```
library(tidyverse)
library(stringr)
library(dplyr)
library(rpart)
library(rpart.plot)
library(sf)

# Questão 1
dados <- read_csv2('/home/fonta42/Desktop/ICDuR/Listas/bgg_dataset.csv')

# a)
abstract_games <- filter(dados, str_detect(Domains, 'Abstract Games'))
abstract_games <- group_by(abstract_games, Domains, 'Rating Average')
abstract_games$Domains <- str_wrap(abstract_games$Domains, width = 20)

grafico_abstract_games <- ggplot(data = abstract_games) +
  geom_col(
    position = position_dodge(),
    mapping = aes(
      x = Domains,
      y = `Rating Average`,
      fill = Domains,
    )
  ) +
  scale_y_continuous(n.breaks = 10) +
  labs(x = "Jogo", y = "Avaliação Média") +
  theme(
    axis.title = element_text(size = 10),
    plot.title = element_text(size = 12, face = "bold")
  ) +
  ggtitle("Abstract Games - Avaliações Médias")
grafico_abstract_games

# b)
strategy_1980 <- filter(dados,
                        Domains =='Strategy Games' &
                          `Year Published` >= 1980)
strategy_1980 <- summarise(strategy_1980,
                           count(strategy_1980, `Year Published`))

grafico_strategy_1980 <- ggplot(data = strategy_1980) +
  geom_line(
    mapping = aes(
      x = `Year Published`,
      y = n,
    )
  ) +
  scale_y_continuous(n.breaks = 10) +
  scale_x_continuous(n.breaks = 10) +
  labs(x = "Ano", y = "Jogos Publicados") +
  theme(
    axis.title = element_text(size = 10),
    plot.title = element_text(size = 12, face = "bold")
  ) +
```

```
  ggtitle("Stratregy Games - Publicados por ano, a partir de 1980")
grafico_strategy_1980

# c)
grafico_idade_complexidade <- ggplot(data = dados) +
  geom_point(
    mapping = aes(
      x = `Complexity Average`,
      y = `Min Age`,
    )
  ) +
  scale_y_continuous(n.breaks = 14) +
  scale_x_continuous(n.breaks = 20) +
  labs(x = "Idade Mínima", y = "Dificuldade Média") +
  theme(
    axis.title = element_text(size = 10),
    plot.title = element_text(size = 12, face = "bold")
  ) +
  ggtitle("Idade mínima X Nível de Complexidade")
grafico_idade_complexidade


# Questão 2
dados <- read_csv('/home/fonta42/Desktop/ICDuR/Listas/insurance.csv')

# a)
# Valores menores que 10.000,00;
intervalo_um <- filter(dados, charges < 10000)
ggplot(data = intervalo_um, aes(x = age, y = charges)) +
  geom_point() +
  geom_smooth()
cor(intervalo_um$age, intervalo_um$charges)


# Valores entre 15.000,00 e 30.000,00
intervalo_dois <- filter(dados, charges >= 15000 & charges <= 30000)
ggplot(data = intervalo_dois, aes(x = age, y = charges)) +
  geom_point() +
  geom_smooth()
cor(intervalo_dois$age, intervalo_dois$charges)

# Valores entre 35.000,00 e 45.000,00
intervalo_tres <- filter(dados, charges >= 35000 & charges <= 45000)
ggplot(data = intervalo_tres, aes(x = age, y = charges)) +
  geom_point() +
  geom_smooth()
cor(intervalo_tres$age, intervalo_tres$charges)
# Análise: Para a faixa de valores de seguro menores que 10.000,00 o
coeficiente
# Pearson apresentado foi o maior, sendo ~0.96, por uma larga margem em
relação
# aos demais. Pela análise gráfica esse mesma faixa de valor apresenta um
# comportamento bastante próximo a uma reta linear crescente no eixo y
conforme
# o eixo x aumenta. Por outro lado, a faixa de valores entre 15.000,00 e
```

```
# 30.000,00 e a faixa de valores entre 35.000,00 e 45.000,00, apresentaram
em
# seus gráficos um tendência curvilinea, pois o eixo y aumenta junto com o
eixo
# x até um certo ponto, porém a partir de determinados valores para x,
conforme
# o mesmo aumenta, y demonstra uma tendência de diminuir, descaracterizando
um
# comportamento linear.
# Dessa forma, conforme os argumentos supracitados a faixa de valores
menores que
# 10.000,00 demonstrou ser mais eficiente para utilização da regressão
linear.


