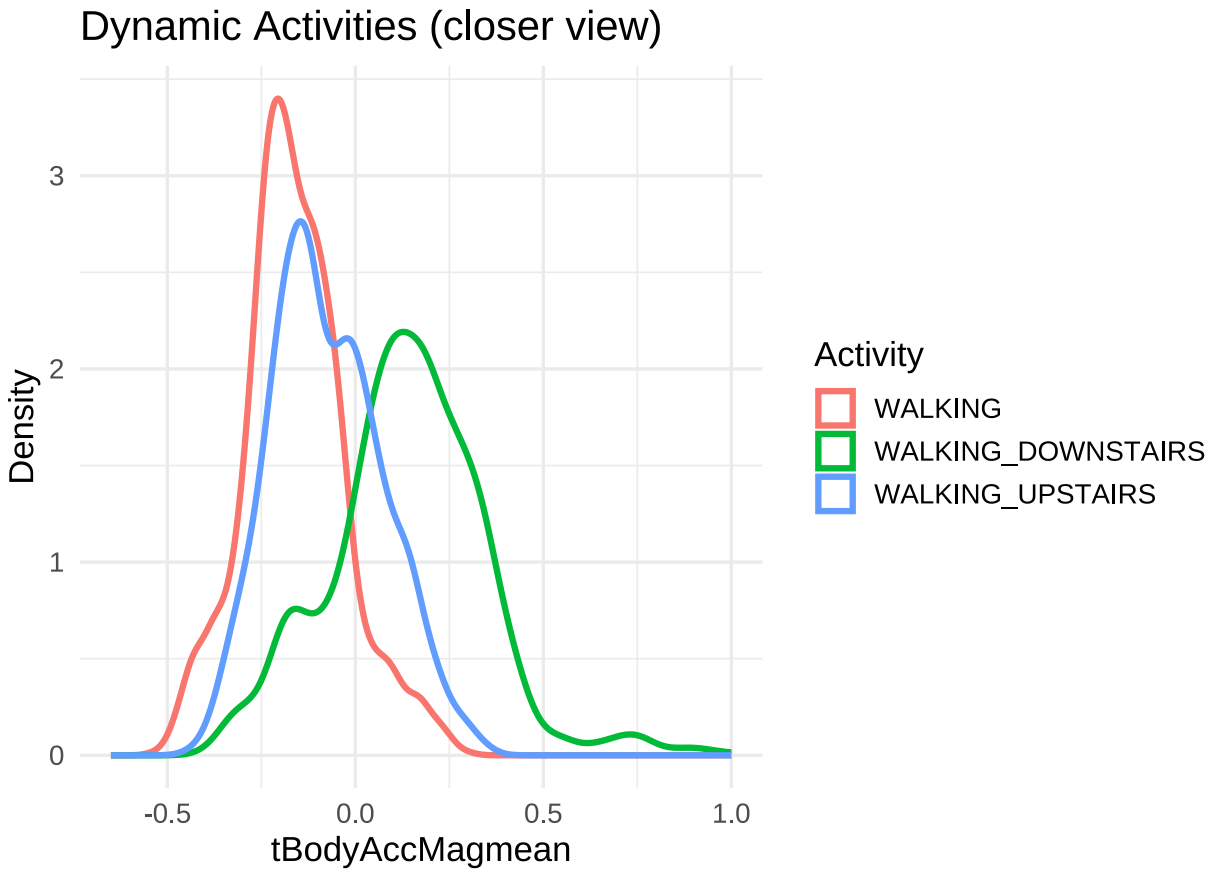
```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

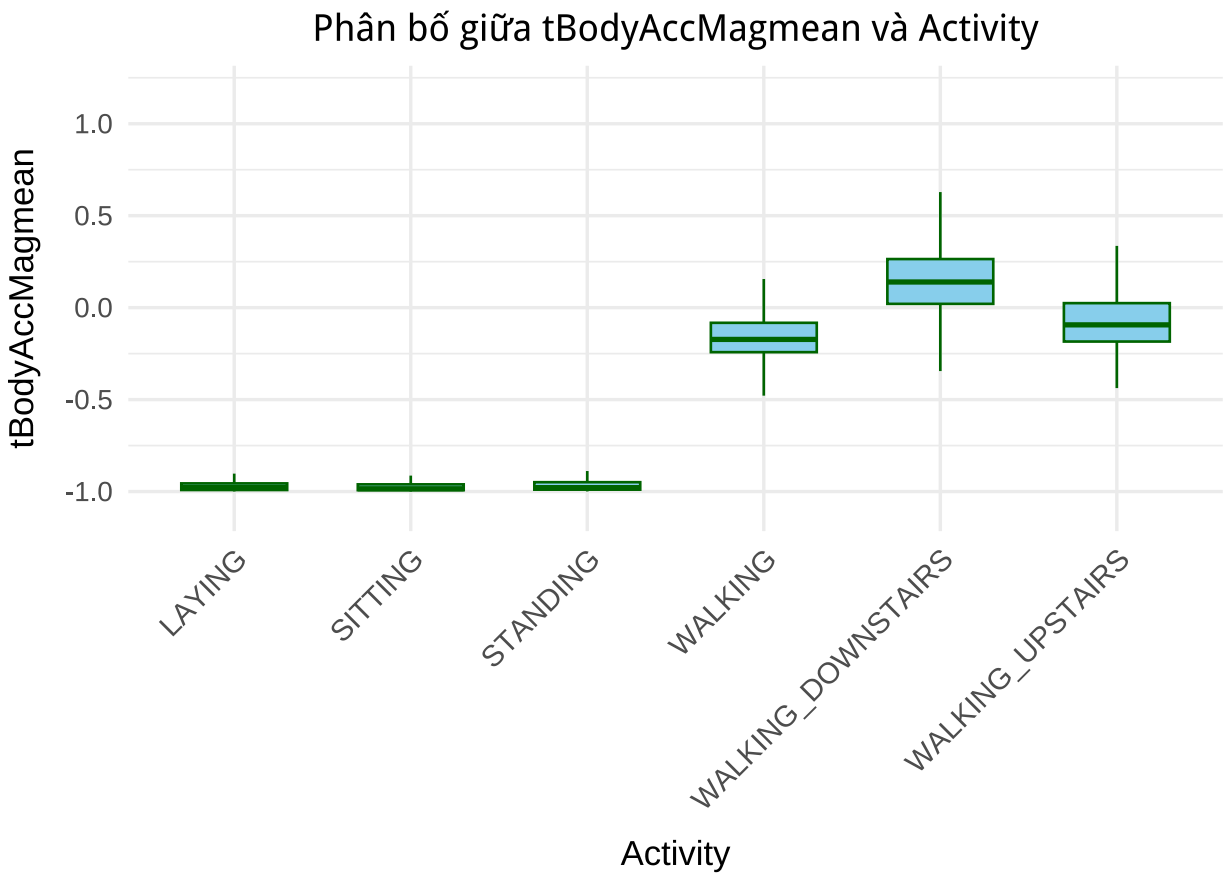
```
p1 <- train %>%
  filter(Activity %in% c("SITTING", "STANDING", "LAYING")) %>%
  ggplot(aes(x = tBodyAccMagmean, color = Activity)) +
