Đề thi:

R PROGRAMMING LANGUAGE FOR DATA SCIENCE

Thời gian làm bài : từ 7h30 đến 16h30, Chủ Nhật ngày 20/09/2020

Đọc kỹ các thông tin dưới đây trước khi làm bài :

- HV tạo một folder là LDS7_HoVaTen_Thi (nằm trong folder LDS7_HoVaTen đã share trên Google Drive), lưu tất cả bài làm vào để GV chấm điểm.
- HV phải ký tên vào danh sách thi, nếu không ký tên sẽ không có điểm.
- Đến deadline, HV gửi mail cho giáo viên kèm link của folder LDS7_HoVaTen_Thi, HV không gửi bài thi sẽ không có điểm thi.
- HV được sử dụng tài liệu.
- HV sẽ bị trừ điểm nếu bài làm giống nhau.

Chú ý, với mỗi câu:

- Lần lượt thực hiện các bước làm bài như đã được hướng dẫn làm bài tập trong lớp.
- Mỗi câu là 1 file, các yêu cầu nhận xét kết quả trong từng câu được viết trong cell dưới định dạng Markdown.

1. houses_to_rent.csv (2.0 điểm)

- *Tạo tập tin: question_1.R hoặc question_1.ipynb (toàn bộ code của câu 1 sẽ được viết trong file này)*
- Cho dữ liệu **houses_to_rent.csv**
- Yêu cầu :
  1. Đọc dữ liệu, hiển thị thông tin chung của dữ liệu : head(), tail(), str(), summary()
  2. Cho biết số dòng, số cột của dữ liệu. Gợi ý :

```
[1] "Number of rows 6080 ; Number of columns: 13"
```

  3. Cho biết 5 loại rooms xuất hiện nhiều nhất trong dataset (5 loại phòng được chọn thuê nhiều nhất). Gợi ý :

rooms	room_counts
3	1994
2	1621
1	1398
4	879
5	143

  4. Cho biết các cột có dữ liệu bị thiếu (na). Mỗi cột thiếu bao nhiêu giá trị? Tỷ lệ thiếu (lấy 2 số lẻ)? Gợi ý :

```
'area'   'bathroom'   'floor'
```

columns	count_missings	missing_percents
floor	1555	25.58
bathroom	121	1.99
area	17	0.28

5. Xóa bỏ các cột property.tax, fire.insurance, total. In head() để xem kết quả. Gợi ý :

	city	area	rooms	bathroom	parking.spaces	floor	animal	furniture	hoa	rent.amount
0	1	240	3	3	4	NA	acept	furnished	R$0	R$8,000
1	0	64	2	1	1	10	acept	not furnished	R$540	R$820
2	1	443	5	5	4	3	acept	furnished	R$4,172	R$7,000
3	1	73	2	NA	1	12	acept	not furnished	R$700	R$1,250
4	1	19	1	NA	0	NA	not acept	not furnished	R$0	R$1,200
5	1	13	1	NA	0	2	acept	not furnished	R$0	R$2,200

6. Tạo cột **rent_amount** từ cột rent.amount. Sau đó, bỏ ký tự "R$", và ',' ở rent_amount và đổi dữ liệu của cột này sang kiểu số. In head() để xem kết quả. Gợi ý :

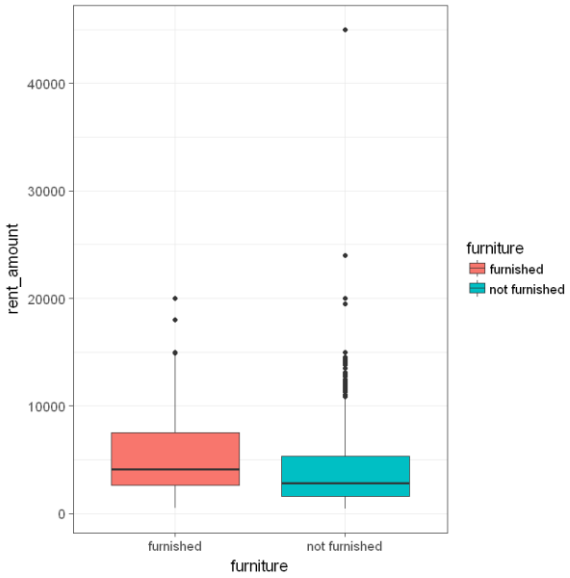
city	area	rooms	bathroom	parking.spaces	floor	animal	furniture	hoa	rent.amount	rent_amount
1	240	3	3	4	NA	acept	furnished	R$0	R$8,000	8000
0	64	2	1	1	10	acept	not furnished	R$540	R$820	820
1	443	5	5	4	3	acept	furnished	R$4,172	R$7,000	7000
1	73	2	NA	1	12	acept	not furnished	R$700	R$1,250	1250
1	19	1	NA	0	NA	not acept	not furnished	R$0	R$1,200	1200
1	13	1	NA	0	2	acept	not furnished	R$0	R$2,200	2200

