# Logistic Regression with Generalized Linear Models (GLM) in R

## 1. Introduction to Generalized Linear Models (GLM)

A Generalized Linear Model (GLM) is a flexible extension of the traditional linear regression model. It helps model different types of outcomes like counts, proportions, or binary values—not just continuous numbers.

Every GLM has **three main parts**:

* **A probability distribution for the response variable**
  + This could be a normal distribution (like in regular linear regression), or other types like Bernoulli (for binary outcomes) or Poisson (for counts).
* **A linear predictor**
  + A combination of input variables and their weights (coefficients), usually written as:

η = X^T β

* **A link function**
  + This function connects the linear predictor η to the expected value of the response variable μ = E(Y|X).
  + The link function varies with the distribution. For example, the logit function is used for binary outcomes.

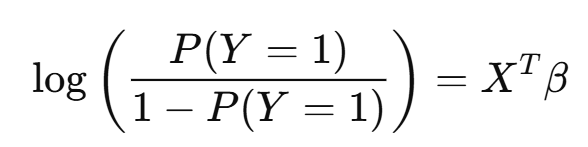
The overall GLM equation is:

g(μ) = X^T β or μ = g⁻¹(X^T β)

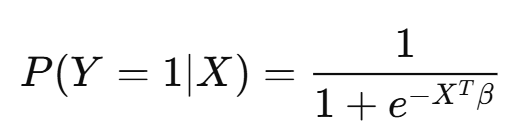
## 2. Logistic Regression Model: Key Characteristics

Logistic Regression (LR) is a type of GLM for modeling binary outcomes (e.g., yes/no, 1/0).  
  
**Key Features of Logistic Regression:**

* Handles binary outcomes:
  + Predicts probabilities for two possible outcomes.
* Uses the logit link function:
  + The model estimates the log-odds of the event occurring:



* + Then converts it back to probability:



* **Estimates coefficients using maximum likelihood**
  + Instead of minimizing squared error like in linear regression, LR finds the set of coefficients that make the observed data most likely.
* **Supports model evaluation through classification metrics**
  + Metrics like AUC (Area Under the Curve), KS (Kolmogorov-Smirnov), and Gini coefficient help measure how well the model distinguishes between classes.
* **Feature interpretability**
  + Each coefficient represents the effect of a variable on the log-odds of the outcome. A positive coefficient increases the odds of the outcome being 1.
* **Can handle categorical and continuous predictors**
  + With the right preprocessing (like dummy encoding), LR can include both types of variables.

## 3. The Code Step-by-Step and Explanation

Here’s a simplified and cleaned-up version of the code used to train and evaluate a logistic regression model in R using GLM.  
**Code- **

**Step 1: Load Data**  
  
train\_data <- read\_csv("train.csv")  
test\_data <- read\_csv("test.csv")  
  
  
**Step 2: Prepare Variables**  
  
model\_var <- c("var1", "var2", "var3")  
x\_train <- train\_data %>% mutate\_at(model\_var, ~ifelse(is.na(.) | . < 0, -99, .))  
  
  
**Step 3: Binning and WOE Calculation**  
bins <- newBinning(data = x\_train, target = "target", varsToInclude = model\_var)  
seg\_bin <- interBin(bins)  
train\_bin <- applyBinning(binning = seg\_bin, data = x\_train, target = "target")  
  
  
**Step 4: Calculate WOE**  
for (var\_name in model\_var) {  
 query <- sprintf("SELECT %s, SUM(weight) AS total, SUM(target \* weight) AS event, SUM(non\_event\_flag \* weight) AS non\_event FROM train\_bin GROUP BY %s", var\_name, var\_name)  
 a1 <- sqldf(query)  
 output <- bind\_rows(output, a1)  
}  
output$woe <- log(output$event / sum(output$event) / (output$non\_event / sum(output$non\_event)))

**Step 5: Applying WOE to Training Data**

train\_woe <- train\_bin

for (var in model\_var) {

df <- output %>%

filter(var\_name == var) %>%

select(bin, woe)

colnames(df)[1] <- var

train\_woe <- left\_join(train\_woe, df, by = var)

new\_col\_name <- paste0(var, "\_woe")

colnames(train\_woe)[colnames(train\_woe) == "woe"] <- new\_col\_name

}  
  
  
**Step 6: Build Model**  
woe\_vars <- paste0(model\_var, "\_woe")

train\_woe\_data <- train\_woe %>% select(all\_of(woe\_vars), target)