  geom_density(size = 1.2) +
  labs(
    title = "Static Activities (closer view)",
    x = "tBodyAccMagmean",
    y = "Density"
  ) +
  xlim(-1.05, -0.1) +
  ylim(0, 35) +
  theme_minimal(base_size = 14)
p1
```



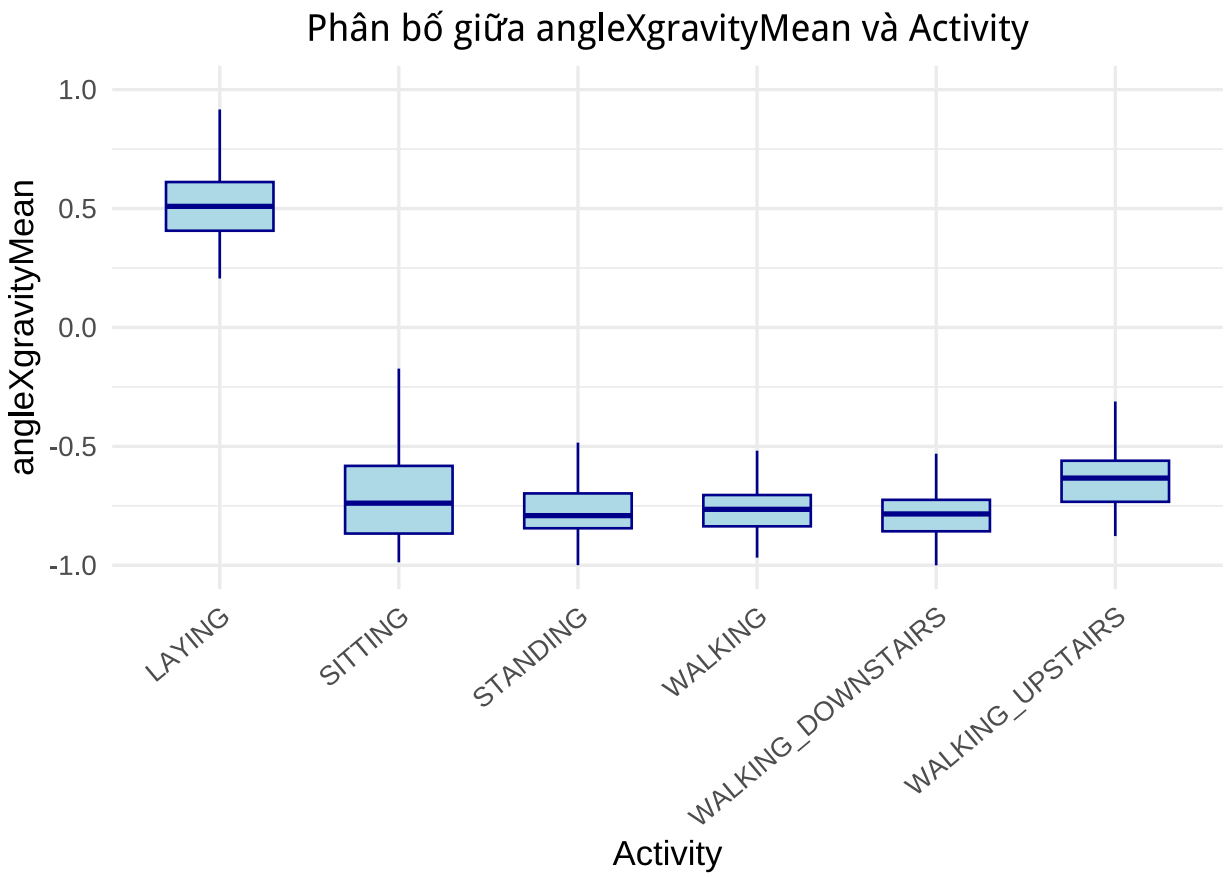