7. Cho biết bao nhiêu nhà có nội thất, bao nhiêu nhà không có nội thất? Trong nhà có nội thất bao nhiêu nhà cho phép nuôi thú, bao nhiêu nhà không? Trong nhà không có nội thất, bao nhiêu nhà cho phép nuôi thú, bao nhiêu nhà không? Gợi ý :

furniture	counts
furnished	1582
not furnished	4498

furniture	animal	counts
furnished	acept	1130
furnished	not acept	452
not furnished	acept	3536
not furnished	not acept	962

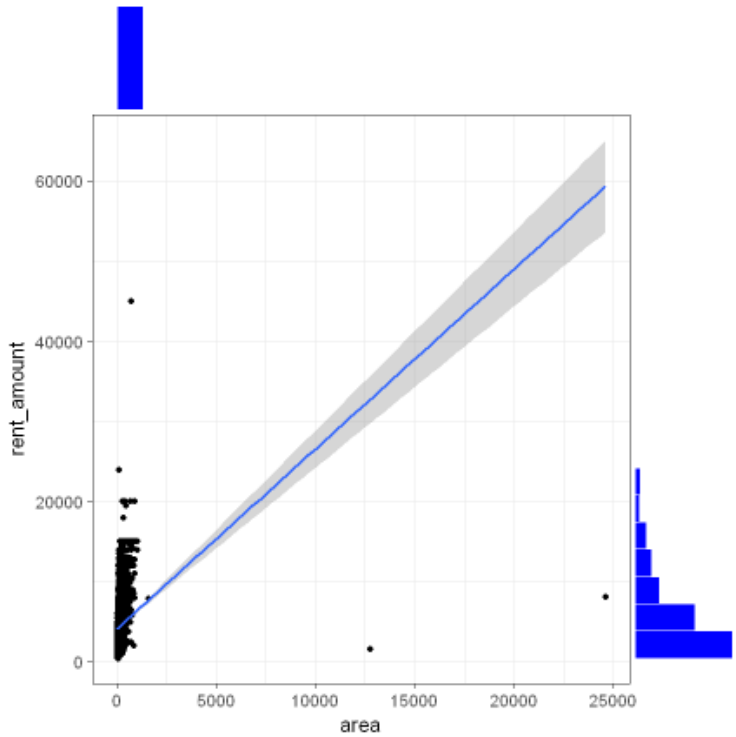
8. Vẽ boxplot của cột rent_amount theo furniture và nhận xét. Gợi ý :



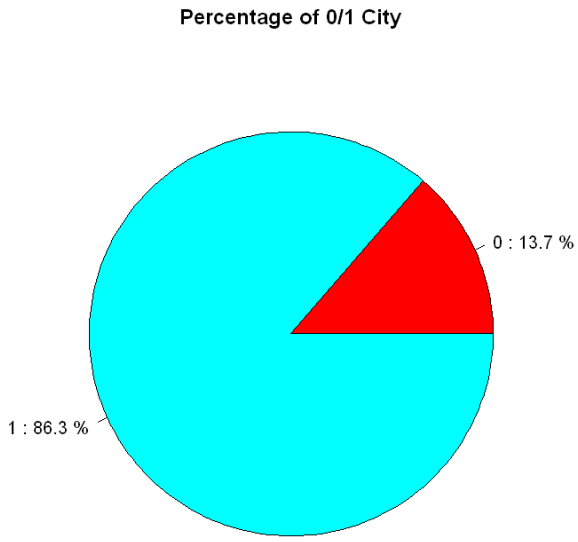
9. Cột area có các giá trị na, hãy thay thế các giá trị na bằng giá trị median.
10. Vẽ boxplot của cột area và nhận xét. Cột area có outlier không? Nếu có cho biết tổng số mẫu outlier. Gợi ý :

```
[1] "Data with area outlier: 6080"
[1] "Data without area outlier: 5835"
[1] "Number of area outlier rows: 245"
```

11. Vẽ biểu đồ thể hiện mối liên hệ của area và rent_amount có bổ sung thêm histogram phụ trên mỗi cột. Nhận xét biểu đồ. Gợi ý:



12. Vẽ pie chart thể hiện % giữa 1 và 0 của cột city. Nhận xét. Gợi ý :



13. Cho biết rent_amount lớn nhất và bé nhất? Liệt kê những căn nhà có rent_amount lớn nhất và bé nhất. Gợi ý :

```
[1] "Max rent amount: 45000"
[1] "Min rent amount: 420"
```

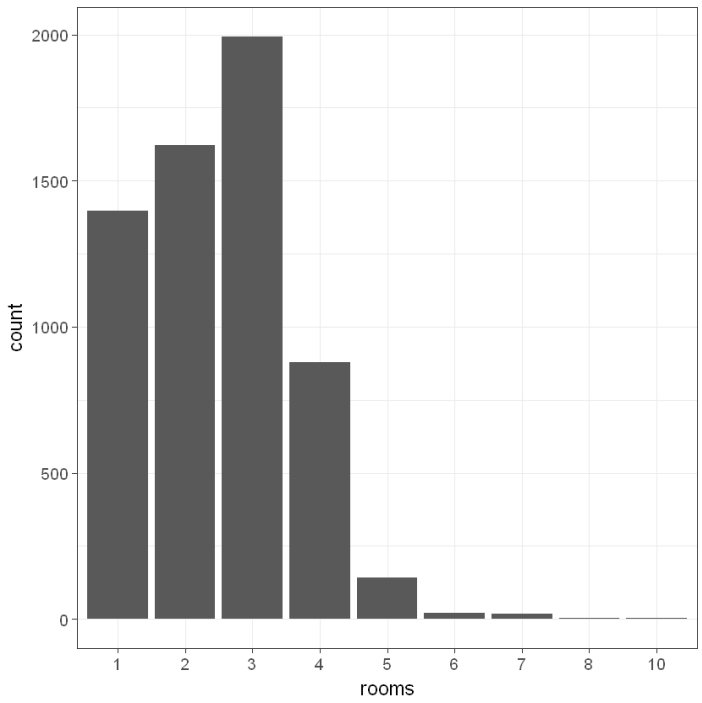
Max rent amount:

city	area	rooms	bathroom	parking.spaces	floor	animal	furniture	hoa	rent.amount	rent_amount
1	700	4	7	8	NA	acept	not furnished	R$0	R$45,000	45000

Min rent amount:

city	area	rooms	bathroom	parking.spaces	floor	animal	furniture	hoa	rent.amount	rent_amount
0	47	1	1	0	2	acept	not furnished	R$426	R$420	420

14. Vẽ biểu đồ thống kê rooms. Nhận xét. Gợi ý :



15. Có mối liên hệ nào giữa area và rent_amount hay không?

2. Titanic (2.0 điểm)

- *Tạo tập tin: **question_2.R** hoặc **question_2.ipynb** (toàn bộ code của câu 2 sẽ được viết trong file này)*
- Cài đặt **install.packages("titanic")**, đưa **library("titanic")** vào, sau đó sử dụng bộ dữ liệu có tên là **titanic_train** của package "titanic" để làm việc.
- Yêu cầu:
  1. Đọc dữ liệu titanic_train. Xem thông tin dữ liệu với head(), tail(), str(), summary().
  2. Tạo titanic_sub từ titanic_train chỉ chứa các cột: 'Survived', 'Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare', 'Embarked'. Cho biết dữ liệu có bao nhiêu dòng, cột?
  3. Trong titanic_sub có na (dữ liệu thiếu/ null) không? Nếu có xóa bỏ tất cả các dòng có chứa na. Cho biết dữ liệu lúc này còn bao nhiêu dòng? In thống kê chung.

     Từ câu 4. trở đi, sử dụng dữ liệu titanic_sub :
  4. Vẽ biểu đồ phân phối tần suất của Age. Nhận xét.
  5. Thực hiện các thống kê cơ bản cho Age và Fare (mean, median, mode, max, min, range)
  6. Cho biết các giá trị ở phân vị thứ 5%, 30%, 60% và 95% của Fare. Biểu diễn phân vị và giá trị tương ứng trên biểu đồ.
  7. Vẽ boxplot cho Age và Fare. Age có outlier hay không? Fare có outlier hay không? Nếu có thì mỗi cột có bao nhiêu outliers?
  8. Vẽ pie chart thống kê hành khách theo từng cột Survived, PClass, Sex. Nhận xét.
  9. Tính phương sai của tất cả các thuộc tính số trong titanic_sub.
  10. Tính standard deviation của tất cả các thuộc tính số trong titanic_sub.
  11. Tính skewness của tất cả các thuộc tính số trong titanic_sub. Nhận xét cho từng thuộc tính.
  12. Tính kurtosis của tất cả các thuộc tính số trong titanic_sub. Nhận xét cho từng thuộc tính.
  13. Vẽ biểu đồ thể hiện mối quan hệ giữa Age và Fare theo Sex, có bổ sung thêm histogram phụ trên mỗi cột. Nhận xét biểu đồ.
  14. Tính giá trị covariance, correlation giữa Age và Fare. Nhận xét.
  15. Cho biết số lượng mẫu có giá trị Fare >= 50, xác suất mẫu có Fare>=50 là bao nhiêu? Nhận xét.
  16. Tìm xác suất của P(30 ≤ Fare ≤ 50). Nhận xét.