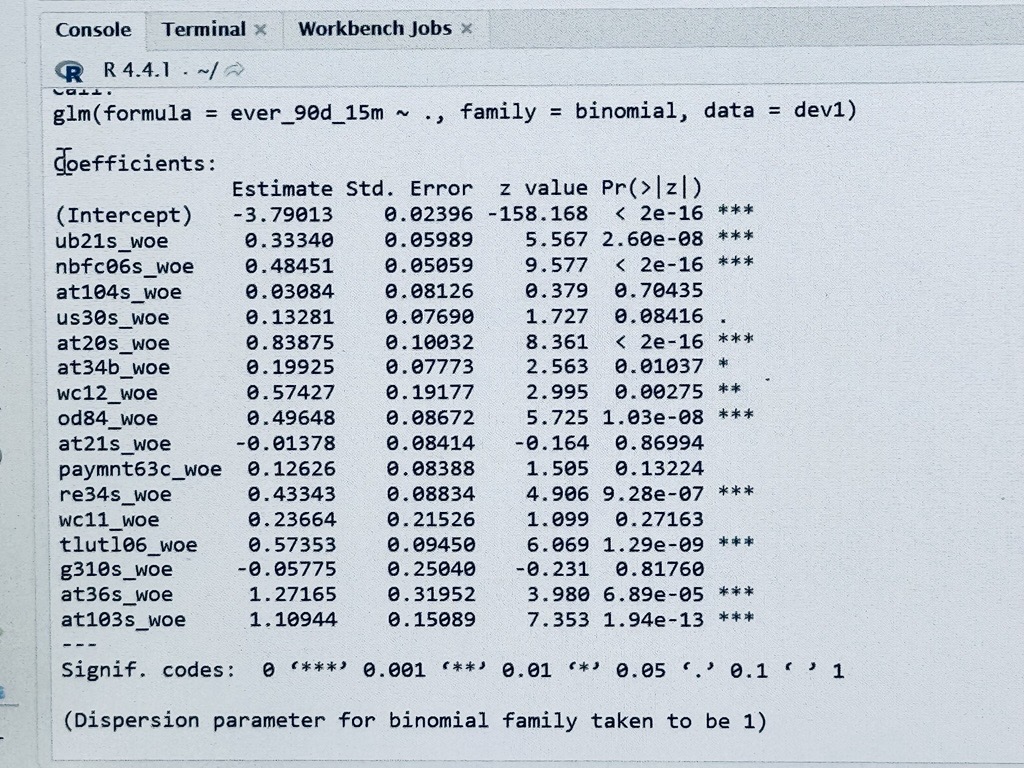
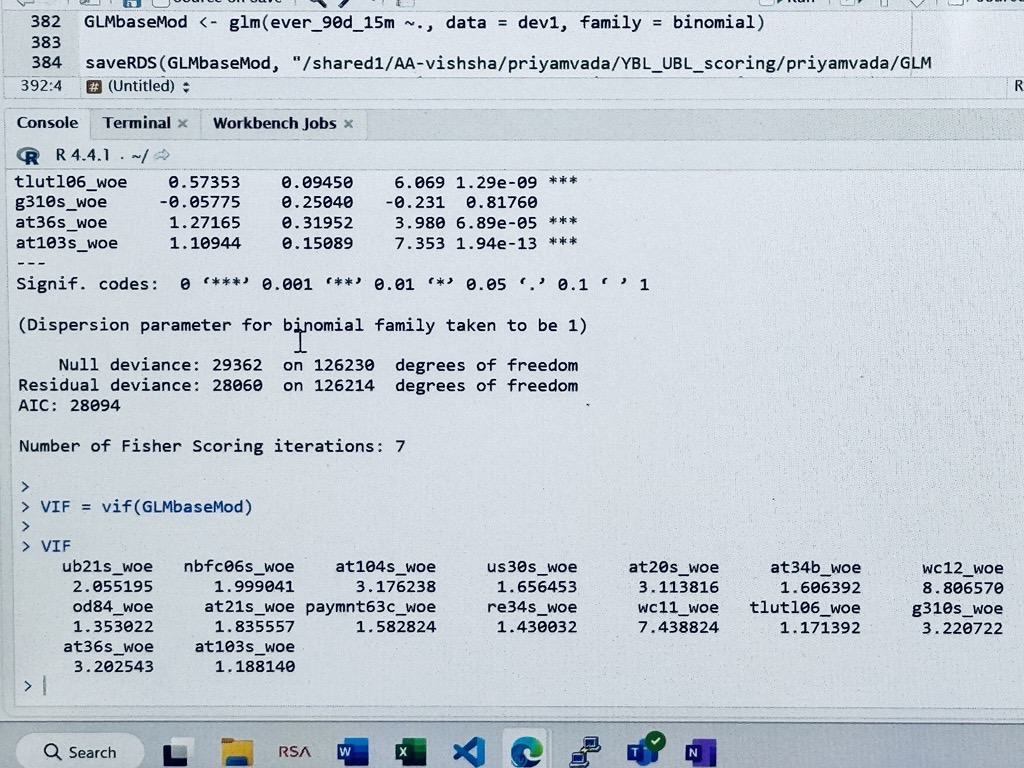
glm\_model <- glm(target ~ ., data = train\_woe\_data, family = binomial(link = "logit"))

summary(glm\_model)

**Step 7: Model Evaluation Functions**

## 4. Variable Selection Criteria

* **Coefficient ≥ 0**
  + The variable has a non-negative effect on the log-odds of the target event occurring.
* **p-value < 0.001**
  + A low p-value implies that there's strong statistical evidence the variable is related to the target.
  + Reduces noise in the model
  + Ensures reliability of predictors
* **VIF < 2**
  + A VIF of 1 means no correlation; VIF > 2 suggests a moderate correlation.
  + Lower VIF ensures your model isn't affected by highly correlated variables, which can make coefficients unstable and hard to interpret.
  + Helps improve model generalizability and reduces overfitting.



## Reference

* <https://medium.com/@sahin.samia/a-comprehensive-introduction-to-generalized-linear-models-fd773d460c1d>
* <https://www.researchgate.net/publication/228396785_Generalized_Linear_Models>