```
p2 <- train %>%
  filter(Activity %in% c("WALKING", "WALKING_DOWNSTAIRS", "WALKING_UPSTAIRS")) %>%
  ggplot(aes(x = tBodyAccMagmean, color = Activity)) +
  geom_density(size = 1.2) +
  labs(
    title = "Dynamic Activities (closer view)",
    x = "tBodyAccMagmean",
    y = "Density"
  ) + xlim(-0.65, 1) +
  theme_minimal(base_size = 14)

p2
```



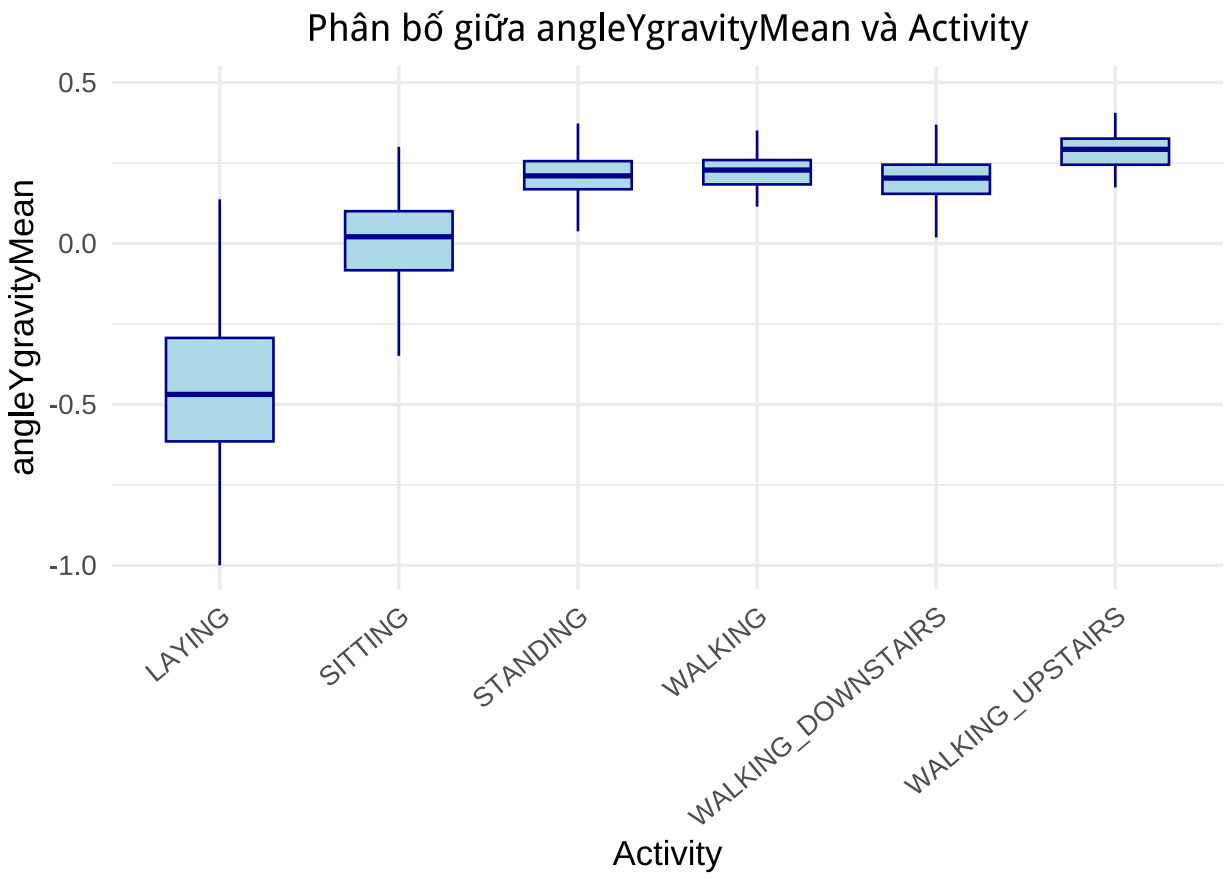

```r
library(ggplot2)

ggplot(train, aes(x = Activity, y = tBodyAccMagmean)) +
  geom_boxplot(
    outlier.shape = NA,
    fill = "skyblue",
    color = "darkgreen",
    width = 0.6
  ) +
  labs(
    title = "Phân bố giữa tBodyAccMagmean và Activity",
    y = "tBodyAccMagmean",
    x = "Activity"
  ) +
  coord_cartesian(ylim = c(-1.1, 1.2)) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 15, face = "bold"),
    axis.text.x = element_text(angle = 45, hjust = 1, size = 12),
    axis.title.x = element_text(margin = margin(t = 10)),
    axis.title.y = element_text(margin = margin(r = 10))
  )
```




```r
library(ggplot2)

ggplot(train, aes(x = Activity, y = angleXgravityMean)) +
  geom_boxplot(
    fill = "lightblue",
    color = "darkblue",
    outlier.shape = NA,
    width = 0.6
  ) +
  labs(
    title = "Phân bố giữa angleXgravityMean và Activity",
    x = "Activity",
    y = "angleXgravityMean"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 15, face = "bold"),
    axis.text.x = element_text(angle = 40, hjust = 1, size = 11)
  )
```

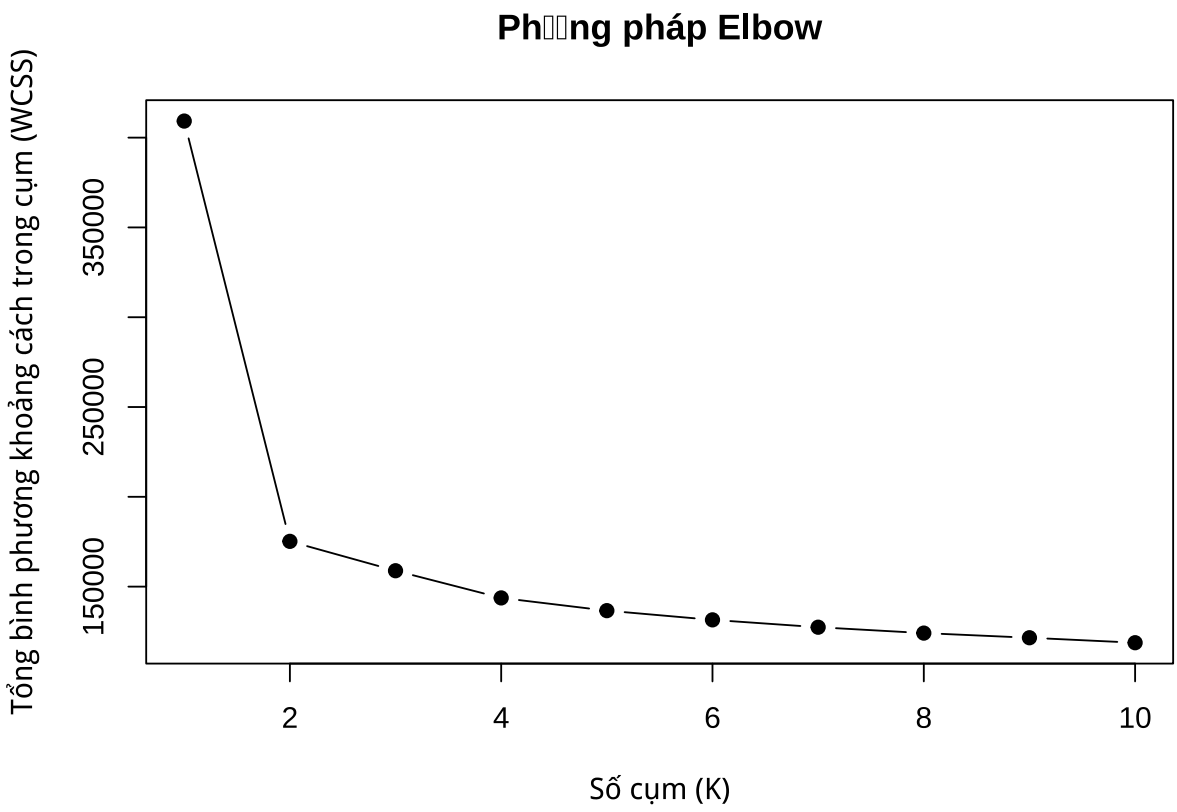