3. Dự đoán số giờ nắng – Time series Analysis (1.0 điểm)

- *Tạo tập tin: **question_3.R** hoặc **question_3.ipynb** (toàn bộ code của câu 3 sẽ được viết trong file này)*
- Cho dữ liệu số giờ nắng ở Hà Nội trong 60 tháng của các năm từ 2013 -> 2017 như sau :

  c(12.2,38.9,75.4,69.0,158.0,161.7,119.9,140.9,89.4,134.9,68.8,158.7,
  119.2,31.9,14.9,13.5,181.5,120.3,133.0,107.6,137.7,134.6,86.3,87.5,
  98.9,43.8,32.4,114.3,204.7,178.0,124.0,157.7,101.0,139.0,83.6,44.6,
  39.6,91.7,22.7,64.6,143.5,192.8,152.4,129.4,119.4,144.5,104.2,135.0,
  49.7,72.9,45.6,81.7,147.9,123.9,111.6,107.6,97.9,93.7,75.1,67.6)
- Yêu cầu:

1. Hãy copy dữ liệu trên (cung cấp trong tập tin **time_series.txt**) và dán vào tập tin *question_3.R hoặc question_3.ipynb* vừa tạo để làm dữ liệu ban đầu (nhớ khai báo biến để nhận dữ liệu trên)
2. In dữ liệu vừa tạo.
3. Chuyển dữ liệu này thành Time Series object => in Time Series object.
4. Vẽ Time Series object vừa tạo.
5. Thực hiện việc dự báo và vẽ biểu đồ so sánh với thực tiễn.
6. Dự đoán số giờ nắng cho 6 tháng tiếp theo.

4. Normal – Binormial Distribution (0.5 điểm)

- *Tạo tập tin: **question_4.R** hoặc **question_4.ipynb** (toàn bộ code của câu 4 sẽ được viết trong file này)*
- Thực hiện các yêu cầu sau :
  1. Giả sử chỉ số IQ thường được phân phối với giá trị trung bình là 100 và độ lệch chuẩn là 15.
     a. Vậy tỷ lệ bao nhiêu phần trăm người có IQ nhỏ hơn 125 ?
     b. Vậy tỷ lệ bao nhiêu phần trăm người có IQ lớn hơn 110 ?
     c. Vậy tỷ lệ bao nhiêu phần trăm người có IQ trong khoảng từ 110 và 125 ?
  2. Xúc xắc có 6 mặt :
     a. Tìm xác suất để có được 2 lần mặt 4 nút trong 5 lần đổ xúc xắc.
     b. Có bao nhiêu mặt 4 nút khi có xác suất 25% xuất hiện khi một xúc xắc được đổ 50 lần?

5. Cars (1.5 điểm)

- *Tạo tập tin: **question_5.R** hoặc **question_5.ipynb** (toàn bộ code của câu 5 sẽ được viết trong file này)*
- Cho dữ liệu **cars** có sẵn trong R (dùng head(cars) để xem một số dòng dữ liệu trong cars)
- Yêu cầu: Sử dụng **cả Linenear Regression và Decision Tree** để thực hiện việc dự đoán car **dist** (khoảng cách được thực hiện để dừng xe lại) dựa trên car **speed** (tốc độ xe). Trong hai thuật toán trên thì thuật toán nào phù hợp hơn cho bộ dữ liệu này ? Vì sao ?
- Gợi ý các bước thực hiện cho từng thuật toán :
  1. Đọc dữ liệu và gán cho biến data.
  2. Xem thông tin data: head(), số dòng, số cột, summary.
  3. Tiền xử lý dữ liệu (nếu cần).
  4. Vẽ biểu đồ quan sát mối liên hệ giữa dist và speed.
  5. Kiểm tra outliers trong data => loại outliers nếu có.
  6. Tạo train và test từ dữ liệu data.
  7. Xây dựng model với train.
  8. In summary của model.
  9. Dự đoán y_pred từ test => so sánh với y_test.
  10. Đánh giá model.
  11. Với speed lần lượt là 5, 10, 15, 20 thì dist lần lượt là bao nhiêu ?

