setwd(“c:/FCD/Hamburgueria209/”) getwd()

vend <- read.csv(‘data207’, stringsAsFactors = FALSE)

# dataset de segurança

View(vend1)

dim(vend1)

str(vend1)

glimpse(vend1)

# Medidas de tendencia

summary(vend1)

Uma imagem contendo interior, parede, mesa

Descrição gerada automaticamente

#Pacote para ler graficos e corplot

library(psych)

library(corrplot)

library(gmodels)

library(class)

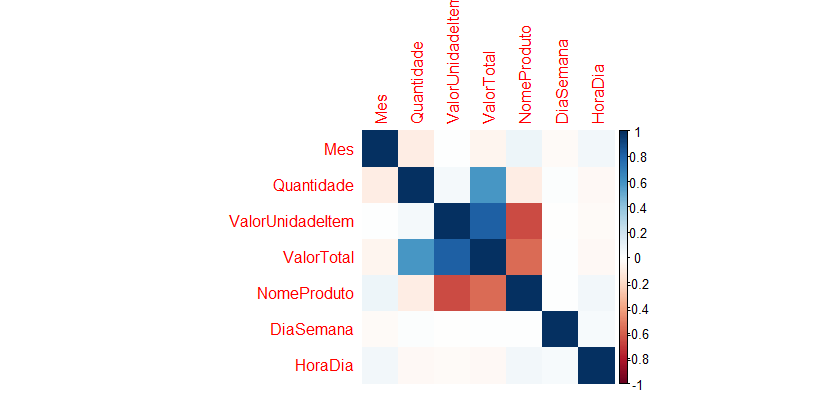
library(dplyr)

library(tidyverse)

# Correlacao entre as variaveis

colunas\_cor <- cor(vend1[c(“Mes”, “Quantidade”, “ValorUnidadeItem”, “ValorTotal”, “NomeProduto”, “DiaSemana”, “HoraDia”)]) cor(vend1[c(“Mes”, “Quantidade”, “ValorUnidadeItem”, “ValorTotal”, “NomeProduto”, “DiaSemana”, “HoraDia”)])

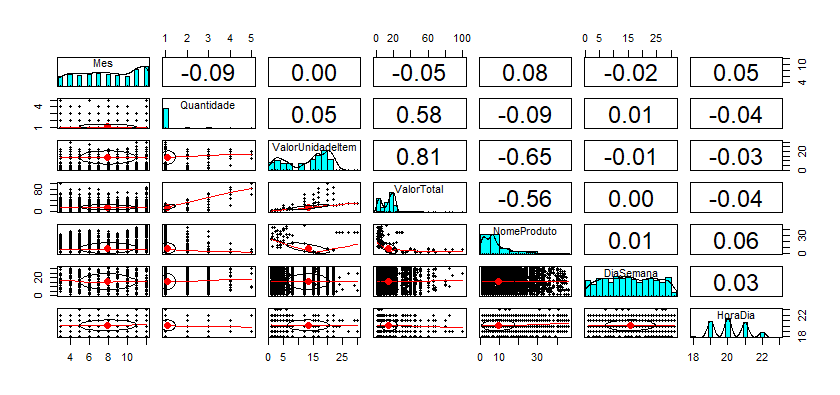
# Grafico de Correlacao

corrplot(colunas\_cor, method = “color”)

# 

# Representacao no grafico pairs

pairs(vend1[c(“Mes”, “Quantidade”, “ValorUnidadeItem”, “ValorTotal”, “NomeProduto”, “DiaSemana”, “HoraDia”)]) #Repesentacao do grafico pairs colorido pairs.panels(vend1[c(“Mes”, “Quantidade”, “ValorUnidadeItem”, “ValorTotal”, “NomeProduto”, “DiaSemana”, “HoraDia”)])



Histogramas

hist(vend1$Mes, main = "Meses x Vendas", xlab = "Meses", ylab = "Volume", col = "blue") hist(vend1$DiaSemana, main = “Por Dia”, xlab = “Dia da Semana”, ylab = “Volume”, col = “blue”) hist(vend1$HoraDia, main = “Por Horario”, xlab = “Hora”, ylab = “Volume”, col = “blue”)

Uma imagem contendo captura de tela

Descrição gerada automaticamenteUma imagem contendo captura de tela

Descrição gerada automaticamente

# 

# Uma imagem contendo captura de tela Descrição gerada automaticamente

# Treinando o modelo

library(caTools)

# Criando dados de treino e dados de teste

amostra <- sample.split(vend1, SplitRatio = 0.70)

dados\_treino <- subset(vend1, amostra == TRUE) dados\_teste <- subset(vend1, amostra == FALSE)

# Criando modelo de treino

modelov1 <- lm(ValorTotal ~ ., dados\_treino)

# Sumario do modelo

summary(modelov1)

Uma imagem contendo texto

Descrição gerada automaticamente

# Prevendo com dados de testes

previsao <- predict(modelov1, dados\_teste) View(previsao)

resultado <- cbind(dados\_teste$ValorTotal, previsao) colnames(resultado) <- c(“Real”, “Previsto”) resultado <- as.data.frame(resultado) View(resultado)

#ggplot(vend1, aes(x = NomeProduto, y = Mes)) + # geom\_line() + # ylab(“Numero de Hanburgueres”) + # xlab(“Linha do Tempo”) + # ggtitle(“Demanda de Hamburgueres por Mes”) + # theme(text = element\_text(size = 15))