```r
library(ggplot2)

ggplot(train, aes(x = Activity, y = angleYgravityMean)) +
  geom_boxplot(
    fill = "lightblue",
    color = "darkblue",
    outlier.shape = NA,
    width = 0.6
  ) +
  labs(
    title = "Phân bố giữa angleYgravityMean và Activity",
    x = "Activity",
    y = "angleYgravityMean"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 15, face = "bold"),
    axis.text.x = element_text(angle = 40, hjust = 1, size = 11)
  )
```



Kmean

```
train_kmean <- train
X <- train_kmean  %>% select(-Activity, -subject)
wcss <- numeric(10)
for(i in 1:10) {
  kmeans_model <- kmeans(X, centers=i, nstart=25)
  wcss[i]  <- kmeans_model$tot.withinss
}

plot(1:10, wcss, type = "b", pch = 19,
     xlab = "Số cụm (K)",
     ylab = "Tổng bình phương khoảng cách trong cụm (WCSS)",
     main = "Phương pháp Elbow")
```




```
# # Tính điểm Silhouette trung bình cho các giá trị K từ 2 đến 10
# library(cluster)
#
# avg_sil <- numeric(9)
# for(k in 2:10) {
#   km <- kmeans(X, centers = k, nstart = 25)
#   ss <- silhouette(km$cluster, dist(X))
#   avg_sil[k-1] <- mean(ss[, 3])
# }
#
# # Vẽ biểu đồ Silhouette
# plot(2:10, avg_sil, type = "b", pch = 19,
#      xlab = "So luong cum (K)",
#      ylab = "Diem Silhouette trung binh",
#      main = "Phuong phap Silhouette")
```

Đánh giá kết quả phân cụm với k = 2

```
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.4.3
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
library(cluster)

# Áp dụng K-Means với K = 2
km_result <- kmeans(X, centers = 2, nstart = 100)

# Tính và vẽ biểu đồ Silhouette
sil <- silhouette(km_result$cluster, dist(X))
fviz_silhouette(sil, print.summary = TRUE)
```