#b)
# divisao em conjunto de treino e de teste
prepare_hold_out <- function(tbl, training_perc) {
  # misturando as observacoes
  tbl_mixed <- tbl[sample(1:nrow(tbl)), ]
  nrow <- nrow(tbl_mixed)

  nrow_train <- ceiling(training_perc * nrow)
  data_trn <- tbl_mixed[1:nrow_train, ]
  data_tst <- tbl_mixed[(1 + nrow_train):(nrow), ]

  # retorna como uma lista nomeada
  list(training = data_trn, test = data_tst)
}

# Configurando uma seed para possibilitar reprodução exata dos resultados
set.seed(12345)

# Dados divididos em conjunto de treino(70%) e teste(30%)
intervalo_um_misturado <- prepare_hold_out(intervalo_um, 0.7)

# Modelo
linear_model <- lm(charges ~ age, intervalo_um_misturado$training)

# Predições do modelo
valores_preditos <- predict(linear_model, intervalo_um_misturado$test)

# c)
# Calcula a acurácia do modelo
accuracy_measures <- function(predicted, observed) {
  e <- observed - predicted
  mae <- mean(abs(e), na.rm = TRUE) # mean absolute error
  mse <- mean(e^2, na.rm = TRUE) # mean squared error
  rmse <- sqrt(mse) # root mean squared error

  rss <- sum(e^2) # residual sum of squares
  tss <- sum((observed - mean(observed))^2) # total sum of squares
  r2 <- 1 - rss / tss

  pe <- e / observed * 100
```

```
  mape <- mean(abs(pe), na.rm = TRUE) # mean absolute percentage error

  list(MAE = mae, RMSE = rmse, MAPE = mape, R2 = r2)
}

# Resultado da análise
vetor_test <- pull(intervalo_um_misturado$test[ , 7])
accuracy_measures(valores_preditos, vetor_test)
# Como o valor de R2 é de 0.9300128, ou seja, muito próximo de 1, é possível
# afirmar que o modelo é uma boa representação do conjuntos de dados.


# Questao 3
# a)
# Configurando uma seed para possibilitar reprodução exata dos resultados
set.seed(12345)

# Dados divididos em conjunto de treino(70%) e teste(30%)
dados_misturados <- prepare_hold_out(dados, 0.7)

# Construindo a árvores
tree <- rpart(smoker ~ age + sex + bmi + children + region + charges,
              data = dados_misturados$training)
rpart.plot(tree)

# b)
# predições do modelo
classes_preditas <- predict(tree, dados_misturados$test, type = "class")

# quantidade de elementos de cada classe em cada vetor
table(dados_misturados$test$smoker)
table(classes_preditas)

# matriz de confusão
confusion_matrix <- table(dados_misturados$test$smoker, classes_preditas)
confusion_matrix
# Quando testado localmente os resultados obtidos foram:
# Falsos positivos = 4
# Falsos negativos = 8

# Questão 4
cidades_brasil <- st_read('/home/fonta42/Desktop/ICDuR/Listas/
BR_Municipios_2020')
cidades_sp <- filter(cidades_brasil, SIGLA_UF == 'SP')
ggplot() +
  geom_sf(data = cidades_sp, mapping = aes(fill = NM_MUN)) +
  scale_fill_discrete(guide="none") + # retirando, pois são muitas cidades
  ggtitle("Cidades do estado de São Paulo")
```