6. Mushroom (1.5 điểm)

- *Tạo tập tin: **question_6.R** hoặc **question_6.ipynb** (toàn bộ code của câu 6 sẽ được viết trong file này)*
- Cho dữ liệu mushroom trong tập tin **mushrooms.csv** chứa thông tin của các mẫu nấm, nấm ăn được và không ăn được.
  - Dữ liệu có thể tham khảo và download tại: https://www.kaggle.com/jnduli/decision-tree-classifier-for-mushroom-dataset/data

  Data Infromation : Bộ dữ liệu chứa 23 thuộc tính. Thuộc tính "class" là class attribute:
  - Attribute Information: (classes: edible=e, poisonous=p)
  - cap-shape: bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s
  - cap-surface: fibrous=f, grooves=g, scaly=y, smooth=s
  - cap-color: brown=n, buff=b, cinnamon=c, gray=g, green=r, pink=p, purple=u, red=e, white=w, yellow=y
  - bruises: bruises=t, no=f
  - odor: almond=a, anise=l, creosote=c, fishy=y, foul=f, musty=m, none=n, pungent=p, spicy=s
  - gill-attachment: attached=a, descending=d, free=f, notched=n
  - gill-spacing: close=c, crowded=w, distant=d
  - gill-size: broad=b,narrow=n
  - gill-color: black=k, brown=n, buff=b, chocolate=h, gray=g, green=r, orange=o, pink=p, purple=u, red=e, white=w, yellow=y
  - stalk-shape: enlarging=e, tapering=t
  - stalk-root: bulbous=b, club=c, cup=u, equal=e, rhizomorphs=z, rooted=r, missing=?
  - stalk-surface-above-ring: fibrous=f, scaly=y, silky=k, smooth=s
  - stalk-surface-below-ring: fibrous=f, scaly=y, silky=k, smooth=s
  - stalk-color-above-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
  - stalk-color-below-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
  - veil-type: partial=p,universal=u
  - veil-color: brown=n, orange=o, white=w, yellow=y
  - ring-number: none=n, one=o,two=t
  - ring-type: cobwebby=c, evanescent=e, flaring=f, large=l, none=n, pendant=p, sheathing=s, zone=z
  - spore-print-color: black=k, brown=n, buff=b, chocolate=h, green=r, orange=o, purple=u, white=w, yellow=y
  - population: abundant=a, clustered=c, numerous=n, scattered=s, several=v, solitary=y
  - habitat: grasses=g, leaves=l, meadows=m, paths=p, urban=u, waste=w, woods=d
- Yêu cầu: Sử dụng **cả Logistic Regression và Decision Tree** để thực hiện việc xác định một mẫu nấm là **nấm ăn được** hay **nấm độc** dựa vào các thông tin còn lại. Trong hai thuật toán trên thì thuật toán nào phù hợp hơn cho bộ dữ liệu này? Vì sao ?
- Gợi ý các bước thực hiện cho từng thuật toán :

1. Đọc dữ liệu và gán cho biến data.
2. Xem thông tin data: head(), số dòng, số cột, summary.
3. Tiền xử lý dữ liệu (nếu cần)
4. Tạo train và test từ dữ liệu data.
5. Xây dựng model với train.
6. In summary của model.
7. Dự đoán y_pred từ test => so sánh với y_test.
8. Đánh giá model.
9. Trực quan hóa model.

7. Attitude (1.5 điểm)

- *Tạo tập tin: **question_7.R** hoặc **question_7.ipynb** (toàn bộ code của câu 7 sẽ được viết trong file này)*
- Cho dữ liệu **attitude.csv**
- Yêu cầu: Đọc dữ liệu, chuẩn hóa dữ liệu (nếu cần) và sử dụng KMeans để thực hiện việc **phân cụm** dữ liệu dựa trên hai cột là **privileges** và **learning**.
- Gợi ý các bước thực hiện:
  1. Đọc dữ liệu và gán cho biến data.
  2. Xem thông tin data: head(), số dòng, số cột, summary.
  3. Tiền xử lý dữ liệu (nếu cần).
  4. Vẽ hình để xem mối liên hệ giữa privileges và learning. Cho nhận xét dựa trên biểu đồ.
  5. Xây dựng model từ dữ liệu privileges và learning.
  6. Tìm kết quả => có bao nhiêu cụm => mẫu nào thuộc cụm nào?
  7. Vẽ hình (với mỗi cụm là một màu) => xem kết quả.
  8. Đưa ra một số nhận xét dựa trên kết quả.

--- ☺ Chúc các bạn làm bài tốt ☺ ---