```
##   cluster size ave.sil.width
## 1       1 4055          0.52
## 2       2 3297          0.45
```

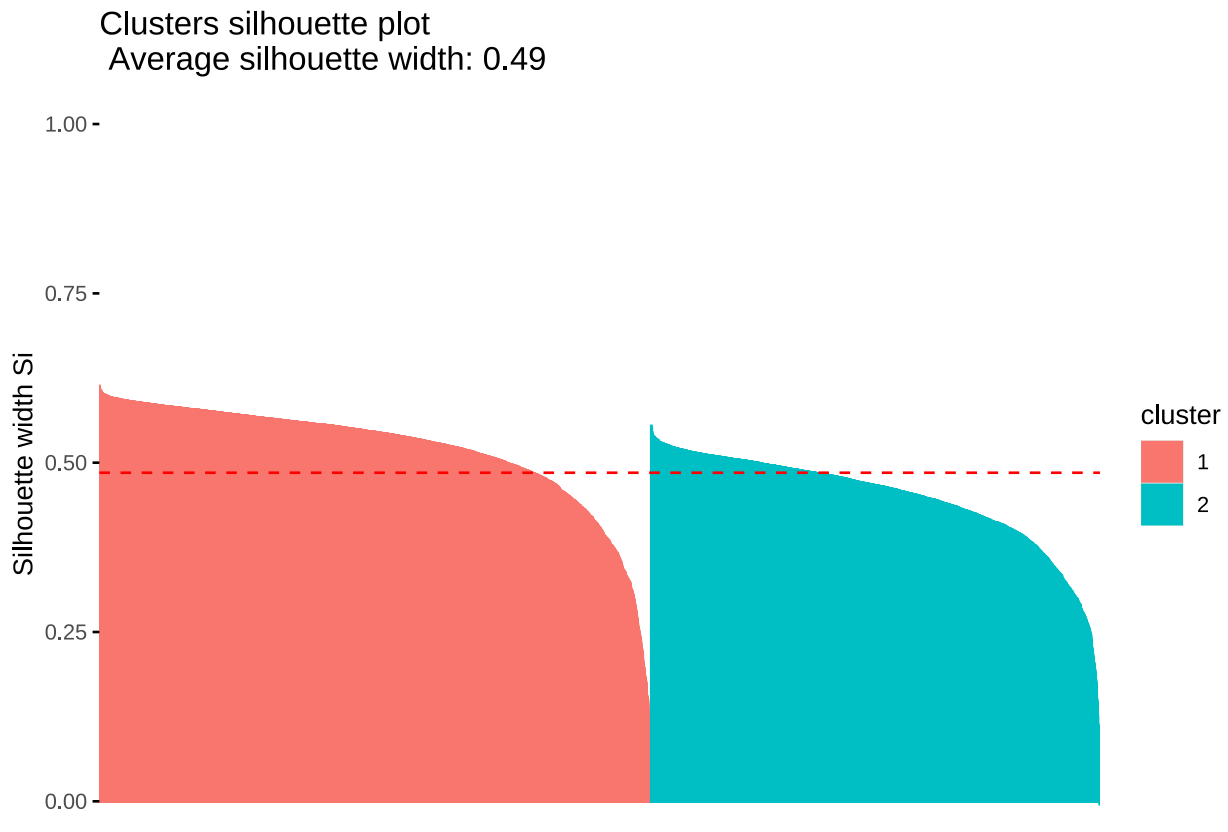
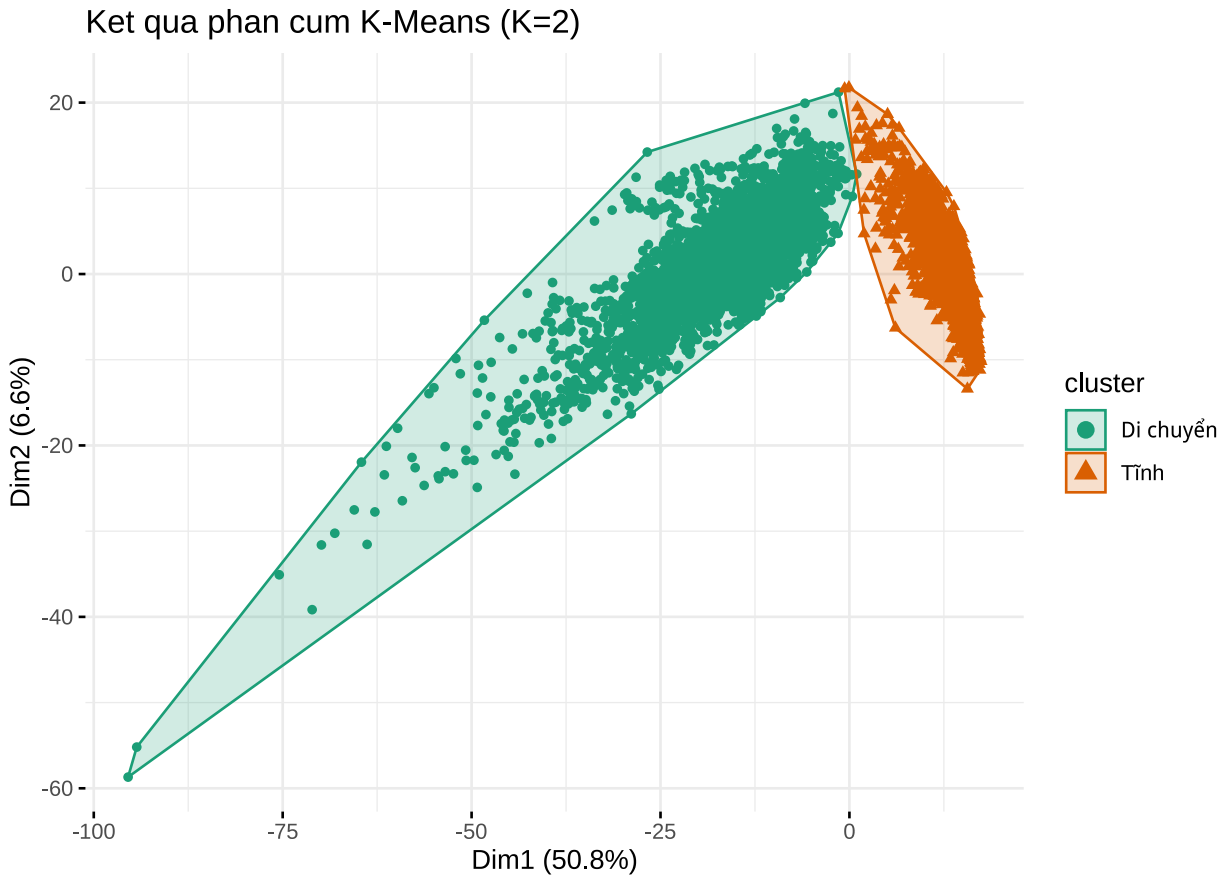


Figure 1: Bieu do Silhouette cho ket qua phan cum

Trực quan kết quả phân cụm với k = 2

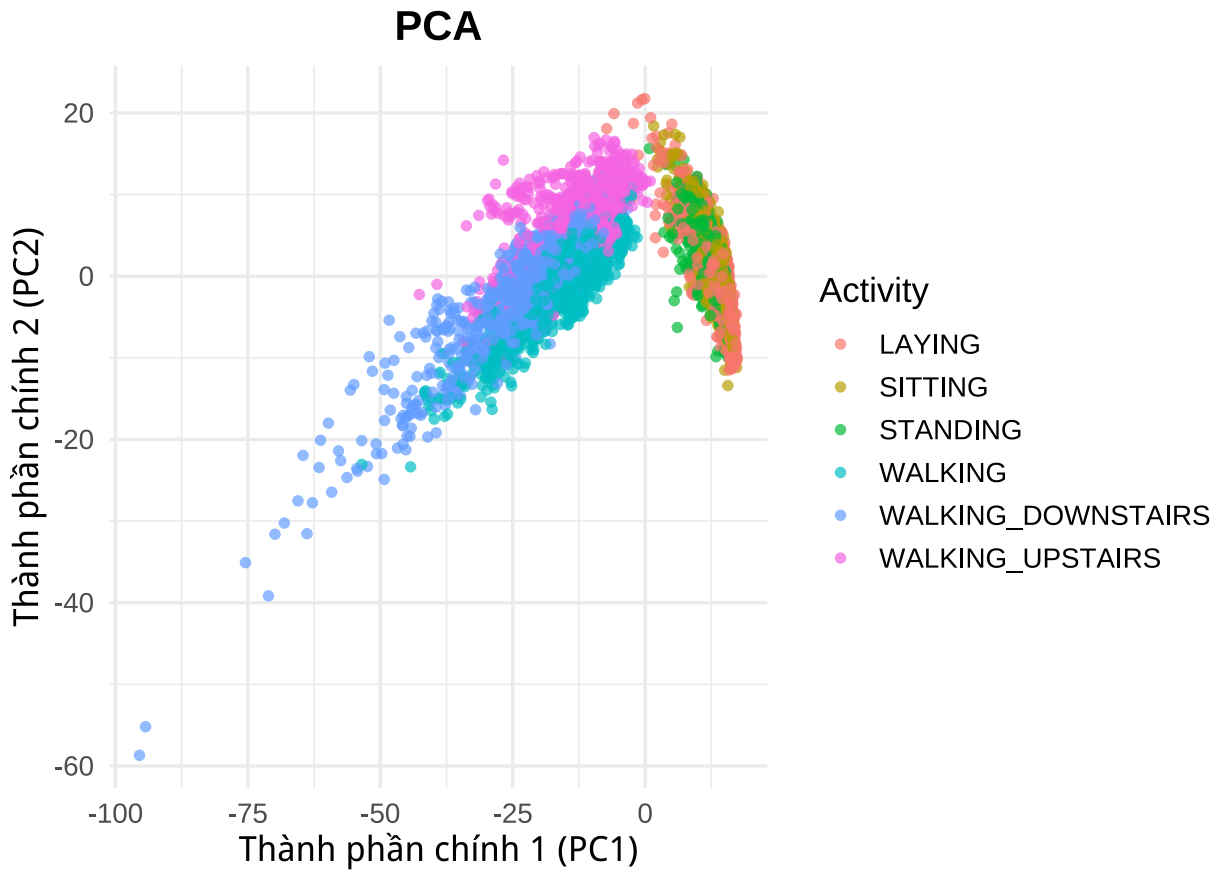
```r
label_cluster <- ifelse(km_result$cluster == 1, "Tĩnh", "Di chuyển")
km_labeled <- km_result
km_labeled$cluster <- as.factor(label_cluster)
fviz_cluster(km_labeled, data = X,
             palette = c("#1B9E77", "#D95F02"),
             geom = "point",
```

