Isabela Gerdes Gyuricza

Sou formada em Biologia pela Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto (USP-RP). Atualmente, sou mestranda no Departamento de Genética (IB-USP). Meu projeto consiste na busca por genes modificadores do fenótipo da Sindrome de Marfan utilizando modelos murinos para a doença. Utilizamos uma série de análises estatísticas e matemáticas para obter animais com diferentes níveis de severidade da doença e, então, comparar as diferenças genéticas entre elas.

exec

TRABALHO FINAL

Plano A) Teste de agrupamentos PCA

O PCA (do inglês Principal Component Analysis) é uma análise multivariada de redução de dimensionalidade, criada por Pearson em 1901 (PEARSON, 1901). A redução da dimensionalidade é feita utilizando-se da ortogonalidade de vetores na conversão de variáveis correlacionadas em eixos cartesianos não correlacionados, denominados componentes principais (PCs). Cada PC é responsável por parte da variação das amostras, o que torna possível encontrar o conjunto de PCs que melhor explicam a variância dos dados. Em outras palavras, o PCA foi desenvolvido na tentativa de demonstrar como um conjunto amostral multidimensional comporta-se frente a determinados parâmetros (variáveis) quantificados. Essa análise apresenta uma série de vantagens e, uma delas, é a possibilidade de utilizar, para uma mesma amostra, variáveis de diferentes magnitudes. Isso é possível porque a PCA normaliza essas variáveis, dando a cada uma delas valores de coordenadas no eixo cartesiano, chamados de scores. Além disso, as amostras distribuídas no plano podem ser visualizadas em diferentes dimensões com o intuito de observar como os dados variam de forma mais compreensível. Uma das desvantagens do PCA é que ela é apenas uma análise exploratória, de forma que a partir dela é necessário realizar, manualmente, uma série de análises estatísticas subsequentes. Como o objetivo da PCA, em geral, é utilizar-se das dimensões e das variáveis mensuradas para verificar graficamente a separação de determinados grupos amostrais, é possível calcular as distâncias entre pontos no eixo cartesiano (distância euclidiana). Esse cálculo pode ser feito a partir da função dist () do R, utilizando o parâmetro euclidian. A fórmula para o cálculo é:

$dist\ ij=\sqrt{(xi-xj)^{(2}+(yi-yj)^{2}}$

A partir dessas distâncias é possível, então, verificar se a variação inter-grupo é maior do que a variação existente dentro de cada grupo. Então, podemos determinar se as variáveis analisadas foram capazes de definir e separar os grupos previamente selecionados.

Arquivos de entrada:

- Packages: “ggbiplot”e “grid”;

- data= dados para construção do PCA (classe=dataframe);

- class= coluna dos dados indicando cada grupo amostral (variável categórica, classe=caracter);

- qt= valor do quantil escolhido (default=0.95);

Verificações:

Data é um dataframe? Se não for – stop (“data precisa ser da classe dataframe”);

Existem dados faltantes em data? Se sim – stop (“valores faltantes em data”);

Data constituído de apenas variáveis contínuas? Se for não – stop (“data deve conter apenas valores da classe numérica”);

Class é uma variável categórica? Se não for – stop (“class deve ser da classe caracter”);

Pseudo-código:

1. Criar objeto pca a partir da função prcomp () utilizando objeto data como dados;

2. Criar objeto pca_centroid contendo as médias (μ) dos scores do pca por cada classe (class);

3. Criar objeto dist_cent com as distâncias entre centroides de cada classe (class) par a par pela função dist(), utilizando o argumento euclidian;

4. Cria objeto dist_class com as distâncias de cada amostra ao centroide da sua respectiva classe (class) função dist(), utilizando o argumento euclidian;

5. Criar um objeto dist_class_qt que armazena o quantil de 95% de dist_class para cada classe;

6. Criar um vetor dist_test contendo os resultados do teste lógico para todas as combinações possíveis par-a-par de classe. Teste:

dist_(cent i j) **>** dist_(class qt i)+ dist_(class qt j), sendo i e j classes (class) diferentes.

