Fundamental Areas of Differential Calculus

# Limits

Limits are a way to evaluate a function as the variable approaches a particular value. They allow us to observe the behaviour of a function as x gets close to a specific point, such as determining what happens when x approaches 2 in the function f(2).

# Derivatives

Derivatives are functions that provide the slope of the original function at a given value. They are effective for calculating the rate of change and help to understand how a function changes in response to small adjustments in its input.

* Calculate the rate of change; tell how fast it changes per unit of time
* Slope of the tangent line slope = y2-y1/x2-x1

# Integration

Integration is the process that is the opposite of differentiation. It involves finding the area under the curve of a function, which is useful for summing up quantities that accumulate continuously.

* Help to determine how much something accumulate over a period of time
* Useful calculating the area under the curve
* A = x(y)

 ---------------------------

# Example in R

library(dplyr)

data <- data %>%

mutate(

avg\_speed = actual\_distance / finish\_time,

over\_distance = actual\_distance - official\_distance,

efficiency = (official\_distance / actual\_distance) \* 100

)

# x and y are positions at different timestamps

data <- data %>%

arrange(time) %>%

mutate(

delta\_x = c(NA, diff(x)),

delta\_y = c(NA, diff(y)),

delta\_t = c(NA, diff(time)),

speed = sqrt(delta\_x^2 + delta\_y^2) / delta\_t

)

library(caret)

# Split data

set.seed(123)

trainIndex <- createDataPartition(data$finish\_time, p = .8, list = FALSE)

train <- data[trainIndex, ]

test <- data[-trainIndex, ]

# Fit model (e.g., random forest regression)

library(randomForest)

rf\_model <- randomForest(finish\_time ~ avg\_speed + over\_distance + efficiency + ..., data=train)

# Predict

pred <- predict(rf\_model, newdata=test)

# Evaluate

postResample(pred, test$finish\_time)

rf\_class <- randomForest(as.factor(winner) ~ avg\_speed + efficiency + ..., data=train)

pred\_class <- predict(rf\_class, newdata=test)

confusionMatrix(pred\_class, as.factor(test$winner))

**Pipeline Summary**

1. **Collect data:** actual distance, finish time, and optional GPS coordinates.
2. **Compute features:** average speed, efficiency, over distance, max speed, acceleration.
3. **Prepare target:** time, winner, top 3, etc.
4. **Split data:** training and testing.
5. **Train model:** regression or classification.
6. **Evaluate & tune:** RMSE for regression, accuracy / AUC for classification.

**Week 1: Install RStudio, download the CSV file from the Velocity group on Microsoft Teams, refresh differential calculations (limits, derivatives, integration), and create a formulated**

**Week 2: Meet with Prof. Ant to catch up on the project and familiarise yourself with the existing studio code from the velocity folder.**