```
            ellipse.type = "convex",
            ggtheme = theme_minimal(),
            main = "Ket qua phan cum K-Means (K=2)")
```




```
train_pca <- train
df_pca <- train_pca %>% select(-subject, -Activity)
pca_result <- prcomp(df_pca, center = TRUE, scale. = TRUE)
df_pca_result <- as.data.frame(pca_result$x[, 1:2])
df_pca_result$Activity <- train$Activity

# df_pca_result
```

```
ggplot(df_pca_result, aes(x = PC1, y = PC2, color = Activity)) +
  geom_point(alpha = 0.7) +
  labs(
    title = " PCA ",
    x = "Thành phần chính 1 (PC1)",
    y = "Thành phần chính 2 (PC2)"
  ) +
  theme_minimal(base_size = 14) +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))
```



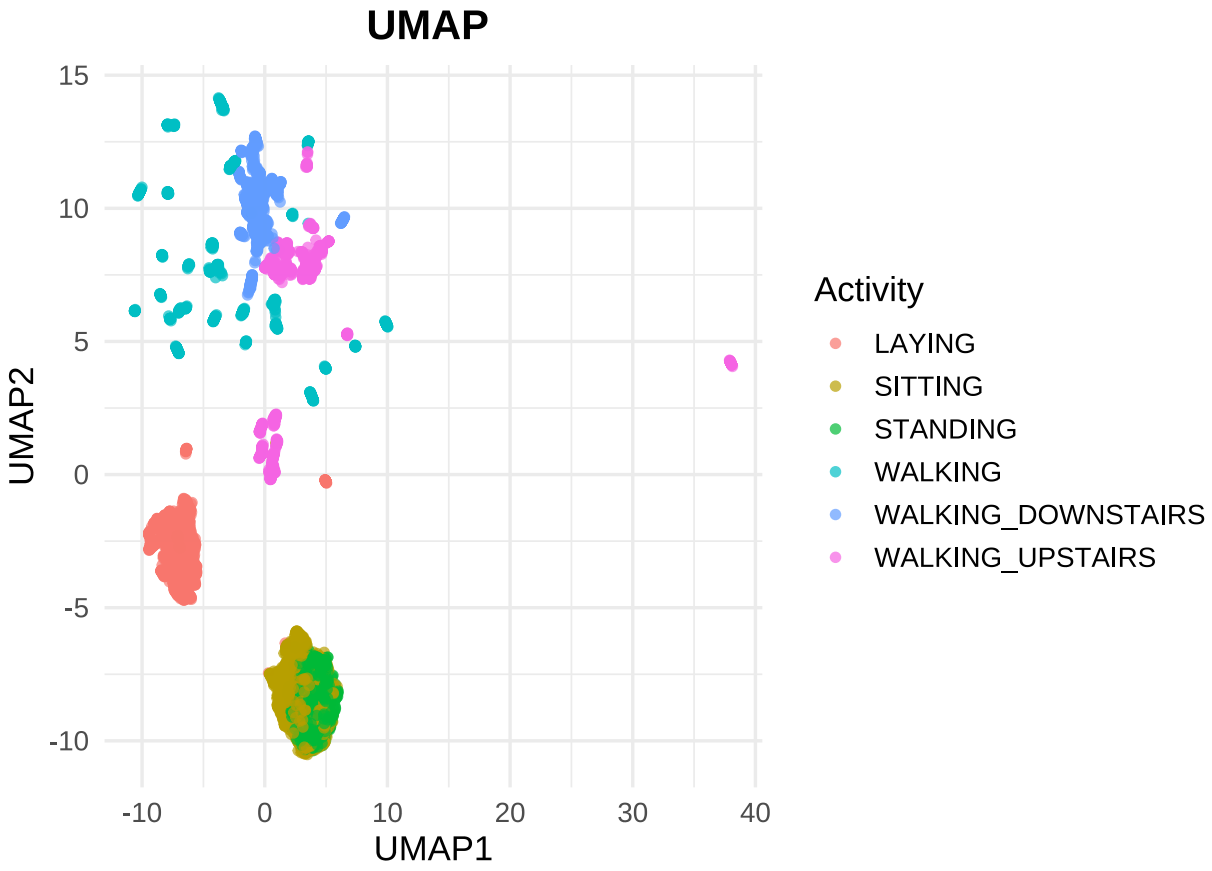