Arquivos de saída:

- Gráfico PCA pela função ggbiplot() tomando como dados o objeto pca e como parâmetro o objeto class.

- Dataframe dist_table contendo uma coluna com todas as combinações par a par dos grupos (class), as distâncias respectivas (dist_cent) e o resultado do teste lógico (dist_test).

Figura 1. Esquema ilustrando as variáveis testadas pela função. Para que haja um agrupamento, a distância entre as médias (centroides) de dois grupos (dist_cent) deve ser maior do que a soma das distâncias entre o centroide de cada grupo ao seu quantil amostral de referência (dist_class_qt).

Referências:

Pearson, K. “On Lines and Planes of Closest Fit to Systems of Points in Space”. Philosophical Magazine. V.2, n.11, p. 559–572, 1901.

LEVER, J.; KRZYWINSKI, M.; ALTMAN, N. Points of Significance: Principal component analysis. Nature Methods, v. 14, n. 7, p. 641–642, 29 jun. 2017.

Plano B) Lotação bandejões

A USP campus São Paulo possui quatro restaurantes universitários, também chamados de bandejões (restaurante central, da química, da física e da prefeitura). Por ser extremamente acessível, a maioria dos alunos optam por realizar as suas refeições (preferencialmente os almoços) nestes restaurantes. Tendo isso em vista, um problema que afeta a vida dos alunos e também dos funcionários dos restaurantes é a superlotação. Dependendo, principalmente, do cardápio e dos dias da semana, as filas são extensas e os alunos, muitas vezes, se deslocam de ônibus até os bandejões e acabam perdendo o almoço por não poderem esperar a fila. Apesar de existir um aplicativo em que os alunos podem ter acesso ao cardápio que será servido em cada bandejão, a previsão do nível de lotação passa a ser uma análise empírica dos alunos, do quão determinado cardápio parece agradar. Sendo assim, o desenvolvimento de um método de prever com maior precisão o nível de lotação dos bandejões em determinado dia pode auxiliar na rotina e no planejamento não apenas dos alunos, mas também dos funcionários que trabalham nas unidades e de toda a comunidade USP que faz uso desses restaurantes. Eu e o professor Alexandre entramos em contato com a Divisão de Alimentação da USP e pedimos acesso aos dados de frequência de usuários em cada bandejão por dia durante o período de 1 ano, bem como os cardápios referentes a estes dias. Caso obtivermos esses dados a tempo, pretendemos criar uma tabela com os seguintes dados:

data | dia_da_semana | prato_principal | sobremesa | frequencia

Passos pré-função:

1. Criar pacote “bandejao”;

2. Carregar pacote “nnet”;

3. Carregar objeto dados contendo a tabela acima;

4. Criar objeto regressao contendo o modelo de regressão multinominal usando a função multinom(), utilizando como variável resposta a “frequencia” e como variáveis preditoras, o “dia_da_semana”, “prato_principal” e “sobremesa”.

5. Criar objetos cardapio_RC, cardapio_RF, cardapio_PUSP, cardapio_RQ contendo o cardápio de almoço dos respectivos bandejões da USP obtidos por meio de web scraping (ex: função rvest()) do site oficial da usp.

Verificações:

Dados é um dataframe? Se não for – stop (“dados deve ser da classe dataframe”);

Dia_da_semana é uma variável categórica? Se não for – stop (“dia_da_semana deve ser da classe fator”);

Prato_principal é uma variável categórica? Se não for – stop (“prato_principal deve ser da classe fator”);

Sobremesa é uma variável categórica? Se não for – stop (“sobremesa deve ser da classe fator”);

Frequencia é uma variável contínua? Se não for – stop (“frequencia deve ser da classe numérica”);

Os objetos cardapio_ são dataframe? Se não for – stop (“cardapio_ deve ser da classe dataframe”);

OBS: O pacote “bandejao” deve conter a função, o objeto dados e os objetos cardapio_RC, cardapio_RF, cardapio_PUSP e cardapio_RQ.

Arquivos de entrada:

- dia=dia da semana em que se quer prever a lotação (classe=fator).

