Modèle linéaire Taux de conformité

# Le code suivant, qui crée un dataframe et supprime les lignes dupliquées, est toujours exécuté et sert de préambule à votre script :

# dataset <- data.frame(undefined, undefined.1)

# dataset <- unique(dataset)

# Collez ou tapez votre code de script ici :

library(dplyr)

library(gridExtra)

library(grid)

library(dplyr)

# à installer sur ta machine R avec install.packages("gridExtra")

# Transformer les colonnes character en factors

dataset <- dataset %>% mutate\_if(is.character, as.factor)

# Construire le modèle

modele <- lm(Taux\_conformite ~ tarif + operateur + compte.nom, data = dataset)

# Extraire le résumé du modèle

res <- summary(modele)

# Extraire les coefficients en data.frame pour affichage

tab\_coef <- as.data.frame(res$coefficients)

tab\_coef$Variable <- rownames(tab\_coef)

rownames(tab\_coef) <- NULL

# Réarranger colonnes pour meilleure lisibilité

tab\_coef <- tab\_coef[, c("Variable", names(tab\_coef)[1:(ncol(tab\_coef)-1)])]

# Afficher la table avec gridExtra::tableGrob

grid.newpage()

grid.table(tab\_coef)

Modèle linéaire taux de sous facturation

# Le code suivant, qui crée un dataframe et supprime les lignes dupliquées, est toujours exécuté et sert de préambule à votre script :

# dataset <- data.frame(undefined, undefined.1, undefined.2, undefined.3)

# dataset <- unique(dataset)

# Collez ou tapez votre code de script ici :

# Collez ou tapez votre code de script ici :

library(dplyr)

library(gridExtra)

library(grid)

library(dplyr)

# à installer sur ta machine R avec install.packages("gridExtra")

# Transformer les colonnes character en factors

dataset <- dataset %>% mutate\_if(is.character, as.factor)

# Construire le modèle

modele <- lm(Taux\_sousfacturation ~ operateur +compte.nom+nom, data = dataset)

# Extraire le résumé du modèle

res <- summary(modele)

# Extraire les coefficients en data.frame pour affichage

tab\_coef <- as.data.frame(res$coefficients)

tab\_coef$Variable <- rownames(tab\_coef)

rownames(tab\_coef) <- NULL

# Réarranger colonnes pour meilleure lisibilité

tab\_coef <- tab\_coef[, c("Variable", names(tab\_coef)[1:(ncol(tab\_coef)-1)])]

# Afficher la table avec gridExtra::tableGrob

grid.newpage()

grid.table(tab\_coef)

Modèle linéaire taux de surfacturation

# Le code suivant, qui crée un dataframe et supprime les lignes dupliquées, est toujours exécuté et sert de préambule à votre script :

# dataset <- data.frame(Taux\_surfacturation, operateur)

# dataset <- unique(dataset)

library(dplyr)

library(gridExtra)

library(grid)

dataset <- dataset %>% mutate\_if(is.character, as.factor)

modele <- lm(Taux\_surfacturation ~ operateur, data = dataset)

res <- summary(modele)

tab\_coef <- as.data.frame(res$coefficients)

tab\_coef$Variable <- rownames(tab\_coef)

rownames(tab\_coef) <- NULL

tab\_coef <- tab\_coef[, c("Variable", names(tab\_coef)[1:(ncol(tab\_coef)-1)])]

# Ne garder que les coefficients estimés

tab\_coef\_simple <- tab\_coef[, c("Variable", "Estimate")]

grid.newpage()

grid.table(tab\_coef\_simple)