```
#install.packages("dplyr")
#install.packages("umap")

# Nạp thư viện
library(dplyr)
library(umap)
```

```
## Warning: package 'umap' was built under R version 4.4.3
```

```
library(ggplot2)

train_umap <- train
df_umap <- train_umap %>% select(-subject, -Activity)

umap_result <- umap(df_umap)
df_umap_result <- as.data.frame(umap_result$layout)
df_umap_result$Activity <- train_umap$Activity

# train_umap
# df_umap_result
```

Trực quan hóa kết quả giảm chiều bằng umap

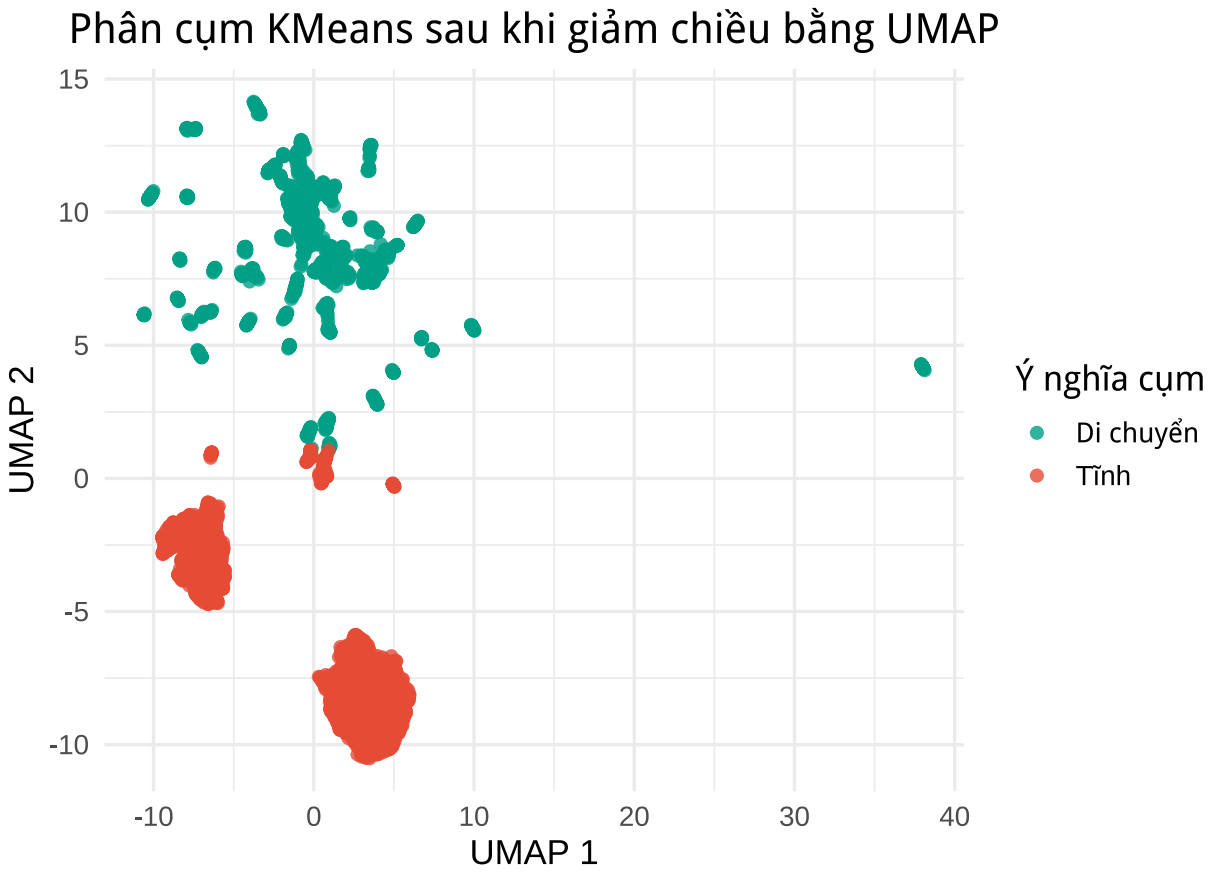
```r
ggplot(df_umap_result, aes(x = V1, y = V2, color = Activity)) +
  geom_point(alpha = 0.7) +
  labs(
    title = "UMAP ",
    x = "UMAP1",
    y = "UMAP2"
  ) +
  theme_minimal(base_size = 14) +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))
```



Trực quan kết quả phân cụm umap với k = 2

```r
set.seed(42)
kmeans_result <- kmeans(df_umap_result[, 1:2], centers = 2, nstart = 25)
df_umap_result$Cluster <- as.factor(kmeans_result$cluster)
df_umap_result$Label <- ifelse(df_umap_result$Cluster == 1, "Tĩnh", "Di chuyển")

ggplot(df_umap_result, aes(x = V1, y = V2, color = Label)) +
  geom_point(alpha = 0.8, size = 2) +
  labs(
    title = "Phân cụm KMeans sau khi giảm chiều bằng UMAP",
    x = "UMAP 1", y = "UMAP 2",
    color = "Ý nghĩa cụm"
  ) +
  scale_color_manual(values = c("Tĩnh" = "#E64B35", "Di chuyển" = "#00A087")) +
```