Verificações: Dia é um fator? Se não for – stop (“dia deve ser da classe fator”);

Dia está escrito como padrão “[...]-Feira? Se não – stop (“dia deve ser escrito como [...]-Feira”).

Planejamento da função: 1. Carregar objetos cardapio_RC, cardapio_RF, cardapio_PUSP e cardapio_RQ;

2. Fazer um subset de cada objeto acima contendo apenas a linha referente ao cardápio do dia escolhido e salvar cada um em um novo objeto cardapio_funcao_ (classe=dataframe);

3. Criar objeto prato principal para cada cardapio_funcao_, contendo a segunda linha do dataframe, nomeando como prato_principal_RC, prato_principal_RF, ... (classe = fator);

4. Criar objeto sobremesa para cada cardapio_funcao_, contendo a quinta linha do dataframe, nomeando como sobremesa_RC, sobremesa_RF, ... (classe = fator);

Figura 2. Esquema ilustrando as linhas que serão extraídas dos dados, a partir do web scraping do site da USP (http://sites.usp.br/sas/).

5. Carregar objeto regressao;

6. Criar um objeto para cada restaurante (restaurante_central, restaurante_da_fisica, restaurante_PUSPC_ e restaurante_das_quimicas) contendo os valores de predição da função regressão. Utilizar a função predict (), utilizando como variáveis preditoras os objetos dia e prato_principal_ e sobremesa_ respectivos a cada restaurante;

7. Salvar os valores preditos de cada restaurante em um objeto chamado score;

Verificações:

Objetos cardapio_funcao_ são dataframe? Se não - stop (“cardapio_funcao_ deve ser da classe dataframe”);

Objetos prato_principal_ são fatores? Se não – stop (“prato_principal deve ser da classe fator”);

Objetos sobremesa_ são fatores? Se não – stop (“sobremesa_ deve ser da classe fator”);

Objeto cardapio_funcao_ constituído dos caracteres “fechado”? Se sim – warning (“restaurante fechado”);

Arquivos de saída:

- Cardápios do dia (cardapio_RC, cardapio_RF, cardapio_PUSP e cardapio_RQ);

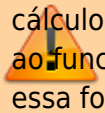
- Objeto score referente a cada restaurante;

As duas propostas estão bem delimitadas e parecem um bom desafio que você está apta a fazer. Entretanto acho que a proposta B é mais interessante exatamente por ser mais dinâmica no controle do que é usado a partir do input e do que é retornado pela função. Sugiro começar por ela e, se acontecer algum problema grande, só aí vc ir pra proposta A.

Se vc encontrar algum poblema crítico, no dia 15 ou 16 vou entrar aqui de novo pra olhar se tem alguma modificação da sua página (se houver tente deixar bem evidente, por favor) e te dar algum outro feedback se vc precisar. Depois desse prazo, se rolar algum problema grande, vc também pode tentar me contatar por whatsapp (por favor não mande áudio - (11) 9-9199-3842).

Matheus Januario

TRABALHO FINAL - justificativa



Como não obtive os dados da Divisão de Alimentação referente ao bandejão da USP/SP. Decidi seguir com a proposta A. Durante o desenvolvimento do projeto A, me deparei com alguns problemas, principalmente no que diz respeito a determinados conceitos de análise multivariada. Muito tempo foi gasto tentando avaliar como a função dist trabalhava os cálculos com mais de duas dimensões e também quais as estatísticas envolvidas por trás dos cálculos dos scores no PCA. Enquanto pesquisava em relação ao funcionamento do PCA, me surgiram algumas dúvidas se essa forma de calcular a distância de agrupamentos seria a mais correta. Dessa forma, dada a complexidade da análise multivariada e a escassez de tempo, decidimos realizar apenas a parte da função. Sendo assim, a função nova, consiste em automatizar e facilitar a análise exploratória dos dados em diferentes dimensões, gerando e salvando gráficos obtidos pela PCA.

Link para minha função finalizada: Trabalho Final

Link para o HELP da minha função: HELP