Começo esse script em R dando um print(“Hello, World!”), como diz a lenda da programação.

Começo minha análise dando um attach no arquivo csv pois, eu joguei o ele pelo o enviroment, guardo meu arquivo na variável ‘df’, substituo os nomes das colunas, logo após eu excluo a coluna ‘price’ e seguida eu tento mudar o tipo primitivo de algumas colunas, mas n obtive êxito pois, não consegui remover o símbolo $ e a vírgula.

Em seguida eu tiro a média de avaliação dos usuários e dou um print para a visualização mais precisa, na quantidade de avaliações eu tento tirar a variação delas, mas n obtive êxito justamente porque não consegui trocar o tipo primitivo.

Após minha análise básica do dataframe, eu faço uma soma de todos os preços dos livros

Pois, queria criar uma coluna com o total de preços e avaliações, mas também não consegui.

Na minha análise eu queria fazer a porcentagem de livros que foram vendidos de cada gênero, criar uma coluna para a soma dos preços e o total de avaliações, qual o gênero que mais vendeu, qual o ano do livro que mais fez sucesso e plotar os gráficos para as respectivas variáveis.

Obs: Não entreguei de 9h pois, foi muito mais difícil do que imaginei kkkk e ainda passei a noite toda tentando

Feedback:

Dando uma primeira olhada no banco, existe uma dificuldade que em termos normais você teria de tratar primeiro de tudo: perceba no glimpse( ) que você deu que os valores monetários não são numeric (ou double, no caso). São String (character) porque quem quer que tabulou os dados incluiu um símbolo de $ pra representar o dólar.

Felizmente existe uma maneira de contornar isso que, no caso desse banco se não é tão difícil assim. Ele fez a inclusão de um atributo chamado Price\_r, que eu imagino que seja price com o parâmetro round, ou seja, arredondamento. Se pegarmos o atributo Price\_r, nós só precisamos nos livrar do $ e podemos typar a variável em numeric ou int.

O código é: gsub("[[:punct:]]", "", df$Price\_r)

Gsub é uma função do R base que faz uma substituição de todos os padrões de string por outro parâmetro em um objeto. Então traduzindo:

gsub(“character que você quer substituir”, “character que você quer no lugar”, nome do seu objeto).

Perceba que para o character que queremos substituir estamos usando [[:punct]], na documentação do gsub isso significa que QUALQUER pontuação ou character especial será substituído – incluindo o . ou , se o valor monetário tivesse casa decimal.

Já para o segundo parâmetro, estamos passando um espaço vazio entre as aspas. Isso sinaliza que queremos substituir por um elemento nulo. Obs: é as aspas sem espaço entre elas. Caso você coloque um espaço, o gsub substituirá por um espaço vazio.

Depois disso, podemos fazer a transformação do tipo de variável – transformando de string para numeric.

df$Price\_r <- as.numeric(df$Price\_r)

Shazam carai! Agora você tem uma variável numérica manipulável para fazer análises.

O que você descreveu no primeiro parágrafo é maior desafio que você teria pra manipular esse banco: ajeitar uma tabulação mal feita (ou de alguém que não tem o conhecimento que isso seria um problema para o estatístico).

Fazendo essa modificação você vai conseguir fazer a soma de todos os preços, tirar um summary( ) e ver a média e outras medidas de posição central.

Então vamos pra análise que você gostaria de ter feito. Vou tentar ao máximo usar só as coisas que a gente viu no workshop:

Usando o comando Unique(df$Genre) o R nos dá quantas variáveis diferentes aquele objeto (ou atributo) tem: Fiction e Non Fiction. Quando é dicotomizado (possui apenas 2) fica bem mais fácil. Então ótimo. A gente só vai precisar criar um filtro com 2 parâmetros e criar um df diferente pra cada um dos gêneros.

Por exemplo:

genre\_fiction = df %>% filter(Genre=="Fiction")

genre\_non\_fiction = df %>% filter(Genre=="Non Fiction")

agora no lugar de um, tempos dois Dataframes, um só com os livros de gênero Fiction, e outro de gênero Non Fiction.

Para saber quantos livros foram vendidos de cada título, basta fazer uma tabela de frequência – e joga um View ( ) pra melhorar a visualização. Então:

View(table(df$Name))

Mas nós dividimos em dois df, então pra saber o montante de livros vendidos de cada gênero, basta olhar a quantidade de observações de cada um dos dois dataframes que acabos de criar!



Para saber a proporção em %, basta fazer uma regra de 3 sabendo que 100% = 650 livros.

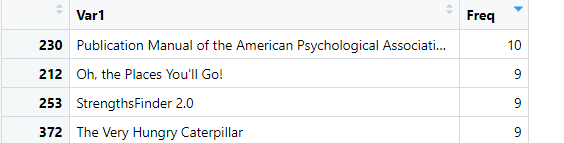
Então Fiction são 44.155% das vendas e Non Fiction são 55.845%.

Segundo passo, criar uma coluna com a soma dos preços não é exatamente necessário, pois você pode saber a soma dos preços com sum(genre\_fiction$Price\_r).

Sabemos assim que Fiction lucrou $3044 e Non Fiction $5274.

O Best Seller é o livro com maior frequência no objeto df$Names. Então precisamos fazer uma tabela de frequência com table(df$Names).

O livro mais vendido foi:



best\_seller = df %>% filter(Name=="Publication Manual of the American Psychological Association, 6th Edition")

View(best\_seller)

sum(best\_seller$Price\_r)

Por aí vai. Acredito que a coisa mais importante aqui é como tudo se abriu quando a gente conseguiu tratar apenas um objeto do banco (Price\_r) e como a gente conseguiu fazer isso no R. A partir daí plots também se tornam possíveis.