```
  theme_minimal(base_size = 14) +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# install.packages("Rtsne")  # nếu chưa cài
library(Rtsne)
```

```
## Warning: package 'Rtsne' was built under R version 4.4.3
```

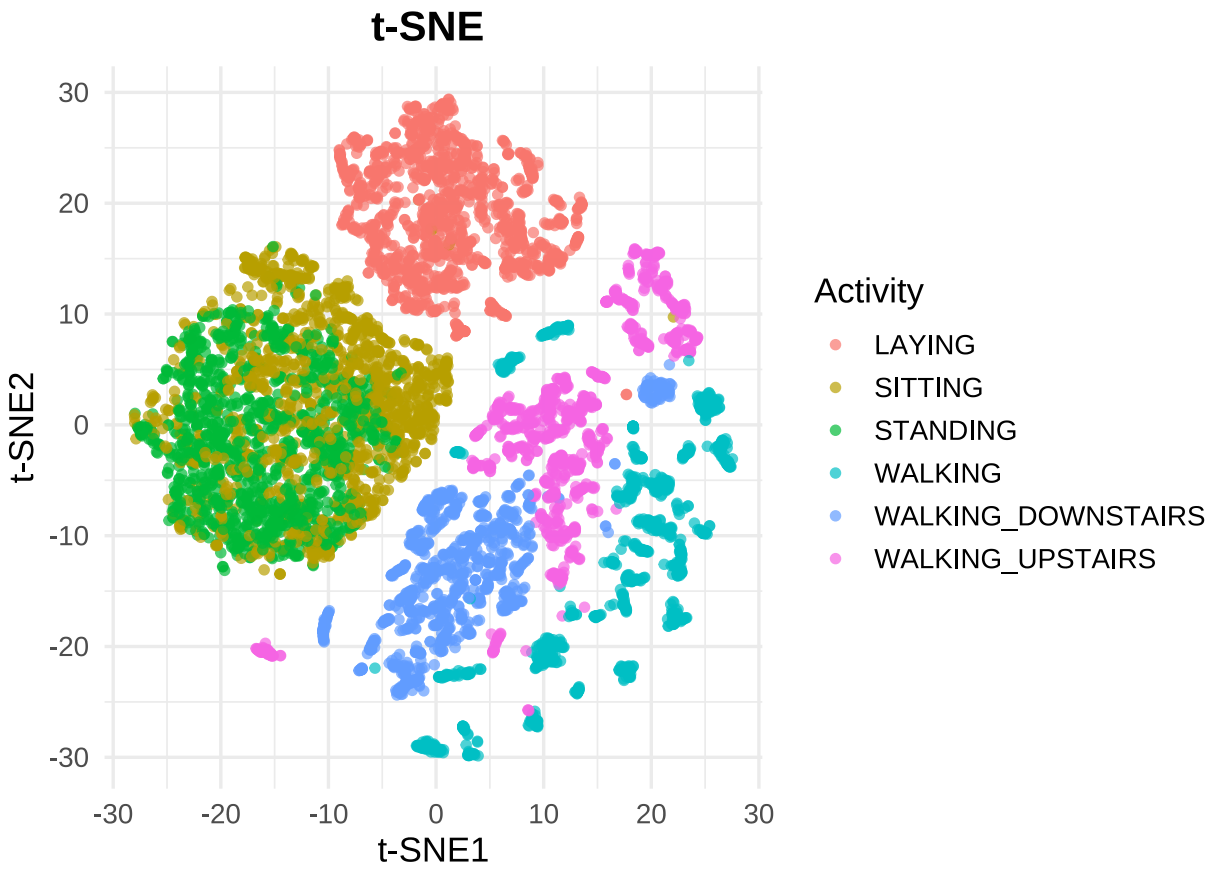
```
train_tsne <- train
df_tsne <- train_tsne %>% select(-subject, -Activity)

set.seed(42)  # đảm bảo tái lập kết quả
tsne_result <- Rtsne(as.matrix(df_tsne), dims = 2, perplexity = 30, verbose = TRUE, max_iter = 500)
```

```
## Performing PCA
## Read the 7352 x 50 data matrix successfully!
## OpenMP is working. 1 threads.
## Using no_dims = 2, perplexity = 30.000000, and theta = 0.500000
## Computing input similarities...
## Building tree...
## Done in 2.08 seconds (sparsity = 0.017122)!
## Learning embedding...
## Iteration 50: error is 93.861229 (50 iterations in 1.29 seconds)
```

```
## Iteration 100: error is 81.745900 (50 iterations in 1.31 seconds)
## Iteration 150: error is 79.073438 (50 iterations in 0.55 seconds)
## Iteration 200: error is 78.158564 (50 iterations in 0.56 seconds)
## Iteration 250: error is 77.781913 (50 iterations in 0.56 seconds)
## Iteration 300: error is 2.660835 (50 iterations in 0.51 seconds)
## Iteration 350: error is 2.268763 (50 iterations in 0.49 seconds)
## Iteration 400: error is 2.046950 (50 iterations in 0.49 seconds)
## Iteration 450: error is 1.904216 (50 iterations in 0.49 seconds)
## Iteration 500: error is 1.804544 (50 iterations in 0.49 seconds)
## Fitting performed in 6.75 seconds.
```

```r
df_tsne_result <- as.data.frame(tsne_result$Y)
df_tsne_result$Activity <- train_tsne$Activity

# df_tsne_result
```

```r
ggplot(df_tsne_result, aes(x = V1, y = V2, color = Activity)) +
  geom_point(alpha = 0.7) +
  labs(
    title = "t-SNE ",
    x = "t-SNE1",
    y = "t-SNE2"
  ) +
  theme_minimal(base_size = 14) +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))
```



Methodology (Phương pháp)